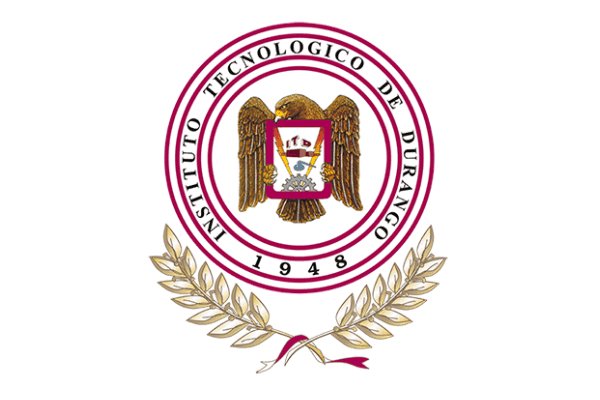
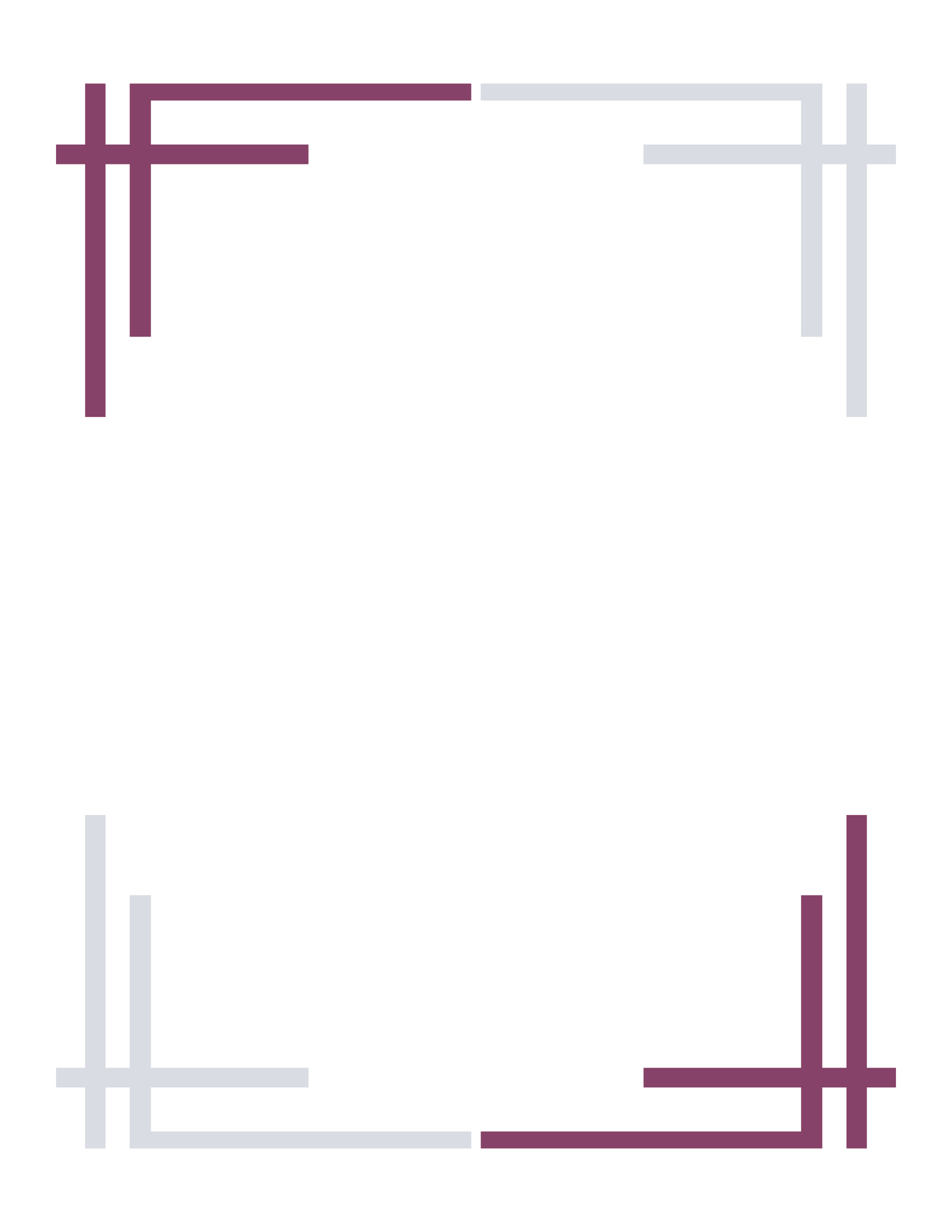
****

**INSTITUTO TECNOLÓGICO DE DURANGO**

**REPORTE DEL PROYECTO:**

**“PROTOTIPO FUNCIONAL DE PLATAFORMA DE TELEMEDICINA”**

**Caso para Titulación**

**Opción VI Examen Global por Áreas de Conocimiento**

**Alumno:** Masiel Elihú Rivas Ortega

**Número de control:** 08040294

**Carrera:** Ingeniería en Sistemas Computacionales

**Área de Especialidad:** Consultoría en Desarrollo de Aplicaciones Web

**Asesores Evaluadores:**

Araceli Domínguez Flores

Dora Luz González Bañales

Carlos Valenzuela Martínez

**Victoria de Durango, Durango, diciembre de 2025.**

**ÍNDICE**

**1. Introducción**

1.1 Contexto y problemática

1.2 Justificación del proyecto

1.3 Objetivo general

1.4 Objetivos específicos

1.5 Alcance del prototipo

1.6 Limitaciones

**2. Marco Teórico y Referencias Normativas**

2.1 Telemedicina: concepto y antecedentes  
2.2 Telemedicina en México  
2.3 Marco legal y ético aplicable

* Ley General de Salud (MX)
* NOM-004-SSA3 (expediente clínico)
* Protección de datos personales (LFPDPPP)
* Consideraciones de consentimiento informado  
  2.4 Plataformas digitales de salud y tendencias  
  2.5 Referencias académicas (mínimo 10, formato APA 7)

**3. Metodología de Desarrollo**

3.1 Enfoque: SCRUM  
3.2 Justificación de adaptación para proyecto individual  
3.3 Roles asignados (Alumno = Product Owner + Dev + Scrum Master)  
3.4 Artefactos utilizados

* Product Backlog
* Sprint Backlog
* Incrementos

3.5 Ciclo de trabajo: sprints semanales  
3.6 Métricas y herramientas utilizadas (Git, tablero, burndown)

**4. Análisis y diseño**

4.1 Requerimientos funcionales y no funcionales  
4.2 Historias de usuario  
4.3 Criterios de aceptación por HU  
4.4 Reglas de negocio  
4.5 Casos de uso del sistema  
4.6 Modelo de datos (conceptual y lógico)  
4.7 Diagrama de arquitectura

**5. Diseño del Sistema**

5.1 Arquitectura lógica y técnica  
5.2 Diseño de base de datos (MySQL)  
5.3 Diseño de componentes (WordPress + Plugin)  
5.4 Diagramas UML (según aplique)

* Diagrama de clases simplificado
* Diagrama de secuencia para agendar cita y videollamada  
  5.5 Diseño UX/UI (wireframes)  
  5.6 Políticas de privacidad y consentimiento

**6. Implementación**

6.1 Tecnologías y herramientas

* WordPress
* PHP, MySQL
* JavaScript
* Jitsi para videollamada  
  6.2 Estructura del plugin telemed-mx  
  6.3 Integración calendario/citas  
  6.4 Chat y videollamada  
  6.5 Gestión de historia clínica y archivos  
  6.6 Funcionalidad de pagos simulados  
  6.7 Seguridad básica (validación, roles, nonces)

**7. Resultados**

7.1 Capturas de pantalla del prototipo  
7.2 Funcionalidades logradas vs planificadas  
7.3 Flujo completo de demo  
7.4 Video de demostración (URL)  
7.5 Pruebas realizadas

* Funcionales
* Roles y permisos
* Flujo paciente–médico

**8. Conocimientos Aplicados por Área**

(Tabla solicitada en el documento oficial)

* Fundamentos de desarrollo de sistemas
* Programación web
* Tópicos web
* Gestión de proyectos
* Planificación y modelado
* Negocios por Internet
* Consultoría
* Negocios inteligentes
* Bases de datos avanzadas

**9. Conclusiones y Trabajo Futuro**

9.1 Conclusiones generales  
9.2 Valor del proyecto  
9.3 Mejoras futuras y escalabilidad

**10. Referencias**

(APA 7, mínimo 10 fuentes verificables)

**11. Anexos**

* Product Backlog completo
* Sprint Backlogs y evidencias
* Bases de datos (modelo y scripts)
* Manual de usuario
* Código base del plugin (fragmentos representativos)

# 1. RESUMEN EJECUTIVO

El presente proyecto tiene como objetivo el diseño y desarrollo de un prototipo funcional de una plataforma de telemedicina orientada a pacientes en México, que facilite la atención médica a distancia a través de una página web.

El prototipo permitirá las siguientes funciones principales:

* Registro y autenticación de pacientes y médicos.
* Gestión de solicitud de cita para médico especialista.
* Chat en vivo.
* Videollamada.
* Consulta de historia clínica.
* Consentimiento informado por parte del paciente Formulario para pago de la consulta.
* Envío de recordatorio de cita al paciente.

Para la demostración del prototipo se habilitarán tres especialidades médicas: Medicina General, Psicología y Cardiología, las cuales permitirán validar la funcionalidad del sistema en diferentes contextos de atención.

# 2. INTRODUCCIÓN

## 2.1 Planteamiento del problema

En México, aún hay muchos retos que impiden a algunos miembros de la población el recibir atención médica oportuna y de calidad, constituyendo esto un serio problema para la salud de nuestra población. Entre estos retos, están las largas distancias, el costo de transporte, comunidades sin especialistas. El problema se hace más evidente cuando los pacientes necesitan consultas frecuentes y tienen que prescindir de ellas por la poca facilidad.

La telemedicina surge como una alternativa para acceder a los servicios de salud. Gracias a las herramientas digitales, hoy es posible que un paciente pueda solicitar consultas y obtener seguimiento de su historia clínica, sin tener que desplazarse a un consultorio físico, o al menos, desplazándose solamente a donde exista acceso a internet.

2.2 Objetivos

Para dar respuesta al problema planteado y orientar el desarrollo del prototipo funcional, se establecen los objetivos que guían este proyecto. Estos objetivos permiten definir de manera clara qué se pretende alcanzar, así como las funciones esenciales que deberá cumplir la plataforma de telemedicina durante su operación y demostración.

2.2.1 Objetivo general

Diseñar y desarrollar un prototipo funcional de plataforma de telemedicina orientada a pacientes en México, que combine la comunicación entre el médico y paciente, la gestión de citas y el manejo de información, dentro de un entorno digital que sea seguro y fácil de utilizar.

### 2.2.2 Objetivos específicos:

* Registro y autenticación por correo electrónico para Paciente y Médico (roles).
* Perfil de usuario con datos mínimos (paciente: nombre, CURP, número de teléfono y e-mail
* de contacto; médico: cédula, especialidad).
* Agenda de citas por médico/especialidad.
* Formulario de pago y emisión de comprobante.
* Consentimiento informado: aceptación previa a la consulta
* Historia clínica mínima, ejemplo: antecedentes, alergias, medicamentos, signos/nota de
* evolución y archivos adjuntos limitados.
* Opciones para comunicación sincrónica: chat en vivo y videollamada durante la cita
* Recordatorios: email/SMS antes de la consulta
* Panel del médico: agenda, acceso a historia clínica y notas: consulta y captura de diagnóstico
* y prescripción de receta del paciente.

### 2.2.3 Alcance del prototipo funcional

El alcance incluye las funciones básicas de una plataforma web de telemedicina para pacientes mexicanos.

El sistema permitirá el registro y autenticación de usuarios (pacientes y médicos), la gestión de citas médicas, y la comunicación en tiempo real simulada, mediante chat y videollamada demostrativa entre médico y paciente. También incluirá la consulta de historia clínica, la validación del consentimiento informado, y la integración con servicios externos como la pasarela de pagos, el servidor de base de datos y el servicio de recordatorios automáticos.

No se busca desarrollar un producto final de telemedicina, sino una versión demostrativa, que sirva como base para futuras mejoras e implementaciones.

# 3. MARCO DE REFERENCIA

Para poder desarrollar correctamente el prototipo de telemedicina, es necesario conocer primero las bases teóricas y legales que lo sustentan. En este marco de referencia se explica qué es la telemedicina, cómo se aplica en México y qué normas deben cumplirse para proteger la información de los pacientes y asegurar que el servicio sea confiable y ético.

## 3.1 Definición de telemedicina

La telemedicina usa la tecnología para brindar atención médica sin que el paciente y el médico estén en el mismo lugar.

Según la Organización Mundial de la Salud (OMS, 2010):

“La telemedicina es la prestación de servicios de atención sanitaria, en los casos en que la distancia es un factor determinante, por profesionales de la salud que utilizan tecnologías de la información y la comunicación para el intercambio de información válida para el diagnóstico, tratamiento y prevención de enfermedades y lesiones, la investigación y la evaluación, y para la formación continua de los profesionales de la salud, todo ello en interés de mejorar la salud de las personas y de sus comunidades.” (p. 9)

Por su parte, la Secretaría de Salud de México (2023) la define como:

“La prestación de servicios de atención médica en la cual el profesional de la salud y el paciente se encuentran en diferentes ubicaciones geográficas y utilizan herramientas tecnológicas para la comunicación y transmisión de datos clínicos.” (párr. 1)

Ambas definiciones muestran que la telemedicina no solo se trata de una consulta por video, sino de un conjunto de servicios que facilitan el acceso a la atención médica mediante herramientas digitales, lo que representa un gran beneficio para la salud.

Después de la pandemia, su uso en México aumentó considerablemente, lo que evidenció tanto sus ventajas como la necesidad de un marco legal más claro (CENETEC, 2024; Revista Abogacía, 2025).

## 3.2 Consideraciones legales y normas aplicables a la telemedicina en México

En México no existe todavía una ley específica sobre telemedicina; es decir, ninguna ley federal regula de manera exclusiva todos los aspectos que esta abarca.

No obstante, sí hay normas generales y leyes que se pueden aplicar como principios a seguir en este tema, ya que regulan el manejo de la información médica y la protección de los pacientes.

A continuación, se mencionan algunos ejemplos importantes a tomar en cuenta durante este proyecto:

**a) NOM-168-SSA1-1998**

La Norma Oficial Mexicana NOM-168-SSA1-1998, del expediente clínico, señala que “el paciente debe ser tratado con confidencialidad y no se divulgará información sin su autorización expresa, incluso aquella derivada de un estudio de investigación al cual haya accedido voluntariamente” (Secretaría de Salud, 1998, p. 6).

**b) NOM-004-SSA3-2012**

Esta norma actualiza la anterior y establece que “los datos proporcionados al personal de salud por el paciente o por terceros son motivo de confidencialidad, en términos del secreto médico profesional” (Secretaría de Salud, 2012, p. 4).

**c) Ley Federal de Protección de Datos Personales en Posesión de los Particulares (LFPDPPP)**

La LFPDPPP complementa las normas sanitarias al establecer las obligaciones sobre el manejo de datos personales dentro de los sistemas digitales, incluidos los de telemedicina.

Según el Instituto Nacional de Transparencia, Acceso a la Información y Protección de Datos Personales (INAI, 2020), “los datos personales sensibles son aquellos que se refieren a la esfera más íntima de su titular, cuya utilización indebida puede dar origen a discriminación o conlleva un riesgo grave para éste” (p. 3).

En esta categoría se incluyen los datos médicos, por lo que su tratamiento requiere el consentimiento expreso y por escrito del paciente, tal como lo establece el artículo 9 de la misma ley.

El artículo 9 de la LFPDPPP señala que:

“El tratamiento de datos personales sensibles requiere el consentimiento expreso y por escrito del titular, a través de su firma autógrafa, firma electrónica o cualquier mecanismo de autenticación que al efecto se establezca.” (INAI, 2020, p. 8)

Esto significa que una plataforma de telemedicina debe solicitar autorización explícita antes de procesar información clínica, lo cual puede lograrse mediante un aviso de privacidad que el paciente acepte antes de usar el sistema.

Además, el artículo 15 de la ley dispone que el responsable del tratamiento de los datos debe poner a disposición del usuario un aviso de privacidad “previo al tratamiento de sus datos personales, de forma física, electrónica o en cualquier otro formato que permita su consulta de manera fácil y gratuita” (INAI, 2020, p. 10).

Estas medidas garantizan que el paciente conozca con claridad qué información se recaba, con qué finalidad y durante cuánto tiempo se conservará, cumpliendo con los principios de transparencia y seguridad establecidos por la ley.

Aplicar estos lineamientos dentro de una plataforma de telemedicina no solo asegura el cumplimiento legal, sino que también fortalece la confianza del usuario en los servicios digitales de salud.

## 3.3 Consideraciones y aspectos éticos

La práctica de la telemedicina no solo requiere bases legales, sino también el cumplimiento de principios éticos que garanticen la calidad y la seguridad del servicio.

La Organización Panamericana de la Salud (OPS, 2024) señala que la ética en la salud digital se basa en el respeto a la autonomía del paciente, la justicia en el acceso, la confidencialidad de los datos y la responsabilidad del personal de salud. Esto implica que, aun cuando la atención sea a distancia, el compromiso ético del médico con el bienestar del paciente debe mantenerse intacto.

Además, la OPS (2024) explica que “la confianza del paciente en las herramientas digitales depende de que se apliquen los mismos estándares éticos que en la atención presencial” (p. 12). Por ello, los profesionales de la salud deben garantizar que la información del paciente sea manejada con respeto, discreción y total transparencia.

El Centro Nacional de Excelencia Tecnológica en Salud (CENETEC, 2024) recomienda que los proyectos de telesalud incluyan medidas para asegurar la privacidad y promover la equidad, evitando que la falta de acceso a internet o a dispositivos tecnológicos se convierta en una barrera para recibir atención médica.

De igual forma, Saluddigital.io (2024) advierte que en México uno de los retos más importantes es la capacitación ética y técnica del personal médico. Esto es esencial para que el uso de las plataformas digitales no sustituya la empatía, y la sensibilidad humana que caracterizan a una consulta médica de calidad.

Las consideraciones éticas muestran que la tecnología no debe reemplazar el trato humano, sino fortalecerlo.

La confianza, la empatía y el respeto por la privacidad del paciente son elementos fundamentales para que la telemedicina sea no solo funcional, sino también humana.

Un prototipo de telemedicina que combine seguridad técnica, cumplimiento legal y sensibilidad ética contribuirá de manera real al bienestar de los pacientes y al desarrollo responsable de la salud digital en México.

# 4. METODOLOGÍA DE DESARROLLO

Para el desarrollo del prototipo se utiliza la metodología ágil Scrum, debido a su enfoque iterativo e incremental, que permite construir software funcional de manera progresiva y adaptarlo conforme se obtiene retroalimentación.

La Guía Scrum (Schwaber & Sutherland, 2020) describe el marco como:

“Scrum es un marco ligero que ayuda a las personas, equipos y organizaciones a generar valor mediante soluciones adaptativas para problemas complejos” (párr. 1).

Este enfoque permite planificar entregas funcionales en ciclos semanales, facilitando el control del progreso, la inspección continua y la mejora del producto.

En el desarrollo del prototipo de telemedicina, donde la interacción con usuarios y la validación continua de funciones como agenda médica y videollamadas son esenciales, Scrum permite obtener versiones funcionales tempranas y ajustar rápidamente los componentes según la retroalimentación obtenida.

## 4.1 Enfoque: SCRUM

Scrum promueve la planificación continua y la entrega frecuente de incrementos funcionales. Según la guía:

“Scrum emplea un enfoque iterativo e incremental para optimizar la previsibilidad y el control del riesgo” (Schwaber & Sutherland, 2020, párr. 8).

Su enfoque flexible facilita el abordaje de requerimientos que pueden evolucionar y permite mantener una comunicación clara y ciclos cortos de mejora.

Dado que el prototipo consiste en una plataforma funcional de telemedicina que requiere iteraciones sobre componentes como agenda médica, historia clínica, videollamada y chat, Scrum proporciona un enfoque adecuado para controlar y fortalecer el avance progresivo del proyecto.

## 4.2 Justificación de adaptación para proyecto individual

Aunque Scrum está diseñado para equipos, su aplicación puede adaptarse al contexto académico individual, manteniendo principios y artefactos esenciales. Como señalan Pressman y Maxim:

“Las metodologías de ingeniería de software pueden adaptarse a proyectos individuales conservando sus principios fundamentales para fines educativos” (Pressman & Maxim, 2019, p. 48).

Por ello, en este proyecto se utiliza una versión ajustada de Scrum donde el desarrollador asume los roles principales, enfocándose en:

* Planificación estructurada.
* Documentación de avances iterativos.
* Revisión después de cada sprint.
* Retrospectivas breves orientadas a mejora continua.

Se conserva, además, la estructura iterativa, los artefactos y la revisión constante de avances, asegurando que cada sprint generara un incremento verificable del producto.

## 4.3 Roles asignados

Scrum define tres roles principales: Product Owner, Scrum Master y Development Team.

Aunque este proyecto es realizado de forma individual, se mantiene la esencia de cada rol para conservar la estructura y la organización que propone la metodología.

A continuación, se enlistan:

|  |  |
| --- | --- |
| ROL | RESPONSABLE |
| Product Owner | Alumno |
| Scrum Master | Alumno |
| Development Team | Alumno |

### 4.3.1 Product Owner

En este rol el alumno asume la responsabilidad de definir la visión general del prototipo de telemedicina, priorizar las funciones más importantes y mantener claro el objetivo del producto.

El alumno se encarga de elaborar el Product Backlog y decidir qué historias de usuario se desarrollan primero, tomando en cuenta el valor que aportan al sistema y la secuencia lógica del desarrollo.

### 4.3.2 Scrum Master

Como Scrum Master, el alumno organiza el trabajo y asegura que cada sprint se complete según lo planeado. Esto implica revisar avances, anotar obstáculos técnicos y buscar soluciones rápidas para no frenar el progreso. También realiza pequeñas retrospectivas al final de cada semana, con el fin de analizar qué podía mejorar en el siguiente ciclo.

### 4.3.3 Development Team

En este rol, el alumno lleva a cabo las tareas técnicas del proyecto: analiza los requerimientos, diseña la base de datos, programa las funciones y realiza las pruebas de cada módulo.

Cada entrega semanal se documenta y controla mediante GitHub, mientras que las tareas se organizan en un tablero digital en la aplicación Trello, lo que permite mantener una visión clara y ordenada del avance general del desarrollo.

## 4.4 Artefactos utilizados

* Product Backlog: Contiene las historias de usuario y requerimientos priorizados del sistema de telemedicina.
* Sprint Backlog: Define las tareas seleccionadas para cada iteración semanal.
* Incremento: Representa la funcionalidad operativa generada al final de cada sprint, lista para revisión y documentación.

## 4.5 Ciclo de trabajo

El proyecto se organizó en **tres sprints**:

* **Sprint 1:** análisis, levantamiento de requerimientos, historias de usuario, modelo de datos y diseño de las interfaces.
* **Sprint 2:** implementación de la base funcional (registro de usuarios, login, disponibilidad médica y conexión con la base de datos).
* **Sprint 3:** desarrollo de funciones avanzadas como reserva de citas, consentimiento informado, historia clínica, chat simulado y generación de evidencias finales.

Durante el desarrollo fue necesario realizar una **replanificación**, dividiendo el Sprint 2 original en dos iteraciones más pequeñas, con el fin de ajustarse a la carga real de trabajo y a la fecha límite establecida en el caso. Este ajuste permitió mantener entregas continuas y asegurar que cada incremento del producto fuera funcional y demostrable.

## 4.6 Herramientas utilizadas

* Visual Studio Code: editor de código utilizado para programar y estructurar los componentes del sistema.
* WordPress: plataforma empleada para diseñar y desarrollar la interfaz web del prototipo, permitiendo integrar las funciones principales de telemedicina de forma visual y accesible.
* Git y GitHub: herramientas para el control de versiones, registro de commits y almacenamiento del proyecto en un repositorio remoto.
* Trello: tablero Kanban digital utilizado para planificar y dar seguimiento a las tareas de cada sprint.
* CorelDRAW: software empleado para elaborar diagramas UML, de flujo y elementos gráficos del sistema.
* Photoshop: software empleado para elaborar wireframes del sistema.
* MysQl Workbench DER
* MySQL
* Microsoft Word: procesador de texto utilizado para la redacción del reporte técnico y la documentación de cada fase del proyecto.

# CAPÍTULO 5. ANÁLISIS Y DISEÑO

## 5.1 Requerimientos del sistema

El presente apartado describe los requerimientos que guían el desarrollo del sistema de telemedicina.

Estos requerimientos fueron identificados a partir de las necesidades planteadas en el caso de titulación, las entrevistas con los interesados y el análisis del proceso de atención médica remota.

### 5.1.1 Requerimientos funcionales

Los requerimientos funcionales representan las funcionalidades principales que el sistema debe cumplir. A continuación, se enlistan los requerimientos que se encuentran directamente relacionados con las Historias de Usuario (HU):

* **Registro y autenticación por correo electrónico** para paciente y médico (roles).
* **Perfil de usuario con datos mínimos:**
  + Paciente: nombre, CURP, número de teléfono y correo electrónico de contacto.
  + Médico: cédula profesional y especialidad.
  + Agenda de citas por médico y especialidad.
* **Formulario de pago y emisión de comprobante.**
* **Consentimiento informado:** aceptación previa a la consulta.
* **Historia clínica mínima**, con campos de antecedentes, alergias, medicamentos, signos o nota de evolución y archivos adjuntos limitados.
* **Opciones para comunicación sincrónica:** chat en vivo y videollamada durante la cita.
* **Recordatorios automáticos** por correo electrónico o SMS antes de la consulta.
* **Panel del médico:** agenda, acceso a historia clínica y notas de consulta, captura de diagnóstico y emisión de receta del paciente.

### 5.1.2 Requerimientos funcionales adicionales

Durante el proceso de análisis funcional y diseño del sistema, se identificaron requerimientos complementarios necesarios para la operación integral del prototipo.

Estos no contradicen el documento original, sino que amplían su alcance técnico para garantizar la funcionalidad completa del sistema.

* **Cancelación y reembolso de citas**: el paciente puede cancelar una cita y solicitar devolución del pago (modo de prueba sandbox).
* **Descarga de comprobante:** el paciente puede descargar el comprobante de pago desde su perfil.
* **Gestión de perfil:** tanto el paciente como el médico pueden actualizar su información personal.
* **Adjuntar archivos clínicos:** el médico puede subir documentos o estudios complementarios a la historia clínica del paciente.
* **Registro automático en bitácora (auditoría básica):** el sistema almacena fecha, usuario y acción de cada evento importante para trazabilidad.
* **Acceso diferenciado por rol:** el sistema restringe funcionalidades de acuerdo con el tipo de usuario (paciente o médico).

### 5.1.3 Requerimientos no funcionales

Los requerimientos no funcionales establecen condiciones de calidad y restricciones técnicas del sistema.

* Usabilidad: la interfaz debe ser intuitiva, clara y accesible desde navegadores comunes.
* Disponibilidad: el sistema debe estar disponible las 24 horas del día, salvo mantenimientos programados.
* Seguridad: se debe proteger la información médica mediante cifrado y autenticación de usuarios.
* Rendimiento: el tiempo de respuesta de las operaciones principales no debe superar los tres segundos.
* Compatibilidad: el sistema debe ser accesible desde computadoras y dispositivos móviles.
* Escalabilidad: debe ser posible agregar nuevos servicios médicos sin rediseñar la estructura.

## 5.2 Historias de Usuario

Las Historias de Usuario (HU) describen las funcionalidades del sistema desde la perspectiva de los actores involucrados.

Cada HU especifica una necesidad y su resultado esperado, y fueron definidas siguiendo el marco de trabajo Scrum, garantizando que cada funcionalidad refleje una necesidad real de los actores del sistema: paciente, médico y sistema automatizado.

### 5.2.1 Historias de Usuario Oficiales

A continuación, se muestran las cinco historias de usuario definidas originalmente para el sistema de telemedicina:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Historia de Usuario | Descripción | Criterios de Aceptación | Prioridad |
| HU-01 | Registro de paciente | Como paciente, quiero registrarme con mi correo y contraseña para agendar y dar seguimiento a mis consultas. | • El sistema valida el correo electrónico y la contraseña.  • Se envía confirmación de registro.  • El usuario puede iniciar sesión posteriormente. | Alta |
| HU-02 | Publicar disponibilidad | Como médico, quiero publicar mi disponibilidad por bloques para recibir reservas ordenadas. | • El médico puede definir fechas y horarios.  • Los pacientes solo pueden reservar en horarios disponibles.  • El calendario se actualiza en tiempo real. | Alta |
| HU-03 | Pago de consulta | Como paciente, quiero pagar de forma segura la consulta para confirmar mi cita y garantizar la atención. | • El sistema procesa el pago mediante sandbox.  • Se emite comprobante digital.  • Se actualiza la agenda del médico. | Alta |
| HU-04 | Consentimiento informado | Como paciente, quiero aceptar el consentimiento informado para comprender el alcance y riesgos del servicio. | • El documento se muestra antes de confirmar la cita.  • No se permite continuar sin aceptación.  • La aceptación queda registrada. | Alta |
| HU-05 | Comunicación sincrónica | Como médico y paciente, quiero acceder al chat o videollamada en la hora programada para realizar la consulta sin fricciones. | • El sistema habilita chat/video en el horario agendado.  • La conexión se establece correctamente.  • Se registra el inicio y fin de la sesión. | Alta |

***Tabla 5.1. Historias de Usuario oficiales***

### 5.2.2 Historias de Usuario Complementarias

Durante el análisis funcional y modelado del sistema se identificaron historias adicionales necesarias para garantizar la operación completa del prototipo y su coherencia con los requerimientos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Historia de Usuario | Descripción | Criterios de Aceptación | Prioridad |
| HU-06 | Historia clínica mínima | Como médico, quiero registrar antecedentes, alergias y medicamentos del paciente para mantener una historia clínica básica. | • El sistema muestra los campos requeridos.  • Solo el médico puede editar.  • Los datos quedan guardados por cita. | Alta |
| HU-07 | Recordatorios automáticos | Como sistema, quiero enviar recordatorios de cita al paciente antes de la consulta. | • El recordatorio se envía por correo/SMS 24h antes.  • Incluye nombre del médico, fecha y hora.  • Se registra el envío. | Media |
| HU-08 | Cancelación y reembolso | Como paciente, quiero cancelar una cita y solicitar reembolso cuando no pueda asistir. | • Solo puede cancelarse con ≥ 12 h de anticipación.  • Se actualiza la agenda del médico.  • Se genera reembolso (sandbox). | Media |
| HU-09 | Gestión de perfil | Como usuario, quiero actualizar mis datos personales para mantener mi información vigente. | • Los campos editables dependen del rol.  • El sistema valida formatos.  • Se confirman los cambios. | Media |
| HU-10 | Ver agenda del día | Como médico, quiero visualizar mis citas programadas para organizar mi jornada. | • El sistema muestra listado filtrado por fecha.  • Las citas pasadas se diferencian de las futuras.  • Permite acceder a detalles del paciente. | Alta |
| HU-11 | Descargar comprobante | Como paciente, quiero descargar mi comprobante de pago en cualquier momento. | • El comprobante se genera en PDF.  • Solo disponible para citas pagadas.  • Incluye fecha, monto y folio. | Media |
| HU-12 | Adjuntar archivos clínicos | Como médico, quiero adjuntar estudios o documentos clínicos a la historia del paciente. | • Se permiten archivos PDF/JPG/DOCX ≤ 10 MB.  • Se guardan con fecha y tipo.  • Solo el médico puede subirlos. | Media |
| HU-13 | Emitir receta médica | Como médico, quiero generar una receta digital con los medicamentos indicados. | • Se genera receta PDF con datos del médico.  • Se envía automáticamente al paciente.  • Se almacena en el historial. | Alta |
| HU-14 | Bitácora de auditoría | Como sistema, quiero registrar cada acción relevante para mantener trazabilidad. | • Cada evento guarda usuario, acción y fecha.  • Solo el administrador puede consultarla.  • No se permite modificación. | Alta |
| HU-15 | Cancelación automática por falta de pago | Como sistema, quiero cancelar una cita automáticamente si el pago no se completa. | • Si el pago no se confirma en sandbox, la cita se invalida.  • Se notifica al paciente por correo.  • El horario se libera. | Media |

***Tabla 5.2. Historias de Usuario complementarias***

### 5.2.3 Justificación de Historias de Usuario

Las historias de usuario oficiales cubren el alcance mínimo establecido en el caso de titulación. Sin embargo, durante el análisis se identificaron necesidades adicionales que permiten reflejar escenarios reales de uso, mejorar la trazabilidad del sistema y complementar el flujo operativo.

Por ello se incorporaron historias de usuario complementarias que fortalecen la continuidad del proceso médico, la interacción con el paciente y la administración del sistema. En conjunto, las 15 historias de usuario representan una visión integral del comportamiento esperado del prototipo.

## 5.3 Diagramas del Sistema

El análisis del sistema se complementa con una serie de diagramas que permiten comprender, desde diferentes perspectivas, el funcionamiento del prototipo de telemedicina. Estos diagramas ayudan a visualizar los actores involucrados, las interacciones del usuario, el flujo general de actividades y el alcance funcional del sistema antes de avanzar a su diseño detallado.

A continuación, se presentan los diagramas elaborados durante el proceso de análisis.

### 5.3.1 Diagrama de contexto

El diagrama de contexto muestra una vista general del sistema y su interacción con los actores principales: el paciente, el médico y los servicios externos necesarios para la operación del prototipo (por ejemplo, correo electrónico, mensajería SMS o pasarela de pago simulada).

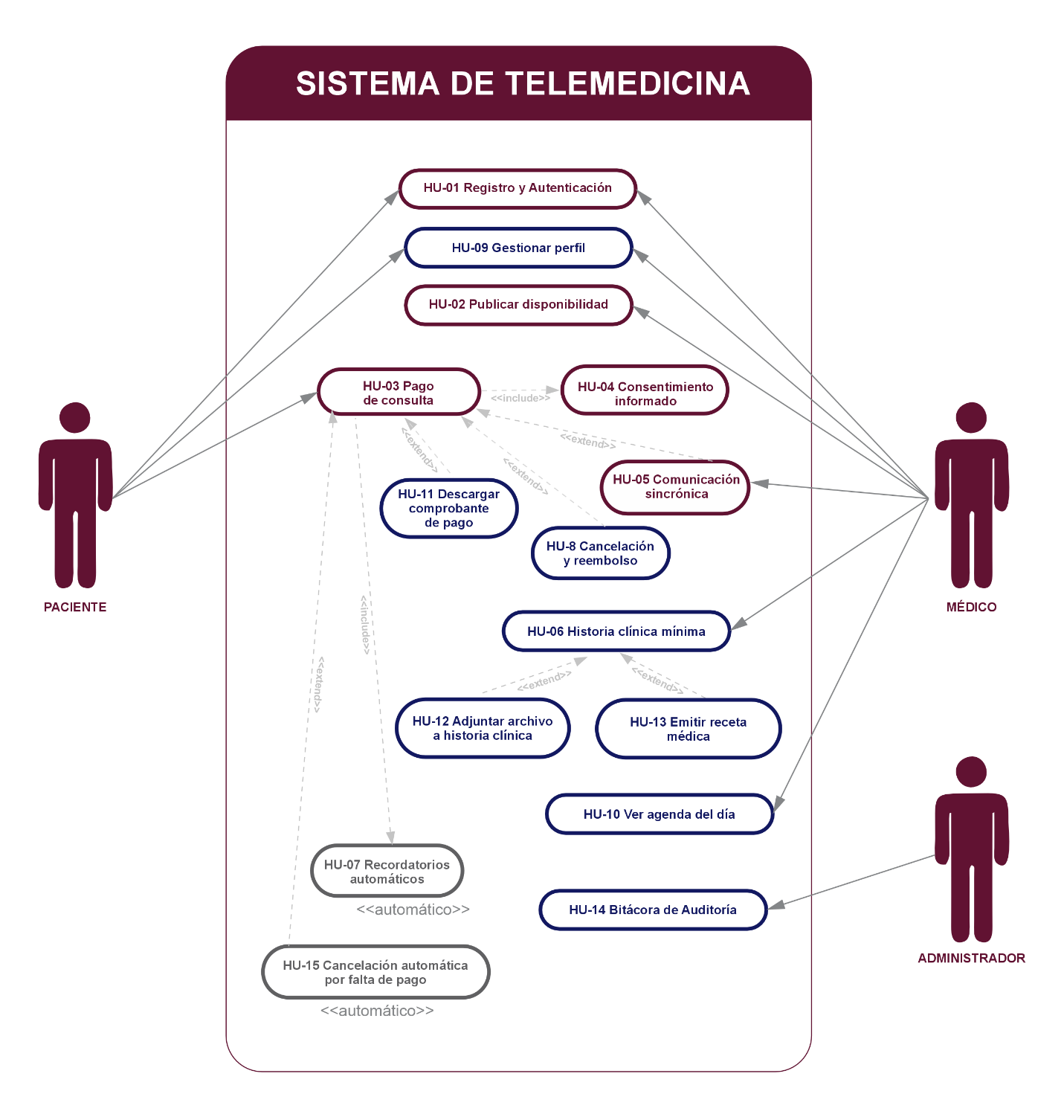
Este diagrama permite identificar los límites del sistema y los flujos de información que se generan en cada interacción.

***Figura 5.1 Diagrama de Contexto***

### 5.3.2 Diagrama de casos de uso

El diagrama de casos de uso representa las funcionalidades principales del sistema desde la perspectiva del usuario. En él se muestran los actores involucrados (paciente, médico y sistema automatizado) y las acciones que cada uno puede realizar, tales como registrarse, solicitar una cita, consultar la agenda, aceptar el consentimiento informado o acceder a la comunicación sincrónica.

Este diagrama permite visualizar el alcance funcional del prototipo y la relación entre los diferentes módulos del sistema.

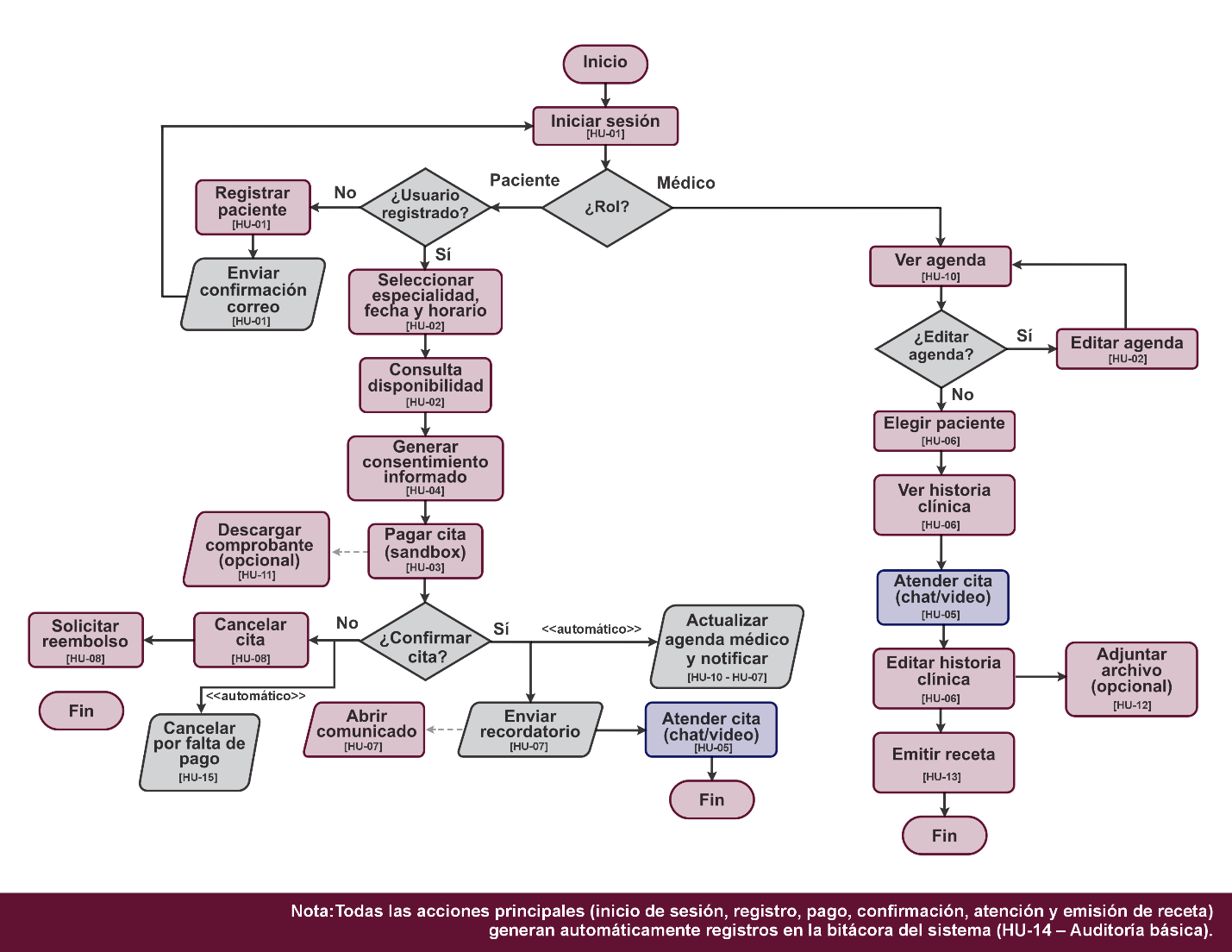


***Figura 5.2 Diagrama de Casos de Uso***

### 5.3.3 Diagrama de flujo del proceso general

El diagrama de flujo describe la secuencia lógica de actividades desde el momento en que el usuario accede al sistema hasta la finalización de una consulta médica. Incluye pasos como registro, reserva de cita, pago, consentimiento informado, atención por chat o videollamada y registro de la historia clínica.

Este diagrama ayuda a comprender el recorrido completo del usuario y la interacción entre módulos, facilitando la identificación de decisiones, entradas, salidas y puntos críticos del proceso.



***Figura 5.3 Diagrama de Flujo***

## 5.4. Modelo de Datos

El modelo de datos del prototipo de telemedicina fue diseñado siguiendo un enfoque relacional, con el objetivo de garantizar integridad, trazabilidad y consistencia en el manejo de la información médica, administrativa y operativa del sistema. Este modelo organiza los datos en entidades que representan a los usuarios, las citas, los pagos, la historia clínica y los elementos de comunicación, permitiendo una operación coherente con los requerimientos definidos en las historias de usuario.

### 5.4.1 Entidades principales

En la siguiente tabla, se describen las entidades principales y sus relaciones:

|  |  |  |
| --- | --- | --- |
| Entidad | Descripción | Relaciones principales |
| Usuario | Representa a todos los usuarios del sistema: pacientes, médicos y administradores. Incluye datos básicos de autenticación y contacto. | • 1:1 con Paciente o Médico  • 1:N con Log\_Auditoria |
| Paciente | Información específica del paciente: fecha de nacimiento, sexo y contacto de emergencia. | • N:1 con Usuario  •1:1 con Historia\_Clínica  • 1:N con Cita y Consentimiento |
| Médico | Información profesional del médico: cédula, especialidad y biografía. | • N:1 con Usuario  • 1:N con Disponibilidad y Cita |
| Disponibilidad | Bloques de tiempo que el médico habilita para citas. | • N:1 con Médico |
| Cita | Reserva entre paciente y médico. | • N:1 con Paciente y Médico  • 1:1 con Pago  • 1:N con Chat\_Mensaje y Video\_Sesion |
| Pago | Información del pago de la consulta. | • 1:1 con Cita |
| Consentimiento | Aceptación del consentimiento informado. | • N:1 con Paciente |
| Historia\_Clinica | Alergias, antecedentes, medicamentos y notas. | • 1:1 con Paciente  • 1:N con Adjunto |
| Adjunto | Archivos clínicos adicionales. | • N:1 con Historia\_Clinica |
| Chat\_Mensaje | Mensajes enviados durante la cita. | • N:1 con Cita |
| Video\_Sesion | Sesiones de videollamada. | • N:1 con Cita |
| Log\_Auditoria | Registro de acciones del usuario. | • N:1 con Usuario |

***Tabla 5.3. Relaciones principales del modelo de datos***

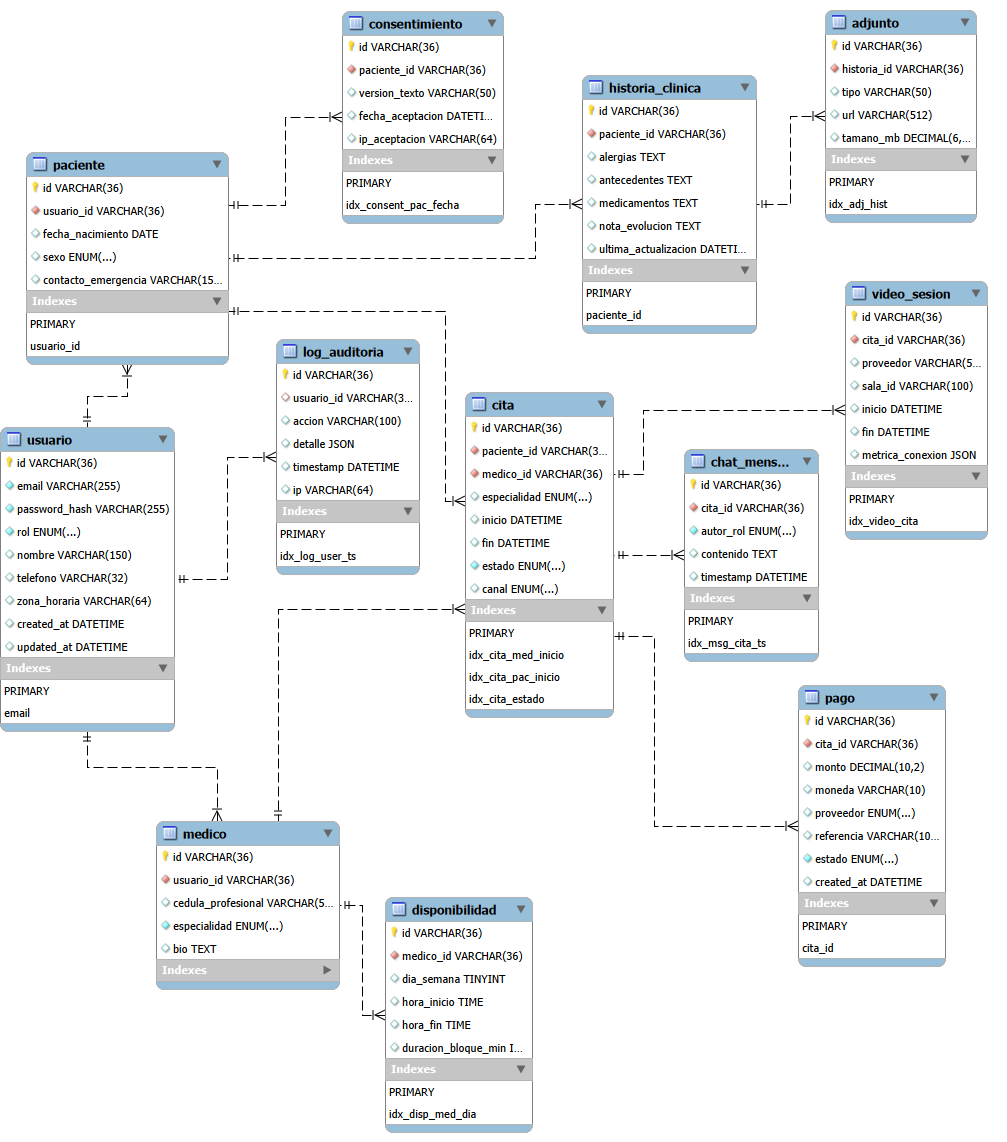
5.4.2 Relaciones Principales

La siguiente tabla resume las relaciones que estructuran el modelo y permiten el flujo correcto de información en el sistema:

|  |  |  |
| --- | --- | --- |
| Relación | Tipo | Descripción |
| Usuario ↔ Paciente | 1:1 | Cada usuario puede ser un paciente. |
| Usuario ↔ Médico | 1:1 | Cada usuario puede ser un médico. |
| Médico ↔ Disponibilidad | 1:N | Un médico publica varios bloques de disponibilidad. |
| Paciente ↔ Cita | 1:N | Un paciente puede generar múltiples citas. |
| Médico ↔ Cita | 1:N | Un médico atiende múltiples citas. |
| Cita ↔ Pago | 1:1 | Cada cita tiene un único registro de pago. |
| Paciente ↔ Consentimiento | 1:N | Cada firma del consentimiento se guarda como un registro separado. |
| Paciente ↔ Historia\_Clinica | 1:1 | Cada paciente tiene una única historia clínica. |
| Historia\_Clinica ↔ Adjunto | 1:N | La historia clínica puede contener varios archivos. |
| Cita ↔ Chat\_Mensaje | 1:N | Cada cita incluye varios mensajes. |
| Cita ↔ Video\_Sesion | 1:N | La cita puede incluir una o más sesiones de video. |
| Usuario ↔ Log\_Auditoria | 1:N | Cada acción importante realizada por el usuario queda registrada. |

### 5.4.3 Diagrama Entidad-Relación

El Diagrama Entidad–Relación representa de forma visual las entidades descritas y las relaciones entre ellas, permitiendo comprender la estructura general de la base de datos y su lógica interna.



***Figura 5.4 Diagrama Entidad-Relación***

## 5.5 Diseño de interfaces

El diseño de interfaces del prototipo de telemedicina tiene como propósito definir de manera clara y visual cómo interactúan los usuarios con el sistema.

Antes de implementar estas pantallas en WordPress, se elaboraron wireframes en Photoshop, lo que permite estandarizar criterios visuales, ahorrar trabajo y facilitar la programación posterior en WordPress y PHP. Cada subsección describe el objetivo de la pantalla, los campos involucrados, las reglas de validación y su relación con el flujo del sistema.

Es importante señalar que los wireframes presentados constituyen una propuesta inicial de diseño y funcionalidad. Su propósito es orientar la estructura visual y anticipar posibles interacciones del usuario. No obstante, durante la implementación se realizaron ajustes derivados de consideraciones técnicas, de usabilidad y de coherencia con las herramientas empleadas. Asimismo, algunos wireframes incluyen funciones que exceden los requisitos mínimos establecidos para el prototipo; dichas características fueron conceptualizadas para una fase posterior y no forman parte del alcance actual. Aun con estos ajustes, el diseño final mantiene el objetivo funcional previsto para cada pantalla.

### 5.5.1 Inicio de Sesión y Registro

El módulo de Inicio de Sesión y Registro constituye la puerta de entrada al sistema de telemedicina y permite que los usuarios accedan a los servicios según su rol correspondiente. Para este prototipo, el registro está habilitado únicamente para pacientes, mientras que los médicos son creados por el administrador del sistema.

|  |  |
| --- | --- |
| **LOGIN ESCRITORIO** | **MÓVIL** |

***Figura 5.5 Wireframes Login***

**Objetivo de las pantallas Login:**

Inicio de sesión: habilitar el acceso seguro a la plataforma mediante correo y contraseña.

**Campos generales:**

* + - * Correo electrónico
      * Contraseña

**Reglas de Validación:**

* Ambos campos son requeridos.
* Se verifica que el usuario exista y que la contraseña coincida con el hash registrado.

|  |  |
| --- | --- |
| **REGISTRO ESCRITORIO** | **MÓVIL** |

***Figura 5.6 Wireframes Registro***

**Objetivo de las pantallas Registro**

Registro de paciente: permitir que un usuario nuevo cree su cuenta proporcionando los datos generales y personales necesarios para su uso dentro del sistema.

**Elementos generales**

* Nombre completo
* Teléfono (10 dígitos)
* Correo electrónico
* Confirmar correo
* Contraseña
* Confirmar contraseña
* Fecha de nacimiento
* Sexo (F, M, Otro)
* Contacto de emergencia

**Reglas de validación:**

* Todos los campos son obligatorios.
* El correo debe tener formato válido y no estar registrado previamente.
* La contraseña debe tener un mínimo de 8 caracteres y coincidir con la confirmación.
* Se solicita la aceptación del aviso de privacidad.

### 5.5.2 Dashboard del Paciente

El dashboard es la pantalla principal del paciente después del inicio de sesión. Desde aquí accede a su agenda, especialidades disponibles y próximas citas.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.7 Wireframes Dashboard del Paciente***

**Objetivo**

Centralizar el acceso a todas las funciones clave del paciente: agenda de citas, especialidades, historial básico y perfil.

**Elementos generales**

* Nombre del paciente
* Tarjetas de especialidades
* Próximas citas
* Botón “Entrar a la sala” (si aplica)

**Reglas de validación**

* Mostrar solo citas futuras
* Las especialidades disponibles se basan en las habilitadas por el prototipo (mínimo 3)
* Botón “Entrar a la sala” solo aparece cuando la cita está Confirmada

### 5.5.3 Selección de Especialidad

La selección de especialidad se presenta como parte del panel principal del paciente. Desde aquí, el usuario elige entre las especialidades disponibles para iniciar el proceso de reserva de una cita.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.8 Wireframes Selección de Especialidad***

**Objetivo**

Permitir que el usuario elija una especialidad para continuar con el proceso de reserva.

**Elementos generales**

* Tarjetas de especialidades (General, Psicología, Cardiología)
* Botón “Agendar” dentro de cada tarjeta

**Reglas de validación**

* Listar solo especialidades habilitadas según el caso de titulación
* Al hacer clic → redirige a pantalla de “Seleccionar médico” filtrada por especialidad

### 5.5.4 Selección de Médico

Pantalla donde el paciente visualiza la lista de médicos disponibles únicamente para la especialidad seleccionada.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.9 Wireframes Selección de Médico***

**Objetivo**

Filtrar médicos por especialidad y permitir elegir uno para continuar con el proceso de cita.

**Elementos generales**

* Nombre del médico
* Especialidad
* Cédula profesional
* Correo
* Biografía

**Reglas de validación**

* Solo mostrar médicos que correspondan a la especialidad seleccionada
* Botón “Ver disponibilidad” abre la pantalla de horarios

### 5.5.5 Disponibilidad del Médico

Muestra los días y horarios disponibles del médico seleccionado.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.10 Wireframes Disponibilidad del Médico***

**Objetivo**

Permitir seleccionar un bloque horario disponible para agendar una cita.

**Campos generales**

* Fecha seleccionada
* Lista de bloques disponibles (inicio-fin)
* Etiquetas de estado (Disponible, Ocupado, No disponible)

**Reglas**

* Solo permitir seleccionar bloques con estado Disponible
* Validar que la fecha no sea anterior a la actual

### 5.5.6 Confirmación de Cita + Consentimiento

Modal que muestra el resumen de la cita seleccionada y el consentimiento informado.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.11 Wireframes Confirmación de Cita + Consentimiento***

**Objetivo**

Permitir al paciente revisar los datos de la cita y aceptar el consentimiento antes de proceder al pago.

**Campos generales**

* Médico
* Especialidad
* Fecha
* Hora
* Tipo de consulta
* Texto de consentimiento informado
* **Checkbox “Acepto el consentimiento informado”**

**Reglas**

* El botón “Continuar” se habilita solo si el consentimiento es aceptado
* Debe almacenarse la aceptación en BD (tabla Consentimiento)

### 5.5.7 Pago de Consulta (Sandbox)

Modal que permite ingresar datos de tarjeta en modo sandbox para simular la pasarela de pago.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.12 Wireframes Pago de Consulta (Sandbox)***

**Objetivo**

Procesar el pago simulado para confirmar la cita.

**Campos generales**

* Nombre del titular
* Número de tarjeta (demo)
* Fecha de vencimiento
* CVV
* Resumen de la cita
* Costo

**Reglas de validación**

* Campos obligatorios
* No es necesario validar formato real de tarjeta (modo sandbox)
* Tras pagar: Se crea el registro en Pago y la cita se marca como Confirmada

### 5.5.8 Cita Confirmada + Comprobante

Pantalla final que informa al paciente que su cita ha sido confirmada y permite descargar el comprobante en PDF.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.13 Wireframes Cita Confirmada + Comprobante***

**Objetivo**

* Mostrar detalles finales de la cita y permitir la descarga del comprobante.

**Campos generales**

* Médico
* Especialidad
* Fecha
* Hora
* Monto pagado
* Folio (ejemplo: TP-25-A9F3B2)

**Reglas**

* Descargar PDF genera un documento con los datos de la cita
* Botón “Ir al Panel” regresa al dashboard

### 5.5.9 Sala de Videollamada (Chat/Video)

Interfaz simulada de videollamada y chat. Usada por el paciente y el médico en la hora programada.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.14 Wireframes Sala de Videollamada***

**Objetivo**

* Permitir comunicación sincrónica durante la consulta médica (simulada).
* Campos / elementos
* Área de video
* Chat
* Botón para activar/desactivar video/micrófono
* Botón “Salir de la sala”

**Reglas**

* Disponible solo para citas Confirmadas
* Funcionalidad simulada

### 5.5.10 Dashboard del Médico

El Panel Médico es la pantalla principal a la que accede el profesional de la salud después de iniciar sesión. Este dashboard centraliza la información operativa del día, permitiendo visualizar rápidamente las citas próximas y acceder a los módulos principales del sistema.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.15 Wireframes Dashboard del Médico***

**Objetivo**

Mostrar al médico un resumen claro y ordenado de sus próximas consultas, incluyendo paciente, horario, estado de la cita y accesos directos a la sala de consulta y a la historia clínica.

**Elementos generales**

* Nombre del médico.
* Filtros de citas
* Tabla de próximas citas con:
  + Fecha
  + Hora
  + Nombre del paciente
  + Acceso a Historia Clínica
  + Acceso a Receta
  + Botón “Entrar a la Sala”
  + Botones inferiores:
  + Ver Agenda
  + Mis Pacientes

**Reglas de validación**

* Mostrar únicamente citas futuras, ordenadas por fecha y hora.
* El botón “Entrar a la Sala” se habilita sólo cuando la cita está Confirmada.
* La opción “Receta” únicamente aparece una vez que el paciente ha tenido al menos una consulta previa o un registro clínico mínimo.

### 5.5.11 Agenda del Médico

La agenda permite al médico administrar los bloques de disponibilidad semanales para recibir citas. El diseño se orienta a facilitar consultas rápidas y edición directa.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.16 Wireframes Agenda del Médico***

**Objetivo**

Permitir que el médico consulte y gestione sus horarios disponibles, agregando bloques nuevos o confirmando los ya existentes.

**Elementos generales**

* Selector semanal con botones: “Anterior” y “Siguiente”.
* Días de la semana del rango mostrado.
* Bloques existentes representados como etiquetas de color.
* En cada día:
  + Texto “Sin disponibilidad”, cuando aplica.
  + Botón “+ Agregar bloque”.

**Reglas de Validación**

* Sólo se pueden agregar bloques en días del rango actual.
* No se permiten bloques superpuestos.
* Los cambios deben reflejarse en tiempo real sobre el calendario utilizado en el flujo del paciente.

### 5.5.12 Mis Pacientes

Esta pantalla lista a todos los pacientes que el médico ha atendido o que tienen una cita activa con él.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.17 Wireframes Mis Pacientes***

**Objetivo**

Centralizar el acceso al historial de cada paciente atendido por el médico, permitiendo consultar historial, adjuntos y recetas.

**Elementos generales**

* Buscador por nombre del paciente.
* Tabla con:
  + Paciente
  + Última cita
  + Historia Clínica
  + Adjuntos
  + Recetas

**Reglas de validación**

* Ordenar por fecha de última cita, descendente.
* Mostrar únicamente pacientes relacionados con el médico actual.
* El botón “Historia Clínica” abre la vista detallada del paciente.

### 5.5.13 Historia Clínica del Paciente

La vista de historia clínica muestra la información clínica registrada del paciente, organizada en secciones editables.

Nota aclaratoria: La pantalla de Historia Clínica está disponible tanto para el paciente como para el médico.

* El paciente solo puede consultar su información.
* El médico puede consultar y editar alergias, antecedentes, medicamentos, notas y adjuntos.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.18 Wireframes Historia Clínica del Paciente***

**Objetivo**

Permitir al médico consultar y registrar datos relevantes del historial clínico del paciente, garantizando continuidad y trazabilidad en las consultas subsecuentes.

**Elementos y secciones**

* Datos generales del paciente
* Última cita
* Alergias
* Botón “Agregar alergia”
* Antecedentes médicos
* Botón “Agregar antecedente”
* Medicamentos actuales
* Botón “Agregar medicamento”
* Notas de evolución
* Botón “Agregar nota”
* Botón Adjuntos
* Botón Recetas
* Volver a Mis Pacientes

**Reglas de validación**

* Todos los cambios deben registrarse como parte de la historia clínica del paciente.
* El sistema almacena fecha y hora de cada anotación.
* El paciente sólo puede visualizar; el médico puede editar.

### 5.5.14 Modal: Agregar Nota de Evolución

Permite registrar observaciones clínicas del médico respecto a la consulta, quedando almacenadas en orden cronológico.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.19 Wireframes Modal Agregar Nota de Evolución***

**Objetivo**

Registrar hallazgos, observaciones, recomendaciones y seguimiento de la consulta.

**Campos**

* Fecha y hora (auto-generado)
* Nota de evolución (texto largo)

**Reglas de Validación**

* El campo debe contener al menos 10 caracteres.
* Se asocia a la historia del paciente y queda registrada por orden cronológico.

### 5.5.15 Modal: Agregar Alergia

Captura alergias relevantes del paciente para mantener advertencias médicas actualizadas

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.20 Wireframes Modal Agregar Alergia***

**Objetivo**

Registrar alergias relevantes comunicadas por el paciente.

**Campo**

Nombre de la alergia

**Reglas de validación**

* No permitir cadenas vacías.
* Registrar como item dentro del apartado Alergias.

### 5.5.16 Modal: Agregar Medicamento

Registra medicamentos actuales o prescritos, incluyendo nombre, dosis y frecuencia.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.21 Wireframes Modal Agregar Medicamento***

**Objetivo**

Registrar medicamentos actualmente prescritos o en uso.

**Campo**

Medicamento (incluye nombre, dosis y frecuencia)

**Reglas de validación**

* Formato libre pero requerido.
* Se añade a la lista de Medicamentos Actuales.

### 5.5.17 Modal: Agregar Antecedente

Añade antecedentes médicos importantes que den contexto a la condición del paciente.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.22 Wireframes Modal Agregar Antecedente***

**Objetivo**

Registrar antecedentes clínicos relevantes del paciente.

**Campo**

Antecedente (texto corto)

**Reglas de validación**

* No permitir cadenas vacías.
* Agregar a la sección Antecedentes Médicos.

### 5.5.18 Modal: Adjuntos

Permite subir y consultar archivos clínicos (PDF/imagen) asociados al historial del paciente.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.23 Wireframes Modal Adjuntos***

**Objetivo**

Permitir la carga y consulta de archivos clínicos complementarios como análisis, imágenes o estudios de laboratorio.

**Elementos**

* Tabla con adjuntos pasados
* Campo para subir archivo
* Botón “Subir”
* Botón “Guardar”

**Reglas de validación**

* Archivos permitidos: PDF, JPG, PNG.
* Tamaño máximo: 10 MB.
* Se guarda con sello de fecha y nombre original.

### 5.5.19 Recetas del Paciente

Pantalla modal que permite al médico consultar las recetas previas del paciente y generar nuevas.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.24 Wireframes Recetas del Paciente***

**Objetivo**

Emitir recetas digitales durante o después de una consulta, entregando un archivo descargable al paciente.

**Elementos**

* Tabla con recetas previas
* Sección para “Emitir nueva receta”

**Reglas de validación**

* Todos los campos del área de texto son obligatorios.
* Al guardar, el sistema genera un PDF que queda disponible para descarga inmediata.
* El paciente puede visualizarla desde su portal.

### 5.5.20 Mi Perfil

El médico visualiza información profesional necesaria para la atención.

El paciente visualiza sus datos personales básicos.

Aclaración: La pantalla de perfil existe tanto para paciente como para médico.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.25 Wireframes Mi Perfil***

**Objetivo**

Permitir que el médico consulte los datos principales de su perfil profesional.

**Elementos**

* Nombre
* Especialidad
* Correo electrónico
* Teléfono
* Botón “Editar Perfil”

### 5.5.21 Editar Perfil

Pantalla donde el médico puede modificar los campos permitidos de su perfil.

|  |  |
| --- | --- |
| **ESCRITORIO** | **MÓVIL** |

***Figura 5.26 Wireframes Editar Perfil***

**Objetivo**

Actualizar la información de contacto y datos profesionales del médico.

**Campos editables**

* Nombre
* Especialidad
* Correo electrónico
* Teléfono

**Reglas de Validación**

* El correo debe tener formato válido.
* Teléfono debe tener entre 10 y 12 dígitos.
* La especialidad debe coincidir con las habilitadas en el sistema.

# 6. IMPLEMENTACIÓN

Esta sección describe cómo se llevó a cabo la construcción del prototipo funcional. El enfoque del desarrollo fue mantener una estructura sencilla, comprensible y fácil de probar, sin perder de vista los requerimientos principales del caso.

Para esto, se combinó el uso de un sitio en WordPress, encargado de las pantallas y la experiencia visual (frontend), con un backend desarrollado en PHP que se comunica directamente con la base de datos MySQL alojada en el mismo servidor del dominio.

Esta separación entre vista y lógica permitió trabajar de manera ordenada y mantener un mejor control sobre cada funcionalidad del sistema.

A continuación, se describen las tecnologías utilizadas y la arquitectura que integra todos los componentes del prototipo.

## 6.1 Tecnologías y herramientas utilizadas

Para desarrollar el prototipo se emplearon herramientas accesibles y ampliamente utilizadas en el desarrollo web, buscando un equilibrio entre simplicidad, funcionalidad y facilidad de mantenimiento. Las principales tecnologías fueron las siguientes:

* WordPress: Plataforma utilizada para construir la interfaz visual del prototipo. Permite crear páginas, formularios y pantallas de usuario de manera rápida mediante Elementor y bloques personalizados, manteniendo un diseño limpio y uniforme.
* PHP: Lenguaje principal del backend. Los procesos clave del sistema —registro, inicio de sesión, disponibilidad del médico, reserva de cita, consentimiento informado, historia clínica y chat simulado— se implementaron en archivos PHP ubicados en la carpeta /api dentro del servidor del proyecto.
* MySQL: Base de datos alojada directamente en el servidor del dominio telemedproyecto.com. En ella se almacenan entidades como usuarios, médicos, pacientes, citas, pagos simulados, consentimiento, historia clínica, mensajes de chat y auditoría.
* JavaScript: Utilizado en la capa de presentación para validaciones rápidas, dinámicas visuales y envío de datos hacia la API mediante AJAX cuando fue necesario.
* MySQL Workbench: Aplicación utilizada para administrar la base de datos, realizar consultas, revisar relaciones e insertar el dataset oficial solicitado en el caso de titulación.
* Visual Studio Code: Editor de código empleado para la organización y desarrollo de los archivos PHP, JavaScript y estructura del backend.
* cPanel y File Manager: Herramientas del hosting utilizadas para gestionar las carpetas del dominio, cargar archivos y administrar la estructura del proyecto.
* FileZilla: Cliente FTP usado para sincronizar la carpeta /api del backend con el servidor, facilitando la actualización de archivos durante el desarrollo.

## 6.2 Arquitectura del Sistema

El prototipo emplea una arquitectura modular dividida en capas, lo cual permite mantener el código organizado, facilitar pruebas y mejorar el mantenimiento general del sistema. Cada componente cumple una función específica dentro del flujo de funcionamiento de la plataforma.

### 6.2.1 Capa de Presentación (Frontend en WordPress)

La capa de presentación está construida en WordPress mediante plantillas y bloques generados con Elementor. En esta capa se muestran todas las vistas con las que interactúan pacientes y médicos: inicio de sesión, registro, paneles, disponibilidad, reserva de cita, consentimiento informado, historia clínica y chat simulado.

Su función es únicamente capturar datos y mostrarlos, sin ejecutar lógica de negocio.

### 6.2.2 Capa de Lógica de Negocio (API en PHP)

Cada endpoint PHP procesa una función específica del prototipo: registro de pacientes, autenticación, gestión de disponibilidad, reservas, pagos simulados, consentimiento informado, historia clínica, adjuntos, chat y bitácora de auditoría.

Esta capa recibe solicitudes provenientes del frontend, valida datos, ejecuta reglas y define el comportamiento del sistema antes de interactuar con la base de datos.

### 6.2.3 Capa de Acceso a Datos

Esta capa está compuesta por funciones PHP que se encargan exclusivamente de comunicarse con la base de datos MySQL.

Sus responsabilidades principales son:

* Establecer la conexión con la base de datos.
* Ejecutar operaciones SELECT, INSERT, UPDATE y DELETE.
* Manejar excepciones y errores.
* Regresar datos limpios y seguros a la capa de lógica.

Con esto se evita mezclar consultas SQL dentro de la lógica de negocio, lo que mejora la seguridad y mantiene el código organizado.

### 6.2.4 Capa de Datos (MySQL en hosting)

La base de datos MySQL se encuentra alojada en el mismo servidor que hospeda el dominio telemedproyecto.com. En ella se almacenan las entidades principales definidas en el caso: Usuario, Paciente, Médico, Disponibilidad, Cita, Pago, Consentimiento, Historia Clínica, Adjuntos, ChatMensaje, VideoSesion y LogAuditoria.

La administración se realiza desde MySQL Workbench, lo que permite aplicar consultas, revisar relaciones y cargar el dataset oficial del caso.

### 6.2.5 Comunicación entre capas

El funcionamiento del prototipo sigue un flujo simple:

* El usuario interactúa con una página en WordPress.
* La página envía los datos a un endpoint PHP mediante formulario o AJAX.
* La API valida la información y ejecuta la lógica correspondiente.
* La capa de acceso a datos realiza las consultas necesarias en MySQL.
* El resultado regresa a la API, que genera una respuesta para WordPress.
* WordPress muestra el resultado al usuario.

Este flujo mantiene separadas las responsabilidades de cada capa, haciendo que el sistema sea más claro y fácil de mantener.

### 6.2.6 Uso de JavaScript en la capa de presentación

Para mejorar la interacción del usuario, se integró JavaScript principalmente en:

* Validación visual de formularios
* Mostrar/ocultar secciones dinámicas
* Respuestas visuales según la API
* Envío de peticiones AJAX cuando fue necesario

JavaScript se mantiene en la capa de presentación y no forma parte de la lógica de negocio.

## 6.3 Implementación de la base de datos en el servidor

Antes de implementar la base de datos, fue necesario preparar el entorno donde operaría el prototipo. Para ello se habilitó el dominio telemedproyecto.com dentro del servicio de hosting contratado en HostGator. Desde dicho panel se creó el espacio de archivos del sitio, la carpeta pública del dominio y los recursos necesarios para alojar tanto WordPress como la carpeta /api destinada al backend.

Esto permitió contar con un entorno accesible en línea, con capacidad para interactuar con la base de datos remota y recibir solicitudes desde la interfaz visual.

La base de datos del prototipo se implementó directamente en este servidor, con base en el modelo entidad–relación desarrollado en el capítulo 5. Se creó un esquema MySQL y se habilitó un usuario con permisos adecuados para conexiones remotas desde MySQL Workbench, herramienta utilizada durante el desarrollo para revisar la estructura, aplicar consultas y administrar la información.

La estructura inicial de tablas se definió en un script SQL propio (Telemedicina\_Camel.sql). En él se declararon explícitamente las tablas, tipos de datos, claves primarias, llaves foráneas e índices. A continuación, se muestra un fragmento representativo del archivo:

CREATE TABLE usuario (

idUsuario VARCHAR(36) PRIMARY KEY,

email VARCHAR(255) NOT NULL UNIQUE,

passwordHash VARCHAR(255) NOT NULL,

rol ENUM('PACIENTE','MEDICO','ADMIN') NOT NULL,

nombre VARCHAR(150),

telefono VARCHAR(32),

zonaHoraria VARCHAR(64),

createdAt DATETIME,

updatedAt DATETIME

);

ALTER TABLE paciente

ADD CONSTRAINT fk\_Paciente\_Usuario

FOREIGN KEY (idUsuario) REFERENCES usuario(idUsuario)

ON UPDATE CASCADE;

Una vez creada la estructura, se importó un conjunto de datos de demostración mediante el archivo dataset\_demo.sql, el cual incluye usuarios, médicos, pacientes, citas, historia clínica, mensajes de chat, adjuntos, pagos simulados y registros de auditoría necesarios para reproducir escenarios del caso.

## 6.4 Conjunto de datos de demostración

Para realizar pruebas funcionales sin depender de la creación manual de registros, se elaboró el archivo dataset\_demo.sql. Este archivo contiene los datos necesarios para validar los flujos principales del prototipo, como registro, inicio de sesión, reserva de cita, visualización de historia clínica y chat simulado.

A continuación, se presentan fragmentos representativos del dataset:

INSERT INTO usuario (idUsuario, email, passwordHash, rol, nombre)

VALUES

('u1', 'paciente1@example.com', 'demoHash1', 'PACIENTE', 'Ana Pérez'),

('u2', 'paciente2@example.com', 'demoHash2', 'PACIENTE', 'Luis Gómez');

INSERT INTO cita (idCita, idPaciente, idMedico, especialidad, inicio, fin, estado)

VALUES

('c1', 'p1', 'm1', 'General', '2025-11-26 10:00:00', '2025-11-26 10:30:00', 'Confirmada');

El archivo completo se encuentra en la carpeta /db del proyecto dentro del repositorio, permitiendo reproducir el mismo entorno de pruebas.

6.5 Conexión a la Base de Datos

Para establecer la comunicación entre la API y la base de datos MySQL alojada en HostGator, se creó un archivo central llamado config.php, que se encuentra dentro de la carpeta /api.

Este archivo utiliza PHP Data Objects (PDO) para manejar la conexión de forma segura y evitar el uso de funciones obsoletas.

A continuación, se muestra el fragmento del código utilizado:

<?php

$DB\_HOST = "mx58.hostgator.mx";

$DB\_NAME = "amartimx\_telemed";

$DB\_USER = "amartimx\_teleuser";

$DB\_PASS = "\*\*\*\*\*\*\*\*\*\*"; //Contraseña protegida

try {

$conn = new PDO(

"mysql:host=$DB\_HOST;port=3306;dbname=$DB\_NAME;charset=utf8",

$DB\_USER,

$DB\_PASS

);

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

die("Error de conexi贸n: " . $e->getMessage());

}

El archivo config.php se incluye en todos los endpoints para asegurar que cualquier operación (crear usuario, iniciar sesión, publicar disponibilidad, etc.) utilice siempre la misma conexión centralizada.

## 6.6 Prueba inicial de conexión

Para comprobar que la API del sistema se comunicaba correctamente con la base de datos, se creó el archivo test\_db.php, el cual ejecuta una consulta simple sobre la tabla usuario.

Código utilizado:

<?php

require\_once "config.php";

try {

$stmt = $conn->query("SELECT idUsuario, nombre FROM usuario LIMIT 5");

$usuarios = $stmt->fetchAll(PDO::FETCH\_ASSOC);

echo "<h2>Conexión exitosa✔️</h2>";

echo "<pre>";

print\_r($usuarios);

echo "</pre>";

} catch (Exception $e) {

echo "Error en consulta: " . $e->getMessage();

}

Al ejecutar: <https://telemedproyecto.com/api/test_db.php> el sistema mostró la leyenda *“Conexión exitosa ✔️”*, confirmando que la API estaba lista para comenzar a recibir y procesar datos.

### 6.7 Implementación del registro de pacientes (HU-01)

El proceso de registro de pacientes constituye el primer punto de interacción directa con el sistema. Esta funcionalidad se implementó durante el Sprint 2 y corresponde al flujo en el que un nuevo usuario proporciona sus datos básicos para crear su cuenta dentro del prototipo.

La página Registrar Paciente se desarrolló en WordPress y se encuentra disponible en:

https://telemedproyecto.com/registrar-paciente/

En esta página se incluyó un formulario construido mediante un bloque HTML, siguiendo el diseño propuesto en los wireframes del capítulo 5.

El formulario solicita únicamente la información necesaria para crear un nuevo usuario y su respectivo registro como paciente: nombre, teléfono, fecha de nacimiento, sexo, correo electrónico y contraseña. Las confirmaciones de correo y contraseña permiten validar la coherencia de los datos antes de enviarlos.

Al enviar el formulario, la información es procesada por el endpoint:

/api/registro\_paciente.php

Este archivo valida los datos recibidos, verifica que el correo no se encuentre previamente registrado y posteriormente crea los registros correspondientes en las tablas usuario y paciente.

El siguiente extracto muestra una parte representativa del código encargado de procesar el registro. En este fragmento se aprecia el uso de encriptación de contraseñas y la inserción del nuevo usuario en la tabla *usuario* empleando consultas preparadas de PDO, lo que contribuye a una comunicación segura con la base de datos.

// Encriptar la contraseña antes de guardarla

$passwordHash = password\_hash($password, PASSWORD\_DEFAULT);

// Inserción del usuario en la base de datos

$sql = "INSERT INTO usuario (

idUsuario, email, passwordHash, rol,

nombre, telefono, zonaHoraria, createdAt, updatedAt

) VALUES (

:idUsuario, :email, :passwordHash, 'PACIENTE',

:nombre, :telefono, :zonaHoraria, :createdAt, :updatedAt

)";

$stmt = $conn->prepare($sql);

$stmt->execute([

':idUsuario' => $idUsuario,

':email' => $email,

':passwordHash' => $passwordHash,

':nombre' => $nombre,

':telefono' => $telefono,

':zonaHoraria' => $zonaHoraria,

':createdAt' => $ahora,

':updatedAt' => $ahora

]);

Una vez finalizado el proceso, el sistema notifica al usuario que su cuenta ha sido creada exitosamente, quedando habilitado para iniciar sesión. Esta funcionalidad establece los cimientos para los flujos posteriores de autenticación y gestión de perfiles.



***Figura 6.1 Registro Exitoso***

## 6.8 Implementación del inicio de sesión y gestión de roles

El módulo de inicio de sesión constituye la puerta de acceso al prototipo y permite diferenciar las funcionalidades disponibles para pacientes, médicos y administradores. Su implementación se realizó combinando una interfaz construida en WordPress mediante Elementor, con un endpoint desarrollado en PHP que se conecta directamente a la base de datos del sistema.

### 6.8.1 Página de inicio de sesión

La interfaz de inicio de sesión se diseñó con base en los wireframes establecidos durante el Sprint 2, manteniendo una estructura sencilla para facilitar la autenticación del usuario. El formulario solicita dos campos principales:

* Correo electrónico
* Contraseña

Cuando el usuario presiona el botón Iniciar sesión, el formulario envía la información al backend mediante una solicitud POST hacia el endpoint /api/login.php.

El diseño visual de esta pantalla se integró en WordPress utilizando Elementor como constructor, permitiendo mantener coherencia con la apariencia general del prototipo.

### 6.8.2 Endpoint de autenticación (/api/login.php)

El proceso de autenticación se implementó a través de un endpoint que valida las credenciales contra la tabla usuario de la base de datos.

* Este endpoint realiza las siguientes tareas:
* Valida que los datos obligatorios hayan sido enviados.
* Consulta el correo electrónico en la base de datos.
* Verifica la contraseña mediante el hash almacenado.
* Inicia una sesión segura mediante $\_SESSION.
* Identifica el rol del usuario (PACIENTE, MEDICO o ADMIN).
* Regresa al frontend una respuesta en formato JSON con el resultado.

Si las credenciales son correctas, el servidor indica cuál es la ruta correspondiente al rol del usuario. Esta respuesta es recibida por JavaScript en WordPress, el cual realiza la redirección.

Este enfoque permite mantener separada la vista (WordPress) de la lógica (PHP), respetando la arquitectura de capas definida para el proyecto.

|  |  |
| --- | --- |
|  |  |

***Figura 6.2 Pruebas de Autenticación***

### 6.8.3 Redirección por rol

Una vez autenticado, el sistema dirige al usuario hacia el panel que le corresponde:

* Paciente → /dashboard-paciente
* Médico → /dashboard-medico
* Administrador → /panel-admin

Esta segmentación garantiza que cada tipo de usuario ingrese únicamente a las opciones que le pertenecen según su rol dentro del sistema de telemedicina.

### 6.8.4 Protección de páginas privadas

Para evitar accesos no autorizados a los paneles internos, se implementó un archivo de protección llamado proteger.php, ubicado dentro de la carpeta /api/.

Este módulo verifica que:

* El usuario tenga una sesión activa.
* El rol coincida con la sección a la que intenta acceder.

Si alguna condición no se cumple, el usuario es redirigido automáticamente a la página de inicio de sesión.

Las páginas internas del sistema (dashboard paciente, dashboard médico y panel de administrador) fueron vinculadas a plantillas PHP personalizadas dentro del tema activo de WordPress. Cada plantilla incluye la verificación correspondiente antes de mostrar el contenido generado por Elementor.

Esto permite combinar seguridad a nivel de servidor con la flexibilidad visual del constructor de páginas.

### 6.8.5 Cierre de sesión

También se creó el endpoint /api/salir.php, cuya función es cerrar la sesión del usuario y regresarlo al inicio de sesión. Esta opción aparece dentro del menú lateral de los paneles internos.

### 6.8.6 Pruebas funcionales realizadas

Para validar la correcta operación del módulo de autenticación, se realizaron las siguientes pruebas:

* Inicio de sesión exitoso con un paciente registrado.
* Inicio de sesión exitoso con un médico creado por el administrador.
* Acceso denegado a paneles que no correspondan al rol del usuario.
* Acceso denegado cuando no existe una sesión activa.
* Redirección automática después de iniciar sesión según el rol.
* Cierre de sesión y bloqueo de rutas privadas al intentar acceder nuevamente.

Los resultados confirmaron que la gestión de autenticación opera de forma correcta y segura dentro del prototipo.

## 6.9 Implementación de la Agenda Médica (HU-02)

La Agenda Médica corresponde a la funcionalidad mediante la cual el médico define los horarios en los que puede ofrecer consultas. Esta sección forma parte del panel del médico y se desarrolló siguiendo estrictamente la estructura visual presentada en los wireframes del capítulo 5 (pantallas “Agenda médica”). La implementación se basó en la combinación de WordPress como capa visual y PHP como backend, respetando la arquitectura modular descrita previamente.

Esta funcionalidad depende directamente de la tabla disponibilidad de la base de datos, definida en el modelo entidad–relación del prototipo, donde se registran los bloques horarios que cada médico habilita para recibir consultas.

### 6.9.1 Página Agenda Médica

La pantalla de Agenda Médica se construyó dentro del panel del médico, utilizando Elementor para recrear la estructura presentada en los wireframes.

Siguiendo el diseño, la página contiene:

* Un encabezado con el título “Agenda médica”.
* Una tabla donde se muestran los bloques de disponibilidad del médico.
* Un enlace tipo botón con la opción “Agregar bloque”.
* Un formulario oculto que se despliega al activar dicha opción.

El formulario se implementó mediante un bloque HTML dentro de Elementor, y contiene únicamente los campos necesarios para completar la tabla disponibilidad:

* Día de la semana
* Hora de inicio (TIME)
* Hora de fin (TIME)
* Duración del bloque (INT, minutos)

La tabla de la Agenda se actualiza de forma automática: cuando la página se carga, un script consulta al servidor los bloques de disponibilidad del médico y llena la tabla con esa información, sin necesidad de recargar o editar manualmente la página.

### 6.9.2 Endpoint para registrar disponibilidad

El registro de bloques se implementó en el archivo: /api/guardar\_disponibilidad.php

Este endpoint:

* Verifica que el usuario tenga sesión activa y rol MEDICO.
* Valida que los datos del formulario sean correctos (campos obligatorios, horas válidas y duración mayor que cero).
* Obtiene el idMedico correspondiente al usuario autenticado.
* Inserta el bloque en la tabla disponibilidad.
* Registra la operación en la tabla logauditoria.

El siguiente fragmento muestra una parte representativa del proceso de inserción:

$sqlMedico = $conn->prepare("SELECT idMedico FROM medico WHERE idUsuario = ?");

$sqlMedico->execute([$idUsuario]);

$medico = $sqlMedico->fetch(PDO::FETCH\_ASSOC);

$idDisponibilidad = uniqid("disp\_", true);

$insertar = $conn->prepare("

INSERT INTO disponibilidad (idDisponibilidad, idMedico, diaSemana, horaInicio, horaFin, duracionBloqueMin)

VALUES (?, ?, ?, ?, ?, ?)

");

$insertar->execute([

$idDisponibilidad,

$idMedico,

$diaSemana,

$horaInicio,

$horaFin,

$duracion

]);

### 6.9.3 Registro en bitácora de auditoría

Toda acción realizada en el sistema debe quedar registrada según los requerimientos del caso. Por ello, se añadió un registro automático para cada bloque creado por el médico, almacenando:

* idUsuario
* Acción realizada (CREAR\_DISPONIBILIDAD)
* Detalle en formato JSON
* IP del usuario
* Fecha y hora

Este registro permite rastrear acciones críticas en el sistema y se integra con el resto de eventos de la plataforma (registro, autenticación, consultas, etc.).

### 6.9.4 Endpoint para mostrar la disponibilidad en la agenda

Para que el médico pueda ver los horarios que ya registró, se creó el endpoint:

/api/listar\_disponibilidad.php.

Este archivo obtiene de la base de datos todos los bloques de disponibilidad del médico autenticado y los entrega en formato JSON.

Al abrir la página de Agenda Medica, un pequeño script en JavaScript pide esa información al servidor y la usa para llenar automáticamente la tabla. Lo mismo ocurre cada vez que el médico agrega un nuevo bloque, por lo que la tabla se actualiza sin necesidad de recargar toda la página.

Fragmento representativo:

$consulta = $conn->prepare("

SELECT diaSemana, horaInicio, horaFin, duracionBloqueMin

FROM disponibilidad

WHERE idMedico = ?

ORDER BY diaSemana, horaInicio

");

### 6.9.5 Pruebas funcionales realizadas

Las pruebas para validar esta funcionalidad incluyeron:

* Acceso a la Agenda médica únicamente por usuarios con rol MÉDICO.
* Registro exitoso de bloques de disponibilidad con validación de horarios.
* Visualización inmediata de los bloques registrados en la tabla.
* Verificación de que los bloques de disponibilidad se almacenan correctamente en la BD.
* Registro automático de cada operación en logauditoria.
* Manejo adecuado de errores (horarios inválidos, campos vacíos, acceso no autorizado).

Las pruebas confirmaron que la Agenda Médica permite registrar y consultar disponibilidad de forma segura y coherente con la estructura del sistema.

|  |  |
| --- | --- |
|  |  |

***Figura 6.3 Pruebas de Agenda Médica***

## 6.10 Validación del comportamiento del backend (1° etapa)

En esta etapa se revisaron los endpoints implementados durante el Sprint 2:

* **Registro de paciente.** Se verificó la correcta inserción en la base de datos y el manejo adecuado de contraseñas mediante algoritmos de hash.
* **Autenticación.** Se confirmó que el proceso de login gestionara la sesión de manera segura y asignara correctamente el rol correspondiente, evitando accesos no autorizados.
* **Disponibilidad del médico.** Se evaluó la inserción de bloques de disponibilidad y la recuperación de estos a través de consultas AJAX, así como la integración con la tabla logauditoria, donde se registran las acciones relevantes.

Los endpoints mostraron un funcionamiento estable y coherente con el modelo de datos, sin errores de sintaxis, consultas inválidas o respuestas inesperadas.

### 6.10.1 Ajustes adicionales

Durante la validación se detectaron algunos comportamientos relacionados con la caché del servidor que permitían visualizar temporalmente contenido privado después de cerrar sesión. Este problema se resolvió ajustando la configuración de caché del hosting (Endurance Cache), además de agregar cabeceras de control de caché en el archivo proteger.php. Con ello se garantizó que las páginas protegidas no puedan ser mostradas a un usuario sin sesión activa.

Asimismo, se realizaron ajustes menores en las plantillas de WordPress para evitar duplicación de contenido y asegurar que únicamente Elementor se encargara de renderizar el nombre del usuario dentro del dashboard.

## 6.11 Implementación de la reserva de cita, consentimiento informado y pago simulado (HU-03 y HU-04)

Esta sección describe en detalle la implementación de las historias de usuario relacionadas con el proceso que sigue un paciente para reservar una cita, aceptar el consentimiento informado y realizar el pago simulado correspondiente.

Este conjunto de funcionalidades integra varios elementos del sistema: consultas de disponibilidad, registro formal de aceptación del consentimiento, creación de la cita, registro del pago y emisión de un comprobante final.

Todo el flujo se desarrolló respetando estrictamente la estructura de la base de datos oficial, sin añadir tablas ni campos adicionales.

### 6.11.1 Selección de especialidad y consulta de médicos

El proceso inicia cuando el paciente ingresa al módulo Agendar Cita. Desde esta pantalla, el usuario selecciona la especialidad deseada y el sistema debe mostrar a los médicos que pertenecen a dicha área.

Para ello se utiliza el endpoint:

/api/get\_medicos.php

Este endpoint tiene como objetivo consultar la base de datos y devolver la lista de médicos cuya especialidad coincide con el parámetro recibido. La información se obtiene combinando las tablas medico y usuario.

Fragmento representativo:

// Extracto del endpoint get\_medicos.php

$especialidad = $\_GET["especialidad"] ?? "";

$sql = $conn->prepare("

SELECT m.idMedico, u.nombre, u.email, m.especialidad, m.cedulaProfesional, m.bio

FROM medico m

INNER JOIN usuario u ON u.idUsuario = m.idUsuario

WHERE m.especialidad = ?

");

$sql->execute([$especialidad]);

$medicos = $sql->fetchAll(PDO::FETCH\_ASSOC);

echo json\_encode([

"ok" => true,

"medicos" => $medicos

]);

Esta respuesta es interpretada por la vista de WordPress mediante JavaScript, generando dinámicamente tarjetas con los datos de cada médico. Esto permite que la pantalla respete el diseño de los wireframes y que pueda reaccionar a la selección del usuario sin recargar la página.

### 6.11.2 Consulta de disponibilidad del médico

Una vez seleccionado un médico, la pantalla de Disponibilidad ejecuta otro endpoint:

/api/get\_disponibilidad.php

Este endpoint tiene como finalidad consultar la tabla disponibilidad y devolver los bloques horarios organizados por día de la semana. La tabla estructura la disponibilidad en función del día (0–6), hora de inicio, hora de fin y duración del bloque.

Fragmento representativo:

// Extracto del endpoint get\_disponibilidad.php

$idMedico = $\_GET["idMedico"] ?? "";

$stmt = $conn->prepare("

SELECT idDisponibilidad, diaSemana, horaInicio, horaFin, duracionBloqueMin

FROM disponibilidad

WHERE idMedico = ?

ORDER BY diaSemana, horaInicio

");

$stmt->execute([$idMedico]);

$bloques = $stmt->fetchAll(PDO::FETCH\_ASSOC);

echo json\_encode([

"ok" => true,

"bloques" => $bloques

]);

En la pantalla, cada bloque se transforma en un botón seleccionable. La construcción dinámica permite mantener el diseño del wireframe, donde los horarios se distribuyen por día y reaccionan a la interacción del usuario:

bloques.forEach(b => {

const boton = document.createElement("button");

boton.textContent = b.horaInicio + " - " + b.horaFin;

boton.onclick = () => seleccionarHorario(b);

dias[b.diaSemana].appendChild(boton);

});

La coherencia entre backend y frontend permite que la selección de horario active el modal de confirmación.



***Figura 6.4 Disponibilidad del Médico***

### 6.11.3 Registro del consentimiento informado (HU-04)

El consentimiento informado es un requisito que debe cumplirse antes de reservar una cita. En la interfaz, esto se implementó mediante un modal que solicita al paciente leer el documento y marcar una casilla de aceptación.

Una vez marcado, el sistema ejecuta el endpoint:

/api/registrar\_consentimiento.php

Este endpoint registra la aceptación en la tabla consentimiento, siguiendo la estructura oficial.

Fragmento representativo:

// Extracto del endpoint registrar\_consentimiento.php

$insert = $conn->prepare("

INSERT INTO consentimiento (idConsentimiento, idPaciente, versionTexto, fechaAceptacion, ipAceptacion)

VALUES (?, ?, ?, NOW(), ?)

");

$insert->execute([$idConsentimiento, $idPaciente, "v1.0-telemedicina", $ip]);

Además de cumplir la historia de usuario, este registro permite que el sistema mantenga evidencia formal de la aceptación del paciente.

### 6.11.4 Creación de la cita (HU-03, primera fase)

El siguiente paso consiste en crear la cita en la base de datos. Esto se realiza mediante el endpoint:

/api/crear\_cita.php

La lógica aplicada aquí es una de las más importantes del módulo, ya que la disponibilidad se define por día de la semana y no por fecha. Por ello, el sistema calcula automáticamente la fecha de la próxima ocurrencia del día seleccionado.

El proceso incluye:

* Obtener el idPaciente desde la sesión.
* Calcular la fecha de la cita.
* Generar un folio incremental tipo c1, c2, c3, etc.
* Registrar la cita en estado PendientePago.
* Crear un registro de auditoría CREAR\_CITA.

Fragmento representativo del cálculo de fecha:

$diaHoy = (int)$hoy->format("w");

$diferencia = ($diaSeleccionado - $diaHoy + 7) % 7;

$fechaCita = clone $hoy;

$fechaCita->modify("+$diferencia day");

$inicio = $fechaCita->setTime($hIni, $hIniMin)->format("Y-m-d H:i:s");

Este enfoque permite mantener una agenda semanal reutilizable sin necesidad de insertar fechas específicas en la tabla de disponibilidad.

### 6.11.5 Pago simulado y actualización del estado de la cita

Después de crear la cita, el paciente debe realizar un pago simulado. El formulario visual se basa en los wireframes, pero los datos de tarjeta no se almacenan, ya que solo se trata de una demostración.

El procesamiento se realiza en:

/api/registrar\_pago.php

Este endpoint:

* Valida que la cita realmente pertenezca al paciente.
* Comprueba que la cita se encuentre en estado PendientePago.
* Genera un idPago incremental (p1, p2, p3…).
* Inserta el pago en la tabla pago.
* Cambia el estado de la cita a Confirmada.
* Registra la acción CREAR\_PAGO en logauditoria.

Fragmento representativo:

$insertPago = $conn->prepare("

INSERT INTO pago (idPago, idCita, monto, moneda, proveedor, referencia, estado, createdAt)

VALUES (?, ?, 500.00, 'MXN', 'Stripe', ?, 'Autorizado', NOW())

");

$insertPago->execute([$idPago, $idCita, $referencia]);

Entonces, la cita se confirma oficialmente y entonces podrá ser atendida.

### 6.11.6 Pantalla de cita confirmada y comprobante

Tras registrarse el pago, el sistema redirige al paciente a: /cita-confirmada

En esta página se muestra un comprobante conformado por información obtenida de las tablas:

* Cita
* Médico (medico + usuario)
* Pago

El comprobante se genera dinámicamente y puede imprimirse mediante la función nativa window.print(). Para esta versión del prototipo no se almacena un archivo PDF, ya que no es un requisito del caso.

Fragmento ilustrativo del consumo del endpoint:

fetch("/api/detalle\_cita.php?idCita=" + idCita)

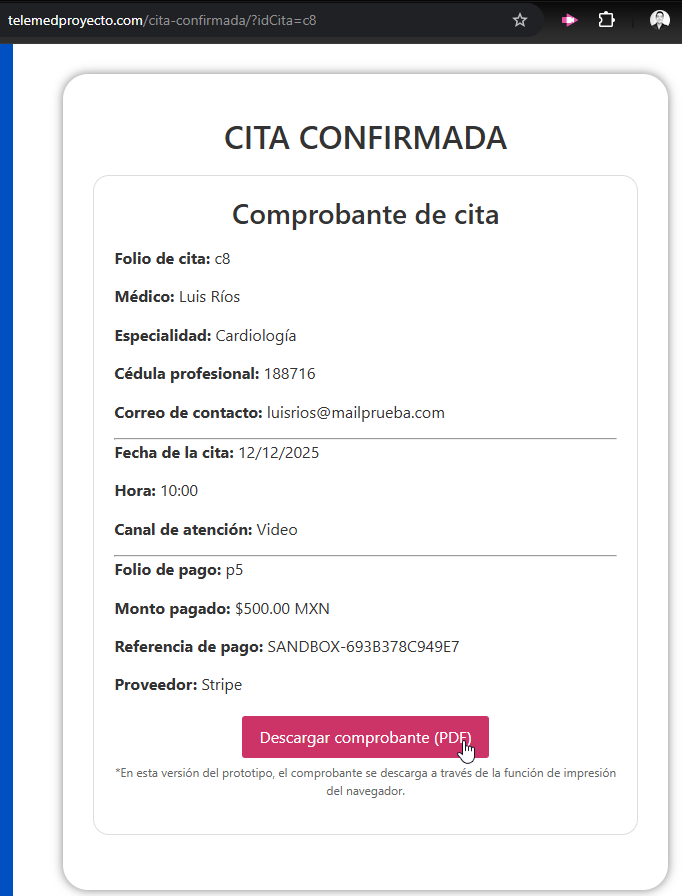
.then(r => r.json())

.then(data => {

document.getElementById("folioCita").textContent = data.cita.idCita;

document.getElementById("nombreMedico").textContent = data.cita.nombreMedico;

});



***Figura 6.5 Cita Confirmada***

### 6.11.7 Consideraciones finales

El flujo completo integra varios módulos de manera coherente, respetando:

* El modelo de datos oficial.
* Las reglas del caso de titulación.
* La separación entre frontend (WordPress/Elementor) y backend (PHP).
* Los wireframes establecidos para el paciente.
* La trazabilidad de eventos a través de la tabla logauditoria.

El resultado es un flujo funcional sólido, que abarca desde la selección del médico hasta la emisión de un comprobante de pago, incorporando además la validación del consentimiento informado.

# 10. REFERENCIAS

Schwaber, K., & Sutherland, J. (2020). *La Guía Scrum™*. Scrum.org.

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>

Pressman, R. S., & Maxim, B. R. (2019). *Ingeniería de Software: Un enfoque práctico* (9ª ed.). McGraw-Hill.

Centro Nacional de Excelencia Tecnológica en Salud [CENETEC]. (2024). Boletín de Telesalud 15 (May-Ago 2024).

<https://cenetec-difusion.com/observatoriotelesalud/wp-content/uploads/2024/09/Boletin-en-Telesalud-15_May-Ago-2024.pdf>

Instituto Nacional de Transparencia, Acceso a la Información y Protección de Datos Personales [INAI]. (2020). Ley Federal de Protección de Datos Personales en Posesión de los Particulares. Diario Oficial de la Federación. https://www.dof.gob.mx/nota\_detalle.php?codigo=5150631

Organización Mundial de la Salud [OMS]. (2010). Telemedicine: Opportunities and developments in Member States (Report of the second global survey on eHealth).

<https://apps.who.int/iris/handle/10665/44497>

Organización Panamericana de la Salud [OPS]. (2024). Salud digital en América Latina: legislación actual y desafíos.

https://iris.paho.org/bitstream/handle/10665.2/59521/v48e402024.pdf

Revista Abogacía. (2025). Telemedicina en México: regulación pendiente para garantizar el derecho a la salud. https://www.revistaabogacia.com/telemedicina-en-mexico-regulacion-pendiente-para-garantizar-el-derecho-a-la-salud

Saluddigital.io. (2024). Telemedicina en México: barreras y oportunidades. https://www.saluddigital.io/blog/telemedicina-en-m%C3%A9xico-barreras-y-oportunidades

Secretaría de Salud. (1998). Norma Oficial Mexicana NOM-168-SSA1-1998, del expediente clínico. Diario Oficial de la Federación. https://dof.gob.mx/nota\_detalle.php?codigo=4881768&fecha=30/09/1999

Secretaría de Salud. (2012). Norma Oficial Mexicana NOM-004-SSA3-2012, del expediente clínico. Diario Oficial de la Federación. https://www.dof.gob.mx/normasOficiales/4415/salud/salud.htm

Secretaría de Salud. (2023). ¿Qué es la telesalud y la telemedicina? https://www.gob.mx/salud/dgmoss/acciones-y-programas/que-es-la-telesalud-y-la-telemedicina

Centro Nacional de Excelencia Tecnológica en Salud [CENETEC]. (2024). Boletín de Telesalud 15 (May-Ago 2024). https://cenetec-difusion.com/observatoriotelesalud/wp-content/uploads/2024/09/Boletin-en-Telesalud-15\_May-Ago-2024.pdf

Organización Panamericana de la Salud [OPS]. (2024). Salud digital en América Latina: legislación actual y desafíos. https://iris.paho.org/bitstream/handle/10665.2/59521/v48e402024.pdf

Saluddigital.io. (2024). Telemedicina en México: barreras y oportunidades. https://www.saluddigital.io/blog/telemedicina-en-m%C3%A9xico-barreras-y-oportunidades