

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE HÁBITAT INFRAESTRUCTURA Y

CREATIVIDAD

CARRERA DE: SISTEMAS DE INFORMACIÓN



TRABAJO DE TITULACIÓN

TEMA: DESARROLLO DE UN PROTOTIPO DE SISTEMA  
WEB PARA LA GESTIÓN DE PACIENTES PARA UN  
CONSULTORIO MÉDICO INDEPENDIENTE

CASO DE ESTUDIO: CONSULTORIO DR. MIGUEL BRAVO

TUTOR: ING. ALARCÓN MENA JORGE ALEJANDRO

AUTOR:

BRAVO VALENCIA BENJAMÍN ISAÍAS

QUITO, ENERO DE 2026

## **Dedicatoria**

*Quiero empezar dedicando este trabajo a toda mi familia, hermanos, cuñados y mis sobrinos, y sobre todo a mi madre, que fue un apoyo incondicional en todo momento, acompañándome en este camino, y a mi padre, que me enseñó el valor del esfuerzo y la dedicación.*

*Dedico a todas las personas que confiaron en mí y, sin duda alguna, también dedico este logro a Dios.*

## **Agradecimiento**

*En primer lugar, me agradezco a mí mismo por la constancia, la confianza que siempre me tuve y me tendré en realizar las cosas y saber que las puedo lograr esforzándome, a pesar de los momentos difíciles que da la vida. Yo nunca renuncio y yo no retrocederé a mi palabra, porque ese es mi camino.*

*Agradezco a mi familia, en especial a mis sobrinos y a mi madre; junto con ellos, en este proceso aprendí que aun en los momentos difíciles cada caída fortalece y que ningún logro es posible sin esfuerzo ni sacrificio.*

*Agradezco a los tutores que fueron una guía en todo este proceso y a los compañeros que con el tiempo se convirtieron en amigos incondicionales.*

## **Abstract**

The main objective of this degree project is to develop a web system prototype for patient management in an independent medical clinic, taking as a case study the independent clinic of Dr. Miguel Bravo, located in Tumbaco, which also provides home medical care depending on the patient's context or condition.

The analysis conducted in this case study identifies the main issues resulting from the manual management of patient and appointment information, such as data loss, duplication, and reduced quality of care due to the absence of a system that facilitates and accelerates patient service. In this context, the purpose of this project is to address and solve these issues by designing and developing a web application prototype that automates manual patient and appointment management tasks, implementing a relational database to ensure the integrity and security of stored data.

For the development of this prototype, the Scrum agile methodology was implemented, allowing effective collaboration and feedback between the user and the project author, as well as the progressive delivery of functional system versions to Dr. Miguel Bravo. The prototype was developed using modern technologies such as backend frameworks, a relational database engine and management system, and a microservices-based architecture to promote future maintainability and scalability.

In conclusion, the proposed web system prototype provides a solution to the problems of the independent medical clinic by improving administrative efficiency and patient information management, reducing the errors inherent in manual processes, and ensuring the protection of sensitive data. Therefore, this development contributes to the implementation of technologies that foster automation in independent medical practices.

## Tabla de Contenidos

Dedicatoria.....	I
Agradecimiento.....	II
Abstract.....	III
CAPÍTULO 1: Introducción.....	7
1.1 Justificación.....	7
1.1.1 Problemas operativos.....	7
1.1.2 Impacto en la calidad del servicio.....	8
1.1.3 Necesidad tecnológica.....	9
1.2 Planteamiento del problema.....	11
1.2.1 Problema central.....	11
1.2.2 Causas.....	11
1.2.3 Consecuencias.....	12
1.2.4 Preguntas que surgen en este planteamiento del problema.....	13
1.3 Objetivos.....	13
1.3.1 Objetivo General.....	13
1.3.2 Objetivos Específicos.....	14
1.4 Metodología.....	14
1.4.1 Tipo de Investigación.....	15
1.5 Alcance.....	15
CAPÍTULO 2: Marco Teórico.....	16
2.1 Antecedentes.....	16
2.2 Observaciones de los estudios.....	17
2.3 Marco Teórico.....	17
2.3.1 Sistemas de gestión de pacientes y tecnologías aplicadas.....	18
2.3.1.1 Justificación de la arquitectura seleccionada.....	19
2.3.2 Metodologías Ágiles.....	20
2.3.3 Metodologías Ágiles vs Metodologías Tradicionales.....	21
2.3.4 Scrum.....	22
2.3.4.1 Beneficios de Scrum.....	23
2.3.4.2 Aplicación de Scrum para el desarrollo del prototipo de sistema web para el consultorio del Dr. Miguel Bravo.....	23
2.3.5 Pruebas de software en el desarrollo del prototipo.....	24
2.3.5.1 Pruebas de Integración.....	25
2.3.5.2 Pruebas de Aceptación.....	25
2.4.1 SQL Server.....	27

2.4.2 SQL Server Management Studio.....	28
2.4.3 Visual Studio 2022.....	29
2.4.4 C#.....	30
2.4.5 ASP.NET Core.....	31
2.4.6 Visual Studio Code.....	31
2.4.7 HTML/CSS.....	31
2.4.8 JavaScript.....	32
2.4.9 Figma.....	32
2.4.10 Power Designer.....	33
2.5 Marco Conceptual.....	34
2.5.1 Consultorio médico independiente.....	34
2.5.2 Requerimientos funcionales y no funcionales.....	34
2.5.3 Base de datos.....	34
2.5.4 IDE.....	34
2.5.5 Framework.....	35
2.5.6 Arquitectura de Microservicios.....	35
2.5.7 Prototipo.....	35
2.5.8 Api.....	35
2.5.9 Backend.....	36
2.5.10 Frontend.....	36
CAPÍTULO 3: Análisis de Requerimientos y Modelado del Sistema.....	37
3.1 Requerimientos del prototipo de sistema web.....	37
3.2 Casos de Uso.....	37
3.2.1 CU-01 Sesión del usuario (Médico).....	37
3.2.2 CU-02 Gestionar Paciente.....	39
3.2.3 CU-03 Gestionar la Cita para el paciente.....	41
3.2.4 CU-04 Generar informe de la cita.....	43
3.3 Funcionalidades.....	45
3.3.1 Requerimientos funcionales.....	46
3.3.2 Requerimientos no funcionales.....	47
3.4 Modelos de la Base de Datos.....	48
3.4.1 Modelo Conceptual.....	48
3.4.2 Modelo físico.....	48
3.5 Tabla de trazabilidad del sistema.....	49
CAPÍTULO 4: Desarrollo del prototipo de sistema web.....	50
4.1 Implementación de la base de datos.....	50
4.2 Implementación del backend.....	52
4.3 Implementación del frontend.....	53
4.3.1 Interfaz de Login.....	54

4.3.2 Interfaz de Menú Principal.....	54
4.3.3 Interfaz para la gestión de los pacientes.....	55
4.3.4 Interfaz para agregar un nuevo paciente.....	56
4.3.5 Interfaz de la modificación de paciente.....	56
4.3.6 Interfaz de la agenda.....	57
4.3.7 Interfaz de modificación de las citas.....	57
4.3.8 Interfaz para asignar una nueva cita al paciente.....	58
4.3.9 Interfaz de los informes.....	58
4.3.10 Interfaz para crear nuevos informes.....	59
4.3.11 Interfaz de recuperar la contraseña.....	59
4.4 Desarrollo del proyecto por Sprints.....	60
4.4.1 Sprint 1 Modelado y Diseño del sistema.....	60
4.4.2 Sprint 2 Creación de la base de datos y Backend.....	62
4.4.3 Sprint 3 Creación del frontend e integración con el backend.....	63
4.5 Pruebas del Prototipo de Sistema web.....	65
4.5.1 Pruebas de Integración.....	65
4.5.1.1 Inicio de Sesión y navegación inicial.....	65
Flujo probado:.....	65
Resultado esperado:.....	65
Evidencia:.....	66
4.5.1.2 Gestión de Pacientes (CRUD).....	67
Objetivo:.....	67
Flujo probado:.....	67
Resultado esperado:.....	67
Evidencia:.....	67
4.5.1.3 Gestión de Citas y Calendario.....	69
Objetivo:.....	69
Flujo probado:.....	69
Resultado esperado:.....	69
Evidencia:.....	69
4.5.1.4 Gestión de Informes.....	72
Objetivo:.....	72
Flujo probado:.....	72
Resultado esperado:.....	72
Evidencia:.....	72
4.5.2 Pruebas de Aceptación.....	74
4.5.2.1 Resultados de pruebas de aceptación.....	74
4.5.2.2 Conclusión de las pruebas.....	74
4.6 Justificación de las decisiones técnicas.....	75

4.6.1 Base de datos relacional.....	75
4.6.2 Backend basado en ASP.NET Web API.....	75
4.6.3 Frontend basado en tecnologías web estándar.....	75
4.7 Análisis del prototipo desarrollado.....	76
4.7.1 Escalabilidad.....	76
4.7.2 Mantenibilidad.....	76
4.7.3 Limitaciones del prototipo.....	76
CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES.....	77
5.1 Conclusiones.....	77
5.2 Recomendaciones.....	78
5.2.1 Recomendaciones técnicas.....	78
5.2.2 Recomendaciones organizacionales.....	78
5.2.3 Recomendaciones metodológicas.....	79
Bibliografía.....	80
ANEXOS:.....	83

## Contenido de Tablas

<b>Tabla 1</b>	Metodologías ágiles vs metodologías tradicionales	22
<b>Tabla 2</b>	Componentes y tecnologías de SQL Server	27
<b>Tabla 3</b>	SQL Server Management Studio	29
<b>Tabla 4</b>	Tabla de descripción de los requerimientos funcionales del sistema	46
<b>Tabla 5</b>	Tabla de descripción de los requerimientos no funcionales del sistema	47
<b>Tabla 6</b>	Tabla de trazabilidad entre objetivos, requerimientos, casos de uso y pruebas	49
<b>Tabla 7</b>	Resultados de las pruebas de aceptación con código de RF	74

## Contenido de Figuras

<b>Figura 1</b>	Caso de uso para la sesión de usuario	40
<b>Figura 2</b>	Caso de Uso para la gestión de paciente	42
<b>Figura 3</b>	Caso de Uso para la gestión de citas	44
<b>Figura 4</b>	Caso de Uso para la gestión de informes	46
<b>Figura 5</b>	Modelo Conceptual	51
<b>Figura 6</b>	Modelo Físico	51
<b>Figura 7</b>	Base de datos montada en el motor	52
<b>Figura 8</b>	Registros de prueba (Paciente)	53
<b>Figura 9</b>	Registros de prueba(Cita)	53
<b>Figura 10</b>	Registros de prueba (Informe)	53
<b>Figura 11</b>	Registros de prueba(Médico)	53
<b>Figura 12</b>	Registros de prueba(Catálogo)	53
<b>Figura 13</b>	Backend	54
<b>Figura 14</b>	Herramienta Swagger para probar las Apis	55
<b>Figura 15</b>	Estructura FronEnd	55
<b>Figura 16</b>	Interfaz de Login	56
<b>Figura 17</b>	Interfaz de Menú Principal	56
<b>Figura 18</b>	Interfaz de Menú Principal parte inferior	57
<b>Figura 19</b>	Interfaz Pacientes	57
<b>Figura 20</b>	Interfaz para agregar nuevos pacientes	58
<b>Figura 21</b>	Interfaz de la modificación del paciente	58
<b>Figura 22</b>	Interfaz de la agenda	59
<b>Figura 23</b>	Interfaz para la modificación de la cita seleccionada	59
<b>Figura 24</b>	Interfaz para crear una nueva cita	60
<b>Figura 25</b>	Interfaz para visualizar los informes generados por el usuario	60
<b>Figura 26</b>	Interfaz para crear nuevos informes	61
<b>Figura 27</b>	Interfaz para recuperar la clave	61
<b>Figura 28</b>	Sprint 1 Modelado y Diseño del Sistema	63
<b>Figura 29</b>	Sprint 2 Creación de la BDD y Backend	64
<b>Figura 30</b>	Sprint3 Creación del Frontend e integración con el Backend	66
<b>Figura 31</b>	Prueba de Integración Inicio de Sesión	68
<b>Figura 32</b>	Prueba de Integración Navegación inicial con la sesión establecida	68
<b>Figura 33</b>	Prueba de Integración registro de nuevo paciente	69
<b>Figura 34</b>	Pruebas de Integración paciente guardado en la base	70
<b>Figura 35</b>	Prueba de Integración Modificación de los datos del paciente	70
<b>Figura 36</b>	Prueba de Integración Confirmación de cambio de datos en la base	70
<b>Figura 37</b>	Prueba de Integración Creación de nueva cita	71
<b>Figura 38</b>	Prueba de Integración cita agendada	72
<b>Figura 39</b>	Prueba de Integración Validación del horario	72
<b>Figura 40</b>	Prueba de Integración Modificación de la cita	73
<b>Figura 41</b>	Prueba de Integración cita modificada en la agenda	73
<b>Figura 42</b>	Pruebas de Integración Filtro de búsqueda por paciente	74
<b>Figura 43</b>	Prueba de Integración Creación del informe por la cita	75
<b>Figura 44</b>	Prueba de Integración Visualización de informe creado y modificación de este	75

# **CAPÍTULO 1: Introducción**

## **1.1 Justificación**

### **1.1.1 Problemas operativos**

“La gestión administrativa es una de las piedras angulares para el éxito de cualquier clínica o consultorio médico. Sin embargo, muchos propietarios de clínicas y equipos administrativos enfrentan retos que pueden poner en riesgo la eficiencia operativa, la satisfacción de los pacientes y, en última instancia, la rentabilidad del negocio. En este artículo, exploraremos los cinco errores más comunes en la gestión administrativa de clínicas y ofreceremos soluciones prácticas para evitarlos.” (Blau Consulting Group, n.d.)

En este caso de estudio en particular se habló del consultorio del Dr. Miguel Bravo, ubicado en Tumbaco y anteriormente se tenía un consultorio también en Cumbayá, cabe recalcar que no solamente atiende pacientes en su consultorio, a su vez atiende pacientes en parroquias aledañas como Puembo y Pifo. Se identificó que el registro de pacientes y la gestión de citas se realizan de manera manual, mediante agendas personales, notas o, a veces, solamente con memoria.

Estos procesos, como es la gestión tanto del paciente como de su cita, con estos métodos, generan varias dificultades, como citas duplicadas por un mal control de citas, pérdida de información del usuario o errores al agendar turnos.

De esta forma, al hablar de un consultorio médico independiente, el Dr. Miguel Bravo atiende tanto en el consultorio de Tumbaco como en los domicilios de pacientes con problemas de movilidad o transporte o que tengan alguna condición de salud que impide que el paciente asista de manera personal al consultorio. Este flujo del proceso aumenta las dificultades de la gestión de citas y de información del paciente, ya que se debe manipular y coordinar varias ubicaciones, horarios y tipos de atención.

En épocas de la pandemia de COVID-19, por ejemplo, esta necesidad de atención médica a domicilio surgió para evitar que los posibles contagiados salieran y pudieran provocar nuevos contagios, especialmente en personas de la tercera edad o adultos

mayores en general. En la actualidad, existen varias enfermedades con características similares a las de la pandemia e incluso siguen presentándose casos de COVID-19 en algunas personas.

Debido a estas circunstancias, el Dr. Bravo, al igual que otros médicos a cargo de un consultorio independiente, se vio en la necesidad de adaptarse y reorganizar la manera de trabajo y atención al paciente. Este cambio fue uno de los motivos que demostró las dificultades y limitaciones de un manejo manual de la información, tanto la asignación y manejo de citas, direcciones; en sí, toda la gestión del paciente se volvió un proceso de alta complejidad.

En esos casos donde se debe cumplir con varias normas y realizar los procesos requeridos de manera eficiente y rápida, no se podía optar por la manera tradicional o manual de registrar, gestionar y documentar el tratamiento del paciente. (Ministerio de Salud Pública del Ecuador, 2022)

### **1.1.2 Impacto en la calidad del servicio**

“Los casos de negligencias médicas relacionados con una gestión inadecuada de los registros médicos pueden tener impactos graves en la salud y bienestar del paciente. La pérdida o mala gestión de información relevante puede generar consecuencias adversas, desde un mal tratamiento hasta complicaciones médicas severas e, incluso, pueden poner en riesgo la vida del paciente.” (Hidalgo Fernández Abogados, 2020)

“Una documentación deficiente puede provocar diagnósticos inexactos o tratamientos inadecuados. La omisión de datos críticos (alergias, historia previa, dosis de medicación) favorece interacciones peligrosas y eventos adversos. Cada error de transcripción o de registro aumenta el riesgo de errores clínicos (medicación incorrecta, procedimientos equivocados), afectando la seguridad del paciente y pudiendo causar daños evitables.

También se resiente la eficiencia del profesional sanitario. El tiempo invertido en buscar datos en archivos físicos o en escribir notas largas reduce la eficiencia asistencial. Como se señala en un estudio de Perkins et al., esta carga administrativa desplaza recursos

del cuidado directo del paciente. Ello disminuye el tiempo de interacción médico-paciente y contribuye al agotamiento profesional. De hecho, el mismo estudio subraya que esta sobrecarga clerical reduce la calidad percibida de la atención y eleva la insatisfacción y el burnout de los médicos. En la práctica, los equipos de salud que deben interpretar notas confusas o buscar información fragmentada reportan una menor eficiencia en sus procesos diarios.” (Relisten, 2025)

Además, existen épocas del año en las cuales el flujo de enfermedades aumenta exponencialmente, ya sea una época de cambios climáticos o de temporadas de enfermedades, como por ejemplo respiratorias, que afectan a una gran cantidad de personas, lo que promueve la atención en diferentes sectores y de manera simultánea. Esta situación aumenta los riesgos de tener errores en las citas o duplicidad en la información por un mal manejo y almacenamiento de esta.

Desde el punto de vista de algunas parroquias pequeñas en espacio, pero grandes en población, como por ejemplo Tumbaco, existen algunos puntos de atención médica, pero no abastecen a toda la gente que acude a estos espacios. Entonces, un sistema que ayude a gestionar más rápidamente a todas estas personas tendrá un impacto directo en la comunidad, por el hecho de que cuántas veces se ha visto que, en un centro de salud, ya sea público o privado, existan filas y filas interminables de gente que acude a su cita ya establecida.

### **1.1.3 Necesidad tecnológica**

Un sistema web para la gestión de pacientes ayudará no solo en los puntos ya hablados, sino hablando del día a día de un doctor, cuando realiza acciones rutinarias o repetitivas, es fundamental que estos mismos procesos se realicen de manera rápida y eficiente, porque, como se sabe, un doctor no solo atiende a un paciente por día o por hora, son tandas y tandas de pacientes y cada uno es un mundo diferente en enfermedad, en tratamiento; por ejemplo, puede ser el caso de que algún paciente solo acuda a un consultorio médico para una cita rutinaria, pero en otros casos hay pacientes que continúan con su tratamiento ya establecido y se necesita generar o documentar sus avances, por lo que la gestión de esta información debe hacerse de manera que no afecte a la consulta o

tratamiento en sí.

Tal vez la situación de estos centros sea la falta de personal o de insumos médicos; sin embargo, por este ámbito no se indagará, sino por el hecho de que en muchos de estos casos estos centros colapsan por el simple motivo de que el sistema no responde, o porque el sistema se cae, o simplemente porque hubo algún error en el sistema y no es escalable o fácil de mantener, ya sea por la tecnología, que puede ser antigua o mal implementada, o en algunos casos porque no se cuenta con un sistema que realice las acciones necesarias, las cuales, en estos centros, podría estar realizando una sola persona para todos los pacientes.

Ahora, hablando de seguridad, es importante saber que cualquier sistema, sea web, de escritorio, una aplicación móvil o cualquier tecnología, siempre va a almacenar información sensible de los usuarios que, en manos equivocadas, pueden ser usadas con malos fines, como el robo de identidad o algún tipo de delito organizado. Por ese motivo, es necesario asegurar la integridad de los datos o un buen manejo de la información para que el usuario pueda confiar y utilizar estas tecnologías sin temor alguno.

Por otra parte, sin contar con los errores que se generan en la gestión y la pérdida de información, la seguridad de la información de los pacientes es un activo muy importante y sensible. Según un informe de (Avertium, 2024), el sector de la salud sufre una amenaza muy grande por los ataques de ingeniería social, y un gran porcentaje de esos ataques resulta exitoso debido al mal manejo o a la falta de protección de la información del paciente, que a simple vista puede parecer inofensiva.

Por todo este contexto y razones, la implementación de un prototipo de sistema web que gestione la cita del paciente y sus citas médicas resulta necesaria y de vital importancia para poder mejorar la administración de los pacientes, reducir los errores humanos al tener un proceso manual, centralizar y unificar la información que ayude a mejorar el control, la seguridad y la integridad de la misma información, y generar al usuario una atención correcta, eficiente y óptima.

## **1.2 Planteamiento del problema**

### **1.2.1 Problema central**

El consultorio del Dr. Miguel Bravo afronta dificultades en la gestión de la información de los pacientes y de las citas médicas agendadas, por motivo de que los procesos y registros se realizan de manera manual, como, por ejemplo, una nota en el celular o escrita en una hoja de papel. Estos métodos son los causantes de generar errores, en el caso de la información que se genera a partir de los datos de los pacientes y de sus citas, puede haber la posibilidad de que existan dos citas el mismo momento el mismo día o, simplemente, que se olvide de la cita y de la ubicación, de esta ya sea en el consultorio de Tumbaco, o si es a domicilio, la dirección.

La situación actual del consultorio médico es que el usuario, en este caso el doctor, atiende a pacientes en su consultorio en Tumbaco y, a su vez, también atiende a personas enfermas o de la tercera edad en sus propios domicilios. Esta es información imprescindible si se quiere llevar a cabo un control adecuado de las citas y de la información del paciente en sí, y, en este caso, no existe dicho control.

### **1.2.2 Causas**

La raíz intrínseca del problema es la falta de automatización para los procesos cotidianos por los que siempre va a pasar el paciente, independientemente que sea una consulta rutinaria, primera visita o si ya está en un tratamiento continuo, por lo que se genera una dependencia en estos procesos que no son totalmente confiables.

Como ya se habló anteriormente, estos procesos manuales son cotidianos y repetitivos, lo que hace que el personal de trabajo que, a veces, es solo el doctor, sin contar con alguna secretaria tenga que realizar estos procesos y, además, brindar el servicio médico por el que acude el paciente.

Al ser un consultorio médico independiente, no se cuenta con alguna organización externa que administre las citas o gestione la información del paciente en un sistema. Por ello, el Dr. Bravo debe realizar tareas médicas y, al mismo tiempo, tareas administrativas, como lo haría un enfermero, enfermera o secretario(a) en un hospital. Esto representa un recurso limitado y genera un impacto de doble rol, al no contar con otra persona que realice estas tareas, lo que, a su vez, resulta más complicado por no disponer de un sistema

eficiente que realice estas gestiones administrativas, las cuales están directamente relacionadas con la atención correcta que se desea brindar al paciente.

### **1.2.3 Consecuencias**

Estos errores, antes mencionados, tienen un impacto directo con la calidad de atención que se brinda al paciente, generando retrasos para la atención; por obvias razones esto incrementa en las épocas del año con mayor flujo de pacientes o enfermos.

Entonces, sumadas esas dos tareas, es cuestión de tiempo, o de mala suerte, que por fatiga o cansancio exista una mayor posibilidad de cometer errores humanos, como, por ejemplo, la pérdida de libretas o recetas donde exista información fundamental del paciente o de su tratamiento; igualmente, en caso de tener las citas anotadas en algún lugar físico o solo en la memoria. Esto es un problema principal, porque los datos siempre deben estar disponibles en caso de cualquier situación emergente o cotidiana.

Estos errores tienen una consecuencia principal en la calidad del servicio que se quiere brindar al paciente: los problemas ya mencionados de no almacenar la información de manera correcta generan retrasos en la atención, haciendo que los pacientes deban esperar más tiempo para ser atendidos. Este mismo tema puede afectar la continuidad del tratamiento de algunos pacientes, debido a que algunos tienen que acudir periódicamente al consultorio para una revisión o para analizar cuál es el estado de su motivo de tratamiento, lo que se dificulta o puede generar retrasos en el mismo tratamiento.

Un paciente debe confiar plenamente en que, al entregar sus datos o información personal con fines médicos al ser información sensible, esta se almacenará de manera segura para evitar cualquier tipo de inconveniente que pueda ocurrir con su información. Por ejemplo, tal vez uno considere que solo la información del paciente no es suficiente para causar daño; pero aquí podría intervenir la ingeniería social: con esos simples datos, aunque no sean críticos, las consecuencias sí pueden serlo.

Algunos problemas se pueden entender fácilmente a simple vista pero hay algunos problemas que no se comprenden en el momento, existe algunas consecuencias ocultas por

no tener implementando un sistema web, como por ejemplo la citas perdidas estas pueden ser en el consultorio y tener que re agendar la cita en otro momento o en la otra situación que es la de atención a domicilio al perder u olvidar las citas puede generar pérdida de capital por la movilidad que surge al no controlar estos datos de la ubicación no se puede gestionar o administrar de manera correcta la movilidad hacia el domicilio del paciente y los costos que estos generan.

Como ya se habló antes existen épocas de mayor flujo de pacientes entonces al no optar por un sistema se pierde la oportunidad de atender a más pacientes en el consultorio o en el domicilio y esto obviamente genera una pérdida de clientes y por ende de ingresos, aunque haya la posibilidad de atender a más personas.

#### **1.2.4 Preguntas que surgen en este planteamiento del problema**

1.- ¿El desarrollo de este prototipo puede resolver los principales problemas consecuentes al tener una gestión de paciente manual?

2.- ¿Qué requerimientos funcionales y no funcionales son necesarios para garantizar el correcto desarrollo de este prototipo?

3.- ¿Qué metodología para el desarrollo será más beneficioso para la construcción del sistema web?

### **1.3 Objetivos**

#### **1.3.1 Objetivo General**

Desarrollar un prototipo de sistema web funcional para la gestión de pacientes y citas médicas del consultorio del Dr. Miguel Bravo, que permita centralizar la información clínica y administrativa mediante una base de datos relacional, optimizando el tiempo de atención y reduciendo los errores derivados del manejo manual de la información.

#### **1.3.2 Objetivos Específicos**

- Definir los posibles requerimientos funcionales y no funcionales más importantes para el prototipo de sistema web propuesto.
- Definir una arquitectura basada en microservicios, utilizando frameworks para el desarrollo del frontend y/o backend del sistema.

- Definir el flujo del sistema web mediante diagramas estructurados.
- Diseñar la base de datos utilizando un modelo entidad-relación.
- Diseñar una interfaz intuitiva para el usuario.

## **1.4 Metodología**

Al diseñar, construir y ejecutar un sistema web desde cero, es indispensable contar con una metodología que aporte y beneficie al desarrollo de este. Por este motivo, se optó por una metodología ágil, debido a que pueden surgir cambios en los requerimientos funcionales o no funcionales del sistema, o incluso modificaciones en la lógica interna del negocio, y el sistema debe poder mantenerse, escalar y, sobre todo, adaptarse a las necesidades del cambio.

Por otra parte, es óptimo porque, en este caso de estudio que beneficiará al médico, es indispensable contar con su apoyo para retroalimentar de la mejor manera el desarrollo del sistema, ya que quién mejor que el propio usuario para ayudar en el desarrollo de este. En cada entrega funcional se informó al Dr. Bravo sobre los cambios o agregaciones al prototipo web planteado en este proyecto y, de esta manera, se podrán ajustar de forma intrínseca según sus necesidades.

### **1.4.1 Tipo de Investigación**

Este prototipo de sistema web propone una solución general para resolver problemas específicos identificados en la gestión que tiene el caso de estudio al no tener un sistema automatizado para gestionar al paciente, sino un sistema manual. Como ya se habló antes, se optó por una metodología ágil; en este caso, se escogió Scrum como representante de esta.

El tipo de investigación, por ende, es aplicado, por el hecho de que se utilizan conocimientos técnicos para resolver un problema real identificado en el consultorio independiente del Dr. Miguel Bravo. A su vez, es descriptivo y de desarrollo tecnológico, debido a que analiza la situación normal y actual de cómo se gestiona la información del paciente, además de sus citas. Por esta razón, se decidió desarrollar un prototipo funcional de sistema web como resultado y solución práctica del caso de estudio.

En este contexto, el tipo de investigación, vinculada con la metodología ágil

seleccionada (Scrum), permite que el desarrollo de este prototipo funcional se realice de manera iterativa y colaborativa con el usuario, siendo flexible ante los cambios y la retroalimentación proporcionada por el mismo. Esto garantiza que la solución propuesta sea una respuesta directa a las necesidades reales analizadas e identificadas en este consultorio independiente.

## **1.5 Alcance**

El alcance de este proyecto terminará cuando el desarrollo funcional del prototipo de sistema web para el consultorio del Dr. Miguel Bravo realice y posea las siguientes características:

- Gestión de Pacientes (CRUD).
- Generar y agendar citas con ubicación.
- Historial clínico básico.
- Búsqueda rápida por filtros.
- Base de datos relacional de pacientes y citas.
- Interfaz web intuitiva.
- Confirmación formal mediante un acta de aprobación y cumplimiento del proyecto por el Doctor.

## **CAPÍTULO 2: Marco Teórico**

### **2.1 Antecedentes**

Ya realizado el establecimiento de los objetivos generales y específicos, se continuó con la tarea de investigar trabajos o proyectos profesionales cuyos objetivos o justificaciones fueran similares o parecidas al prototipo que se desarrolló. Con honestidad, se esperaba encontrar algunos trabajos que realizarán esta transición o automatización de la gestión manual de pacientes para lograr una mayor eficacia en el proceso, y sorprendió descubrir que no solo se habían desarrollado en el extranjero, sino también en el Ecuador.

El objetivo de la tesis de la (Escuela Superior Politécnica del Litoral, 2023) fue crear y poner en marcha una aplicación web que automatiza la gestión del paciente, el cual fue un gran radical respecto a la forma manual a la que estaban acostumbrados.

Así que el sistema hizo las cosas mucho más fáciles cuando se trataba de manejar el administrador y los asuntos médicos para pacientes este fondo es relevante para este proyecto de grado, ya que las estrategias utilizadas para resolver el problema, como la automatización manual de tareas en la gestión del paciente, son similares al enfoque propuesto en esta investigación.

En el proyecto desarrollado por la (Universidad Carlos III de Madrid, 2021) a la vez que se implementó una aplicación web, también se optó por una plataforma online orientada más a la parte de administración. Por este contexto se habla de una mejoría en la atención del clientes y reducción de los errores en la gestión de información que ocurría por una gestión manal.

En la (Universidad Politécnica Salesiana, 2022) se realizó una aplicación web para poder gestionar las citas y el historial clínico de los pacientes, lo que mejoro todo aspecto en la gestión de los pacientes y al centrar los datos médicos garantizaron la integridad de los datos.

El proyecto forma parte de la (Universidad Regional Autónoma de los Andes, 2021) estableció un sistema para llevar un seguimiento de los pacientes en Teh principal

objetivo aquí era hacer mejor gestión de los registros médicos durante las visitas y racionalizar cómo manejamos los medicamentos y sus documentos relacionados

Como resultado, el sistema permitió optimizar el apoyo y la atención a los pacientes, así como mejorar el seguimiento de los tratamientos.

## **2.2 Observaciones de los estudios**

Se observó que la mayoría de los trabajos académicos, ya sean proyectos, tesis o reportes, la mayoría eran en Ecuador y, en los casos colocados en este antecedente, se tenía problemas para gestionar de manera óptima a los pacientes, se tenían problemas de control y manejo de los datos y la información generada en los procesos que se realizaban, como el agendamiento de citas o de las historias clínicas, y gracias al desarrollo e implementación de estas soluciones tecnológicas se pudieron solventar estos problemas de raíz y se notó un mejoramiento en el flujo cotidiano de todos estos centros de salud.

## **2.3 Marco Teórico**

En el campo de la salud, más a fondo en la atención y soporte que se le da al paciente desde su primera cita hasta sus consultas normales, en este ámbito la tecnología ha mejorado considerablemente, lo cual permitió que los tiempos de espera para el paciente mejoren, la organización de las citas, datos de los pacientes e información en sí sea segura y confiable, debido a los errores humanos que se tenían al tener un sistema manual.

Como indica en su artículo (Pérez Rodríguez, 2024) la unificación de estas herramientas tecnológicas el gestionar clínicas o instituciones de salud ayuda a mejorar los tiempo de respuesta hacia los pacientes y a optimizar los recursos, también enfatiza que aunque está implantación de la tecnología sea algo factible, aún existe centros de atención a los pacientes siguen teniendo una atención o gestión manual lo que genera problemas de inexactitud de los datos y en otros casos pérdidas de información.

### **2.3.1 Sistemas de gestión de pacientes y tecnologías aplicadas**

Hablando de los tipos de sistemas que pueden ayudar a los lugares donde no se tenga un sistema automatizado de gestión, pueden entrar los sistemas web o incluso las

aplicaciones móviles, este trabajo se enfoca en el nicho de los sistemas web. A pesar de todas las ventajas que pueden brindar estos sistemas web, hay que considerar también lo malo o los riesgos o dificultades que se tendrán al implementar estas soluciones.

En el área de la ingeniería o tecnología el desafío que plantea desde el análisis de los requerimientos, el desarrollo del sistema hasta la entrega y validación del sistema completo. Por otro lado, del ámbito del usuario final que será el doctor hay personas que no son muy afines a la tecnología ya sea por su edad, etc. En estos casos la capacitación del usuario es un paso fundamental y necesario. Todos estos aspectos se deben considerar al desarrollar una solución tecnológica para la gestión de los pacientes (AppMaster, 2024)

Actualmente, en el mundo del desarrollo web es beneficioso que un sistema sea fácil de actualizar o mantener con el tiempo. Por esta razón, han surgido arquitecturas como la de microservicios, las API que permiten la comunicación entre sí y el uso de frameworks para la programación. En este contexto, para el desarrollo de este prototipo de sistema web se optará por una arquitectura basada en microservicios, debido a que la tarea de desarrollarlo se hará más fácil al dividir los diferentes servicios que se comunican o interactúan a través de una API.

En este proyecto, los sistemas de gestión de pacientes están perfectamente sincronizados con la necesidad de unificar los datos clínicos y administrativos del sistema del Dr. Miguel Bravo hace que programar y hacer un seguimiento de las historias de los pacientes sea mucho más fácil. El gran aporte es reducir los errores del trabajo de datos manual y facilitar la obtención de información desde varios puntos.

Pero a su vez presenta una dificultad significativa para el usuario al no estar tan enfatizado con la tecnología puede resultar un poco difícil de manipular al inicio y hasta acostumbrarse puede ser un poco demoroso y tedioso para el mismo usuario.

### **2.3.1.1 Justificación de la arquitectura seleccionada**

Cuando se desarrolla un sistema web existen algunos tipos de arquitecturas para

estos casos como por ejemplo la arquitectura monolítica la arquitectura en capas la arquitectura de microservicios.

La monolítica es la que unifica toda la lógica que usa el sistema en una sola capa lo que podría hacer el desarrollo fácil en un principio, pero como se sabe un software nunca va a estar acabado y por eso esta no es la mejor en términos de escalabilidad y mantenimiento de este.

La arquitectura por capas organiza un poco más el sistema por partes estructurándolo de mejor manera que la monolítica pero todavía depende mucho de todos los componentes interactuando entre si lo que complica los cambios que se quieran hacer solo en una parte de la estructura.

La arquitectura de microservicios que permite separar los módulos del sistema que no depende uno de otro y que se hablan por Apis, lo que facilita eficazmente el manteamiento del sistema y la escalabilidad el mismo que conjunto con la metodología de desarrollo se escogió esta arquitectura para desarrollar el prototipo.

### **2.3.2 Metodologías Ágiles**

No se debe cometer el error de delimitar las metodologías ágiles como una simple herramienta o plantilla, por el hecho de que es una estrategia funcional que ayuda a toda organización para poder definir, gestionar, desarrollar y finalizar sus proyectos de manera rápida, eficiente y de manera elástica (ADEN Business Magazine, 2025).

En toda organización o empresas siempre van a tener requerimientos proyectos ya sean internos o externos de la misma. En este contexto cualquier empresa busca la manera o forma en que se pueda realizar dicho proyecto de manera flexible lo que quiere decir que no se defina todo desde el principio del desarrollo como las fechas los resultados finales, fechas de entrega de ciertas partes del proyecto o en algunos casos esperar a la fecha límite de entrega final con el proyecto totalmente culminado, entonces surge la siguiente pregunta:

¿Por qué se busca una estrategia flexible?

“En el mundo empresarial actual, la agilidad y la flexibilidad se han convertido en pilares fundamentales para el éxito y la supervivencia de las organizaciones. La capacidad de adaptarse rápidamente a los cambios del mercado, innovar constantemente y responder a las necesidades de los clientes son características indispensables para cualquier empresa que desee prosperar en el entorno competitivo contemporáneo.” (Mercer, 2024)

En el marco de este proyecto, los métodos ágiles ayudan a cambiar según lo que aparezca al crear el prototipo, viendo que los pedidos pueden variar según la opinión del usuario final. Lo más útil que tienen es que son flexibles y dan funciones poco a poco; pero, algo que no es tan bueno es que piden que el usuario participe siempre, y eso puede ser difícil si no hay tiempo suficiente.

### **2.3.3 Metodologías Ágiles vs Metodologías Tradicionales**

En primera instancia se puede decir que las dos metodologías buscan brindar un marco o una plantilla para que el desarrollo de software tenga una estructura. Sin embargo, tienen diferencias en la manera en la que se trabaja en cada fase de la estructura.

Según (ADEN Business Magazine, 2025), las ya mencionadas metodologías tradicionales promueven una plantilla de trabajo en donde las tareas están formadas de manera secuencial y estrictamente rígida sin optar por algún cambio. Por este motivo surge una extensa dificultad en las primeras fases que inician y planifican el resto del desarrollo y después de esta fase no existe resiliencia al cambio al ya establecer parámetros y objetivos en las primeras fases.

Por otro lado, las metodologías ágiles buscan una forma de trabajo mucho más flexible ante los cambios o adaptaciones que pueden surgir al momento de desarrollar y hablando de las metodologías ágiles es indispensable hablar de las fases iterativas que posee en este apartado de las metodologías el cliente o stakeholder puede ser parte del equipo y poder retroalimentar cada entrega funcional que se realice. Esto reduce los posibles errores que surgirían sin esta colaboración entre todos los miembros del equipo al

igual que el del cliente (ADEN Business Magazine, 2025).

Por ende, para este proyecto conviene un desarrollo basado en una metodología ágil al querer realizar entregas rápidas, funcionales y para poder validar con el Dr. Miguel Bravo y así poder acabar el desarrollo del prototipo y solventar todos los problemas de la consultoría medica independiente planteados en puntos anteriores.

### **Tabla 1**

*Metodologías ágiles vs metodologías tradicionales*

<b>Metodologías Tradicionales</b>	<b>Metodologías Ágiles</b>
Todo el Proyecto es definido y planificado al inicio, por lo que durante el desarrollo es poco flexible.	La planificación puede ajustarse en cualquier fase según el proyecto lo demande.
El costo de los cambios que surgen en el desarrollo es bastante alto y tienen mucha dificultad para implementarlos.	Los cambios son rutinarios y los requerimientos funcionales y no funcionales pueden ir cambiando.
La única entrega que existe aquí es al culminar el desarrollo del proyecto.	Existen varias entregas funcionales en cada fin de sprint.
El cliente o stakeholder interviene en la etapa final y al inicio del proyecto.	El cliente o stakeholder es un miembro más del equipo.
Es primordial presentar una documentación extensa, detallada y formal antes de empezar a desarrollar.	Es primordial el desarrollo del software antes que la documentación.

**Nota:** La Tabla 1 plantea y compara diferencias intrínsecas que se pueden encontrar al hablar de metodologías ágiles y tradicionales. Información propia a partir de (Universidad Nacional de Trujillo, 2017)

#### **2.3.4 Scrum**

“Scrum es un marco de trabajo ágil a través del cual las personas pueden abordar problemas complejos adaptativos a la vez que se entregan productos de forma eficiente y creativa con el máximo valor. Así, Scrum es una metodología que ayuda a los equipos a

colaborar y realizar un trabajo de alto impacto. La metodología Scrum proporciona un plan de valores, roles y pautas para ayudar a tu equipo a concentrarse en la iteración y la mejora continua en proyectos complejos. Por otra parte, deberás saber que en Scrum se trabaja con equipos pequeños multidisciplinares en ciclos iterativos centrados en el cliente y se crea un producto de forma incremental.” (Asana, 2025)

#### **2.3.4.1 Beneficios de Scrum**

Scrum tiene varios beneficios; para empezar, es indispensable hablar de la adaptabilidad que se desglosa al trabajar con este marco de trabajo ágil. Como ya se habló anteriormente en el trabajo, se sabe que las organizaciones y equipos deben tener estrategias de contingencia frente al cambio que se da día a día en el mundo del desarrollo de software.

Las entregas en Scrum, al contrario de otras metodologías que son tardadas y dependen de mucho tiempo, son rápidas y con un constante mejoramiento. Esto quiere decir que, al momento de implementar los sprints, se tiene una entrega al final de este y se pueden hacer aportaciones y, sobre todo, retroalimentar con ayuda del cliente y todo el equipo que colabora en tiempo real.

Otra gran ventaja de Scrum, como ya se habló un poco en el punto anterior, es la colaboración que existe al optar por este marco de trabajo. Como se indica en la Tabla 1, en las metodologías tradicionales la participación del cliente es casi nula, por lo que solo actuaba al inicio y al final del desarrollo, y esto ocasiona falta de confianza entre el equipo y el cliente y podría generar errores que solo se podían ver al final del desarrollo, en la entrega del producto. Por lo tanto, al optar por esta metodología ágil como es Scrum, el cliente está en constante participación con el equipo, retroalimentando y aportando en continuidad con el desarrollo de software.

#### **2.3.4.2 Aplicación de Scrum para el desarrollo del prototipo de sistema web para el consultorio del Dr. Miguel Bravo**

Para el desarrollo en el que está centrado el proyecto de titulación, como ya se mencionó antes, se opta por Scrum, por el hecho de que permite trabajar de manera

colaborativa con el usuario final, de forma iterativa y flexible. En este caso de estudio, el Dr. Miguel Bravo será el usuario final y, a su vez, tendrá el rol de Product Owner.

El proyecto se dividirá en sprints, en los cuales se realizará un desarrollo corto de diferentes segmentos del sistema. Al final de cada uno de estos sprints se presentarán al usuario pequeñas versiones que contarán con una parte funcional del sistema, con el fin de mantener siempre retroalimentación y colaboración con el usuario.

Todo esto con el objetivo de que el producto final desarrollado y presentado se adapte directamente a las necesidades reales del consultorio médico independiente, y que los posibles cambios que surjan al final de cada sprint no afecten el desarrollo en general ni resulten perjudiciales en términos de pérdida de tiempo, entre otros aspectos.

Se definieron los siguientes roles para tener una mejor organización y entendimiento al implementar Scrum:

- Product Owner: Dr. Miguel Bravo, iniciará y definirá lo primordial en el desarrollo, dará su visto bueno y sobre todo su retroalimentación en cada nueva versión del prototipo presentada al final de cada sprint.
- Scrum Master: El autor del proyecto, es el encargado de asignar y definir los tiempos para cada actividad y debe garantizar que se cumpla con la estructura y las normas de Scrum.
- Development Team: El autor del proyecto, es el responsable de diseñar, crear, validar e integrar todo lo referente o necesario del prototipo de sistema web.

En cada sprint se visualizará el desarrollo de una actividad funcional del prototipo, donde se priorizan aquellas funcionalidades críticas. En este contexto, el uso de Scrum permitirá revisar el progreso de cada actividad propuesta y asignada por el Scrum Master. A su vez, permitirá identificar o mitigar de manera rápida los errores que puedan surgir en cada sprint o versión presentable, promoviendo la mejora continua y adaptándose más a la visión del usuario.

### **2.3.5 Pruebas de software en el desarrollo del prototipo**

Dentro del campo de cómo se construye el software, testear es una parte clave para asegurar que un programa cumpla con todo lo que se había planeado, tanto en lo que tiene que hacer como en su calidad. Hay muchas clases de chequeos, por ejemplo, probar piezas sueltas, probar cómo se juntan, ver si funciona lo que debe, si la gente lo acepta, si va rápido o si es seguro, por nombrar algunos; cada uno mira algo concreto y tiene un propósito claro.

Con esta versión preliminar del sistema web para manejar a los pacientes, se le ha dado más importancia a probar cómo se unen las partes y a probar si los usuarios finales están contentos, ya que esto encaja mejor con lo que se quiere lograr y con la urgencia de confirmar que las funciones esenciales del sistema anden bien.

Debido al alcance y estructura del proyecto se especificarán dos tipos de pruebas que más adelante serán las responsables de evaluar el prototipo del sistema final.

### **2.3.5.1 Pruebas de Integración**

Las pruebas de integración son una clase de evaluación de software diseñados para comprobar que las distintas piezas o secciones de un programa trabajan bien cuando interactúan unas con otras. A diferencia de las pruebas unitarias, que solo miran fragmentos separados, estos se enfocan en descubrir problemas que salen al cruzarse el frente, la parte de atrás y el almacén de datos.

En este prototipo de sistema web, estos chequeos verifican recorridos completos, como cuando el usuario registra algún paciente o alguna cita, se procesa esa información y luego se guarda o se saca otra vez. El punto es garantizar que las partes del sistema como el de los paciente, citas o informes funcionen de forma pareja y que todo lo planeado se haga tal como se pidió.

### **2.3.5.2 Pruebas de Aceptación**

Las pruebas de aceptación tienen la meta de ver si el programa cumple o no con lo que se acordó y si lo que hace el software le sirve de verdad al que lo va a usar. Estos

chequeos los hace el cliente o el dueño del producto, revisando que cada cosa que hace sirva para lo que se necesita en el proyecto.

La idea de este chequeo es confirmar que este sistema sirve, se entiende fácil y funciona en situaciones parecidas a las de cuando se usará normalmente. Como es el caso de la metodología rápida estas pruebas se pueden realizar al acabar un sprint y verificar que la entrega de esa semana cumple con lo pactado y poder seguir con el siguiente sprint.

## **2.4 Herramientas**

A continuación, se visualiza una lista de las herramientas a utilizar para poder llevar a cabo el desarrollo del prototipo de sistemas web utilizando tecnología moderna que brindan ayuda en el nivel o etapa de desarrollo en donde se encuentre el proyecto.

- SQL Server (Motor de Base de datos)
- SQL Server Management Studio (Gestor de Base de datos)
- Visual Studio 2022 (IDE)
- C# (Lenguaje de Programación)
- ASP.NET Core (Framework BackEnd)
- Visual Studio Code (IDE)
- HTML/CSS
- JavaScript (Lenguaje de Programación)
- Figma (Diagramas)
- PowerDesigner (Modelado de Datos)
- Entity Framework Core (ORM)

### **2.4.1 SQL Server**

Es un motor de base de datos relacional, fue desarrollado por Microsoft. La función principal de SQL Server es almacenar y organizar mucho volumen de información de manera estructurada y relacional. Una ventaja total de SQL Server es que se puede visualizar, eliminar, actualizar e insertar datos con el uso de SQL qué significa Structured Query Lenguaje.

Es un motor de base que ofrece seguridad, disponibilidad e integridad de la data

almacenada.

**Tabla 2**

*Componentes y tecnologías de SQL Server*

Componentes	Descripción
Motor de base de datos	El motor de base de datos es el servicio principal para almacenar, procesar y proteger datos.
Replicación	SQL Server Replication consiste en un conjunto de tecnologías para copiar y distribuir datos y objetos de base de datos de una base de datos a otra y, a continuación, sincronizar las bases de datos para mantener la coherencia.
Data Quality Services (DQS)	Data Quality Services proporciona una solución de limpieza de datos controlada por conocimiento.
Integration Services (SSIS)	SQL Server Integration Services Incluye paquetes que proporcionan procesamiento de extracción, transformación y carga (ETL) para almacenamiento de datos.

**Nota:** La tabla 2 Indica algunos componentes y tecnologías que SQL Server brinda al usuario. Adaptado de Microsoft. (2024). *¿Qué es SQL Server?* En *Microsoft Learn*. Recuperado el 28 de octubre de 2025, de <https://learn.microsoft.com/es-es/sql/sql-server/what-is-sql-server?view=sql-server-ver17>.

Por todo este contexto se escogió SQL Server para que sea el motor de base de datos del prototipo de sistema web para la gestión de pacientes y citas del consultorio independiente del Dr. Miguel Bravo. No solamente por sus ventajas sino también por el tipo de arquitectura, pero de eso se hablará en cada herramienta.

## 2.4.2 SQL Server Management Studio

”SQL Server Management Studio (SSMS) es un entorno integrado para administrar cualquier infraestructura de SQL. Use SSMS para acceder a todos los componentes de SQL Server, Azure SQL Database, Azure SQL Managed Instance, SQL Server en VM de Azure y Azure Synapse Analytics, así como para configurarlos, administrarlos y desarrollarlos. SSMS ofrece una única utilidad integral que combina un amplio grupo de herramientas gráficas con una serie de editores de script enriquecidos que permiten a desarrolladores y administradores de bases de datos de todos los niveles acceder a SQL Server.” (Microsoft, 2024)

Se optó por SQL Server Management Studio para ser el gestor de base de datos de Microsoft SQL Server debido a la compatibilidad, por todas las herramientas que incluye (aunque en este proyecto no se las vaya a explotar), también porque integra todos los componentes que se necesitan al gestionar la base de datos, cómo escribir consultas SQL y administrar la seguridad por roles de esta.

En este caso, es útil para crear y conectar instancias locales y, a su vez, también brinda un diagrama de relaciones de las tablas para que sea más intuitivo el funcionamiento y las relaciones entre entidades de la base de datos.

A continuación, se visualizará una tabla donde se mostrarán algunos componentes que brinda SQL Server Management Studio (SSMS) al escogerlo como DBMS.

**Tabla 3**  
*SQL Server Management Studio*

Descripción	Componente
Uso del Explorador de objetos para ver y administrar todos los objetos de una o más instancias de SQL Server.	Explorador de objetos
Use el Explorador de plantillas para compilar y administrar archivos de texto reutilizables que se pueden usar para acelerar el desarrollo de consultas y scripts.	Explorador de plantillas
Use el Explorador de soluciones en desuso para compilar proyectos que se emplean para administrar elementos de	Explorador de soluciones

administración, como scripts y consultas.	
Use las herramientas de diseño visual incluidas en SSMS para compilar consultas, tablas y bases de datos de diagrama.	Visual Database Tools
Use los editores de lenguaje de SSMS para compilar y depurar interactivamente consultas y scripts.	Editores de consultas y texto

**Nota:** Fuente: Microsoft. (2024). *SQL Server Management Studio (SSMS)*. En *Microsoft Learn*. Recuperado el 28 de octubre de 2025, de <https://learn.microsoft.com/es-es/ssms/sql-server-management-studio-ssms>

### 2.4.3 Visual Studio 2022

Visual Studio 2022 es una IDE (Integrated development environment) lo significa que es un software o una aplicación que brinda al usuario un ambiente lleno de herramientas que sirve a los desarrolladores para programar, compilar y depurar aplicaciones. Esta IDE fue creada por Microsoft y simplifica en gran porcentaje el trabajo de los programadores al permitir codificar, validar el código y poder publicarlo o desplegarlo de manera rápida y eficaz.

Las principales ventajas de Visual Studio 2022 son la posibilidad de utilizar varios lenguajes de programación dependiendo de su necesidad y uso, como por ejemplo en este proyecto, C#. Al utilizar este lenguaje más esta IDE, brinda un eficaz desarrollo para sistemas basadas en .NET, además de que viene con un depurador integrado y es compatible con algunos frameworks modernos, como es el caso de este proyecto, ASP.NET Core.

El motivo principal por el que se escogió esta IDE es debido a las necesidades del desarrollo del prototipo de sistema web, por la compatibilidad en los frameworks y lenguajes de programación. Brinda ayuda para programar de manera eficiente y rápida, lo cual, al ser un desarrollo implementado con Scrum, es de vital ayuda para poder entregar versiones funcionales que no demoren mucho tiempo. Por otra parte, también es compatible con el motor de base de datos escogido, SQL Server.

#### **2.4.4 C#**

C# Es un lenguaje de programación moderno, usado igualmente para la programación orientada a objetos, permite la creación de sistemas escalables, seguras y con buen rendimiento.

Su sintaxis es totalmente estructurada y desde el punto de vista del autor fuertemente tipeado comparándolo con otros lenguajes de programación como por ejemplo Python, pero esto se ve como una virtud por el hecho de permitir al desarrollador programar de manera ordenada y entendible al revisar el código.

Por todo este contexto y los beneficios se optó a C# como lenguaje de programación para el desarrollo del prototipo de sistema web propuesto en el proyecto.

#### **2.4.5 ASP.NET Core**

ASP.NET Core es un marco de trabajo o framework que permite crear sistemas web rápidos, seguras y modernas. Este framework fue creado para trabajar en el ambiente de .Net que a su vez es compatible con el lenguaje de programación seleccionado.

En resumen, se utilizó ASP.NET Core por las siguientes razones:

- Escalabilidad y óptimo rendimiento.
- Soporte para Apis.
- Compatibilidad con Visual Studio 2022 y C#.

#### **2.4.6 Visual Studio Code**

Visual Studio Code es una IDE igualmente permite codificar, depurar y ejecutar código en varios lenguajes de programación. En este proyecto se utilizó esta IDE para realizar toda la construcción del Frontend. En resumidas cuentas, se escogió esta IDE para ser el ambiente donde se construya el FrontEnd del prototipo por las siguientes razones:

- Afinidad con varios lenguajes de programación
- Multiplataformas
- Implementación de HTML, CSS con facilidad.

#### **2.4.7 HTML/CSS**

En este proyecto se utilizó HTML y CSS para crear la interfaz de usuario (UI) por en resumidas cuentas por lo siguiente:

- Su implementación sirve para cualquier tipo de navegador web.
- Permite crear interfaces que sean escalables y adaptables a largo plazo.
- Es fácil de integrar estas herramientas conjunto a JavaScript para consumir las Apis, pero de JavaScript se hablará más a fondo en el siguiente punto.

### **2.4.8 JavaScript**

JavaScript es un lenguaje de programación de alto nivel que es conocido por ser la herramienta con la que se pueda hacer dinámica o funcional una página web.

En este proyecto, se implementó JavaScript en el frontend para consumir las Apis que se desarrollaron anteriormente en el backend con ASP.NET Core. En sí intrínsecamente JavaScript ayuda a mejorar o asegurar que el usuario tenga una agradable experiencia al utilizar el prototipo de sistema web. Se escogió JavaScript por las siguientes razones:

- Dinamiza las páginas estáticas mejorando la experiencia del usuario.
- Es compatible y se puede consumir las Apis del backend.
- Es compatible con varios navegadores.

### **2.4.9 Figma**

Figma es una herramienta web y también de escritorio que sirve para realizar interfaces gráficas de usuario (UI) de manera unitaria o colaborativa. Esta herramienta también permite la creación de diagramas UML, como por ejemplo diagramas de caso de uso, de secuencia, de clases, etc. Lo que es algo indispensable al realizar un desarrollo de software y todo tenga que ser definido o analizado en las primeras fases y por ende todo debe estar correctamente documentado.

En resumen, se escogió Figma por las siguientes razones:

- Recursos o herramientas incluidas como bibliotecas reutilizables para la

elaboración de UML.

- Fácil de maniobrar ya sea el sistema web o de escritorio.

### **2.4.10 Power Designer**

Power Designer es una hermanita para diseñar, crear y estructurar bases de datos con la opción de hacerlo de manera gráfica. La función intrínseca de esta herramienta es permitir al usuario crear modelos entidad-relación, que son la base fundamental para la creación completa de la base de datos.

En este proyecto propuesto se implementó Power Designer para los diseños de la base de datos, por un lado, el lógico y por otro el físico. Esto genera un modelo que describe las entidades principales, que serían cada tabla, sus atributos, que serían las columnas de cada tabla, y las relaciones que existen entre las tablas. Por estos hechos se pudo obtener una vista entendible y completamente estructurada de cómo se organizará la información que se genere del prototipo de sistema web y, a su vez, sirve para realizar las documentaciones técnicas del caso.

Por todas estas razones, y más aun tratándose de un desarrollo desde cero, es necesario y fundamental partir de una buena base, que en este caso sería la base de datos. Si creamos la base de datos con algún tipo de error o problema al no utilizar esta herramienta, o por no usar ninguna herramienta de este estilo, tendría grandes repercusiones en el futuro del desarrollo, lo que puede alentar los tiempos de entrega y causar más trabajo innecesario.

## **2.5 Marco Conceptual**

En el desarrollo de este sistema web se utilizan varios términos que son necesarios de entender y comprender, cómo el área de la salud es muy extensa, se especifica en el nicho de la gestión del paciente y de igual manera con la tecnología en los sistemas web. Conocer estas terminologías permite establecer un claro entendimiento al desarrollar el prototipo propuesto.

### **2.5.1 Consultorio médico independiente**

“Al hablar de una atención de salud como médico independiente nos referimos a hacer una atención o consulta médica sin una institución de por medio (consultorio, centro

médico, clínica, hospital u otros), es decir, de manera directa con el paciente.” (Equipo Médico QuickHealth, 2025). Como se menciona es la forma en que un médico puede atender a un paciente de manera autónoma sin contar con algún organismo de por medio.

### **2.5.2 Requerimientos funcionales y no funcionales**

Según (QRA Corp, 2023) , los requerimientos funcionales indican que intrínsecamente hará o debe hacer el sistema en sí, es decir las funciones, procesos, que permiten cumplir con las necesidades del usuario al ingresar al sistema.

### **2.5.3 Base de datos**

“Una base de datos es una recopilación de datos sistemática y almacenada electrónicamente. Puede contener cualquier tipo de datos, incluidos palabras, números, imágenes, vídeos y archivos.” (Amazon Web Services (AWS), 2023)

### **2.5.4 IDE**

“Un entorno de desarrollo integrado (IDE) es una aplicación de software que ayuda a los programadores a desarrollar código de software de manera eficiente.” (Amazon Web Services (AWS), 2023)

### **2.5.5 Framework**

“Un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software.” (UNIR Formación Profesional, 2023)

### **2.5.6 Arquitectura de Microservicios**

“Una arquitectura de microservicios divide una aplicación en una serie de servicios implementables de forma independiente que se comunican a través de API. Este enfoque permite implementar y escalar cada servicio individual de forma independiente, así como

la entrega rápida y frecuente de aplicaciones grandes y complejas. A diferencia de una aplicación monolítica, una arquitectura de microservicios permite a los equipos implementar nuevas funciones y hacer cambios más rápido, sin tener que volver a escribir una gran parte del código existente.” (Atlassian, 2023)

### **2.5.7 Prototipo**

“Un prototipo es un primer modelo que sirve como representación o simulación del producto final y que nos permite verificar el diseño y confirmar que cuenta con las características específicas planteada.” (Angeles-Angeles, 2020)

### **2.5.8 Api**

“Una API (application programming interface), o interfaz de programación de aplicaciones, es un conjunto de reglas o protocolos que permiten que las aplicaciones de software se comuniquen entre sí para intercambiar datos, características y funcionalidades” (IBM, 2023)

### **2.5.9 Backend**

“El backend es la parte lógica de un sitio, este se encarga de la lógica de negocio, de recibir y devolver datos procesados a las apps y sitios web, de forma que facilite la navegación y se garantice el funcionamiento y la seguridad de diferentes funciones.” (Ken, 2023). También se puede entender como la parte que hace que cualquier sistema web o aplicación web sea funcional.

### **2.5.10 Frontend**

“El término frontend hace referencia a la interfaz gráfica de usuario (GUI) con la que los usuarios pueden interactuar de forma directa, como los menús de navegación, los elementos de diseño, los botones, las imágenes y los gráficos. En términos técnicos, una página o pantalla que el usuario ve con varios componentes de la interfaz de usuario se denomina modelo de objetos del documento (DOM). (Amazon Web Services, n.d.). Otra forma de entenderlo es la parte visual que sirve para que la parte del Backend tenga sentido para el usuario.

## CAPÍTULO 3: Análisis de Requerimientos y Modelado del Sistema

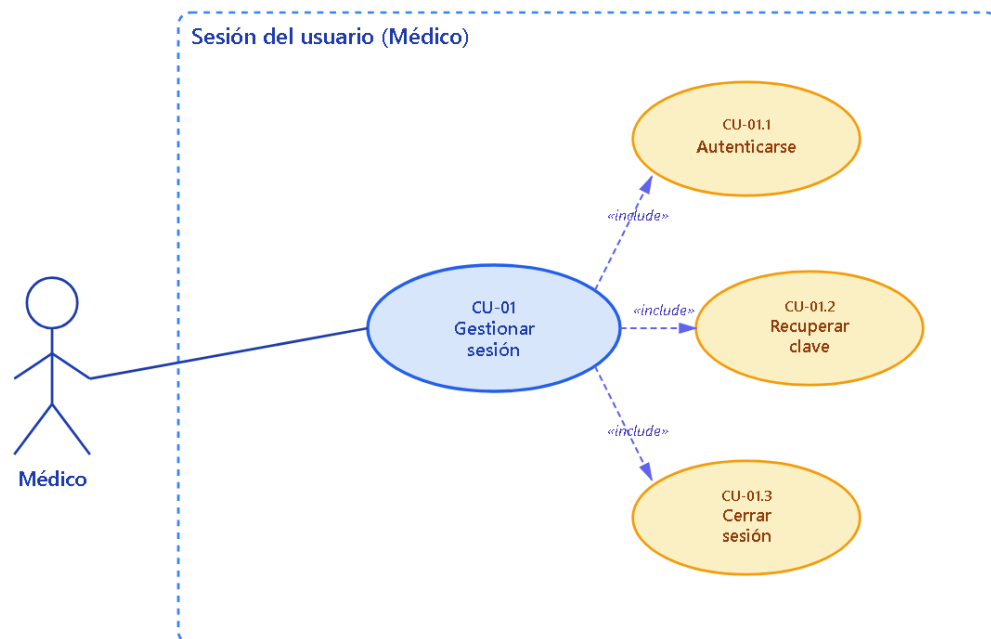
### 3.1 Requerimientos del prototipo de sistema web

Como visión general los requerimientos generales que necesita el prototipo de sistema web es la gestión de la cuenta de los usuarios que cuente con seguridad para poder ingresar y cerrar la sesión, as u vez se necesita la gestión total del paciente y también poder generar una cita para el mismo. Poder monitorear las citas por semana y en cada cita poder generar un informe para el paciente en la cita.

### 3.2 Casos de Uso

Los casos de estudio sirven para identificar a los usuarios y como interactúan con el sistema, también se describe el flujo normal y alterno.

#### 3.2.1 CU-01 Sesión del usuario (Médico)



#### Figura 1

*Caso de uso para la sesión de usuario*

**Nota:** Diagrama del caso de uso para el módulo de sesión de usuario. Elaboración propia en Figma (2025).

**Actor:**

- Médico

**Descripción:**

- El caso de uso CU-01 indica cómo el actor interactúa con el módulo para gestionar su sesión (acceso al sistema) esto incluye las opciones de autenticarse, cerrar la sesión y recuperar la clave.

**Propósito:**

- El objetivo es garantizar un inicio y cierre de sesión correcto y a su vez la opción de recuperar la contraseña funcional en caso de olvidarla.

**Precondiciones:**

- El usuario debe estar previamente registrado en el sistema.

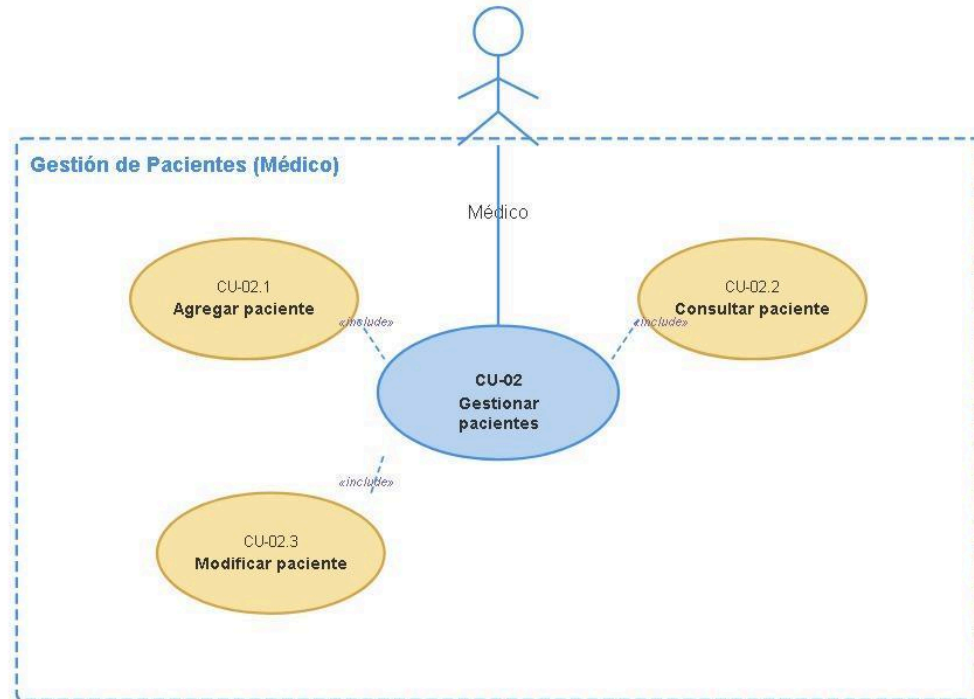
**Flujo normal:**

1. El médico ingresa al sitio web y la primera interfaz será la de Inicio de sesión.
2. Ingresa sus credenciales.
3. Se validan las credenciales ingresadas (Seguridad).
4. Inicio de sesión completo y pasa a la página principal del sistema.

**Flujo alterno:**

1. Si las credenciales no se validan se mostrará un mensaje al usuario de error para que pueda volver a ingresar sus credenciales.
2. Si el usuario no recuerda su contraseña podrá recordar la clave con la opción de recuperar esta.

### 3.2.2 CU-02 Gestionar Paciente



**Figura 2**

*Caso de Uso para la gestión de paciente*

Nota: Diagrama del caso de uso para el módulo de gestionar pacientes. Elaboración propia en Figma (2025).

**Actor:**

- Médico

**Descripción:**

- El caso de uso CU-02 indica cómo el actor describe las acciones que el médico podría realizar gestionar toda la información del paciente dentro del prototipo de sistema web.

**Propósito:**

- Dar funcionalidad al usuario para poder manipular y gestionar toda la información del paciente.

### Precondiciones:

- El usuario debe haber iniciado sesión correctamente para poder utilizar estos apartados.

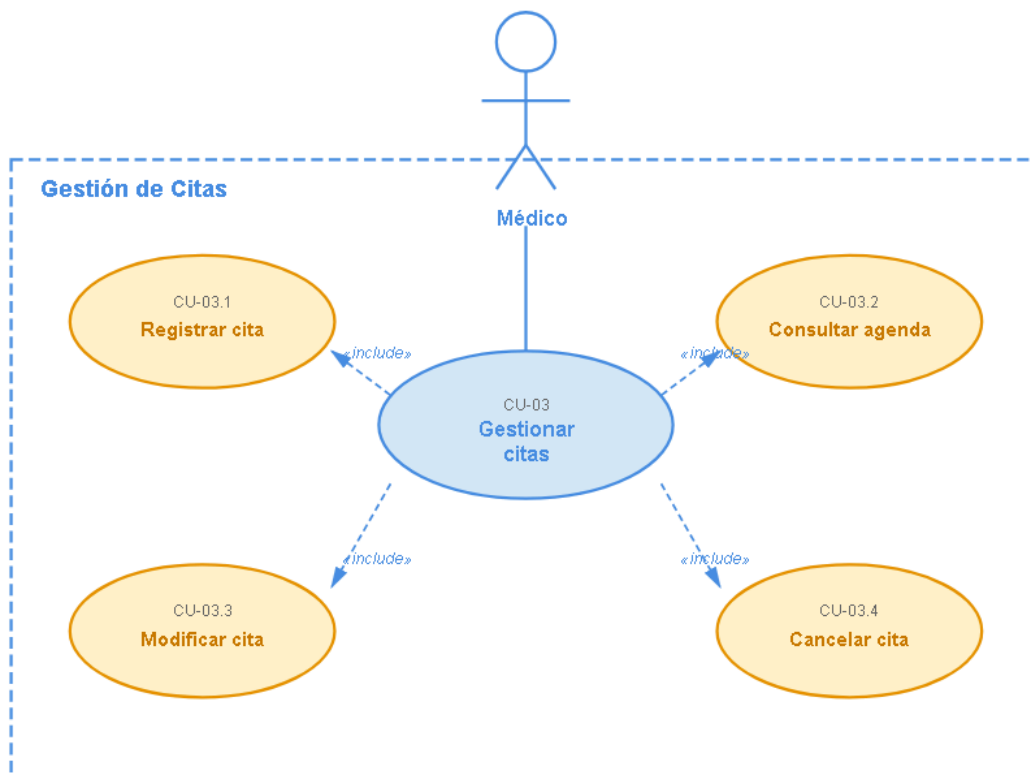
### Flujo normal:

1. El usuario recurre al apartado de ingresar nuevo paciente.
2. El usuario ingresa los datos del paciente.
3. El sistema guarda y registra al paciente en la base de datos usando el sistema.
4. Al crear el usuario correctamente el usuario podrá consultarlo o modificarlo si se requiere.

### Flujo alterno:

1. Si se intentó agregar un paciente nuevo con el mismo número de cédula el sistema indicará un error en el ingreso de los datos.

### 3.2.3 CU-03 Gestionar la Cita para el paciente



### **Figura 3**

#### *Caso de Uso para la gestión de citas*

Nota: Diagrama del caso de uso para el módulo de gestión de citas del paciente.

Elaboración propia en Figma (2025).

#### **Actor:**

- Médico

#### **Descripción:**

- El caso de uso CU-03 indica las acciones que el usuario (médico) puede realizar para las citas de cada paciente. En resumen, este módulo permitirá crear una cita para el paciente, consultar las citas que se tengan a la semana como un calendario, modificar la cita ya sea en hora o en fecha y si por alguna razón se deba cancelar la cita se tendrá la opción de eliminarla para poder agendarla si es el caso.

#### **Propósito:**

- Permite al médico registrar, modificar, consultar o eliminar una cita médica de algún paciente que se tenga en el sistema.

#### **Precondiciones:**

- El usuario debe haber iniciado sesión correctamente para poder utilizar estos apartados.
- El paciente debe estar previamente registrado en el sistema.

#### **Flujo normal:**

1. El usuario recurre al apartado de registrar cita.
2. El usuario deberá ingresar la fecha, hora el tipo de atención si es a domicilio o en el consultorio y el paciente.
3. El sistema guarda la cita.
4. El usuario puede acceder al calendario para ver las citas programadas para la semana o ver anteriores.

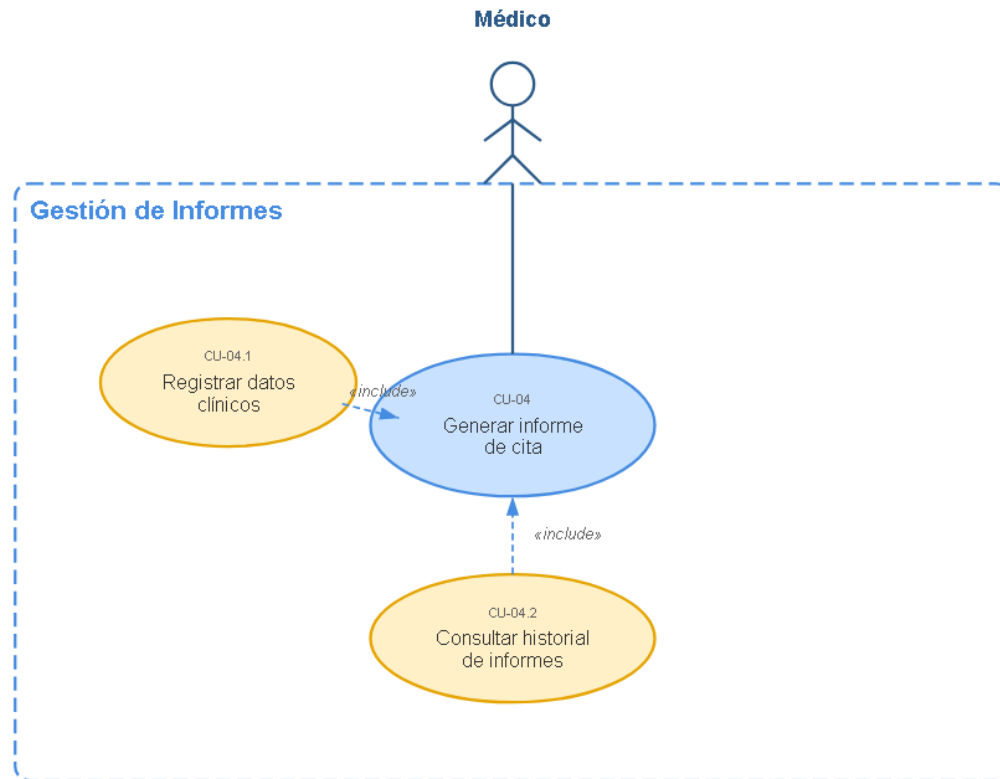
#### **Flujo alterno:**

1. Si el usuario quiere agendar una cita en una fecha u hora ocupadas deberá salir

un mensaje de aviso o error.

2. El sistema permite al usuario modificar la fecha y hora de la cita o eliminarla si es necesario.

### 3.2.4 CU-04 Generar informe de la cita



#### Figura 4

*Caso de Uso para la gestión de informes*

Nota: Diagrama del caso de uso para el módulo de gestión de informes a partir de la cita. Elaboración propia en Figma (2025).

#### Actor:

- Médico

#### Descripción:

- El caso de uso CU-04 describe el proceso por el cual el médico puede registrar y emitir un informe detallado de la cita en particular. Este módulo es intrínseco para la cita.

**Propósito:**

- Permite al médico ingresar datos clínicos de una cita y generar un informe digital.

**Precondiciones:**

- El usuario debe haber iniciado sesión correctamente para poder utilizar estos apartados.
- La cita debe estar previamente registrada en el sistema.

**Flujo normal:**

1. El usuario ingresa los datos clínicos del paciente e información en sí del paciente o tratamiento en la ficha.
2. El sistema guarda esa información en la base de datos y la asocia a la cita correspondiente.
3. El usuario puede imprimir el informe.

**Flujo alterno:**

1. Si el sistema identifica un error al generar el archivo, se mostrará un mensaje de error o aviso.

### 3.3 Funcionalidades

Para poder validar las funcionalidades del prototipo de sistema web para el consultorio médico independiente se realizó una reunión con el Product Owner que a su vez es el médico usuario final del sistema web. Durante esta reunión se revisaron las partes críticas y prioritarias del sistema.

En esta reunión se dieron a conocer las principales funcionalidades del sistema en base a los casos de uso ya establecidos en el punto anterior Como son la gestión tanto de la sesión del usuario como la de la información de los pacientes, citas e informes generados.

Todas estas funcionalidades se convirtieron en el soporte para poder extraer los requerimientos tanto funcionales como no funcionales que se detallarán en los siguientes apartados.

En la parte de anexos, se visualizará el acta de las reuniones correspondientes al

levantamiento de los requerimientos realizando un trabajo colaborativo con el Product Owner.

### 3.3.1 Requerimientos funcionales

A continuación, se detallarán los requerimientos funcionales identificados con el producto owner y a partir de los casos de uso establecidos en puntos anteriores, a su vez se asociará el requerimiento funcional por un código.

**Tabla 4**

*Tabla de descripción de los requerimientos funcionales del sistema*

<b>Código</b>	<b>Descripción del Requerimiento Funcional</b>	<b>Prioridad</b>
<b>RF-01</b>	El sistema debe permitir al médico iniciar sesión mediante el ingreso de credenciales como usuario y contraseña.	Alta
<b>RF-02</b>	El sistema debe validar las credenciales ingresadas para permitir el acceso.	Alta
<b>RF-03</b>	El sistema debe tener la opción de recuperar la contraseña.	Baja
<b>RF-04</b>	El sistema debe permitir cerrar la sesión.	Alta
<b>RF-05</b>	El sistema debe permitir al médico registrar nuevos pacientes ingresando sus datos personales.	Alta
<b>RF-06</b>	El sistema debe guardar y registrar la información del paciente en la base de datos.	Alta
<b>RF-07</b>	El sistema debe permitir consultar la información de los pacientes registrados.	Alta
<b>RF-08</b>	El sistema debe permitir modificar la información de un paciente existente.	Alta
<b>RF-09</b>	El sistema debe permitir registrar nuevas citas médicas para pacientes ya existentes	Alta
<b>RF-10</b>	El sistema debe guardar las citas con su fecha, hora, tipo de atención (consultorio o domicilio) y paciente asignado.	Alta
<b>RF-11</b>	El sistema debe mostrar las citas programadas por semana o día en formato de calendario.	Media
<b>RF-12</b>	El sistema debe permitir modificar o cancelar una cita existente.	Media
<b>RF-13</b>	El sistema debe mostrar un mensaje de error si se intenta registrar una cita en un horario ocupado.	Alta
<b>RF-14</b>	El sistema debe permitir generar informes clínicos asociados a cada cita.	Baja
<b>RF-15</b>	El sistema debe permitir registrar observaciones del paciente en el informe.	Baja
<b>RF-16</b>	El sistema debe guardar la información del informe y asociarla a la cita correspondiente.	Media

Nota: La Tabla 4 describe los requerimientos funcionales extraídos a partir de los casos de uso y los organiza por su código.

### 3.3.2 Requerimientos no funcionales

A continuación, se detallan los requerimientos no funcionales que debe tener el prototipo de sistema web para la gestión de pacientes del Dr. Miguel Bravo.

**Tabla 5**

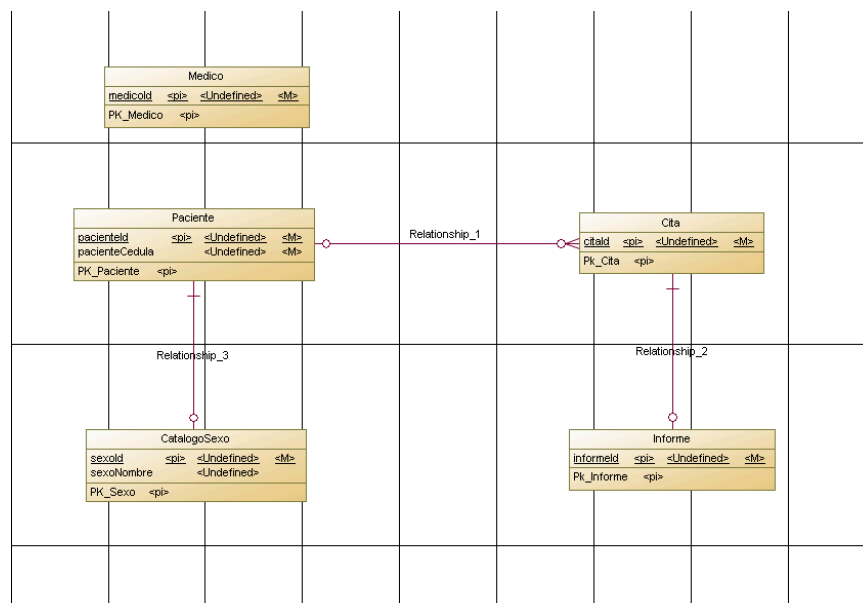
*Tabla de descripción de los requerimientos no funcionales del sistema*

Código	Descripción del Requerimiento No Funcional
RNF-01	El sistema tiene que asegurar la integridad de los datos con medios de verificación, manejo de entrada y salida.
RNF-02	El sistema debe permitir únicamente a usuarios que estén registrados en el sistema poder acceder a las funcionalidades del sistema.
RNF-03	El tiempo de respuesta del sistema al realizar las funcionalidades no debe pasar de 5 segundos para cada acción como insertar y modificar paciente o cita.
RNF-04	El sistema debe almacenar los datos e información de las entidades (pacientes, citas, etc.) de manera íntegra en una base de datos relacional.
RNF-05	El sistema debe tener una interfaz intuitiva.
RNF-06	El sistema debe permitir navegar correctamente y con sentido hacia las funcionalidades del sistema.
RNF-07	El sistema debe mostrar mensajes de error.
RNF-08	El sistema debe tener dashboards interactivos con la base de datos.

Nota: La Tabla 5 describe los requerimientos no funcionales.

### 3.4 Modelos de la Base de Datos

#### 3.4.1 Modelo Conceptual



## Figura 5 Modelo Conceptual

Nota: Modelo Conceptual de la base de datos. Elaboración propia en PowerDesigner (2025).

### 3.4.2 Modelo físico

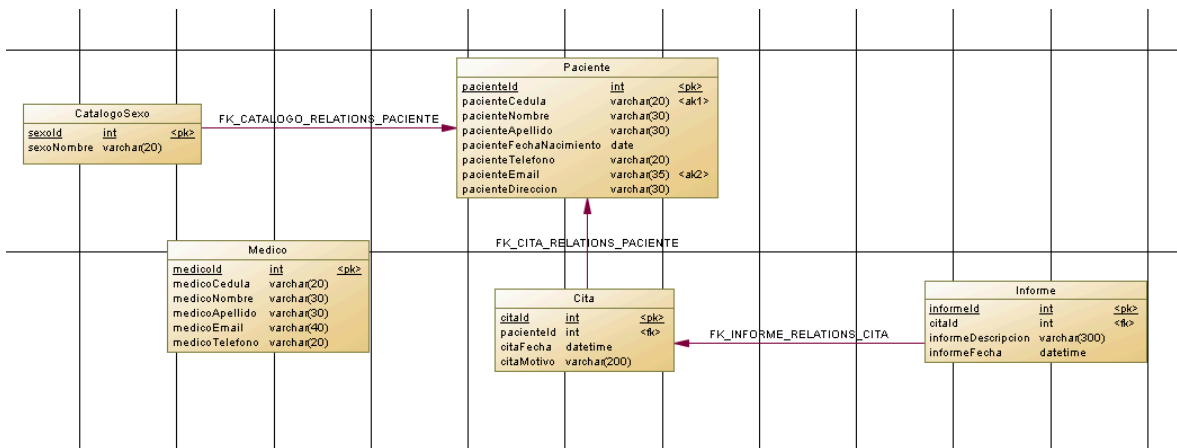


Figura 6 Modelo Físico

Nota: Modelo físico de la base de datos. Elaboración propia en PowerDesigner (2025).

### 3.5 Tabla de trazabilidad del sistema

Tabla 6

Tabla de trazabilidad entre objetivos, requerimientos, casos de uso y pruebas

Objetivo	Requerimiento	Caso de Uso	Tipo de Prueba
Objetivo General	RF-01	CU-01	Integración / Aceptación
Objetivo General	RF-02	CU-01	Integración / Aceptación
Objetivo General	RF-03	CU-01	Integración / Aceptación
Objetivo General	RF-04	CU-01	Integración / Aceptación
Objetivo General	RF-05	CU-02	Integración / Aceptación
Objetivo General	RF-06	CU-02	Integración / Aceptación
Objetivo General	RF-07	CU-02	Integración / Aceptación
Objetivo General	RF-08	CU-02	Integración / Aceptación
Objetivo General	RF-09	CU-03	Integración / Aceptación
Objetivo General	RF-10	CU-03	Integración / Aceptación
Objetivo General	RF-11	CU-03	Integración / Aceptación
Objetivo General	RF-12	CU-03	Integración / Aceptación
Objetivo General	RF-13	CU-03	Integración / Aceptación
Objetivo General	RF-14	CU-04	Integración / Aceptación

Objetivo General	RF-15	CU-04	Integración / Aceptación
Objetivo General	RF-16	CU-04	Integración / Aceptación

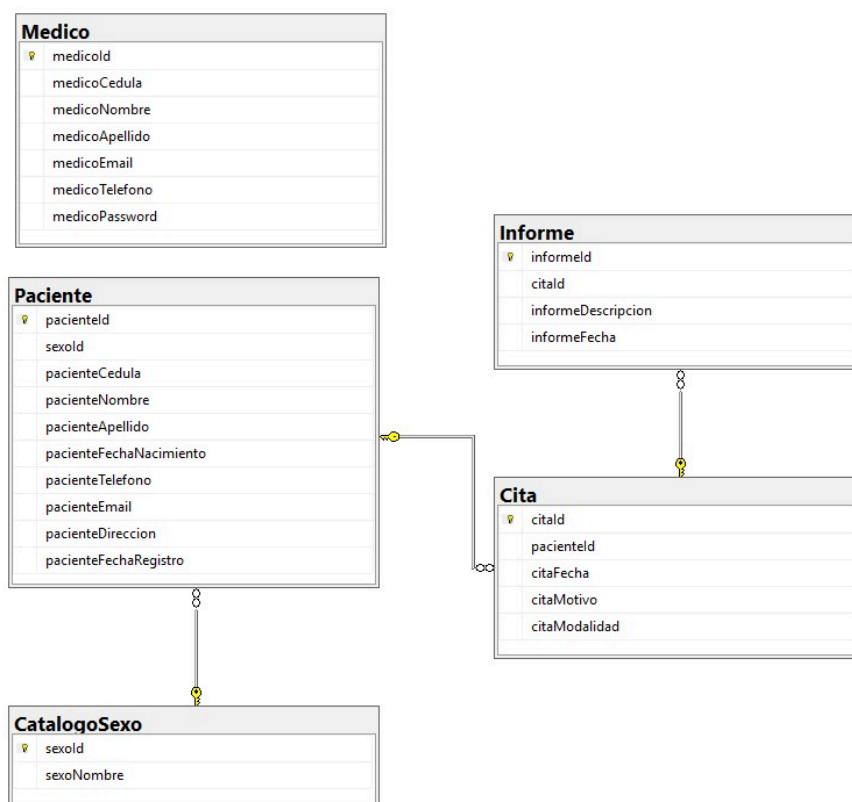
---

Nota: La Tabla 6 ayuda a observar el seguimiento entre la meta principal del proyecto, las necesidades funcionales planteadas, los casos de uso unidos y las clases de pruebas hechas para confirmar la operación correcta del prototipo de sistema web.

## CAPÍTULO 4: Desarrollo del prototipo de sistema web

### 4.1 Implementación de la base de datos

Como se habló en el capítulo 2 como motor de base de datos se utilizó MySQL Server y con Power designer se modeló la base de datos lo que fue esencial para poder visualizar los modelos de la base de datos indicados en el capítulo 3 y por ende también se pudo obtener el script de la base de datos. Por lo que inmediatamente se ejecutó el script y por ende se crearon las tablas, claves primarias, claves foráneas, restricciones, etc.



**Figura 7**

*Base de datos montada en el motor*

Nota: Diagrama de la base de datos montada en MySQL Server (2025).

Las entidades, atributos, claves primarias y foráneas se crearon para asegurar la integridad de los datos respetando las formas normales buscando normalizar la tabla para asegurar que la información del paciente se almacene correctamente.

Para validar escenarios funcionales donde actuará la base de datos se insertaron

algunos registros en las tablas para poder apreciar de mejor manera las relaciones entre cada tabla y el funcionamiento general de la base de datos como tal.

	pacienteld	sexold	pacienteCedula	pacienteNombre	pacienteApellido	pacienteFechaNacimiento	pacienteTelefono	pacienteEmail	pacienteDireccion
1	2	2	1712345678	Andrea	Cárdenas	1997-09-22	0987456321	andrea.cardenas@gmail.com	Av. América y M. de Jesús
2	3	1	0912345678	Carlos	Ramírez	1995-05-10	0987654321	carlos.ramirez@gmail.com	Av. Amazonas y Colón
3	4	2	1754297568	Carla	Tinajero	2011-01-19	0996659867	carla@gmail.com	Armenia
4	5	1	1102233445	Luis	Paredes	1990-03-15	0987654123	luis.paredes@example.com	Av. República y Colón
5	6	2	1305566778	María	Ortiz	1993-07-22	0998877665	maria.ortiz@example.com	Calle Los Álamos 234
6	7	1	1729988776	Pedro	Salinas	1988-11-05	0960011223	pedro.salinas@example.com	Urbanización El Bosque, Mz. B

### Figura 8

Registros de prueba (Paciente)

Nota: Registros de prueba tabla Paciente (2025).

	citald	pacienteld	citaFecha	citaMotivo	citaModalidad
1	2	2	2025-11-11 09:30:00.000	Chequeo general anual	Consultorio
2	3	2	2025-11-13 11:30:00.000	Control post consulta	Virtual
3	4	2	2025-11-12 19:30:00.000	Tratamiento	Domicilio
4	5	5	2025-11-12 15:00:00.000	Dolor lumbar	Consultorio
5	6	6	2025-11-13 10:30:00.000	Control post tratamiento	Virtual
6	7	7	2025-11-10 14:00:00.000	Evaluación inicial	Domicilio

### Figura 9

Registros de prueba (Cita)

Nota: Registros de prueba tabla Paciente (2025).

	informeld	citald	informeDescripcion	informeFecha
1	2	2	Paciente en buen estado general. Se recomienda s...	2025-11-14 00:00:00.000

### Figura 10

Registros de prueba (Informe)

Nota: Registros de prueba tabla Informe (2025).

	medicold	medicoCedula	medicoNombre	medicoApellido	medicoEmail	medicoTelefono	medicoPassword
1	1	1706575048	Miguel	Bravo	miguelbravob@hotmail.com	0998595365	123

### Figura 11

Registros de prueba (Médico)

Nota: Registros de prueba tabla Médico (2025).

	sexold	sexoNombre
1	2	Femenino
2	1	Masculino

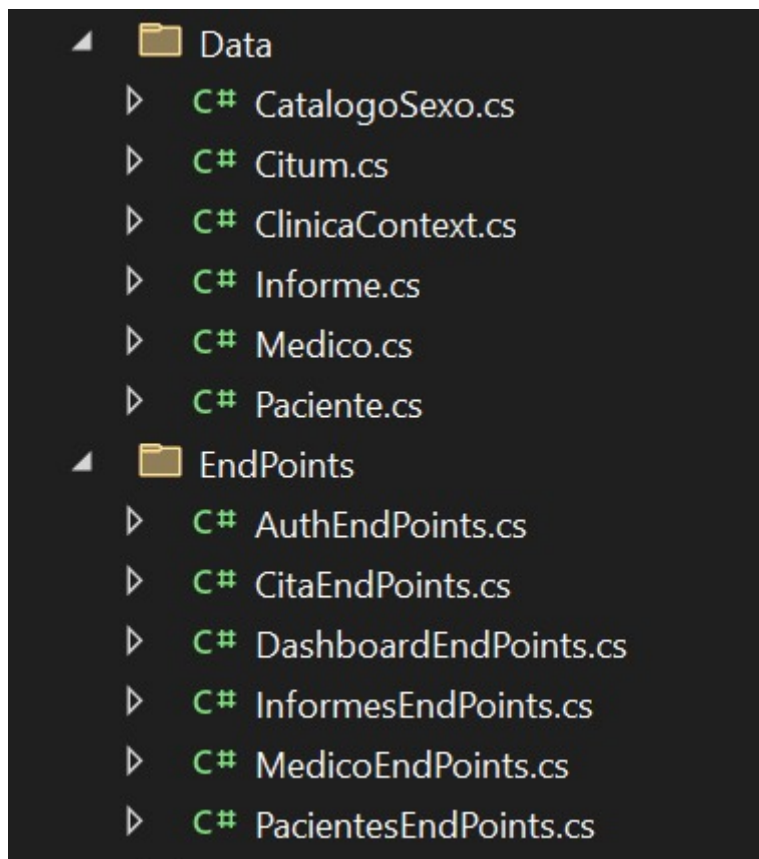
### Figura 12

Registros de prueba (Catálogo)

Nota: Registros de prueba tabla CatálogoSexo (2025).

## 4.2 Implementación del backend

Para la construcción del backend se creó un proyecto ASP.Net Web Api, por el hecho de la herramientas y facilidad de crear endpoints y poder probarlos con swagger que es una herramienta para probar estas Apis sin tener que utilizar un software aparte como postman, etc. Se generaron los modelos con comandos desde el cmd de visual studio 2022 lo que construyó la carpeta de datos con las clases que referencia nuestras entidades en la base de datos para luego proceder a crear los endpoints en otra capa. Esto permite al sistema ser escalable frente a nuevas actualización o funcionalidades sin tener que realizar grandes cambios.

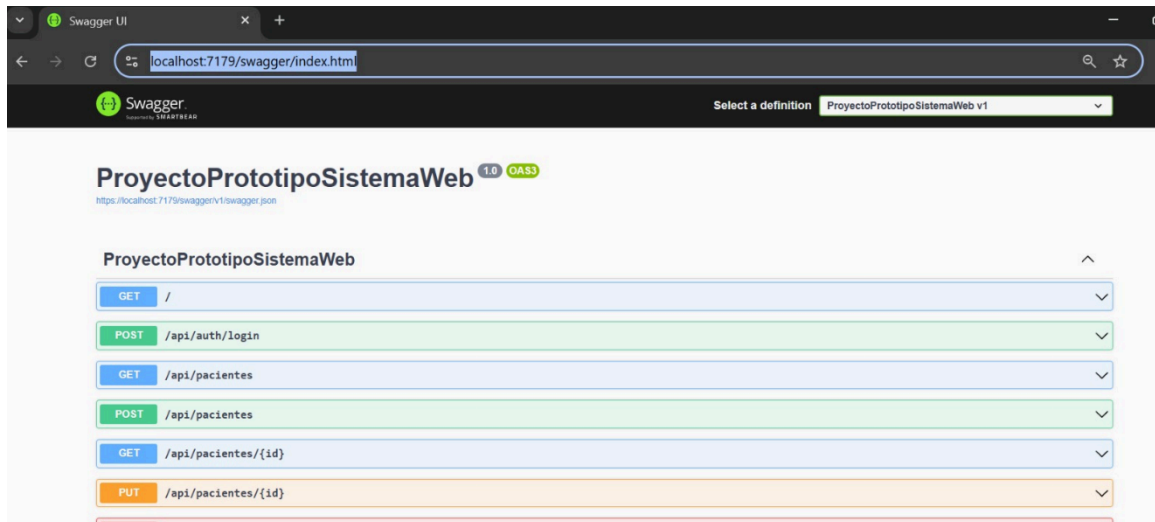


**Figura 13**

*Backend*

Nota: Carpeta data generada a partir de las entidades usando la conexión a la base de datos local y carpeta de endpoints (2025).

Para los endpoints es importante destacar que se creó un CRUD como tal para cada entidad lo que quiere decir que el api brinda servicio de consultar la tabla ya sea completa o por parámetros, también permite eliminar, actualizar e insertar registros en la tabla.

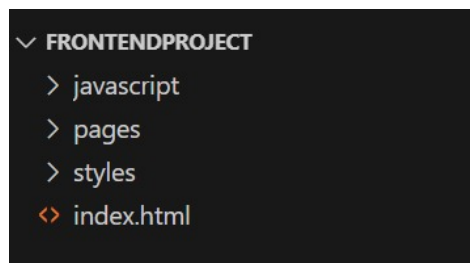


**Figura 14**  
*Herramienta Swagger para probar las Apis*

Nota: Herramienta Swagger donde se visualizan y prueban las Apis generadas para consumirlas (2025).

### 4.3 Implementación del frontend

Para la construcción del frontend como se mencionó en el capítulo 2 se utilizará como IDE a visual studio code entonces lo primero que se realizó fue crear el proyecto y organizar la estructura de este para poder implementar todos los componentes necesarios para la construcción de las interfaces intuitivas y escalables.



**Figura15**  
*Estructura FronEnd*

Nota: Estructura para el FrontEnd en Visual Studio Code (2025).

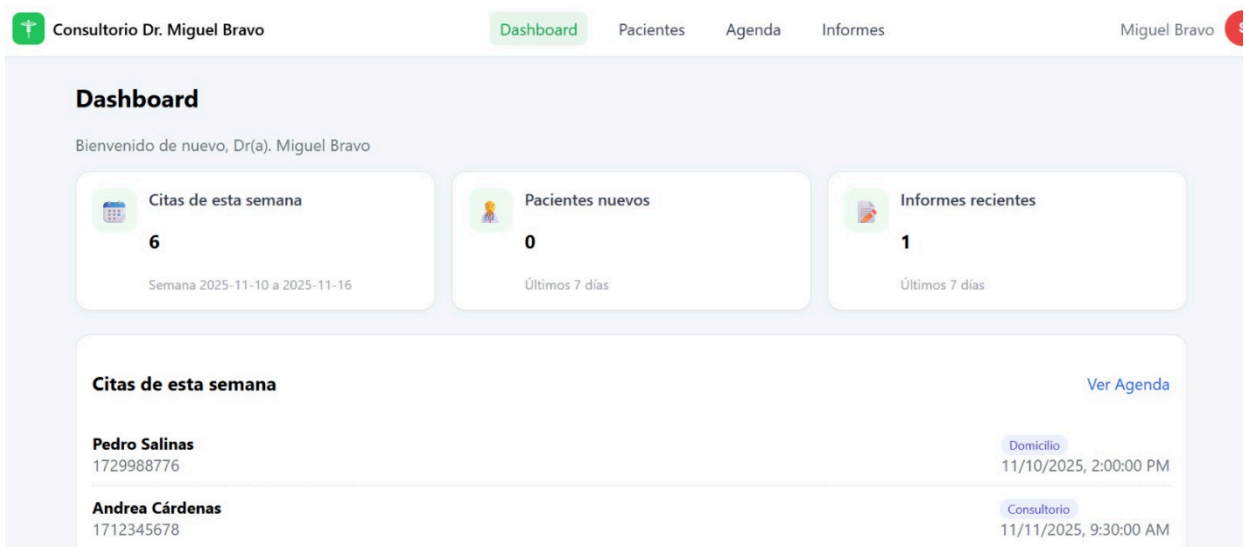
### 4.3.1 Interfaz de Login



**Figura 16**  
*Interfaz de Login*

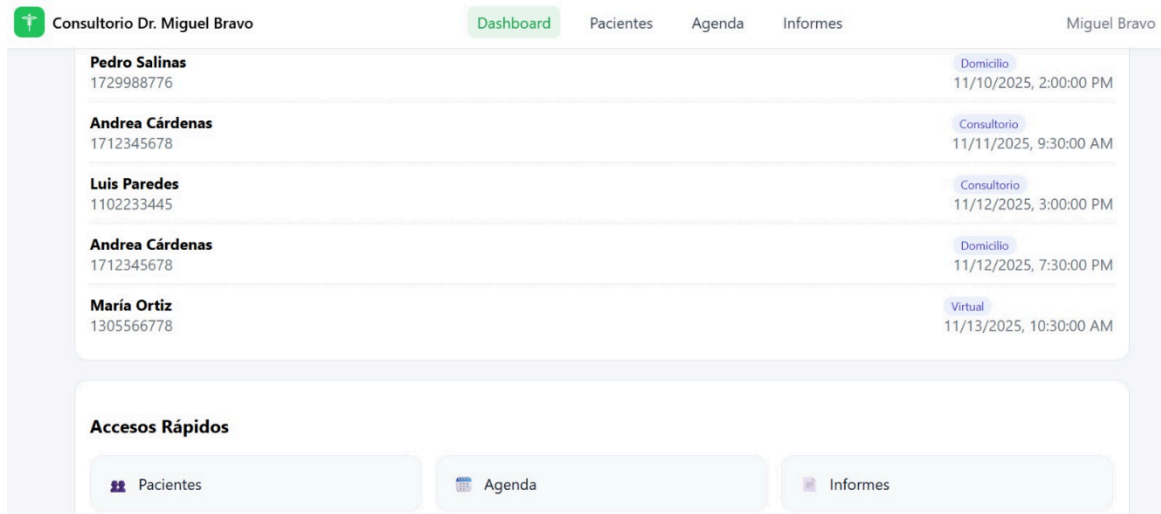
Nota: Interfaz de login(2025).

### 4.3.2 Interfaz de Menú Principal



**Figura 17**  
*Interfaz de Menú Principal*

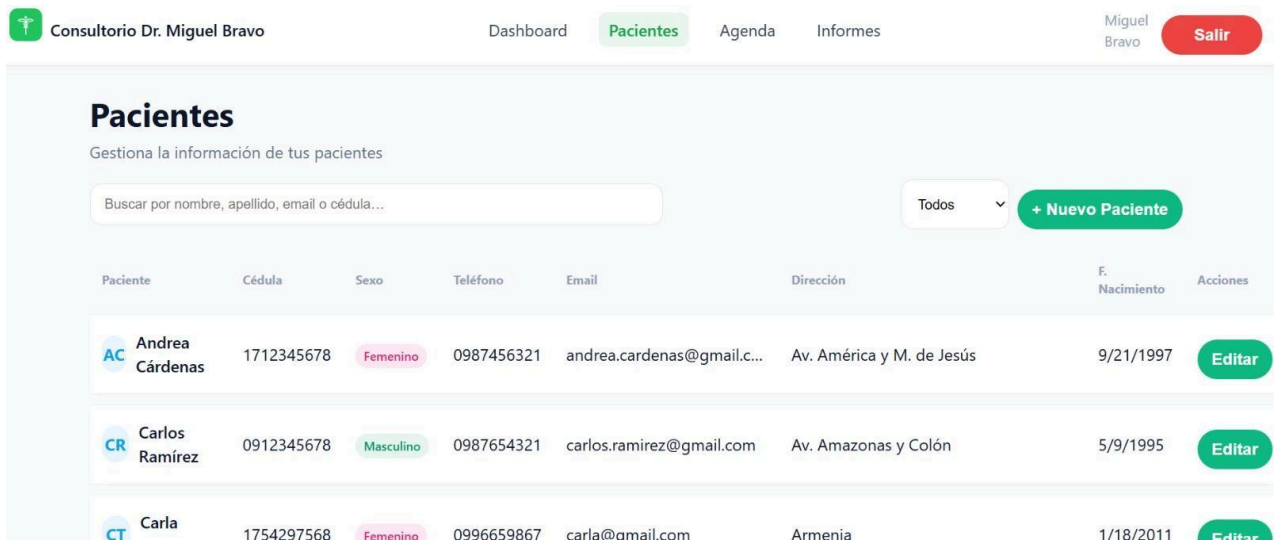
Nota: Interfaz del menú principal después de iniciar sesión correctamente (2025).



**Figura 18**  
*Interfaz de Menú Principal parte inferior*

Nota: Imagen que muestra la parte inferior de la interfaz del menú principal después de iniciar sesión correctamente (2025).

### 4.3.3 Interfaz para la gestión de los pacientes



**Figura 19**  
*Interfaz Pacientes*

Nota: Interfaz donde se gestionará la información de los pacientes (2025).

### 4.3.4 Interfaz para agregar un nuevo paciente

The screenshot shows a form titled "Nuevo Paciente" with the subtitle "Registra los datos del paciente". The form contains several input fields: "Nombre" and "Apellido" (text boxes), "Cédula" (text box), "Sexo" (dropdown menu with "Selecciona..." selected), "Fecha de nacimiento" (text box with "mm/dd/yyyy" placeholder and a calendar icon), "Teléfono" (text box), "Email" (text box), and "Dirección" (text box). At the bottom right, there are two buttons: "Cancelar" (light blue) and "Guardar" (green).

**Figura 20**  
*Interfaz para agregar nuevos pacientes*

Nota: Interfaz donde el usuario podrá agregar nuevos pacientes (2025).

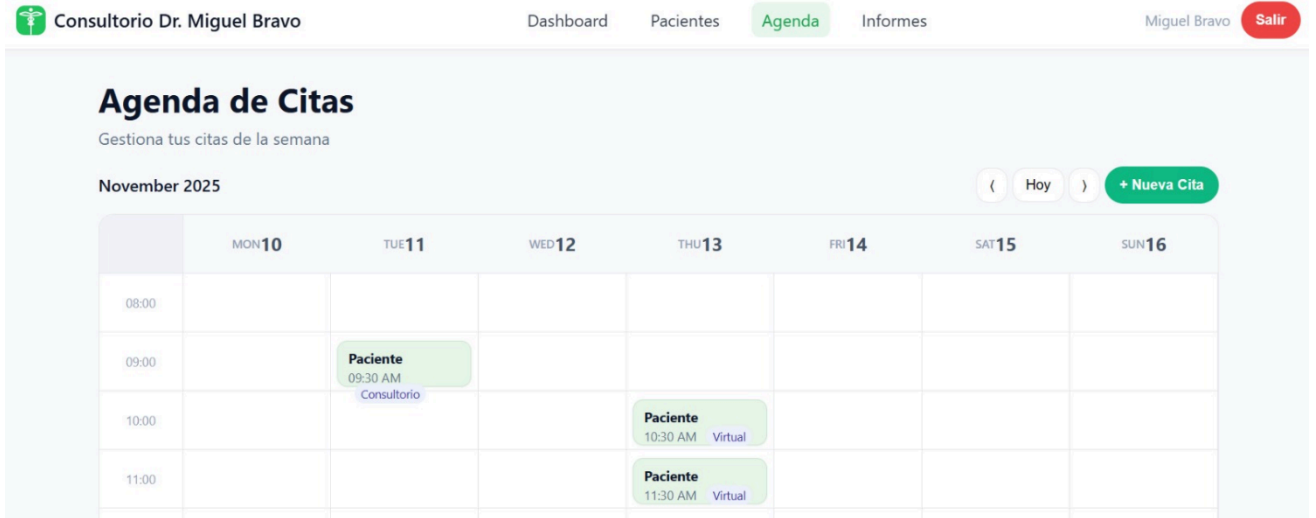
### 4.3.5 Interfaz de la modificación de paciente

The screenshot shows a form titled "Actualizar Paciente" with the subtitle "Modifica los datos del paciente seleccionado". The form contains several input fields with pre-filled data: "Nombre" (Andrea), "Apellido" (Cárdenas), "Documento de Identidad: Cédula / Pasaporte / Ruc" (1712345678), "Sexo" (Femenino), "Fecha de nacimiento" (09/22/1997), "Teléfono" (0987456321), "Email" (andrea.cardenas@gmail.com), and "Dirección" (Av. América y M. de Jesús). At the bottom right, there are two buttons: "Cancelar" (light blue) and "Guardar Cambios" (green).

**Figura 21**  
*Interfaz de la modificación del paciente*

Nota: Interfaz donde el usuario podrá modificar la información del paciente (2025).

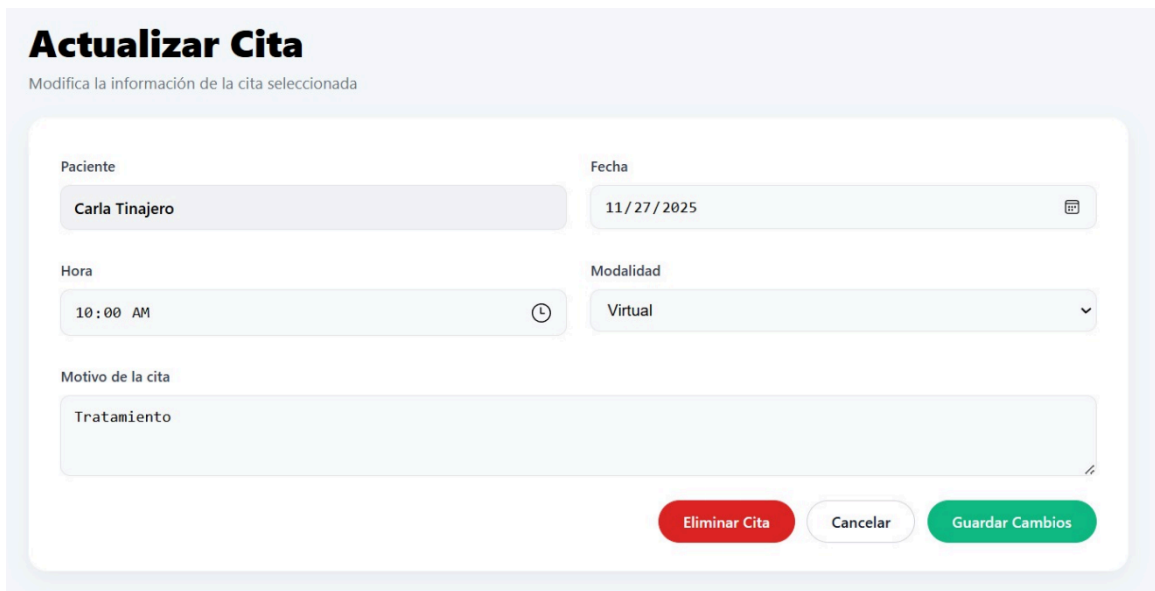
### 4.3.6 Interfaz de la agenda



**Figura 22**  
*Interfaz de la agenda*

Nota: Interfaz donde se visualiza un calendario con las citas del usuario (2025).

### 4.3.7 Interfaz de modificación de las citas



**Figura 23**  
*Interfaz para la modificación de la cita seleccionada*

Nota: Interfaz donde se puede modificar la información de la cita y se puede eliminar si es que se desea (2025).

### 4.3.8 Interfaz para asignar una nueva cita al paciente

**Nueva Cita**  
Crea una cita para un paciente existente.

Paciente  
Busca por nombre o cédula  
Sin paciente seleccionado

Fecha: 11/13/2025  
Hora: 11:35 PM

Modalidad: Consultorio  
Motivo: Breve descripción

← Regresar a Agenda   Ir al Dashboard   Guardar Cita

**Figura 24**  
*Interfaz para crear una nueva cita*

Nota: Interfaz para asignar una nueva cita a un paciente previamente creado (2025).

### 4.3.9 Interfaz de los informes

**Informes Médicos**  
Gestiona los informes de tus pacientes

+ Nuevo Informe

Buscar por paciente...   mm/dd/yyyy   Buscar   Limpiar

- Chequeo general anual**  
Andrea Cárdenas  
14 nov 2025 • Consultorio
- Consulta General**  
Carla Tinajero  
24 nov 2025 • Virtual
- Tratamiento**  
Carlos Ramírez  
25 nov 2025 • Domicilio

**Figura 25**  
*Interfaz para visualizar los informes generados por el usuario*

Nota: Interfaz donde se visualizan todos los informes generados por el usuario(2025).

### 4.3.10 Interfaz para crear nuevos informes

**Crear Informe**

Cita: Benjamin

Fecha del Informe: 11/27/2025

Al hacer clic se muestran las citas de esta semana. Escribe para buscar por nombre.

Benjamin Bravo • 1754207296 • Domicilio • 11/25/2025 (02:30 PM)

Descripción: Ingrese los detalles del informe médico...

Cancelar Guardar Informe

**Figura 26**  
*Interfaz para crear nuevos informes*

Nota: Interfaz donde el usuario puede generar nuevos informes (2025).

### 4.3.11 Interfaz de recuperar la contraseña

**Restablecer contraseña**

Busca tu usuario por cédula y define una nueva contraseña.

Cédula: 1706575048 Buscar

Miguel Bravo — miguelbravob@hotmail.com

Nueva contraseña: Ingrese la nueva contraseña

Confirmar contraseña: Repita la nueva contraseña

Guardar nueva contraseña

← Volver al inicio de sesión

Médico encontrado. Ahora define la nueva contraseña.

**Figura 27**  
*Interfaz para recuperar la clave*

Nota: Interfaz donde el usuario puede recuperar sus credenciales solo si se encuentra en el sistema (2025).

## **4.4 Desarrollo del proyecto por Sprints**

El desarrollo del prototipo del sistema web fue llevado a cabo mediante una metodología ágil estructurada en sprints, los cuales funcionaron como unidades de trabajo que permitieron organizar las tareas de manera ordenada y asegurar una evolución continua del proyecto, de modo que cada ciclo incorporó fases de planificación, ejecución y revisión que facilitaron la construcción gradual de las funcionalidades previstas dentro del sistema y promovieron un proceso de avance coherente con las necesidades técnicas definidas desde el inicio.

En el siguiente apartado se expone la documentación narrativa correspondiente a cada sprint con la intención de ofrecer una visión detallada del proceso seguido, razón por la cual se incluyen los objetivos establecidos para cada etapa, las actividades efectuadas por el equipo, los registros elaborados durante el desarrollo, las modificaciones introducidas a medida que surgieron nuevas exigencias y el conjunto de pruebas destinadas a verificar el adecuado desempeño del sistema en su versión preliminar.

### **4.4.1 Sprint 1 Modelado y Diseño del sistema**

#### **Objetivo del sprint**

Establecer a fondo la estructura del programa, las funciones necesarias y las funciones para mejorar la experiencia del usuario. Por este contexto se realizaron modelos conceptuales y físicos para tener un esbozo de como seria la estructura de la base de datos la organización del repositorio de datos.

#### **Actividades**

- Definir los requerimientos funcionales y no funcionales del sistema.
- Definir el flujo y capacidad del sistema.
- Construcción del modelo conceptual y físico con Power Designer.
- Creación de interfaces con figma.
- Validar el diseño; y estética de las interfaces con el Producto Owner

## Bitácora



### Figura 28

*Sprint 1 Modelado y Diseño del Sistema*

Nota: Sprint 1 Modelado y Diseño del Sistema creado en Claude (2025).

### Cambios realizados durante el Sprint

- Se agregó el catálogo Sexo para la tabla pacientes
- El tamaño de los botones en la interfaz aumentó por problemas oculares del usuario.

### Pruebas del sprint

- Evaluación de la navegación entre las interfaces.

### Resultados del sprint

- Diseño de la base de datos concretado.
- Interfaces listas solo para agregar las funcionalidades.

## 4.4.2 Sprint 2 Creación de la base de datos y Backend

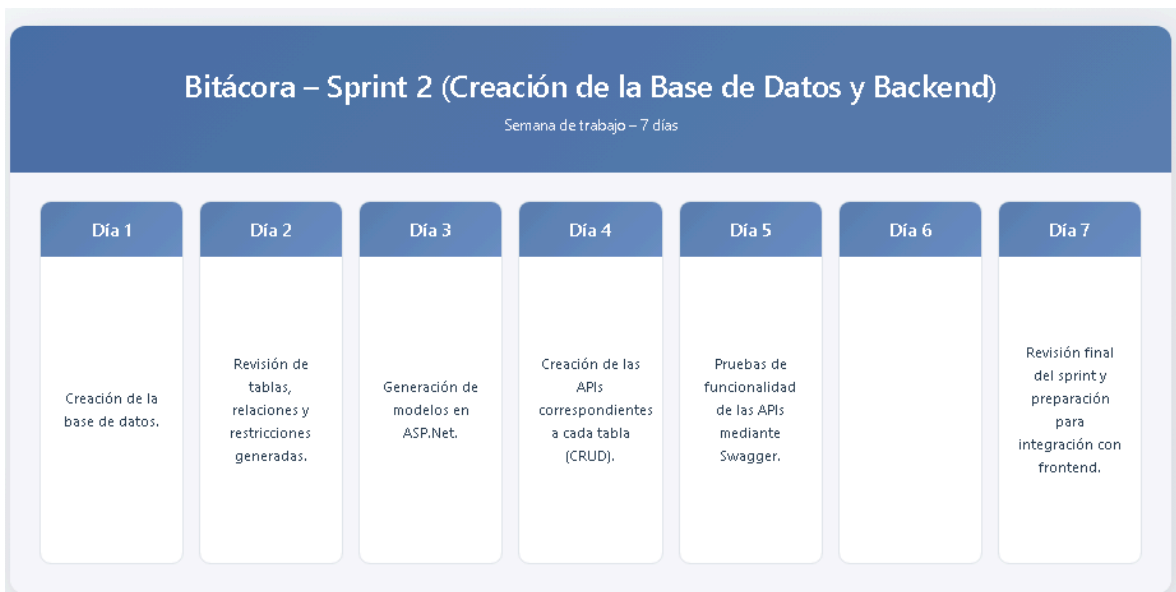
### Objetivo del sprint

En este Sprint se creó la base de datos al tener el modelo físico power designer te facilita el script de creación, y a su vez se programaron las funcionalidades del sistema en base a los requerimientos funcionales y no funcionales y en base a eso se crearon las Apis que próximamente serán consumidas en el frontend.

### Actividades

- Creación de la base de datos en MYSQL Server con el script de power designer.
- Generación de modelos en ASP.Net mediante comandos que conectan la base de datos local con el backend en Visual estudio 2022.
- Creación de las Apis que realizan acciones a partir de cada tabla .
- Pruebas de funcionalidad de la Api mediante Swagger.

### Bitácora



**Figura 29**

*Sprint 2 Creación de la BDD y Backend*

Nota: Sprint 2 Creación de la Base de datos y Backend creado en Claude (2025).

### Cambios realizados durante el Sprint

- Se aumentaron Apis referentes al dashboard y para el resteo de contraseña del médico por cédula.

### **Pruebas del sprint**

- Pruebas del CRUD en swagger.
- Validar la conexión entre la base de datos local y el Backend.

### **Resultados del sprint**

- Backend funcionando correctamente en local.
- Base de datos creada con registros de prueba.

## **4.4.3 Sprint 3 Creación del frontend e integración con el backend**

### **Objetivo del sprint**

Al tener ya las interfaces a partir de figma en este sprint lo que se tenía que realizar es construirlas en base a eso y con el backend ya montado, la parte fundamental para hacer un sistema completamente funcional en base a los requerimientos ya establecidos es integrar el frontend con el backend.

### **Actividades**

- Creación del proyecto FrontEnd en la IDE visual estudio code con la estructura para tener elementos en html, css y javascript.
- Desarrollo de las interfaces (Login, Menú Principal, Gestión de Pacientes, Citas e informes).
- Consumo de las Apis en el Frontend.
- Pruebas con el frontend y backend integrados.

### **Bitácora**



**Figura 30**

*Sprint3 Creación del Frontend e integración con el Backend*

Nota: Sprint 3 Creación del Frontend e integración con el Backend creado en Claude (2025).

### **Cambios realizados durante el Sprint**

- Se vio necesario implementar un nuevo api para la búsqueda de los pacientes por varios filtros (nombre, apellido, cedula, etc.).
- Se colocaron mensajes de error para evitar los errores de base de datos.

### **Pruebas del sprint**

- Pruebas de funcionalidad completa desde el Frontend hasta el backend.
- Validación de la navegación que tendrá el usuario.

### **Resultados del sprint**

- FrontEnd funcionando correctamente con el backend.

## **4.5 Pruebas del Prototipo de Sistema web**

El proceso de pruebas con el objetivo de que la arquitectura frontend y backend se comuniquen y funcionen de manera correcta. Debido a la arquitectura del prototipo de sistema web es necesario pruebas de integración y por otra parte para visualizar el funcionamiento por flujos de manera correcta son necesarias pruebas de aceptación.

## 4.5.1 Pruebas de Integración

Estas pruebas de integración se aplicaron a los 4 módulos principales:

- 1.-Inicio de sesión (Login).
- 2.-Gestión de Pacientes (CRUD).
- 3.-Gestión de Citas (CRUD).
- 4.-Gestión de Informes.

### 4.5.1.1 Inicio de Sesión y navegación inicial

#### **Objetivo:**

- Validar que el usuario pueda ingresar sus credenciales y se validen con las de la base de datos para que el frontend reciba la respuesta correcta.

#### **Flujo probado:**

- El usuario ingresa su correo y clave en la interfaz de login.
- El frontend envía la petición a la API /login vía método POST.
- El backend valida las credenciales en la base de datos.
- La API responde con los datos del médico.
- El frontend redirige al menú principal, cargando el nombre del usuario en pantalla.

#### **Resultado esperado:**

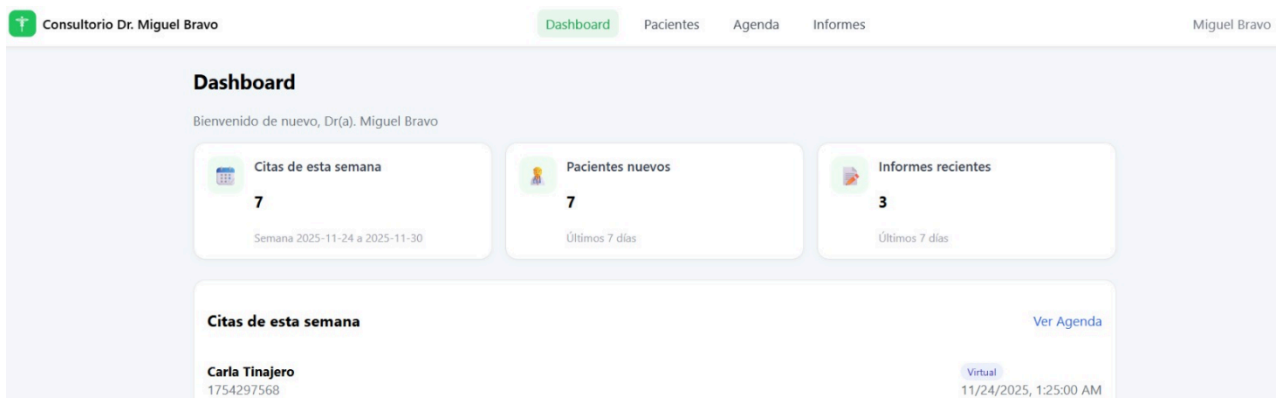
- Acceso exitoso, visualización del menú principal y carga correcta de variables de sesión.

#### **Evidencia:**



**Figura 31**  
*Prueba de Integración Inicio de Sesión*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).



**Figura 32**  
*Prueba de Integración Navegación inicial con la sesión establecida*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).

### 4.5.1.2 Gestión de Pacientes (CRUD)

#### Objetivo:

- Verificar que la creación, consulta, modificación y eliminación de pacientes funciona correctamente desde el frontend hasta la base de datos.

#### Flujo probado:

- Registro de un nuevo paciente desde el formulario.
- Envío de la información a la API /pacientes.
- Almacenamiento del paciente en la base de datos.
- Consulta de la lista de pacientes y visualización del recién creado.
- Modificación de los datos del paciente.
- Eliminación del paciente (si no tiene citas asociadas).

#### Resultado esperado:

- El paciente debe aparecer reflejado en la lista, poder modificarse y eliminarse sin inconsistencias.

#### Evidencia:

### Figura 33

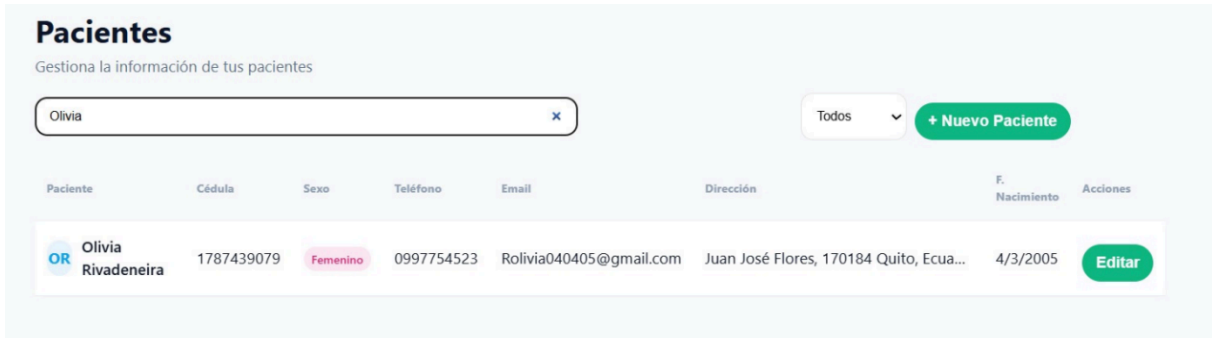
*Prueba de Integración registro de nuevo paciente*

The image shows a web form titled "Nuevo Paciente" with the subtitle "Registra los datos del paciente". The form contains the following fields and values:

Nombre	Apellido
Olivia	Rivadeneira
Cédula	Sexo
1787439079	Femenino
Fecha de nacimiento	Teléfono
04/04/2005	0997754523
Email	
Rolivia040405@gmail.com	
Dirección	
Juan José Flores, 170184 Quito, Ecuador	

At the bottom right of the form are two buttons: "Cancelar" (grey) and "Guardar" (green).

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).



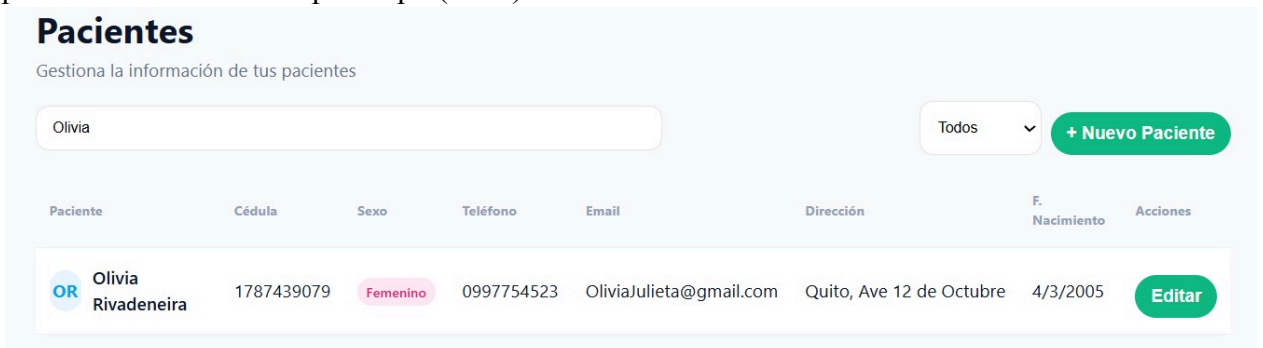
**Figura 34**  
*Pruebas de Integración paciente guardado en la base*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).



**Figura 35**  
*Prueba de Integración Modificación de los datos del paciente*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).



**Figura 36**  
*Prueba de Integración Confirmación de cambio de datos en la base*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).

pruebas funcionales del prototipo (2025).

### 4.5.1.3 Gestión de Citas y Calendario

#### Objetivo:

- Verificar que las citas se registren y se reflejen correctamente en el calendario con la validación de horario por fecha y hora.

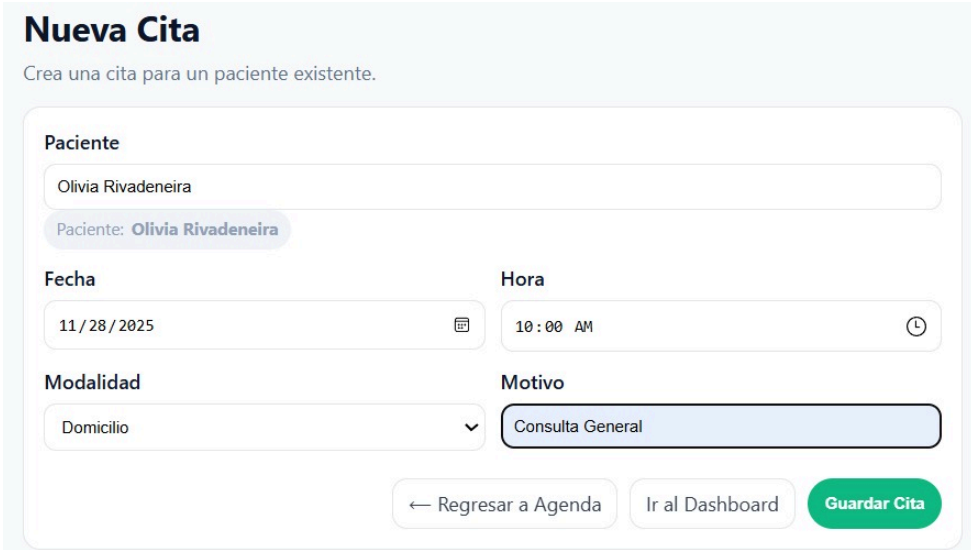
#### Flujo probado:

- Registro de una nueva cita indicando fecha, hora y tipo de atención.
- Validación para evitar duplicidad de horarios.
- Almacenamiento de la cita en la base de datos y visualización de la cita en el calendario semanal.
- Modificación o eliminación de la cita. (si no tiene informes asociados).

#### Resultado esperado:

- La cita debe estar correctamente registrada sin problemas de horario y debe visualizarse en el calendario correctamente con el nombre e información del paciente.

#### Evidencia:



**Nueva Cita**  
Crea una cita para un paciente existente.

**Paciente**  
Olivia Rivadeneira  
Paciente: **Olivia Rivadeneira**

**Fecha** 11 / 28 / 2025 **Hora** 10 : 00 AM

**Modalidad** Domicilio **Motivo** Consulta General

← Regresar a Agenda Ir al Dashboard **Guardar Cita**

#### Figura 37

*Prueba de Integración Creación de nueva cita*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).



**Figura 38**  
*Prueba de Integración cita agendada*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).

**Figura 39**  
*Prueba de Integración Validación del horario*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).

## Actualizar Cita

Modifica la información de la cita seleccionada

Paciente

Fecha

Hora

Modalidad

Motivo de la cita

Eliminar Cita
Cancelar
Guardar Cambios

### Figura 40

*Prueba de Integración Modificación de la cita*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).

## Agenda de Citas

Gestiona tus citas de la semana

November 2025 < Hoy > + Nueva Cita

	MON 24	TUE 25	WED 26	THU 27	FRI 28	SAT 29	SUN 30
08:00							
09:00							
10:00				Carla Tinajero 10:00 AM Virtual			
11:00				Benjamin Bravo 11:00 AM Consultorio			
12:00		Carlos Ramírez 12:00 PM Domicilio			Olivia Rivadeneira 12:30 PM Consultorio		
13:00							

### Figura 41

*Prueba de Integración cita modificada en la agenda*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).

#### 4.5.1.4 Gestión de Informes

##### Objetivo:

- Verificar los informes creados y poder filtrarlos por paciente, y a su vez realizar nuevos informes y modificar los que ya existan.

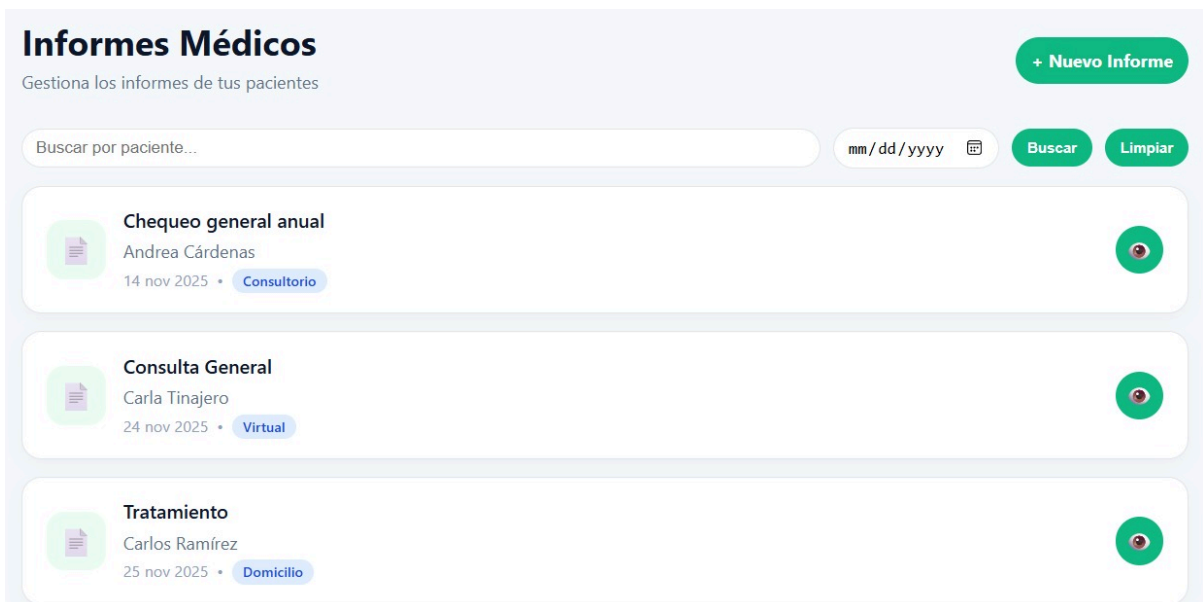
##### Flujo probado:

- Ingreso de un nuevo registro en base a una cita.
- Filtro para buscar por paciente.
- Almacenamiento del informe en la base de datos y visualización en el apartado de informes.
- Modificación o eliminación del informe.

##### Resultado esperado:

- El informe debe estar correctamente registrada y debe visualizarse en el apartado de informes correctamente con el nombre del paciente.

##### Evidencia:



**Figura 42**

*Pruebas de Integración Filtro de búsqueda por paciente*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).

## Crear Informe

**Cita**

Benjamin

Al hacer clic se muestran las citas de esta semana. Escribe para buscar por nombre.

Benjamin Bravo • 1754207296 • Consultorio • 11/27/2025 (11:00 AM) ▾

**Fecha del Informe**

11 / 29 / 2025

La fecha se asigna automáticamente al crear el informe.

**Descripción**

Tratamiento del Covid19 sesión 2

Cancelar Guardar Informe

**Figura 43**

*Prueba de Integración Creación del informe por la cita*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).

## Ver Informe

Imprimir Eliminar Cerrar

**Consultorio Dr. Miguel Bravo**

Sistema de Gestión Médica

**Tratamiento Covid19**

<small>Paciente</small>	<small>Fecha</small>
<b>Benjamin Bravo</b>	<b>29 nov 2025</b>
<small>Modalidad</small>	<small>Motivo</small>
<b>Consultorio</b>	<b>Tratamiento Covid19</b>

Descripción

Tratamiento del Covid19 Sesión 2

**Figura 44**

*Prueba de Integración Visualización de informe creado y modificación de este*

Nota: Evidencia del proceso de integración entre el Frontend y el Backend durante las pruebas funcionales del prototipo (2025).

### 4.5.2 Pruebas de Aceptación

Las pruebas de aceptación se realizar con el Product Owner, revisando si el sistema cumple a los requerimientos funcionales definidos en la Tabla 4. Cada prueba realizada se relaciona con un requerimiento funcional.

### 4.5.2.1 Resultados de pruebas de aceptación

**Tabla 7**

*Resultados de las pruebas de aceptación con código de RF*

<b>Código RF</b>	<b>Requerimiento</b>	<b>Prueba realizada</b>	<b>Resultado</b>
RF-01	Inicio de sesión	Se ingresaron credenciales válidas	Aprobado
RF-05	Registrar paciente	Se agregó un paciente completo	Aprobado
RF-10	Registrar cita	Se creó una cita asociada correctamente	Aprobado
RF-14	Validar horario ocupado	Se intentó crear una cita duplicada	Aprobado
RF-16	Registrar informe	Se completó un informe clínico	Aprobado
RF-17	Asociar informe a cita	Se generó informe vinculado	Aprobado

Nota: Tabla elaborada a partir de las pruebas de aceptación realizadas junto con el Product Owner, verificando el cumplimiento de los requerimientos funcionales establecidos para el prototipo (2025).

### 4.5.2.2 Conclusión de las pruebas

Las pruebas de integración aseguraron que cada parte del software opera de forma unida y armoniosa, siguiendo las pautas de negocio indicadas en los escenarios de uso.

Las pruebas de aceptación hechas con el product Owner mostraron que el borrador satisface las exigencias funcionales puestas, así el programa es considerado bueno, operativo y válido para la meta fijada.

## 4.6 Justificación de las decisiones técnicas

### 4.6.1 Base de datos relacional

La decisión sobre una base de datos relacional hizo posible organizar los datos clínicos y administrativos del sistema de forma lógica, garantizando la integridad referencial entre pacientes, citas e informes. Si se hubiera elegido una base de datos no relacional, el manejo de relaciones complejas habría necesitado más lógica en la aplicación, aumentando la complejidad del backend. Sin embargo, una desventaja de esta estrategia es que, ante un crecimiento masivo del sistema o muchos consultorios simultáneos, habría que estudiar otros mecanismos de escalabilidad y cuando sea el caso hablar de rendimiento.

## **4.6.2 Backend basado en ASP.NET Web API**

La elección de ASP.NET Web API permitió crear un servidor robusto y ordenado, ayudando a hacer puntos finales RESTful y la división de tareas. Si se usara un método único o una tecnología menos fuerte, el soporte y el aumento del sistema serían difíciles ante nuevas funciones.

## **4.6.3 Frontend basado en tecnologías web estándar**

La parte visible se hizo usando HTML, CSS y JavaScript para asegurar que funcione bien con varios navegadores y aparatos. Usar tecnologías comunes ayuda a que el sistema se mantenga fácil; aun así, una posible mejora sería usar estructuras modernas que mejoren el manejo del estado y la reutilización de elementos.

## **4.7 Análisis del prototipo desarrollado**

### **4.7.1 Escalabilidad**

El modelo inicial permite crecer al añadir funciones con nuevos elementos mediante APIs separadas. Sin embargo, al ser un sistema hecho para una consulta individual, hacerlo más grande para muchos usuarios a la vez pediría mejoras extras en su base y soporte.

### **4.7.2 Mantenibilidad**

El dividir el sistema en FrontEnd, Backend y base de datos hace más fácil su cuidado, pues los cambios se pueden hacer sin tocar otras partes. Usar un diseño con APIs ayuda a poner al día funciones precisas sin tener que rehacer todo el sistema.

### **4.7.3 Limitaciones del prototipo**

Entre lo más limitado está que necesita internet para entrar al sistema y que no tiene formas de seguridad muy buenas como pedir más de una clave. Se ven bien estas faltas por

ser un proyecto de estudio, pero se pueden hacer mejoras en versiones que sigan.

## **CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES**

### **5.1 Conclusiones**

1. Respecto al propósito general, se determina que la creación del modelo de sistema web hizo posible automatizar los trámites hechos a mano del consultorio médico aparte del Dr. Miguel Bravo, viéndose una mejoría en el orden de los datos de pacientes, horas e informes médicos. Dicha automatización bajó la posibilidad de fallos humanos, lo cual se probó durante las pruebas de unión y permiso hechas en el Capítulo 4.
2. El uso de una base de datos relacional y una forma basada en APIs ayudó al buen manejo y unión de los datos médicos, dejando una comunicación buena entre el backend y el frontend. Como prueba, se pudo apuntar, mirar y cambiar datos sin errores durante las pruebas prácticas, llenando así los requisitos prácticos dichos en el Capítulo 3.
3. La aplicación de la metodología ágil Scrum dejó un desarrollo seguido y vigilado del modelo, ayudando a la adaptación del sistema a las necesidades reales del consultorio. El apoyo continuo del Product Owner fue clave para aprobar funciones en cada sprint, lo cual se vio en los cambios hechos durante la creación del proyecto.
4. Las pruebas de integración y de aceptación fueron claves para asegurar el buen andar del sistema como un todo. Estas pruebas dejaron ver errores de comunicación entre partes y aprobar que el modelo llenara los propósitos puestos, mostrando que el sistema sirve y es bueno para su uso en un lugar real de consultorio médico aparte.

5. En conclusión, se admite que el modelo creado no piensa en funciones complejas como autenticación multifactor, manejo de muchos médicos o aumento para grandes cantidades de usuarios, ya que estas cosas quedaron fuera del límite dicho para este proyecto académico. Sin embargo, estas faltas no tocan el cumplimiento del propósito mayor del sistema, sino que muestran opciones claras de mejora futura.

## **5.2 Recomendaciones**

### **5.2.1 Recomendaciones técnicas**

1. Recomienda escalar un poco más en la seguridad, como usar varias formas de comprobar quién eres y guardar la información importante de forma secreta, para que los datos de salud guardados estén más seguros.
2. Es aconsejable implementar múltiples usuarios para el sistema y pueda utilizarse en varios consultorios, esto se puede lograr con pequeños cambios técnicos gracias a la escalabilidad actual del sistema.

### **5.2.2 Recomendaciones organizacionales**

3. Es bueno que el consultorio tenga bien organizado cómo guardar copias de seguridad y cómo poner al día la información guardada en el sistema, para que los datos estén disponibles si hay problemas técnicos o cosas inesperadas.
4. Al tratarse de un sistema directamente orientado al ámbito operacional del usuario final, se recomienda instruir al médico para que pueda utilizar el sistema de manera correcta y por ende que sea una herramienta de apoyo más que un desafío.

### **5.2.3 Recomendaciones metodológicas**

5. Al querer realizar sistemas que gestionan información sensible se recomienda utilizar una metodología de desarrollo ágil, como Scrum, debido a su capacidad de adaptarse al cambio y por la importancia de trabajar en conjunto con el cliente en todo el proceso de desarrollo de software.
  
6. Es útil seguir usando herramientas para pensar y diseñar antes de empezar, como Power Designer y Figma, ya que ayudaron a entender bien el sistema desde el principio y ayudaron a que todo fuera más ordenado y eficaz.

## Bibliografía

- ADEN Business Magazine. (2025). *Metodologías ágiles: Qué son y cuáles son las más utilizadas*. Retrieved from ADEN Business Magazine: <https://www.aden.org/business-magazine/metodologias-agiles/>
- ADEN Business Magazine. (2025). *Metodologías ágiles: Qué son y cuáles son las más utilizadas*. Retrieved from ADEN Business Magazine: <https://www.aden.org/business-magazine/metodologias-agiles/>
- Amazon Web Services (AWS). (2023). *¿Qué es Scrum?* Retrieved from Amazon Web Services: <https://aws.amazon.com/es/what-is/scrum/>
- Amazon Web Services (AWS). (2023). *¿Qué es una aplicación web?* Retrieved from Amazon Web Services: <https://aws.amazon.com/es/what-is/web-application/>
- Amazon Web Services (AWS). (2023). *Amazon Web Services*. Retrieved from ¿Qué es una base de datos?: <https://aws.amazon.com/es/what-is/database/>
- Amazon Web Services (AWS). (2023). *Amazon Web Services*. Retrieved from ¿Qué es un entorno de desarrollo integrado (IDE)?: <https://aws.amazon.com/es/what-is/ide/>
- Amazon Web Services. (n.d.). *Front End frente a back-end: diferencia entre el desarrollo de aplicaciones*. Retrieved from AWS: <https://aws.amazon.com/es/compare/the-difference-between-frontend-and-backend/>
- Angeles-Angeles, F. (2020). Prototipo Con-Ciencia. *Boletín Científico de la Escuela Preparatoria No. 3*, 33–34.
- AppMaster. (2024). *5 desafíos comunes en la implementación de registros médicos electrónicos (EHR)*. Retrieved from AppMaster Blog: <https://appmaster.io/es/blog/desafios-de-la-implementacion-de-ehr>
- Asana. (2025). *¿Qué es Scrum?* Retrieved from Asana: <https://asana.com/es/resources/what-is-scrum>
- Atlassian. (2023). *Atlassian*. Retrieved from ¿Qué es la arquitectura de microservicios?: <https://www.atlassian.com/es/microservices/microservices-architecture>
- Avertium. (2024, Febrero 14). *Social Engineering Threats in Healthcare*. Retrieved from Avertium: <https://www.avertium.com/resources/threat-reports/social-engineering-threats-in-healthcare>
- BBVA. (2023). *Metodología Scrum: qué es un sprint*. Retrieved from BBVA: <https://www.bbva.com/es/innovacion/metodologia-scrum-que-es-un-sprint/>
- Blau Consulting Group. (n.d.). *Los 5 errores más comunes en la gestión administrativa de clínicas y cómo evitarlos*. Retrieved from Blau Consulting Group: <https://blau-consultinggroup.com/transformacion-digital/los-5-errores-mas-comunes-en-la-gestion-administrativa-de-clinicas-y-como-evitarlos/>
- Consejo de Europa. (2021). *Marco común europeo de referencia para las lenguas: aprendizaje, enseñanza, evaluación. Volumen complementario*. Ministerio de Educación y Formación Profesional e Instituto Cervantes. Retrieved from [https://cvc.cervantes.es/ensenanza/biblioteca\\_ele/marco\\_complementario/anejo\\_01.htm](https://cvc.cervantes.es/ensenanza/biblioteca_ele/marco_complementario/anejo_01.htm)
- Council of Europe. (2025, Octubre 20). *Common European Framework of Reference for Languages (CEFR)*. Retrieved from The CEFR Levels: <https://www.coe.int/en/web/common-european-framework-reference-languages/level-descriptions>
- Du, J., & Daniel, B. K. (2024). Transforming language education: A systematic review of AI-powered chatbots for English as a foreign language speaking practice. *Computers and Education: Artificial Intelligence*. doi:10.1016/j.caeai.2024.100230
- EF Education First. (2024). *EF English Proficiency Index (EPI) 2024: A Ranking of 116 Countries and Regions by English Skills*. EF Education First. Retrieved from <https://www.ef.com/wwen/epi/>
- Equipo Médico QuickHealth. (2025). *Cómo atender como médico independiente*. Retrieved from QuickHealth: <https://www.quickhealth.cl/recien-egresado/como-atender-medico-independiente>

- Escuela Superior Politécnica del Litoral. (2023). *Desarrollo de un sistema de gestión de pacientes para el Hospital República del Ecuador, Islas Galápagos*. Guayaquil, Ecuador: Escuela Superior Politécnica del Litoral.
- Feng, S., Magana, A. J., & Kao, D. (2021). A Systematic Review of Literature on the Effectiveness of Intelligent Tutoring Systems in STEM. *2021 IEEE Frontiers in Education Conference (FIE)*, (pp. 1-9). doi:10.1109/FIE49875.2021.9637240
- Hidalgo Fernández Abogados. (2020). *Errores en la gestión de registros médicos*. Retrieved from Hidalgo Fernández Abogados: <https://hidalgofernandezabogados.com/errores-mas-habituales-en-la-gestion-de-registros-medicos-y-de-salud/>
- IBM. (2023). *IBM Think Blog*. Retrieved from ¿Qué es una API?: <https://www.ibm.com/mx-es/think/topics/api>
- Ken, A. (2023, Octubre 4). *Backend: ¿Qué es y para qué sirve?* Retrieved from Gluo: <https://www.gluo.mx/blog/backend-que-es-y-para-que-sirve>
- Kuiken, F. (2022). Linguistic complexity in second language acquisition. *Linguistics Vanguard*, 83-93. doi:<https://doi.org/10.1515/lingvan-2021-0112>
- Larsen-Freeman, D. (1997). Chaos/Complexity Science and Second Language Acquisition. *Applied Linguistics*, 2, 141-165. doi:<https://doi.org/10.1093/applin/18.2.141>
- Mercer. (2024). *¿Por qué transformarse en una empresa ágil y flexible?* Retrieved from Mercer: <https://www.mercer.com/es-mx/insights/people-strategy/future-of-work/por-que-transformarse-en-una-empresa-agil-y-flexible/>
- Microsoft. (2024). *¿Qué es SQL Server?* Retrieved from Microsoft Learn: <https://learn.microsoft.com/es-es/sql/sql-server/what-is-sql-server?view=sql-server-ver17>
- Microsoft. (2024). *SQL Server Management Studio (SSMS)*. Retrieved from Microsoft Learn: <https://learn.microsoft.com/es-es/ssms/sql-server-management-studio-ssms>
- Ministerio de Salud Pública del Ecuador. (2022). *Lineamientos para el manejo domiciliario de pacientes con COVID-19*. Ministerio de Salud Pública del Ecuador.
- Pérez Rodríguez, A. S. (2024). Impacto de la digitalización en la administración de servicios de salud. *Polo del Conocimiento*, 4–6.
- Proyectos Ágiles. (2023). *¿Qué es Scrum?* Retrieved from Proyectos Ágiles: <https://proyectosagiles.org/que-es-scrum/>
- QRA Corp. (2023). *Functional vs. Non-Functional Requirements*. Retrieved from QRA Corp: [https://qracorp.com/guides\\_checklists/functional-vs-non-functional-requirements/](https://qracorp.com/guides_checklists/functional-vs-non-functional-requirements/)
- Quizhpe, R., & Yanella, R. (2024). The Use of Self-Produced Videos to Improve Speaking Skills: A Complete Protocol. *Universidad San Francisco de Quito (USFQ)*. Retrieved from <https://repositorio.usfq.edu.ec/handle/23000/14052>
- Relisten. (2025, Agosto 12). *Errores comunes en la documentación clínica manual*. Retrieved from Relisten: <https://relisten.ai/errores-comunes-en-la-documentacion-clinica-manual/>
- Rivas, A. (2025). *La llegada de la IA a la educación en América Latina: en construcción*. Organización de Estados Iberoamericanos (OEI) & Fundación ProFuturo, Madrid. Retrieved from <https://oei.int/oficinas/secretaria-general/publicaciones/la-llegada-de-la-ia-a-la-educacion-en-america-latina-en-construccion>
- Sanz Manzanedo, M. (2025). La IA en la enseñanza de idiomas: chatbots y formación del profesorado. *European Public & Social Innovation Review*. doi:10.31637/epsir-2025-513
- Sasmayunita, & Assiddiq, M. A. (2025). The Role of Artificial Intelligence (AI) in Promoting Autonomous Learning in Language Education. *The 5th International Conference on Linguistics and Cultural Studies 5 (ICLC-5 2024)*, (pp. 349-363). doi:10.2991/978-2-38476-394-8\_40
- UNESCO. (2025). *AI and the future of education: disruptions, dilemmas and directions*. Retrieved from UNESDOC Digital Library: <https://unesdoc.unesco.org/ark:/48223/pf0000395236?posInSet=3&queryId=d1650d47-8561-4475-855e-53a54b31cd36>
- UNIR Formación Profesional. (2023). *Revista de Ingeniería y Tecnología – UNIR FP*. Retrieved

- from ¿Qué es un framework y para qué sirve?:  
<https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/>
- Universidad Carlos III de Madrid. (2021). *Sistema web para la gestión de una clínica médica*. Madrid, España: Universidad Carlos III de Madrid.
- Universidad Nacional de Trujillo. (2017). *Metodologías ágiles*. Retrieved from Universidad Nacional de Trujillo:  
[https://d1wqtxts1xzle7.cloudfront.net/53222887/Metodologias\\_Agiles-libre.pdf?1495404476=&response-content-disposition=inline%3B+filename%3DUniversidad\\_Nacional\\_de\\_Trujillo.pdf&Expires=1761083409&Signature=GNRyn5kT9FbgHR0dQRfnhqW6v02qHtLSvE6x7dFq7OYmtYQz4](https://d1wqtxts1xzle7.cloudfront.net/53222887/Metodologias_Agiles-libre.pdf?1495404476=&response-content-disposition=inline%3B+filename%3DUniversidad_Nacional_de_Trujillo.pdf&Expires=1761083409&Signature=GNRyn5kT9FbgHR0dQRfnhqW6v02qHtLSvE6x7dFq7OYmtYQz4)
- Universidad Politécnica Salesiana. (2022). *Desarrollo de una aplicación web para el control de citas y manejo de historial clínico en la Unidad Médica “Family Care”*. Quito, Ecuador: Universidad Politécnica Salesiana.
- Universidad Regional Autónoma de los Andes. (2021). *Sistema informático para la gestión de pacientes en Uniandes, extensión Babahoyo*. Babahoyo, Ecuador: Universidad Regional Autónoma de los Andes.
- Wang, I. X., Wu, X., Coates, E., Zeng, M., Kuang, J., Liu, S., . . . Park, J. (2024). *Neural Automated Writing Evaluation with Corrective Feedback*. Retrieved from arXiv:  
<https://arxiv.org/abs/2402.17613>
- Wei, L. (2023). Artificial intelligence in language instruction: impact on English learning achievement, L2 motivation, and self-regulated learning. *Frontiers in Psychology*, 14. doi:<https://doi.org/10.3389/fpsyg.2023.1261955>

**ANEXOS:**

**ACTA DE REUNIÓN – SPRINT 1**

Fecha: 05 de noviembre de 2025

Duración: 3 hora

Objetivo del Sprint: Recolectar los requerimientos del sistema y definir los casos de uso principales.

Descripción de la reunión:

Se revisaron las necesidades del consultorio, se levantaron los requerimientos funcionales y no funcionales, y se definieron los casos de uso esenciales que estructurarán el sistema web.

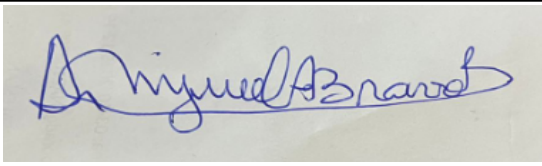

Resultados:

Requerimientos funcionales y no funcionales documentados.

Casos de uso base identificados y aprobados.

Alcance inicial del proyecto claramente delimitado.

Firmas:

Product Owner	
Desarrollador	

## ACTA DE REUNIÓN – SPRINT 2

Fecha: 12 de noviembre de 2025

Duración: 2 hora

Objetivo del Sprint: Presentar y validar las interfaces diseñadas del sistema.

Descripción de la reunión:

Se presentaron las interfaces principales del sistema, revisando el flujo visual, la navegación y la coherencia con los requerimientos levantados. Se registraron comentarios para mejorar la experiencia de usuario.

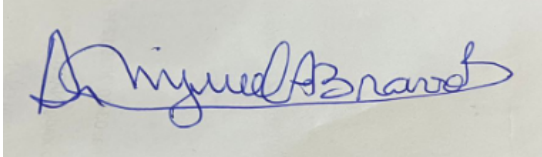

Resultados:

Interfaces preliminares aprobadas con observaciones menores.

Ajustes acordados para elementos visuales y usabilidad.

Validación general del diseño obtenido.

Firmas:

Product Owner	
Desarrollador	

## ACTA DE REUNIÓN – SPRINT 3

Fecha: 1 de diciembre de 2025

Duración: 2 hora

Objetivo del Sprint: Presentar las funcionalidades implementadas junto con las interfaces finales del sistema.

Descripción de la reunión:

Se mostraron las funcionalidades completas, incluyendo gestión de pacientes, citas e informes. Se revisaron las interfaces finales y se validó el flujo general del sistema, verificando su funcionamiento adecuado.

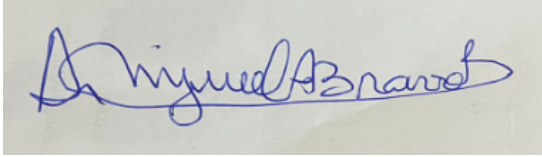

Resultados:

Funcionalidades principales implementadas correctamente.

Interfaces finales validadas por el Product Owner.

Se aprobó el sistema para cierre del sprint y preparación de entrega final.

Firmas:

Product Owner	
Desarrollador	

## Acta de Aceptación

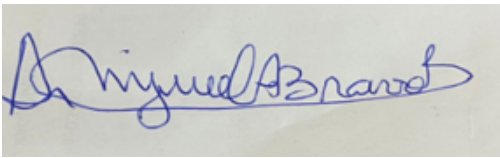
Producto: Prototipo de sistema web para la gestión de los pacientes y de sus citas.

Fecha: 5 de enero de 2026

Yo, Miguel Bravo, con cédula de ciudadanía No. 1706575048, responsable del consultorio médico independiente “Dr. Miguel Bravo”, manifiesto que recibo y acepto el prototipo desarrollado por Benjamín Bravo, identificado con cédula de ciudadanía No. 1754207296, en la fecha indicada.

**Product Owner:** Miguel Bravo

### Firmas

<b>Product Owner:</b> Miguel Bravo C.C. 1706575048	
<b>Desarrollador:</b> Benjamín Bravo C.C. 1754207296	