

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE APLICACIÓN WEB DE GESTIÓN DE
HISTORIAS Y CITAS MÉDICAS
PARA EL CONSULTORIO MÉDICO “DEVITA MEDICAL”**

DESARROLLO DE UNA APLICACIÓN WEB

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO
SUPERIOR EN DESARROLLO DE SOFTWARE**

ALBERTO ALFONSO HEREDIA IZA

DIRECTOR: ING. BYRON LOARTE

DMQ, febrero 2023

CERTIFICACIONES

Yo, Alberto Alfonso Heredia declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Proyecto de titulación



Alberto Alfonso Heredia

alberto.heredia@epn.edu.ec

Certifico que el presente trabajo de integración curricular fue desarrollado por Alberto Alfonso Heredia, bajo mi supervisión.



Ing. Byron Loarte, MSc.

DIRECTOR

byron.loarteb@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Alberto Alfonso Heredia Iza

DEDICATORIA

Dedico esta tesis a mi padre Alfonso Heredia y a mi madre Luz Iza, quienes han sido las personas que siempre me han apoyado en todo entre mal genios y risas han sido un apoyo constante en mi día a día siendo en todo momento un ejemplo para mi vida. Sin su amor, comprensión y ánimo, no habría sido posible llegar hasta aquí. A mis hermanos, por ser mi familia y mi mejor equipo de apoyo, tenemos roces y peleamos, pero sin importar lo enojados que estemos en momentos de dificultad siempre están allí. A mis amigos, por sus consejos, risas y apoyo en los momentos difíciles. Por último, al Ing. Byron Loarte, por su valiosa orientación y constante apoyo que sin su ayuda esto no sería posible. Este logro es nuestro y estoy inmensamente agradecido a todos ustedes por su papel en mi vida y en este proceso.

Alberto Alfonso Heredia Iza

AGRADECIMIENTO

Agradezco principalmente a Dios Todopoderoso, que en su buen sentido del humor me creo y tiene una paciencia infinita conmigo gracias por darme la sabiduría, fortaleza y perseverancia para llevar a cabo esta tesis. Gracias a mi familia, por su amor incondicional y apoyo constante en todos los momentos de mi vida porque sin importar lo que pase ellos siempre creen en mí y me apoyan incondicionalmente, especialmente durante este proceso de investigación y escritura a altas horas de la noche o en la madrugada ellos estaban allí. A mi director de tesis, por su valiosa orientación y paciencia en guiarme y corregirme en las cosas que estaba equivocado.

A la Escuela Politécnica Nacional, por brindarme las herramientas y oportunidades para desarrollar mis habilidades y conocimientos. Agradezco también a todos los profesores y compañeros de clase por su colaboración y contribución en mi formación académica.

Alberto Alfonso Heredia Iza

ÍNDICE DE CONTENIDOS

1	INTRODUCCIÓN.....	1
1.1	Objetivo general	2
1.2	Objetivos específicos.....	2
1.3	Alcance	2
1.4	Marco Teórico	4
2	METODOLOGÍA.....	6
2.1	Metodología de Desarrollo	6
Roles	6	
Artefactos	7	
2.2	Diseño de interfaces.....	10
Herramienta utilizada para el diseño.....	10	
2.3	Diseño de la arquitectura.....	11
Patrón arquitectónico.....	11	
2.4	Herramientas.....	12
3	RESULTADOS Y DISCUSIÓN	14
3.1	<i>Sprint 0.</i> Ambiente de desarrollo para la codificación de la aplicación <i>web</i>	14
Restricciones para la aplicación <i>web</i>	14	
Estructura del proyecto backend y frontend	15	
Diseño de BDD relacional	16	
Asignación de roles	16	
3.2	<i>Sprint 1.</i> Usuario administrador desarrollo e implementación de módulos que se han asignado.....	17
Inicio de sesión para los respectivos usuarios	17	
Gestionar información de perfiles de usuario	18	
Gestión registro de pacientes	19	
Gestionar información de servicios y costos	20	

3.3 Sprint 2. Usuario médico desarrollo e implementación de módulos que se han asignado.....	22
Ingreso a la aplicación web el usuario médico	23
Administrar información del paciente	24
Gestionar Historias médicas	25
Agendamiento de cita médica.....	26
3.4 Sprint 3. Pruebas de la aplicación web.....	27
Unitarias y resultados que se han obtenido	28
Compatibilidad y resultados que se han obtenido	29
Aceptación y resultados que se han obtenido.....	30
3.5 Sprint 4. Despliegue de la aplicación web	31
4 CONCLUSIONES	33
5 RECOMENDACIONES.....	34
6 REFERENCIAS BIBLIOGRÁFICAS.....	35
7 ANEXOS.....	40
ANEXO I	41
ANEXO II	42
ANEXO III	79
ANEXO IV	80

RESUMEN

Cuando comenzó la pandemia del Covid-19 en el Ecuador se implementaron diversas medidas para controlar los casos de infección y de propagación. Estas medidas, como el aislamiento en casa y las medidas de bioseguridad, causaron pánico entre la población. A pesar de que la pandemia todavía no ha terminado, el centro de especialidades Médicas “Devita Medical” comenzó a ofrecer servicios médicos a domicilio. Sin embargo, el registro manual de historiales médicos y de la información del paciente ha causado una serie de malestares y problemas. Por lo que, en este proyecto de Integración Curricular, se ha desarrollado una aplicación web que permita ser un apoyo en la gestión de la información de los pacientes del centro de especialidades médicas. Logrando con ello llevar un control más eficiente de los datos de los pacientes que son atendidos en el centro. Además, la aplicación permite a los médicos tener una organización y acceso en tiempo real al historial médico de los pacientes, gracias al uso de herramientas de desarrollo modernas y escalables, brindando una experiencia de calidad para todos los usuarios que hagan uso de la aplicación web en producción.

El presente Trabajo de Integración Curricular está estructurado de la siguiente manera: en la primera parte se presenta la problemática, objetivos, alcance y marco teórico relacionado con el proyecto. En la segunda sección, se detalla cómo se aplica la metodología *Scrum*, prototipos, herramientas y base de datos relacional en el desarrollo de la aplicación web. En la tercera parte, se presentan los resultados que se han obtenido a través de las tareas de cada *Sprint* y el despliegue final. Por último, en la cuarta sección se presentan las conclusiones y recomendaciones que se han obtenido durante el proyecto de Integración Curricular.

PALABRAS CLAVE: Covid-19, Aplicación web, MySQL, Centro Médico, Laravel, React.

ABSTRACT

When the Covid-19 pandemic began in Ecuador, various measures were implemented to control infection and spread. These measures, such as home isolation and biosecurity measures, caused panic among the population. Despite the fact that the pandemic has not yet ended, the "Devita Medical" Medical Specialties Center began to offer medical services at home. However, the manual registration of medical records and patient information has caused a series of discomforts and problems. Therefore, in this Curricular Integration project, a web application has been developed to support the management of patient information at the medical specialties center, achieving a more efficient control of the data of the patients who are attended at the center. Additionally, the application allows doctors to have real-time organization and access to patients' medical records, thanks to the use of modern and scalable development tools, providing a quality experience for all users who use the web application in production.

This Curricular Integration Project is structured as follows: the first part presents the problem, objectives, scope, and theoretical framework related to the project. In the second section, it details how the Scrum methodology, prototypes, tools, and relational database are applied in the development of the web application. In the third part, the results obtained through the tasks of each Sprint and the final deployment are presented. Finally, in the fourth section, the conclusions and recommendations obtained during the Curricular Integration project are presented.

KEYWORDS: Covid-19, Web application, MySQL, Medical Center, Laravel, React.

1 INTRODUCCIÓN

En el Ecuador, a inicios del 2020 cuando inicio la pandemia mundial conocida como Covid-19 llamada así por la Organización Mundial de la Salud (OMS), se presentaron diversas medidas para mitigar los crecientes casos de personas infectadas, a tal grado que día jueves 12 de marzo el Gobierno Nacional del Ecuador declaro diversas medidas con el objetivo de reducir la expansión del virus entre las cuales se encontraban el confinamiento domiciliario, teletrabajo, estudios online, diversas medidas de bioseguridad tanto al momento de salir de los hogares como al regresar a ellos, entre otras medidas [1], lo cual provocó que la ciudadanía entre en un pánico extremo al salir de sus hogares por miedo a contagiarse e incluso si se sentían enfermos el temor de contagiarse en los centros de salud ya sean estos públicos o privados, lo cual optaban por hacerse atender por otros medios o mecanismos [2].

Actualmente, la pandemia todavía no ha terminado y a causa de la problemática antes mencionada y como una forma de atender a las personas que por temor a los contagios optan por no salir de sus hogares y necesitan atención médica especializada, el centro de especialidades médicas “Devita Medical” con su sede en el Sur de Quito ofrece un servicio integral de medicina a domicilio entre los cuales se encuentra, medicina general, ginecología, fisioterapia física y respiratoria, laboratorio clínico, rayos X, ecógrafo portátil, alquiler de tanques de oxígeno, concentradores de oxígeno, entre otros. Debido a la gran acogida de esta iniciativa el poder llevar un adecuado control de las Historias Médicas de cada uno de los pacientes y el extenso trabajo en el área de recepción que lleva toda la información de forma manual ha ocasionado una serie de problemas para los médicos ya que toda la información la registran en hojas de papel bon, cuadernos académicos y en raras ocasiones utilizando medios ofimáticos, los cuales se pueden perder o al buscar la información esto se realizaría de una forma poco eficiente ocasionando malestar en cada uno de los diferentes pacientes que son atendidos diariamente.

Es común ver hoy en día que gran parte de las empresas utilizan sistemas automatizados que ayudan a la organización a tener un mejor control de sus recursos tanto en servicios como en personal humano, por lo cual se considera necesario establecer un sistema automatizado para poder tener un mejor control de los procesos internos del negocio o empresa. Por otra parte, la mayoría de los sistemas que han sido implementados en las empresas eran sistemas offline los cuales eran instalados en los principales equipos informáticos de la organización y solo era posible acceder a la información desde ese equipo para poder administrar toda la información almacenada, pero los tiempos han

cambiado y ahora las empresas cada vez están migrando a sistema *online* para que la información así como su administración se realice de forma casi instantánea sin necesidad de estar presente en un lugar en particular [3].

Gracias a las ventajas que otorgan los sistemas *online* y en base a la problemática que presenta actualmente el centro de especialidades médicas, por medio del presente Trabajo de Integración Curricular se ha codificado una aplicación *web* que permita llevar un mejor control de los datos de cada uno de los pacientes que son atendidos en el centro de especialidades. Además, la presente aplicación *web* les permite a los médicos llevar el historial médico de los pacientes de una forma mucho más organizada y en tiempo real gracias a la implementación de herramientas de desarrollo modernas y escalables brindando así una aplicación que sea de calidad para los usuarios y médicos que laboran en el centro de especialidades médicas.

1.1 Objetivo general

Desarrollar una aplicación web de gestión de historias y citas médicas para el consultorio médico “Devita Medical”.

1.2 Objetivos específicos

1. Establecer requisitos a un nivel funcional y no funcional para la aplicación *web*.
2. Diseñar cada una de las interfaces y posterior a ello establecer el modelo de base de datos para la etapa de codificación.
3. Establecer la arquitectura, herramientas y librerías requeridas para la etapa de codificación.
4. Codificar los diferentes *endpoints* y módulos para la aplicación *web* en base a los requisitos que se han obtenido.
5. Probar el rendimiento, compatibilidad y aceptación de la aplicación *web*.
6. Desplegar la aplicación *web* para su uso.

1.3 Alcance

El desarrollo e implantación de la aplicación *web* otorga una adecuada administración de la información para el centro de especialidades médicas “Devita Medical” que brinda sus servicios médicos en Quito, específicamente en el sector sur. Además, la aplicación facilita a los médicos llevar la información de sus pacientes de una forma adecuada, así como el historial clínico, agendamiento de citas médicas y costo de las citas médicas de acuerdo con los servicios solicitados, entre otros [4].

La aplicación *web* se encuentra dividida en dos componentes un *backend* y un *frontend* lo que permite su fácil integración con otros sistemas y garantizar la escalabilidad a futuro si se requiere integrar más módulos. Además, proporciona varios mecanismos de seguridad para proteger los datos, una interfaz amigable para que los médicos puedan visualizar toda la información requerida y que el manejo de la información se lo realice de forma fácil e intuitiva gracias al uso de UX [5]. Así mismo, se ha utilizado una metodología ágil para la verificación de los avances del proyecto, un base de datos relacional para que la información sea lo más integra posible, un patrón de arquitectura para garantizar la legibilidad del código, una serie de pruebas para verificar la calidad del producto y una etapa para el despliegue a producción.

Por último, se presenta los perfiles que intervienen en la aplicación *web*, así como los módulos a lo que puede acceder para realizar sus tareas requeridas.

Perfiles de la aplicación *web*

- Administrador.
- Médico.

El perfil administrador puede:

- Iniciar o cerrar sesión.
- Actualización de perfil.
- Crear, editar y eliminar nuevos usuarios.
- Administrar información de pacientes.

El perfil médico puede:

- Iniciar o cerrar sesión.
- Gestionar información de pacientes.
- Gestionar historias clínicas.
- Gestionar agenda médica.

1.4 Marco Teórico

Para el presente desarrollo de la aplicación *web* se lo ha realizado con diferentes herramientas y algunos conceptos como *framework*, *API*, *backend*, entre otros que se describen a continuación para una mejor comprensión y entendimiento.

El *backend* hace referencia a la estructura interna de un sistema *software* o por así llamarlo es el área lógica la cual no es visible para el usuario final puesto que es el código interno y no posee ningún elemento visual, su principal ventaja es proporcionar elementos que sean valiosos y útiles para el procesamiento de la información, escalabilidad, seguridad, integración con otros sistemas, etc. Además, el *backend* se puede desarrollar con diferentes lenguajes de programación como: JavaScript, Python, PHP, C++, Java, etc. [6].

Los *frameworks* pueden ser definidos como pequeñas plantillas o también como la base sobre la cual se asienta un proyecto, ayudando considerablemente al momento de realizar un proyecto de *software*. Además, simplifica en gran medida el proceso de codificación, añadiendo código automáticamente cuando se requiera y presentar un código optimizado y eficiente, de igual forma ayuda a realizar diferentes *test* a la aplicación [7].

Laravel es el *framework* más usado en la actualidad que utiliza como lenguaje base de programación PHP, el cual facilita considerablemente la creación de aplicaciones *web*, API'S RESTful y aplicaciones de escritorio. Este *framework* facilita el desarrollo por medio de un sistema de gestión de paquetes que automáticamente crea instancias, métodos y clases para usarlos directamente, así como también usa el patrón de arquitectura por defecto (MVC) evitando de esta manera escribir muchas líneas de código [8].

ORM también conocido como Mapeo de Objeto Relacional ayuda en la transformación de las tablas de una Base de datos en una serie de diferentes entidades para extraer la información de manera segura. Además, mejora la seguridad contra ataques ya que facilita en gran medida el trabajo para enfocarse en el diseño y creación del producto *software* y no tanto en el proceso de diseño de la Base de datos [9].

La librería *Eloquent* es usada por Laravel que funciona como ORM, que como se mencionó anteriormente, ayuda obtener la información de la Base de datos relacional de forma sencilla y segura, sin tener que utilizar sentencias SQL tan complejas [10].

Una Base de datos relacional (SQL) en si es un conjunto de tablas que están relacionadas entre ellas por un campo que es una llave primaria y que las relaciona entre

si añadiendo a la tabla un campo foráneo, evitando así que algunos datos se repitan y en caso de eliminar alguna tabla no se pierda la información de las demás ayudando así a mantener la integridad de la información [11].

MySQL es un SGBD o comúnmente llamado Sistema Gestor de Base de Datos, el cual ayuda a la administración de la base de datos e información almacenada. Además, permite obtener los datos de una manera rápida gracias a un lenguaje de consulta estructurado. Por otra parte, cuenta con dos tipos de licencias una de código abierto y la otra empresarial la cual pertenece a Oracle [12].

El frontend también conocido la aplicación del lado del cliente, permite desarrollar todas y cada una de las interfaces de un sistema *software*, en otras palabras, es lo que el usuario final puede visualizar para poder obtener la información. Además, existen un sin número de herramientas para trabajar con el *frontend* como: HTML, CSS y JavaScript, para dotar de esta manera colores, tipografías, imágenes texturas, animaciones entre otros que ayudan a que el sistema *software* sea más vistoso [8].

React es una librería de JavaScript, la cual tiene como principal objetivo desarrollar interfaces de usuario que sean visualmente agradables para el usuario final. Además, es una librería que dispone de un completo ecosistema de diferentes herramientas, módulos y componentes permitiendo que el desarrollo sea de forma eficiente y rápida [13].

Visual Studio Code o VsCode es un IDE, o en otras palabras es un editor de código que en los últimos años ha tomado mucha popularidad debido a los múltiples *plugins* que dispone y la posibilidad de añadir aplicaciones extras para testear API's, conexión directa con GitHub, entre otras más. Por último, este IDE al ser de código abierto puede funcionar en diferentes sistemas operativos [14].

API, no es más que una interfaz de programación de aplicación que son en realidad un conjunto de normas o reglas que deben seguirse para que se establezca una comunicación entre diferentes aplicaciones o entre un usuario determinado y una Base de datos, logrando de esta manera obtener información, ya sean estas imágenes, textos, datos numéricos, entre otros. [15].

Rest - Transferencia Estado Representacional, se la puede definir como una arquitectura que impone ciertos requerimientos que debe cumplir una API, ayudando de esta manera a que la información se transfiera de forma segura, confiable y esta pueda tener un buen rendimiento a gran escala. Además, este es un método de comunicación que hace que la parte interna de un sistema *software* [15].

2 METODOLOGÍA

El estudio de casos es considerado como un tipo de metodología aplicable al análisis de fenómenos sociales como a la investigación educativa, clínica, empresaria o al estudio de un tema en específico. Además, busca siempre obtener una mejor comprensión del tema a investigar y hacerlo de una forma sencilla que a su vez permita optimizar tiempo y recursos [16].

En base a lo citado anteriormente, este Trabajo de Integración Curricular integra en su totalidad el estudio de casos ya que preliminarmente se parte de un proceso de investigación completa sobre la problemática que existe en el centro de especialidades médicas “Devita Medical” y posterior a ello, buscar una solución adecuada para que la gestión de historias médicas se lo realice de forma automatizada gracias a la implementación de una aplicación *web*.

2.1 Metodología de Desarrollo

Son ampliamente utilizadas obviamente en el campo de la programación con el propósito de desarrollar en equipo y de manera organizada sistemas escalables. Con el paso del tiempo estas metodologías han ido mejorando dejando de ser una mera formalidad a ser una parte fundamental para el desarrollo de un producto *software*. Por otra parte, en algunas empresas de *software* siguen usando metodologías tradicionales para el diseño y creación de un *software*, aunque existen también metodologías ágiles que se usan bastante en la actualidad [17].

En la actualidad la gran mayoría de las aplicaciones se los realiza con metodologías ágiles que permiten enfocarse en la colaboración del equipo de trabajo y el dueño del producto, creando así un flujo constante y eficiente de trabajo realizando pequeñas tareas definidas o partes del sistema en periodos cortos periodos de tiempo. Gracias a la constante evolución de las metodologías ágiles es posible trabajar de forma más eficiente y proactiva [18]. Por tal motivo se ha utilizado la metodología ágil *Scrum* para la planificación y codificación de la aplicación *web*.

Roles

Scrum al ser una metodología ágil, dispone de una serie de roles que se usan para poder desarrollar aplicaciones de una manera mucho más colaborativa y organizada. Además, una parte importante es entender las responsabilidades de cada integrante del proyecto

para lograr el objetivo propuesto [19]. A continuación, se lista los 3 roles que se han sido necesarios como parte de la planificación para la aplicación *web*.

Product Owner

Tiene como responsabilidad dictaminar todos los requisitos del producto *software* a ser desarrollado y a la vez trabaja en conjunto con el *Scrum Master* [19]. Es este caso la **TABLA I** presenta la persona asignada para comandar dicho rol.

Scrum Master

Persona que actúa como líder y ayuda al equipo u organización a usar de mejor forma la metodología *Scrum* enfocándose sobre todo en la parte del proceso del negocio [19]. Es este caso en la **TABLA I** presenta la persona asignada para comandar dicho rol.

Development Team

Principalmente es un grupo de profesionales o técnicos con los conocimientos necesarios para poder llevar a cabo cada una de las tareas asignadas en el proyecto; trabajando de manera conjunta y comprometiéndose con cada iteración (*Sprint*) [19]. Es este caso en la **TABLA I** presenta la persona asignada para comandar dicho rol.

TABLA I: Determinación de nombres y asignación de roles.

ROLES	NOMBRES
<i>Product Owner</i>	Adm. Daniel Jaqui
<i>Scrum Master</i>	Ing. Byron Loarte, MSc.
<i>Development Team</i>	Alberto Alfonso Heredia

Artefactos

Los artefactos como parte de *Scrum* están diseñados con el fin de garantizar que la información sea transparente para todo el equipo encargado de la creación del producto *software*. De esta manera, se trabaja de una forma eficiente y se evitan retrasos a la hora de implementar una iteración (*Sprint*) [20]. A continuación, se lista los 3 artefactos que se han sido necesarios como parte de la planificación para la aplicación *web*.

Recopilación de requerimientos

Gracias a una adecuada recopilación de requerimientos se puede identificar las diferentes necesidades que el centro de especialidades médicas requiere y con ello poder plasmarlas de forma adecuada al momento de comenzar la etapa de codificación [21]. La **TABLA II** presenta la estructura de la tabla que se ha utilizado para recopilar la información y los principales requerimientos del centro de especialidades médicas, además, la tabla completa se la puede observar en el **ANEXO II**.

TABLA II: Tabla para el levantamiento de requisitos.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	RR – ID	ENUNCIADO DEL ÍTEM
Aplicación Web	RR-001	Tanto el usuario administrador y médico necesitan que la página de inicio cuente con mecanismos para proteger la información tales como un formulario de inicio de sesión.

Historias de Usuario

Una historia de usuario consiste en elaborar pequeñas tarjetas con una pequeña descripción de los requerimientos que el dueño del producto ha proporcionado. La descripción de cada tarjeta debe estar explicada de una forma sencilla de entender para poder dar una solución adecuada a las necesidades del usuario [22]. La **TABLA III** muestra un ejemplar para guiar al equipo de desarrollo en la etapa de codificación, además, las demás tarjetas completas se la puede observar en el **ANEXO II**.

TABLA III: Historia de Usuario - Visualizar página Inicio.

HISTORIA DE USUARIO	
Identificador (ID): HU-001	Usuario: Administrador, médico
Nombre historia: Visualizar página Inicio	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Sprint asignado: SB-001	

Programador responsable: Alfonso Heredia
Descripción: La página de inicio tanto para el usuario administrador y usuario médico será el formulario login, con el cual podrán tener accesos a la aplicación web presentando sus respectivas credenciales “Usuario y contraseña”.
Observación: Solo cuando se ingrese la contraseña se puede tener acceso a las diferentes opciones de la aplicación web.

Product Backlog

Básicamente es un conjunto de tareas creadas después de finalizar la elaboración de las historias de usuario. Además, cuenta con información relevante como la prioridad de ejecución e implementación y a la iteración a la que pertenece para una mejor organización [23]. La **TABLA IV** presenta la estructura de la tabla para elaborar el *Product Backlog* para el centro de especialidades médicas, además, la tabla completa se la puede observar en el **ANEXO II**.

TABLA IV: Determinación de iteraciones y prioridad.

ELABORACIÓN DEL PRODUCT BACKLOG				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU-001	Visualizar página Inicio	1	Finalizado	Terminado

Sprint Backlog

En esta sección se establece las tareas o actividades necesarias para la implementación del proyecto, organizadas en orden de prioridad con el *Product Owner* mediante un tablero de actividades en el cual se muestran todas las tareas a realizar también conocidas como *Sprints* [23]. En la

TABLA V presenta la estructura de la tabla para elaborar *Sprint Backlog* para el centro de especialidades médicas, además, la tabla completa se la puede observar en el **ANEXO II**.

TABLA V: Sprint Backlog.

ELABORACIÓN DE SPRINT BACKLOG						
ID – SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREA	TIEMPO ESTIMADO
SB-001	Diseño e implementación de los módulos para usuario administrador .	Módulo Pagina Inicio	HU-001	Visualizar	<ul style="list-style-type: none"> Definir tanto las páginas públicas como las privadas. métodos para la visualización de inicio. 	15 horas

2.2 Diseño de interfaces

Una interfaz es una parte importante al momento de crear un proyecto *software*, debido a su fácil interacción que existe entre la parte externa del sistema y la interna, en otras palabras, la interfaz permite que el usuario final pueda tener acceso a toda la información de un producto *software* de una forma intuitiva y a su vez que sea de fácil uso e interpretación [24].

Herramienta utilizada para el diseño

Para diseñar todas las interfaces de la aplicación *web* es necesario disponer de una herramienta de fácil uso, por este motivo, se ha utilizado Justindmind, el cual es un instrumento *online* que permite crear diferentes prototipos de una forma sencilla, rápida y colaborativa [25]. La **Fig. 1** se presenta un ejemplo de un prototipo para la aplicación *web*, mientras que los demás prototipos se los puede apreciar de una mejor forma en el **ANEXO II**.



Fig. 1: Prototipo - inicio de sesión en la aplicación web.

2.3 Diseño de la arquitectura

Al momento de diseñar un producto software sea cual sea, siempre es necesario implementar un diseño de arquitectura que ayude a comprender como está constituida la estructura interna del sistema software. Además, la integración de una arquitectura ayuda a delimitar la integración y codificación de algunos módulos, herramientas y librerías, así como garantizar la escalabilidad y un adecuado mantenimiento del código a futuro [26].

Patrón arquitectónico

Al momento de organizar componentes o separarlos en capas se establece como buena práctica la implementación de un patrón de arquitectura de software (MVC), el cual ayuda a que en la aplicación se pueda dar un mantenimiento adecuado en el caso de requerir actualizaciones y principalmente para que sea escalable ya que dispone 3 capas bien definidas:

- **Modelo:** capa para administrar de la información y sobre todo la conexión y persistencia con cualquier (SGBD).
- **Vista:** capa para proporcionar la interfaz y la presentación de los datos.
- **Controlador:** capa para gestionar y procesar las peticiones entrantes del usuario que accede a la aplicación.

En la **Fig. 2** se puede ver la implementación del MVC en el desarrollo de la aplicación web, juntamente con las herramientas para la etapa de despliegue a producción.

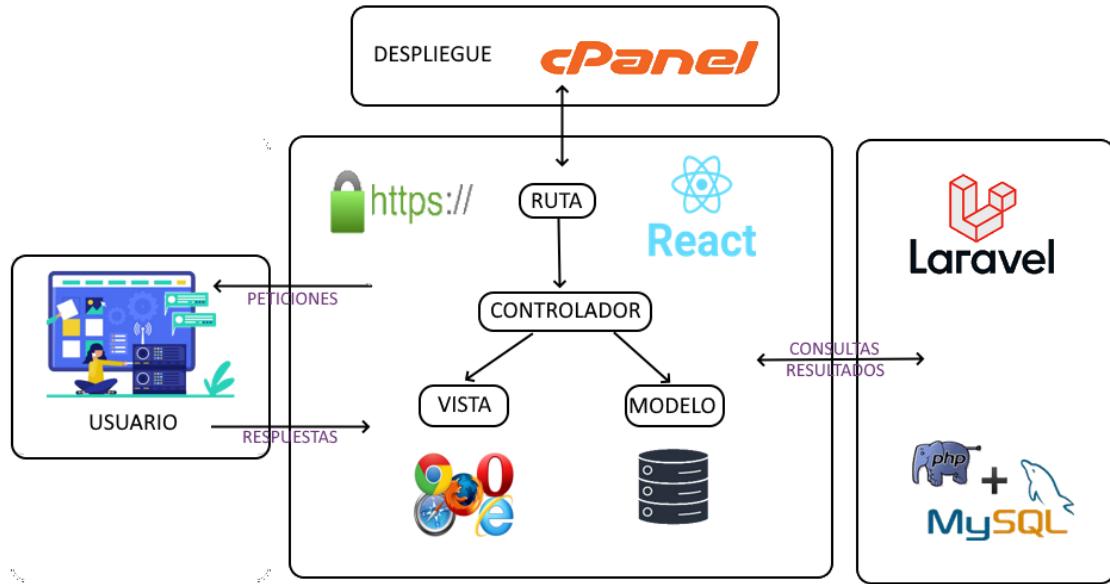


Fig. 2: Patrón arquitectónico de software.

2.4 Herramientas

Para la creación de la aplicación web se han determinado un conjunto de herramientas, las cuales han sido seleccionadas con el fin de ayudar a disminuir el tiempo de codificación y que la integración con otras herramientas, librerías y *Frameworks* sea lo más íntegro posible [27]. Muestra de ello, la **TABLA VI** presenta las herramientas que se han utilizado para la codificación de los módulos de la aplicación web.

TABLA VI: Herramientas que se han utilizado.

HERRAMIENTA	JUSTIFICACIÓN
VSCode	Es un entorno de desarrollo integrado que ha permitido escribir el código fuente para la aplicación web [14].
React	Esta librería ha permitido construir todas las interfaces de una forma fácil e intuitiva [13]
Laravel	Es un <i>framework</i> que ha permitido la creación del <i>backend</i> [8].
Node.js	Principalmente es un entorno que permite ejecutar JavaScript fuera del navegador y que ha permitido crear

	aplicaciones <i>web</i> así como API's RESTful [28].
MySQL	Gracias al uso de este SGBD el mismo ha permitido alojar la Base de datos relacional, almacenar y gestionar toda información de la aplicación <i>web</i> [12]
Cypress	El uso de Cypress ha permitido confirmar y validar que los diferentes componentes de una aplicación están funcionando correctamente. [29]

Librerías

La **TABLA VII**, presenta de forma general a todas las librerías fundamentales para la codificación de la aplicación *web*.

TABLA VII: Librerías que se han utilizado.

LIBRERÍA	DESCRIPCIÓN
tymondesign/jwt-auth	Librería que ayuda a transmitir información de manera segura con ayuda de firmas digitales [30].
Axios	Librería que ayuda a consumir datos JSON a través de peticiones HTML [31].
React Router	Librería que ayuda a definir las rutas a las cuales se dirigirán las páginas al realizar una acción determinada [32].
React Hook Form	Librería que ayuda a crear formularios usando HTML [33].

3 RESULTADOS Y DISCUSIÓN

Con el objetivo de presentar de mejor forma las tareas que se han especificado en cada iteración, así como también los resultados que se han obtenido en esta sección se presenta cada *Sprint* de una forma detallada y los resultados que se han logrado obtener.

3.1 Sprint 0. Ambiente de desarrollo para la codificación de la aplicación web.

Las tareas que conforman este *Sprint* son:

- Restricciones para la aplicación web.
- Estructura del proyecto *backend* y *frontend*.
- Diseño de BDD relacional.
- Asignación de roles.

Restricciones para la aplicación web

Autenticación de usuarios

Tanto el usuario administrador como el usuario médico pueden acceder a la aplicación web a través de un inicio de sesión que se presenta en la página de inicio, para lo cual tienen que ingresar la información como: usuario y contraseña.

Gestión de consultorio médico

El usuario administrador se encarga de asignar roles a cada usuario que se encuentre registrado en la aplicación web, también puede añadir nuevos servicios y los costos de cada uno de ellos, a su vez es responsable de actualizar los costos en caso que estos necesiten ser modificados.

Control de información del centro médico

El usuario administrador se encarga de administrar toda la información, tanto de los pacientes como de los médicos, citas e historias médicas, con el fin de verificar o editar la información de estos. Además, el usuario médico tiene acceso exclusivamente a la información de pacientes para crear, editar y eliminar en caso de que sea necesario.

Cambio de contraseña

Tanto el usuario médico como administrador tienen la función de cambiar la contraseña directamente en el perfil de usuario en el caso de que lo requieran.

Por último, en la **Fig. 3** se presenta a cada uno de los usuarios y las tareas que pueden realizar dentro de la aplicación web.

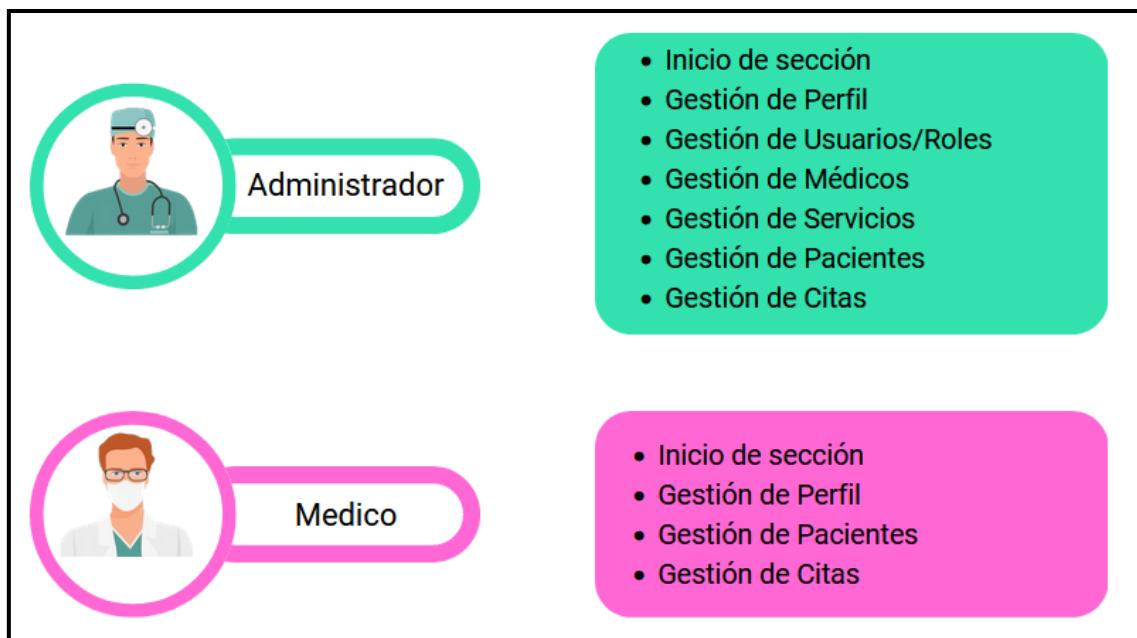


Fig. 3: usuarios (administrador y médico) con sus respectivas tareas.

Estructura del proyecto backend y frontend

Al momento de codificar la aplicación web se ha utilizado un editor de código llamado *Visual Studio Code*, el cual a través de sus diferentes plugins ayuda al momento de escribir el código de una forma ordenada, óptima y ahorrando tiempo. Además, se integra perfectamente con las diferentes herramientas y *frameworks* que se utilizan para la creación de aplicaciones del lado del cliente, tales como *Laravel*, *Spring*, *Express*, etc. Por otra parte, para codificar el *backend* se ha utilizado el *framework* Laravel principalmente para implementar cada uno de los *endpoints* con la ayuda de una Base de datos MySQL, en la cual se almacena toda la información de pacientes, médicos y usuarios. Por último, para la codificación del *frontend* se ha utilizado React el cual es una librería utilizada principalmente para la creación de interfaces de usuario que sean visualmente agradables y la presentación de la información con el consumo de los *endpoints* que se han generado por parte del *backend*.

Diseño de BDD relacional

Con el objetivo de gestionar de mejor forma la información de pacientes y médicos se ha implementado una base de datos relacional, la cual utiliza principalmente tablas y relaciones (claves primarias y foráneas) para garantizar la integridad de la información. Logrando de esta manera que la información se gestione de una forma adecuada a la vez que eficiente. En ese sentido, la **Fig. 4** presenta la estructura de la Base de datos que ha sido diseñada mediante tablas y que las mismas están relacionadas entre sí para que el *backend* tenga acceso a la información. Por último, para una mejor visualización de la Base de datos con todas su relaciones se puede apreciar de mejor manera en el **ANEXO II**.

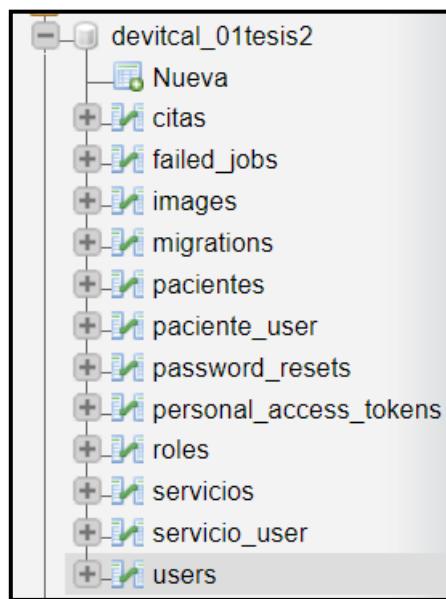


Fig. 4: Estructura de tablas de la Base de datos.

Asignación de roles

Como parte de la aplicación web se puede observar en la **Fig. 5** los roles que se han definido, así como las funciones a las cuales pueden acceder.

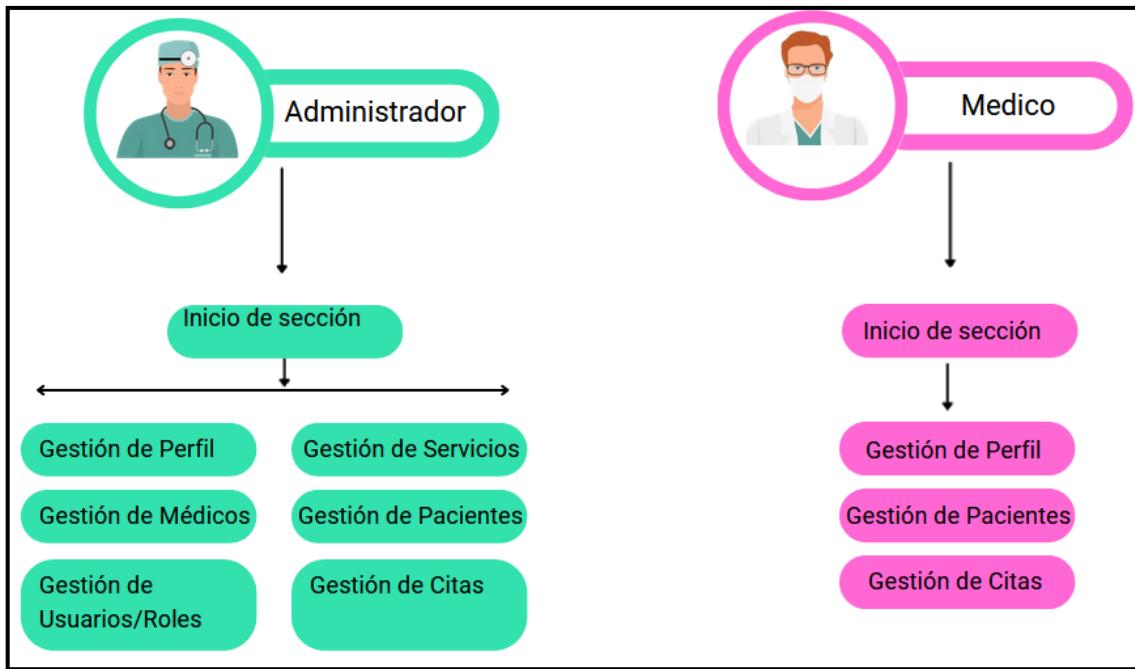


Fig. 5: Establecimiento de funciones para los usuarios administrador y médico.

3.2 Sprint 1. Usuario administrador desarrollo e implementación de módulos que se han asignado.

En base a la información que ha sido recopilada por el personal del centro de especialidades médicas “Devita Medical” se ha realizado la siguiente planificación de tareas para este *Sprint*:

- Inicio de sesión para los respectivos usuarios.
- Gestionar información de perfiles de usuario.
- Gestión registro de pacientes.
- Gestionar información de servicios y costos.

Inicio de sesión para los respectivos usuarios

La aplicación web presenta una página para el inicio de sesión como muestra la **Fig. 6** en la cual se puede apreciar dos *inputs* para que el usuario ingrese el email y contraseña, los cuales se validan y posterior a ello se otorga el acceso a los diferentes módulos acorde a los roles que ya han sido asignados previamente. Adicional a ello, en la **Fig. 7** se presenta el resultado de prueba unitaria y para apreciar de mejor forma este proceso se puede hacer en el **ANEXO II**.

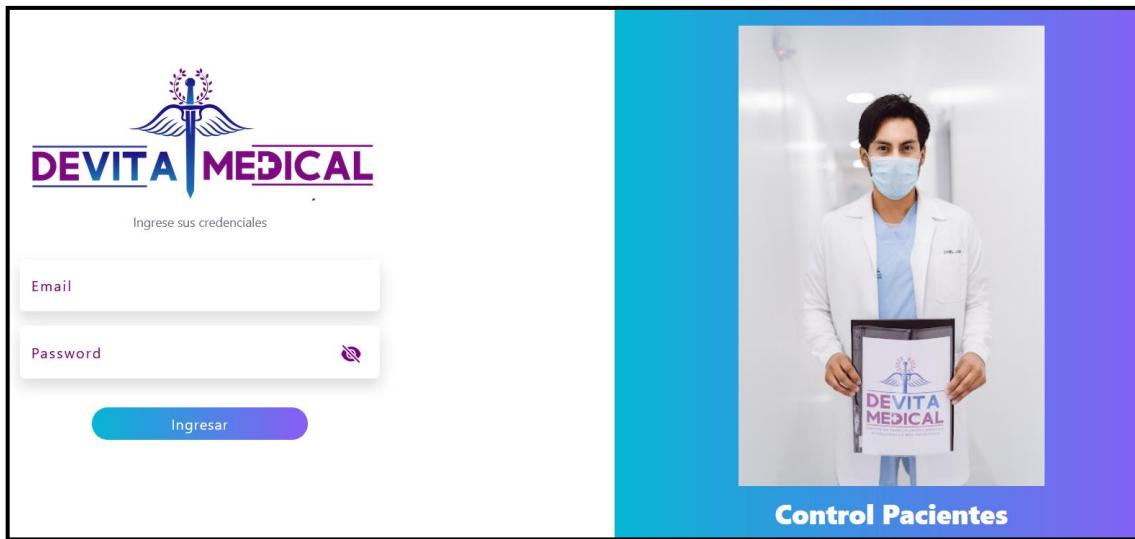


Fig. 6: Página de inicio de sesión para usuarios de la aplicación.

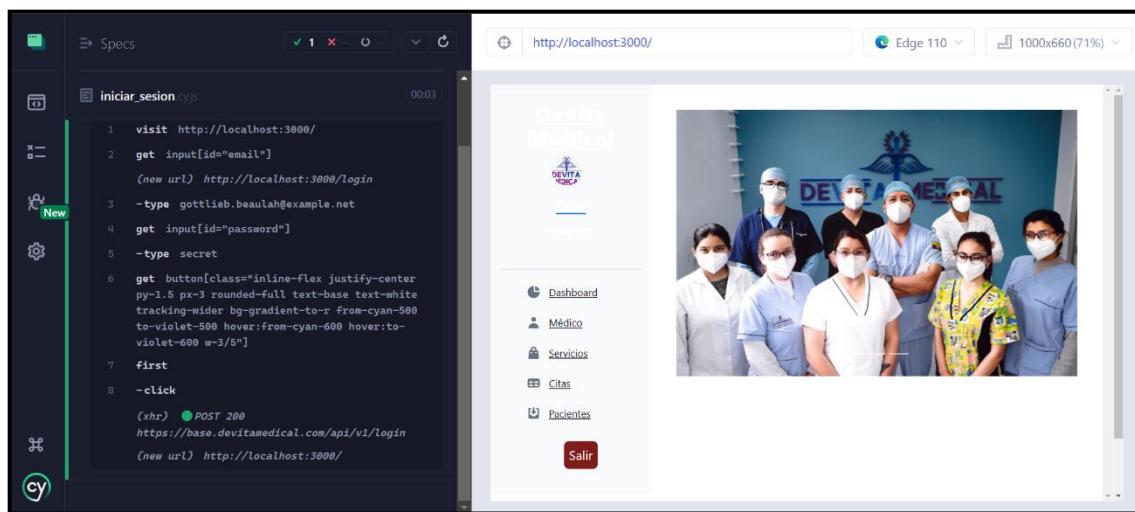


Fig. 7: Resultado de ejecución de la prueba unitaria.

Gestionar información de perfiles de usuario

Como se puede apreciar en la **Fig. 8** se presenta una página para modificar el perfil del usuario que en este caso es para el usuario médico, y que solo el usuario administrador puede realizar dicha acción. Ahora bien, el usuario médico solo podrá visualizar su información y tiene la posibilidad de cambiar la contraseña. Adicional a ello, en la **Fig. 9** se presenta el resultado de prueba unitaria y para apreciar de mejor forma este proceso se puede hacer en el **ANEXO II**.

Usuario

En este módulo encontrarás los datos del Usuario y podrás cambiar la contraseña

[Volver Atrás](#)

<ul style="list-style-type: none"> * NOMBRES: Santiago * APELLIDOS: Robles * EMAIL: myrna86@example.net * NÚMERO CELULAR: * STATE: Inactive * NÚMERO DE CASA: N/A 	<h3>Actualizar o Cambiar la contraseña</h3> <div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 5px;"> <input placeholder="Password" style="width: 100%; height: 30px; margin-bottom: 5px;" type="password"/> <input placeholder="Confirmar Password" style="width: 100%; height: 30px; margin-bottom: 5px;" type="password"/> </div> <div style="text-align: right;"> ✖ </div> <div style="background-color: #0070C0; color: white; padding: 5px 10px; border-radius: 5px; text-decoration: none; font-weight: bold; text-align: center;">ACTUALIZAR</div>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fig. 8: Gestionar los datos del médico.

Specs

actualización_perfil.js

```
(xhr) GET 200 https://base.devitamedical.com/api/v1/profile
14 get input[id="password_confirmation"]
15 -type Secret1*
16 get input[class="bg-sky-800 w-full p-3 text-white uppercase font-bold rounded-lg hover:bg-sky-900 cursor-pointer transition-all"]
17 first
18 -click
19 get button[class="m-auto text-white text-xl block mt-4 hover:text-red-300 text-center bg-red-900 p-2 rounded-lg"]
20 first
21 -click
(xhr) POST 200 https://base.devitamedical.com/api/v1/update-password
(new url) http://localhost:3000/citas2
(xhr) POST 200
```

https://base.devitamedical.com/api/v1/update-passw... Edge 110 1000x660 (71%)

Usuario

En este módulo encontrarás los datos del Usuario y podrás cambiar la contraseña

[Volver Atrás](#)

<ul style="list-style-type: none"> * NOMBRES: Elna * APELLIDOS: Leuschke * EMAIL: lamar52@example.com * NÚMERO CELULAR: [REDACTED] Request * STATE: Inactive * NÚMERO DE CASA: N/A 	<h3>Actualizar o Cambiar la contraseña</h3> <div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 5px;"> <input placeholder="Password" style="width: 100%; height: 30px; margin-bottom: 5px;" type="password"/> <input placeholder="Confirmar Password" style="width: 100%; height: 30px; margin-bottom: 5px;" type="password"/> </div> <div style="text-align: right;"> ✖ </div> <div style="background-color: #0070C0; color: white; padding: 5px 10px; border-radius: 5px; text-decoration: none; font-weight: bold; text-align: center;">ACTUALIZAR</div>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fig. 9: Resultado de ejecución de la prueba unitaria.

Gestión registro de pacientes

En la **Fig. 10** se puede observar el módulo del paciente, así como los datos de cada uno de los pacientes que se encuentran registrados y que al mismo tiempo se dispone de una opción que permite modificar la información de acuerdo a las necesidades del médico o del administrador. Cabe recalcar que este módulo tiene acceso el usuario médico y administrador. Adicional a ello, en la **Fig. 11** se presenta el resultado de prueba unitaria y para apreciar de mejor forma este proceso se puede hacer en el **ANEXO II**.

Pacientes				
Este módulo te permite gestionar los Pacientes del Centro Médico				
Agregar Paciente <input type="text" value="Búsqueda por Nombre o Cédula de Identidad"/>				
NOMBRE	CÉDULA DE IDENTIDAD	FECHA DE NACIMIENTO	TELÉFONO	ACCIÓN
Andrea Robledo	1728736602	2005-06-14	0941250933	VER EDITAR ACTIVO
Dario Ortega	0676086480	1975-07-22	0938990589	VER EDITAR ACTIVO
Doris Torres	0551242769	1994-02-14	0973830290	VER EDITAR ACTIVO

Fig. 10: Gestionar registro de pacientes.

The left side shows the Cypress Test Runner interface with a test spec named 'gestion_pacientes.cy.js'. The spec contains several test cases (cy.assert, cy.click, cy.get) for interacting with the application's login and patient management features. The right side shows a screenshot of a web browser (Edge 110) displaying the 'Pacientes' page from the 'DeVita Medical' website. The page lists patients with columns for Name, ID, Birthdate, Phone, and Action buttons (VER, EDITAR, ACTIVO/INACTIVO). The browser status bar indicates the window size is 1000x660 (71%).

Fig. 11: Resultado de ejecución de la prueba unitaria.

Gestionar información de servicios y costos

Una parte importante de esta aplicación *web* es que se tenga la posibilidad de gestionar servicios y asignar sus precios recordando que por petición del consultorio estos tienen que ser siempre valores enteros, para posteriormente poder entregar los valores al paciente como se puede observar en la Fig. 12. Además, este módulo permite administrar y visualizar la información de los diferentes servicios que dispone el

consultorio y en lo que respecta al acceso solo el administrador tiene acceso a este sin ninguna restricción y a su vez puede modificar los valores de dichos servicio, así como se puede observar en la **Fig. 13** o si fuera el caso deshabilitar los servicios que se han registrado. Adicional a ello, en la **Fig. 14** y **Fig. 15** se presentan los resultados de las pruebas unitarias y para apreciar de mejor forma este proceso se puede hacer en el **ANEXO II.**

Servicios y Especialidades Médicas			
Este módulo te permite gestionar los diferentes Servicios y Especialidades del Centro Médico			
ESPECIALIDAD	DESCRIPCIÓN	PRECIO	ACCIÓN
Cardiología	Se especializa en el diagnóstico y tratamiento de enfermedades del corazón, los vasos sanguíneos y el sistema circulatorio. Estas enfermedades incluyen enfermedad de las arterias coronarias, problemas del ritmo del corazón e insuficiencia cardíaca.	\$ 35	VER EDITAR ACTIVO
Fisioterapia	Conjunto de métodos, actuaciones y técnicas que, mediante la aplicación de medios físicos, curan, previenen, recuperan y adaptan a personas afectadas de disfunciones somáticas o a las que se desea mantener en un nivel adecuado de salud.	\$ 20	VER EDITAR ACTIVO
Ginecológica	La ginecología y los servicios para la salud reproductiva pueden incluir el diagnóstico y tratamiento de diversas afecciones y enfermedades, como: Citologías vaginales anormales. Presencia de VPH de alto riesgo. Sangrado vaginal anormal.	\$ 30	VER EDITAR ACTIVO

Fig. 12: Tabla de servicios.

Servicio
<div style="border: 1px solid #ccc; padding: 10px; width: 100%;"> <div style="text-align: center;">EDITAR SERVICIO</div> <div style="margin-bottom: 10px;"> Especialidad <input type="text" value="Cardiología"/> </div> <div> Descripción <input type="text" value="Se especializa en el diagnóstico y tratamiento de enfermedades del corazón, los vasos sanguíneos y el sistema circulatorio. Estas e"/> </div> <div> Precio (10) Sin decimales <input type="text" value="35"/> </div> <div style="text-align: center; background-color: #0070C0; color: white; padding: 5px 0; margin-top: 10px;"> ACTUALIZAR </div> </div>

Fig. 13: Tabla para edición de precios.

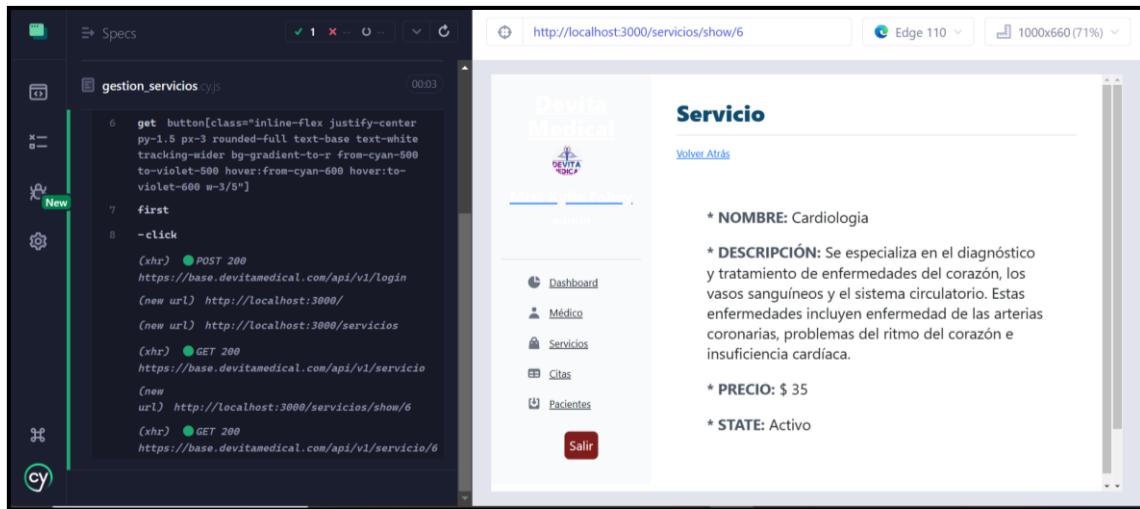


Fig. 14: Resultado de ejecución de la prueba unitaria tabla de servicios.

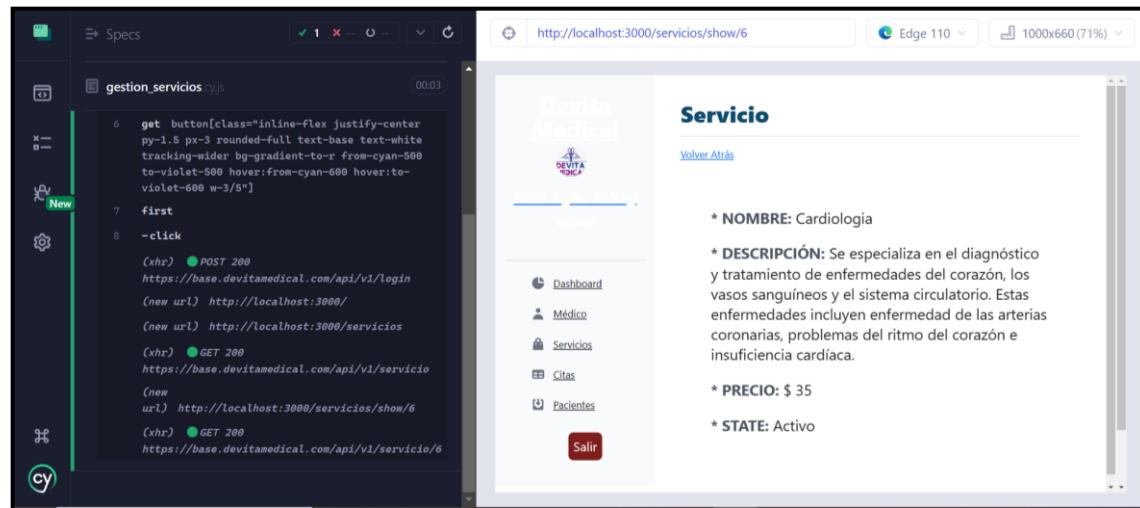


Fig. 15: Resultado de ejecución de la prueba unitaria edición de precios.

3.3 Sprint 2. Usuario médico desarrollo e implementación de módulos que se han asignado.

En base a la información que ha sido recopilada por el personal del centro de especialidades médicas “Devita Medical” se ha realizado la siguiente planificación de tareas para este *Sprint*:

- Ingreso a la aplicación web el usuario médico.
- Administrar información del paciente.
- Gestionar historias médicas.
- Agendamiento de cita médica.

Ingreso a la aplicación web el usuario médico

La aplicación web presenta una página para el inicio de sesión como muestra la **Fig. 16** en la cual se puede apreciar dos inputs para que el usuario ingrese el email y contraseña, los cuales se validan y posterior a ello se otorga el acceso a los diferentes módulos acorde a los roles que ya han sido asignados previamente. Adicional a ello, en la **Fig. 17** se presenta el resultado de prueba unitaria y para apreciar de mejor forma este proceso se puede hacer en el **ANEXO II**.

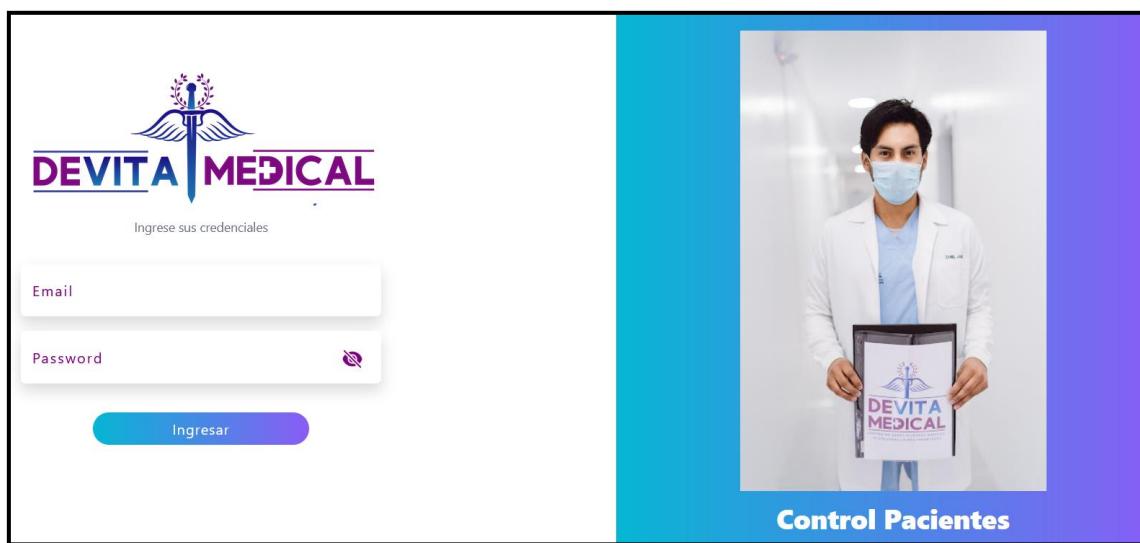


Fig. 16: Página de inicio de sesión para usuarios de la aplicación.

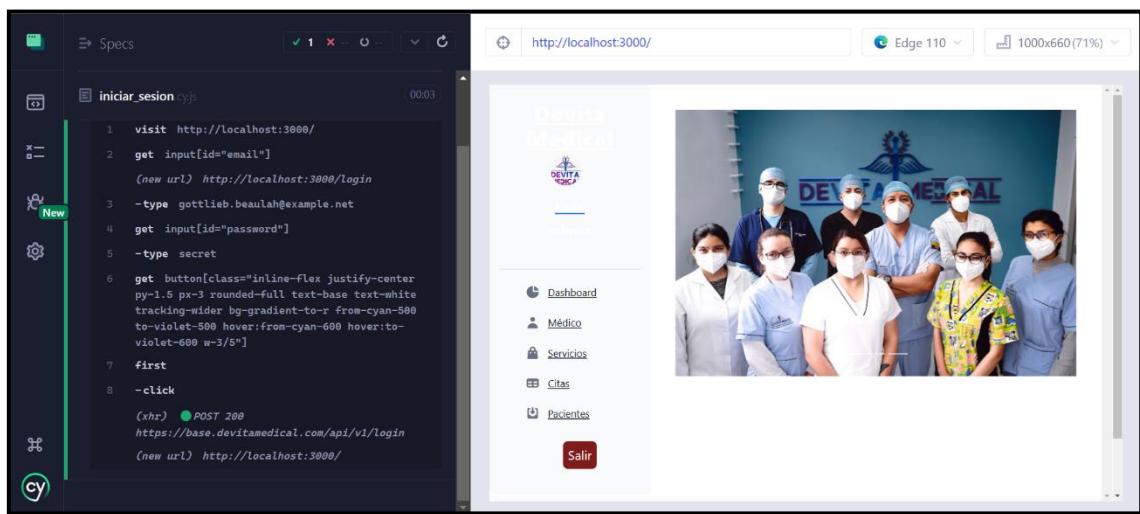


Fig. 17: Resultado de ejecución de la prueba unitaria.

Administrar información del paciente

Como se puede apreciar en la **Fig. 18** existe un módulo específico que permite añadir la información del paciente y con ello llevar un mejor control, esto incluye el poder editar la información del mismo en caso de ser necesario. Adicional a ello, en **Fig. 19** se presenta el resultado de prueba unitaria y para apreciar de mejor forma este proceso se puede hacer en el **ANEXO II**.

NOMBRE	CÉDULA DE IDENTIDAD	FECHA DE NACIMIENTO	TELÉFONO	ACCIÓN
Andrea Robledo	1728736602	2005-06-14	0941250933	<button>VER</button> <button>EDITAR</button> <button>ACTIVO</button>
Dario Ortega	0676086480	1975-07-22	0938990589	<button>VER</button> <button>EDITAR</button> <button>ACTIVO</button>
Doris Torres	0551242769	1994-02-14	0973830290	<button>VER</button> <button>EDITAR</button> <button>ACTIVO</button>

Fig. 18: Administrar información del paciente.

Fig. 19: Resultado de ejecución de la prueba unitaria.

Gestionar Historias médicas

En la **Fig. 20** se puede observar la información del paciente, así como la cita agendada previamente, mientras que la **Fig. 21** se puede apreciar la opción para editar dicha información. Adicional a ello, en la **Fig. 22** y **Fig. 23** se presentan los resultados de las pruebas unitarias y para apreciar de mejor forma este proceso se puede hacer en el **ANEXO II**.

Historia Clínica

[Volver Atrás](#)

* **NOMBRE DEL PACIENTE:** Doris Torres

* **FECHA DE NACIMIENTO:** 1994-02-14

* **SEXO:** Femenino

* **ALERGIAS:** Ninguna

* **F/H DE CITA:** 2023-02-08 06:30:00

* **DESCRIPCION:** Cefalea



Fig. 20: Datos de historias médicas.

EDITAR HISTORIA MEDICA

Signos y Síntomas
Escriba los Signos y Síntomas...

Diagnóstico
Escriba el Diagnóstico...

Prescripción
Escriba la Prescripción o Receta...

Observaciones
Escriba las observaciones (opcional)...



Fig. 21: Opción para editar información.

The screenshot displays the execution results of a unit test for a medical history feature. On the left, the Cypress Test Runner interface shows a test named 'DIR_gestionCitas.cy.js' with step 6 failing. Step 6 involves a POST request to the login endpoint, followed by several GET requests to retrieve appointment and patient information. On the right, a screenshot of a web browser shows the 'Historia Clínica' (Medical History) page. The page header is 'Historia Clínica'. It lists patient details: Nombre del Paciente: Andrea Salvador, Fecha de Nacimiento: 2005-06-15, Sexo: Femenino, Alergias: Penicilina, F/H de Cita: 2023-02-09 07:30:00, and Descripción: Dolor de cabeza, fiebre, malestar general, dolores musculares y articulares generalizados. The page also includes a sidebar with links to Dashboard, Médico, Servicios, Citas, and Pacientes.

Fig. 22: Resultado de ejecución de la prueba unitaria datos de historias médicas.

The screenshot displays the execution results of a unit test for appointment management. On the left, the Cypress Test Runner interface shows a test named 'gestion_citas.cy.js' with step 6 failing. Step 6 involves a POST request to update an appointment. On the right, a screenshot of a web browser shows the 'Citas - Agenda Médica' (Appointments - Medical Agenda) page. The page title is 'Citas - Agenda Médica' and it includes a sub-header: 'Este módulo te permite gestionar las citas agendadas e historias clínicas del Centro Médico'. It features a search bar for 'Búsqueda por Fecha, Paciente o Médico' and a table listing appointments. The table has columns: FECHA DE CITA, PACIENTE, TELÉFONO, MÉDICO, DESCRIPCIÓN, and ACCIÓN. Two rows of data are shown: one for a patient named Dario Ortega and another for Doris Torres. Each row includes buttons for 'HISTORIAS CLÍNICAS', 'EDITAR', and 'ELIMINAR'.

Fig. 23: Resultado de ejecución de la prueba unitaria editar información.

Agendamiento de cita médica

Como se puede apreciar en **Fig. 24** se permite gestionar el agendamiento de citas médicas acorde a la necesidad de los pacientes que se encuentran registrados en la aplicación web de manera que sea una tarea fácil para los usuarios médicos. Adicional a ello, en la **Fig. 25** se presenta el resultado de prueba unitaria y para apreciar de mejor forma este proceso se puede hacer en el **ANEXO II**.

Citas - Agenda Médica

Este módulo te permite gestionar las citas agendadas e **historias clínicas** del Centro Médico

[Agregar Cita](#)

Búsqueda por Fecha, Paciente o Médico

FECHA DE CITA	PACIENTE	TELÉFONO	MÉDICO	DESCRIPCIÓN	ACCIÓN
2023-02-22 10:00:00	Dario Ortega	0938990589	Gladis Tuares	Cefalea	HISTORIAS CLÍNICAS EDITAR ELIMINAR
2023-02-08 06:30:00	Doris Torres	0973830290	Richard Perez	Cefalea	HISTORIAS CLÍNICAS EDITAR ELIMINAR

Fig. 24: Citas médicas.

Fig. 25: Resultado de ejecución de la prueba unitaria.

3.4 Sprint 3. Pruebas de la aplicación web

Al momento de terminar la etapa de codificación de la aplicación web se procede a realizar un conjunto de pruebas:

- Unitarias y resultados que se han obtenido.
- Compatibilidad y resultados que se han obtenido.
- Aceptación y resultados que se han obtenido.

Unitarias y resultados que se han obtenido

El objetivo de las pruebas unitarias es la detección temprana de errores que se han producido al momento de codificar un sistema *software* con el fin de evitar el posible mal funcionamiento al momento de ser implementado en un ambiente de producción. Además, gracias a estas pruebas se garantiza la completa funcionalidad de los módulos y la calidad del código fuente [34].

Para ejecutar las pruebas se ha utilizado la herramienta denominada Cypress como se puede apreciar en la **Fig. 26**, esta herramienta toma pequeñas porciones de código en para un módulo en concreto que en este caso es el módulo *login* y obtener los resultados de la ejecución de la prueba como se puede apreciar en la **Fig. 27**. Adicional a ello, para apreciar de mejor forma las demás pruebas se puede hacer en el **ANEXO II**.

```
1 describe('template spec', () => {
2   it('verificado', () => {
3     cy.visit('http://localhost:3000')
4     cy.get('#email').type("gottlieb.beaulah@example.net")
5     cy.get('#password').type("secret")
6     cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-base text-white tracking-wider bg-gradient-to-r from-cyan-500 to-violet-500 hover:from-cyan-600 hover:to-violet-600 w-3/5"]')
7   })
8 })
```

Fig. 26: Prueba unitaria con la herramienta Cypress.

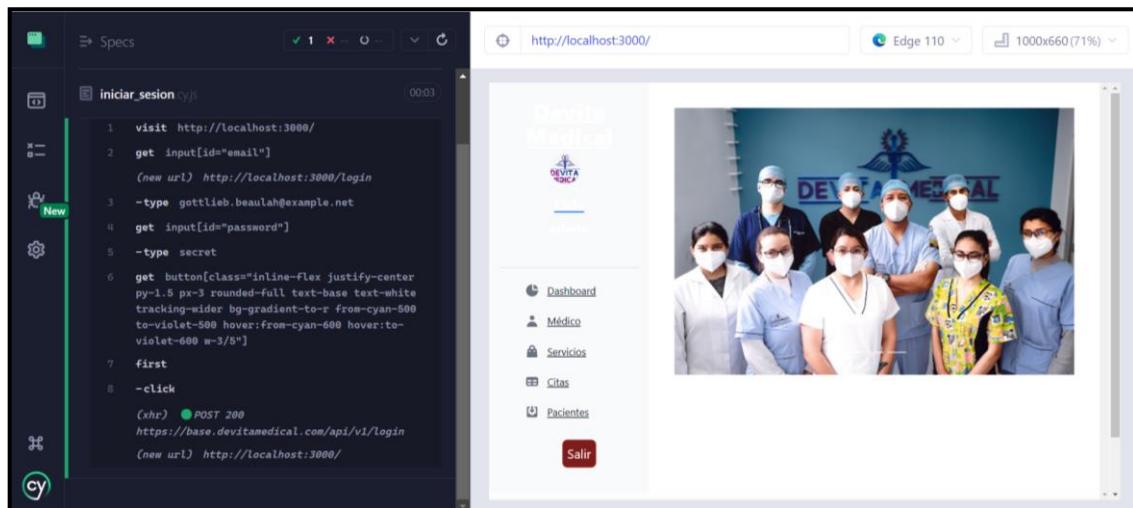


Fig. 27: Resultado de la prueba ejecutada al módulo login.

Al momento de finalizar con todas las pruebas unitarias de los diferentes módulos se ha obtenido un porcentaje del 100% de verificación y validación, lo que indica que la aplicación *web* no muestra ninguna falla a nivel de funcionalidad y código fuente en cada uno de los módulos y que están listos para su implementación a producción.

Compatibilidad y resultados que se han obtenido

Este tipo de prueba ayuda a garantizar como su nombre lo indica la compatibilidad de interfaces de un sistema *software* con diferentes pantallas de dispositivos y sistemas operativos, así como también diferentes navegadores debido a que por las diversas dimensiones de las pantallas la información puede distorsionarse o presentarse de forma desproporcionada. Adicional a ello, con estas pruebas se garantiza que la información se presente de la misma forma que se tenía prevista [35].

Para realizar esta prueba se ha utilizado diferentes dispositivos así como navegadores *web* con el fin de observar como se muestra cada una de las interfaces de la aplicación *web* en diferentes plataformas como se aprecia en la **TABLA VIII**, mientras que en la **Fig. 28** se puede observar el resultado de la prueba que se han realizado en el navegador *Brave*, mientras que en el **ANEXO II** se puede ver los resultados en los otros navegadores.

TABLA VIII: Compatibilidad de navegadores.

NOMBRE	VERSIÓN	OBSERVACIÓN
<i>Brave</i>	1.46.144	Funcional
<i>Google Chrome</i>	110.0.5481.104	Funcional
<i>Opera</i>	94.0.4606.54	Funcional
<i>Firefox</i>	1.7.13	Funcional
<i>Microsoft Edge</i>	104.0.1293.81	Funcional

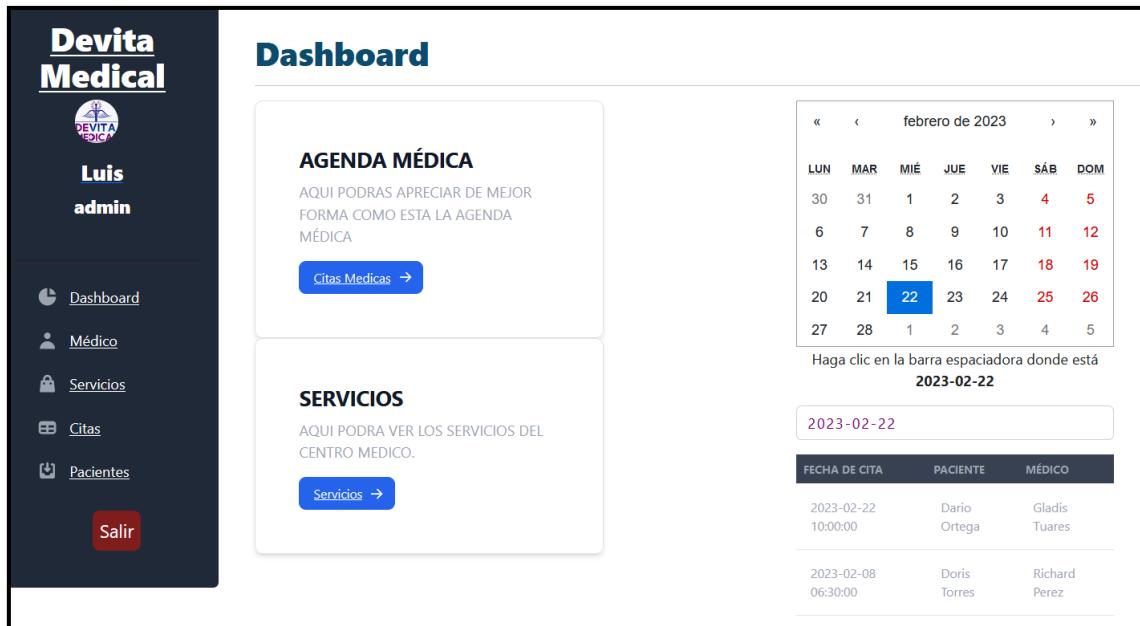


Fig. 28:Prueba de compatibilidad del navegador *Brave*.

Al momento de finalizar de realizar las pruebas de compatibilidad se ha determinado que las interfaces al igual que el contenido de la aplicación *web* es compatible con la mayoría de los navegadores *web* que se han utilizado.

Aceptación y resultados que se han obtenido

Esta prueba permite verificar que los requerimientos del cliente hayan sido completamente cubiertos y en caso de rechazo de la misma, realizar los cambios pertinentes antes de que el sistema *software* sea implementado a producción [36]. En la **TABLA IX** se puede observar la prueba de aceptación la cual se verifica la funcionalidad del módulo y si cumple con los requisitos que han sido proporcionados por el cliente, mientras que en el **ANEXO II** se puede ver a mayor detalle las demás pruebas.

TABLA IX: Prueba de aceptación.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA-011	Identificador historia de Usuario: HU-011
Nombre: Ingreso de usuario médico	
Descripción:	
El usuario médico ingresa a la aplicación <i>web</i> con el fin de visualizar la historia de un usuario determinado.	
Pasos de ejecución:	

- Digitar la URL en el navegador *web*.
- Ingresar el usuario y contraseña.
- Dirigirse a la opción historial médico.

Resultado deseado:

La aplicación *web* permite que el médico visualice el historial médico del paciente, así como también le ofrece la posibilidad de editar la información del mismo y revisar las atenciones previas.

Evaluación de la prueba

La aplicación *web* realiza lo esperado y es 100% funcional.

Al terminar las pruebas de aceptación se puede apreciar que la aplicación *web* cumple con las peticiones del *Product Owner* listos para comenzar la última etapa.

3.5 Sprint 4. Despliegue de la aplicación *web*

Al terminar de completar con todas las tareas del *Sprint* y luego de finalizar la etapa de pruebas, en este *Sprint* se procede al despliegue de la aplicación *web* en un ambiente de producción en este caso se procede al despliegue en un *hosting* conocido como *HostGator*, el cual tiene un costo relativamente bajo y el cual se puede apreciar los accesos respectivos en el **ANEXO II**. Por último, en la **Fig. 29** se puede apreciar a la aplicación *web* alojada bajo un dominio el cual se puede acceder bajo la siguiente url:

<https://front.devitamedical.com/login>

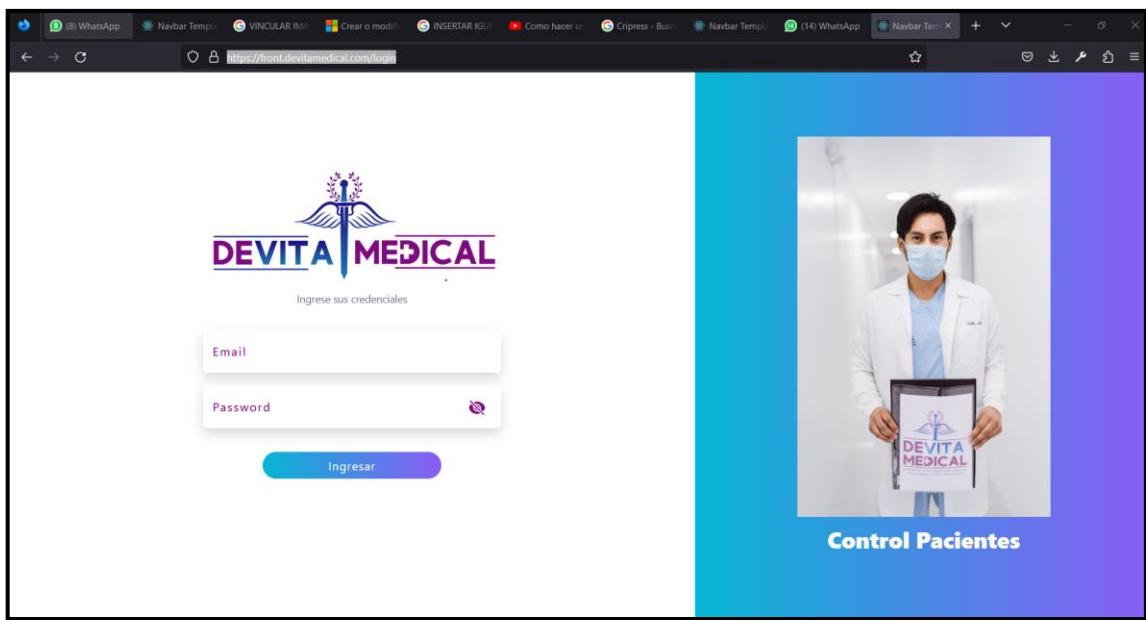


Fig. 29: Despliegue de la aplicación web.

4 CONCLUSIONES

Esta sección presenta todas las conclusiones a las que se han llegado durante el desarrollo de la aplicación *web* como parte del presente Trabajo de Integración Curricular:

- Antes de codificar la aplicación *web* es fundamental tener una idea clara de las necesidades del cliente y el alcance del proyecto, puesto que con la ayuda de esta información inicial se puede seleccionar las herramientas adecuadas para desarrollar de manera más eficiente el producto *software*.
- Es importante hacer uso de las metodologías ágiles que en la actualidad son ampliamente usadas al momento de crear una aplicación *software*. *Scrum* es un claro ejemplo de ellas, ya que ha permitido planificar de manera adecuada cada *Sprint*, con el fin de optimizar tiempos y garantizar el correcto desarrollo de la aplicación reduciendo errores en cada una de las etapas de desarrollo.
- El uso del patrón MVC (Modelo Vista Controlador) ha permitido reducir tiempo al momento de escribir líneas de código y también ayuda a la implementación de nuevos módulos de forma sencilla, con códigos estructurados y limpios.
- La integración de un *Framework* al igual que las herramientas y diferentes librerías que están disponibles ha permitido que el código y la aplicación *web* sea escalable y lista para integrarse a otras aplicaciones si se requiere.
- Gracias a la integración de un Sistema Gestor de Base de Datos (MySQL) ha permitido manejar la información de forma organizada para cada servicio, médico y paciente. Al mismo tiempo ha permitido realizar consultas mucho más fáciles mejorando así el rendimiento y reduciendo errores al momento de implementarla con el servidor.
- Con el fin de verificar que la aplicación *web* funcione de forma adecuada en diferentes aspectos como velocidad, compatibilidad, y rendimiento ha sido necesario realizar diferentes pruebas en los módulos de la aplicación con el fin de evitar posibles errores que se puedan presentar al momento de desplegar a un ambiente de producción.

5 RECOMENDACIONES

Esta sección presenta todas las recomendaciones que se pueden otorgar como parte del desarrollo de la aplicación *web*:

- Es recomendable verificar la versión de los diferentes navegadores ya que, si no está actualizado puede presentar diferentes errores de visualización, afectando de esta forma la experiencia de usuario.
- Se recomienda comprobar si existe nuevas actualizaciones de las herramientas que se han utilizado en la aplicación *web* en un periodo de 6 meses, con la finalidad de mantener los componentes, librerías y ciertas configuraciones actualizadas para asegurar que la aplicación *web* no presente ningún tipo de error.
- Es necesario tener un ambiente de pruebas que ayuden a incorporar nuevas funcionalidades o correcciones a la aplicación *web* con el fin de evitar que la aplicación que se ha desarrollado se vea afectada por posibles errores.
- Es fundamental instruir al administrador para que pueda implementar protocolos de seguridad, debido a que él es el que puede dar permisos de administración y se debe prevenir que personas maliciosas o ajena al consultorio tengan acceso a la información.
- Es recomendable realizar semanal o mensualmente copias de seguridad de la base de datos puesto que la información del paciente es de vital importancia y en caso de pérdida sea factible recuperarla fácilmente.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] M. Mendoza, «www.elcomercio.com,» 12 marzo 2021. [En línea]. Available: <https://www.elcomercio.com/actualidad/ecuador/ecuador-pandemia-covid-emergencia-sanitaria.html>.
- [2] EFE, «https://www.elconfidencial.com,» 01 05 2020. [En línea]. Available: https://www.elconfidencial.com/alma-corazon-vida/2020-05-01/coronavirus-miedo-salir-calle-expertos-analisis_2575484/.
- [3] «https://www.internetya.co,» Aplicativos Web, 14 11 2016. [En línea]. Available: <https://www.internetya.co/ventajas-y-beneficios-de-las-aplicaciones-web/>.
- [4] I. J. B. García, «https://www.servnet.mx,» 30 03 2021. [En línea]. Available: <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web>.
- [5] J. Hurtado, «iebschool,» 15 10 2021. [En línea]. Available: <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>.
- [6] C. Maria, «hubspot,» 19 10 2022. [En línea]. Available: <https://blog.hubspot.es/website/frontend-y-backend>.
- [7] G. Muente, «rockcontent,» 08 01 2020. [En línea]. Available: <https://rockcontent.com/es/blog/framework/>.
- [8] R. A. Vera, «openwebinars,» 31 03 2021. [En línea]. Available: <https://openwebinars.net/blog/que-es-laravel-caracteristicas-y-ventajas/>.
- [9] M. Romanos, «dreams,» 2022. [En línea]. Available: <https://www.dreams.es/transformacion-digital/desarrolladores-paginas-web/que-es-un-orm>.
- [10] A. B. Martínez, «platzi,» 2019. [En línea]. Available: https://platzi.com/clases/1920-eloquent-laravel/28515-que-es-un-orm-y-para-que-sirve-eloquent/?utm_source=google&utm_medium=cpc&utm_campaign=18798607679&utm_adgroup=&utm_content=&gclid=CjwKCAiAs8acBhA1EiwAgRFdwzH58kf4oiv1M GQepDEjbkyjBDFW15Y055HfZxzoL-DIHh.
- [11] oracle, «oracle,» [En línea]. Available: <https://www.oracle.com/ar/database/what-is-a-relational-database/>.
- [12] A. Robledano, «openwebinars,» 29 10 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-mysql/>.
- [13] M. A. Alvarez, «desarrolloweb,» 25 02 2019. [En línea]. Available: <https://desarrolloweb.com/articulos/que-es-react-motivos-uso.html>.
- [14] F. Flores, «openwebinars,» 22 07 2022. [En línea]. Available: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>.

- [15] amazon, «aws.amazon,» 2022. [En línea]. Available: <https://aws.amazon.com/es/what-is/restful-api/>.
- [16] questionpro, «questionpro,» [En línea]. Available: <https://www.questionpro.com/blog/es/que-es-un-estudio-de-caso/>.
- [17] S. Universidades, «becas-santander,» 21 12 2020. [En línea]. Available: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>.
- [18] Redhat, «redhat,» 2022 07 02. [En línea]. Available: <https://www.redhat.com/es/devops/what-is-agile-methodology>.
- [19] J. Roche, «deloitte,» 2022. [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>.
- [20] J. Roche, «deloitte,» 2022. [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html>.
- [21] asana, «asana,» 15 11 2021. [En línea]. Available: <https://asana.com/es/resources/requirements-gathering>. [Último acceso: 18 05 2022].
- [22] digite, «digite,» 2020. [En línea]. Available: <https://www.digite.com/es/agile/historias-de-usuarios/>. [Último acceso: 18 05 2022].
- [23] ealde, «ealde,» 27 08 2019. [En línea]. Available: <https://www.ealde.es/product-backlog-sprint-backlog/>. [Último acceso: 18 05 2022].
- [24] multimedia, «multimedia,» [En línea]. Available: <https://multimedia.uoc.edu/blogs/dii/es/que-es-una-interficie/>. [Último acceso: 18 05 2022].
- [25] D. Caldentey, «unir,» 10 02 2020. [En línea]. Available: <https://www.unir.net/ingenieria/revista/justinmind-la-herramienta-que-se-estudia-en-stanford-y-berkeley-llega-a-unir/>.
- [26] lucidchart, «lucidchart,» [En línea]. Available: <https://www.lucidchart.com/blog/es/como-disenar-una-arquitectura-de-software#:~:text=El%20dise%C3%B1o%20de%20una%20arquitectura,comenzar%20a%20elaborar%20un%20prototipo..> [Último acceso: 18 05 2022].
- [27] okhosting, «okhosting,» [En línea]. Available: <https://okhosting.com/blog/herramientas-de-desarrollo-de-software/>. [Último acceso: 18 05 2022].
- [28] J. Lucas, «openwebinars,» 04 08 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-nodejs/>.
- [29] C. Ekekenta, «logrocket,» 19 04 2022. [En línea]. Available: <https://blog.logrocket.com/implementing-jwt-authentication-laravel-9/>.

- [30] R. Barger, «freecodecamp,» 14 07 2022. [En línea]. Available: <https://www.freecodecamp.org/espanol/news/como-usar-axios-con-react/>.
- [31] R. KeepCoding, «keepcoding,» 20 10 2022. [En línea]. Available: https://keepcoding.io/blog/que-es-react-router/#Como_funciona_React_Router.
- [32] A. Maranto, «medium,» 15 02 2021. [En línea]. Available: <https://medium.com/nowports-tech/introducci%C3%B3n-a-react-hook-form-b3e725b707c4>.
- [33] yeeply, «yeeply,» [En línea]. Available: <https://www.yeeply.com/blog/que-son-pruebas-unitarias/>. [Último acceso: 26 05 2022].
- [34] ebooksonline, «ebooksonline,» 13 03 2021. [En línea]. Available: <https://ebooksonline.es/que-es-una-prueba-de-compatibilidad-prueba-haciaadelante-y-hacia-atras-ejemplo/>. [Último acceso: 26 05 2022].
- [35] manuel, «manuel,» [En línea]. Available: <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/pruebas/aceptacion/>. [Último acceso: 26 05 2022].
- [36] N. unidas, «Naciones unidas,» 27 03 2020. [En línea]. Available: <https://news.un.org/es/story/2020/03/1471932>. [Último acceso: 12 05 2022].
- [37] scielo, «scielo,» 01 03 2021. [En línea]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1028-99332021000200004. [Último acceso: 12 05 2022].
- [38] Q. Devs, «Quality Devs,» 16 09 2019. [En línea]. Available: <https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/>. [Último acceso: 12 05 2022].
- [39] OpenWebinars, «OpenWebinars,» 22 06 2021. [En línea]. Available: <https://openwebinars.net/blog/que-es-firebase-de-google/#:~:text=Firebase%20b%C3%A1sicamente%20es%20una%20plataforma,incrementar%20la%20base%20de%20datos>. [Último acceso: 12 05 2022].
- [40] ichlese, «ichlese,» [En línea]. Available: <https://es.ichlese.at/what-is-angular-material>. [Último acceso: 12 05 2020].
- [41] CEI, «CEI,» [En línea]. Available: <https://cei.es/que-es-figma/>. [Último acceso: 12 05 2022].
- [42] wikipedia, «wikipedia,» 03 05 2022. [En línea]. Available: https://en.wikipedia.org/wiki/Visual_Studio_Code. [Último acceso: 16 05 2022].
- [43] crehana, «crehana,» 08 02 2021. [En línea]. Available: https://www.crehana.com/blog/desarrollo-web/que-es-nodejs/?gclid=EAIalQobChMI7dfz4bT9QIVqwaICR2RZQ9REAAVASAEgJLzvD_BwE. [Último acceso: 16 05 2022].
- [44] loadview, «loadview,» 16 10 2020. [En línea]. Available: <https://www.loadview.com/>

testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/. [Último acceso: 16 05 2022].

- [45] datacentric, «datacentric,» 14 10 2015. [En línea]. Available: <https://www.datacentric.es/blog/bases-datos/importancia-bases-de-datos-2/#:~:text=Las%20principales%20utilidades%20que%20ofrece,la%20organizaci%C3%B3n%20de%20nuestra%20actividad..> [Último acceso: 19 05 2022].
- [46] highscalability, «highscalability,» 27 09 2021. [En línea]. Available: <https://highscalability.wordpress.com/tag/despliegue-de-software/>. [Último acceso: 18 05 2022].
- [47] rockcontent, «rockcontent,» 16 04 2019. [En línea]. Available: <https://rockcontent.com/es/blog/que-es-firebase/>. [Último acceso: 18 05 2022].
- [48] servnet, «servnet,» 30 03 2021. [En línea]. Available: <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programaci%C3%B3n-de-una-aplicaci%C3%B3n-web>. [Último acceso: 18 05 2022].
- [49] M. Coppola, «hubspot,» 19 10 2022. [En línea]. Available: <https://blog.hubspot.es/website/frontend-y-backend>.
- [50] T. Naeem, «astera,» 28 01 2020. [En línea]. Available: <https://www.astera.com/es/type/blog/rest-api-definition/>.
- [51] kaizenia, «kaizenia,» [En línea]. Available: <https://kzi.mx/que-son-los-artefactos-de-scrum/>. [Último acceso: 18 05 2022].
- [52] c. mvp, «campus mvp,» 15 10 2019. [En línea]. Available: <https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx#:~:text=MVC%20era%20initialmente%20un%20patr%C3%B3n,entre%20cada%20uno%20de%20ellos..> [Último acceso: 18 05 2022].
- [53] npmjs, «npmjs,» 09 05 2022. [En línea]. Available: <https://www.npmjs.com/package/firebase>. [Último acceso: 19 05 2022].
- [54] npmjs, «npmjs,» 19 03 2022. [En línea]. Available: <https://www.npmjs.com/package/@angular/fire>. [Último acceso: 19 05 2022].
- [55] libraries, «libraries,» [En línea]. Available: <https://libraries.io/npm/ngx-cookie-service>. [Último acceso: 19 05 2022].
- [56] npmjs, «npmjs,» 17 05 2022. [En línea]. Available: <https://www.npmjs.com/package/@ng-bootstrap/ng-bootstrap>. [Último acceso: 19 05 2022].
- [57] material, «material,» [En línea]. Available: <https://material.angular.io/>. [Último acceso: 19 05 2022].
- [58] loadview, «loadview,» [En línea]. Available: <https://www.loadview-testing.com/es/pruebas-de-carga/>. [Último acceso: 16 06 2022].

- [59] acens, «acens,» [En línea]. Available: <https://www.acens.com/wp-content/images/2016/02/google-pagespeed-insights-wp-acens.pdf>. [Último acceso: 07 07 2022].

7 ANEXOS

A continuación, se presenta cada uno de los Anexos que se ha utilizado para el desarrollo de la aplicación *web*, los cuales se encuentran detallados de la siguiente manera:

- **ANEXO I.** Resultado del programa anti plagio *Turnitin*.
- **ANEXO II.** Manual de Usuario.
- **ANEXO III.** Manual de Instalación.
- **ANEXO IV.** Credenciales de acceso y despliegue.

ANEXO I

A continuación, se presenta el certificado que el Director de Tesis ha emitido y en donde se evidencia el resultado que se ha obtenido en la herramienta antiplagio Turnitin.



CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 24 de febrero de 2023

De mi consideración:

Yo, Loarte Cajamarca Byron Gustavo, en calidad de Director del Trabajo de Integración Curricular titulado Desarrollo de una aplicación web asociado al DESARROLLO DE APLICACIÓN WEB DE GESTIÓN DE HISTORIAS Y CITAS MÉDICAS DEL CONSULTORIO MEDICO “DEVITA MEDICAL” elaborado por el estudiante Alberto Alfonso Heredia Iza de la carrera en Tecnología Superior en Desarrollo de Software, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito secciones: Descripción del componente desarrollado, Metodología, Resultados, Conclusiones y Recomendaciones, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 09%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,

A handwritten signature in blue ink, appearing to read "B. Loarte".

Loarte Cajamarca Byron Gustavo
Profesor Ocasional a Tiempo Completo
Escuela de Formación de Tecnólogos

ANEXO II

Recopilación de requerimientos

En la **TABLA X** se muestra los requerimientos que han sido recopilados al inicio del proyecto en donde se evidencia lo solicitado por el *Product Owner*.

TABLA X: Recopilación de requerimientos.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	RR - ID	ENUNCIADO DEL ÍTEM
Aplicación Web	RR-001	Tanto el usuario administrador y médico necesitan que la página de inicio cuente con mecanismos para proteger la información tales como un formulario de inicio de sesión.
	RR-002	Como administrador necesita: <ul style="list-style-type: none">• Iniciar sesión.• Cerrar sesión.
	RR-003	Como usuario administrador necesita: <ul style="list-style-type: none">• Gestionar perfil de usuarios.
	RR-004	Como usuario administrador necesita: <ul style="list-style-type: none">• Gestionar roles
	RR-005	Como usuario administrador necesita: <ul style="list-style-type: none">• Gestionar servicios y precios.
	RR-006	Como usuario médico necesita: <ul style="list-style-type: none">• Gestionar perfil.
	RR-007	Como usuario médico necesita: <ul style="list-style-type: none">• Gestionar perfil de los pacientes.
	RR-008	Como usuario médico necesita: <ul style="list-style-type: none">• Gestionar de historias clínicas.
	RR-009	Como usuario médico necesita: <ul style="list-style-type: none">• Gestionar citas médicas.

Historias de Usuario

Una vez que se ha realizado y completado el proceso de recopilación de requerimientos, a continuación, se implementan las historias de usuario de la aplicación web. Con ello se muestran 09 de las historias de usuario en base a la recopilación de requerimientos las cuales representan desde la **TABLA XI** corresponde a la historia de usuario 1 hasta la **TABLA XIX** que corresponde a la historia de usuario 8.

TABLA XI: Historia de Usuario – Visualizar página Inicio.

HISTORIA DE USUARIO	
Identificador (ID): HU-001	Usuario: Administrador, médico
Nombre historia: Visualizar página Inicio	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Sprint asignado: SB-001	
Programador responsable: Alfonso Heredia	
Descripción: La página de inicio tanto para el usuario administrador y usuario médico será el formulario login, con el cual podrán tener accesos a la aplicación web presentando sus respectivas credenciales “Usuario y contraseña”.	
Observación: Solo cuando se ingrese la contraseña se puede tener acceso a las diferentes opciones de la aplicación web.	

TABLA XII: Historia de Usuario – Realizar *login* y *logout*.

HISTORIA DE USUARIO	
Identificador (ID): HU-002	Usuario: Administrador
Nombre historia: Realizar <i>login</i> y <i>logout</i> .	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Sprint asignado: SB-001	

Programador responsable: Alfonso Heredia
Descripción:
Tanto el usuario administrador como el usuario médico, dentro de nuestra aplicación web tienen la capacidad de iniciar la sesión a través de un usuario “correo electrónico” y contraseña. Dentro de nuestra aplicación web tenemos la opción de cerrar sesión.
Observación:
Es necesario que al finalizar las tareas en la aplicación web cierre la aplicación, en otras palabras de clic en el logout.

TABLA XIII: Historia de Usuario – Gestionar perfil de usuarios.

HISTORIA DE USUARIO	
Identificador (ID): HU-003	Usuario: Administrador
Nombre historia: Gestionar perfil de usuarios.	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Sprint asignado: SB-001	
Programador responsable: Alfonso Heredia	
Descripción:	
El usuario con perfil administrador en la aplicación web tiene la posibilidad de modificar su información de perfil de usuarios una vez que haya iniciado sesión. Los datos que se pueden modificar son:	
<ul style="list-style-type: none"> • Nombre • Apellido • Correo • Fecha de Nacimiento • Contraseña • Dirección • Teléfono 	

A su vez en el enlace de perfil tenemos la opción de cambiar la contraseña.

Observación:

Los campos son validados de acuerdo a la información solicitada para que de esta forma nos aseguremos que la información ingresada sea la adecuada.

TABLA XIV: Historia de Usuario – Gestionar roles.

HISTORIA DE USUARIO	
Identificador (ID): HU-004	Usuario: Administrador
Nombre historia: Gestionar roles	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Sprint asignado: SB-002	
Programador responsable: Alfonso Heredia	
Descripción: El usuario administrador, tiene la posibilidad de gestionar los roles de cada usuario, con el fin de poder darles acceso a los módulos del administrador, cuando estas las ameriten.	
Observación: Los roles de administrador les dan la posibilidad de tener un acceso completo a la aplicación web.	

TABLA XV: Historia de Usuario – Gestionar precios de los servicios.

HISTORIA DE USUARIO	
Identificador (ID): HU-005	Usuario: Administrador
Nombre historia: Gestionar precios de los servicios	

Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: SB-001	
Programador responsable: Alfonso Heredia	
Descripción: El usuario Administrador es el único con acceso para poder añadir y modificar servicios de acuerdo con las necesidades del consultorio médico. Los datos que se pueden modificar son:	
<ul style="list-style-type: none"> • Nombre. • Descripción. • Precio. 	
Observación: El usuario médico solo puede visualizar los diferentes servicios, así como sus costos.	

TABLA XVI: Historia de Usuario – Visualizar perfil y cambiar contraseña

HISTORIA DE USUARIO	
Identificador (ID): HU-006	Usuario: Administrador, Médico
Nombre historia: Visualizar perfil y cambiar contraseña	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Sprint asignado: SB-001	
Programador responsable: Alfonso Heredia	
Descripción: El usuario con perfil administrador y usuario Médico en la aplicación web tiene la posibilidad de visualizar su información de perfil una vez que haya iniciado sesión. Los datos que se pueden modificar son:	
<ul style="list-style-type: none"> • Contraseña 	

Observación:

Los campos son validados de acuerdo con la información solicitada para que de esta forma nos aseguremos que la información solicitada sea la adecuada.

TABLA XVII: Historia de Usuario – Gestionar perfil paciente.

HISTORIA DE USUARIO	
Identificador (ID): HU-007	Usuario: Médico
Nombre historia: Gestionar perfiles del paciente	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Sprint asignado: SB-002	
Programador responsable: Alfonso Heredia	
Descripción: El usuario médico del consultorio puede visualizar la información del paciente, así como modificarla, los datos que pueden modificar son los siguientes:	
<ul style="list-style-type: none"> • Nombre • Fecha de nacimiento • Sexo • Alergias • Correo • Teléfonos • Dirección 	
Observación: Los campos son validados de acuerdo con la información solicitada para que de esta forma nos aseguremos que la información solicitada sea la adecuada.	

TABLA XVIII: Historia de Usuario – Gestionar historias clínicas.

HISTORIA DE USUARIO	
Identificador (ID): HU-008	Usuario: Médico
Nombre historia: Gestionar historias clínicas	

Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Sprint asignado: SB-001	
Programador responsable: Alfonso Heredia	
Descripción: El usuario médico, tiene la posibilidad de gestionar el historial médico de cada paciente, con el fin de poder visualizarlas y sobre todo realizar modificaciones al historial, cuando estas las ameriten.	
Observación: La información todas las historias medicas siempre tiene que estar disponible para el personal médico.	

TABLA XIX: Historia de Usuario – Gestionar citas médicas.

HISTORIA DE USUARIO	
Identificador (ID): HU-009	Usuario: Médico
Nombre historia: Gestionar citas médicas.	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Sprint asignado: SB-001	
Programador responsable: Alfonso Heredia	
Descripción: El usuario médico, tiene la posibilidad de gestionar las citas médicas de cada paciente, con el fin de poder visualizarlas y sobre todo realizar modificaciones, cuando estas las ameriten.	
Observación: La información las citas médicas siempre deben estar disponible para el personal médico.	

Product Backlog

Este apartado presenta la definición completa del *Product Backlog* siendo así **TABLA XX** la muestra cual es la prioridad que tiene cada historia de usuario en base a la importancia del mismo para el dueño del producto y su complejidad en el desarrollo.

TABLA XX: Product Backlog.

ELABORACIÓN DEL PRODUCT BACKLOG				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU-001	Visualizar página Inicio	1	Finalizado	Terminado
HU-002	Realizar <i>login</i> y <i>logout</i> .	1	Finalizado	Terminado
HU-003	Gestionar perfil de usuarios	1	Finalizado	Terminado
HU-004	Gestionar roles	1	Finalizado	Terminado
HU-005	Gestionar precios de servicios.	1	Finalizado	Terminado
HU-006	Visualizar <i>perfil</i> y cambiar contraseña.	2	Finalizado	Terminado
HU-007	Gestionar perfil del paciente	2	Finalizado	Terminado
HU-008	Gestionar historias médicas	2	Finalizado	Terminado
HU-009	Gestionar citas médicas	2	Finalizado	Terminado

Sprint Backlog

En el presente apartado, se muestra los diferentes *Sprints* que se han desarrollado a lo largo de este proyecto, en la **TABLA XXI** se puede apreciar las prioridades de cada historia de usuario en base a la información que se ha recopilado al comienzo de la creación del proyecto.

TABLA XXI: Sprint Backlog.

ELABORACIÓN DEL SPRINT BACKLOG						
ID – SB	NOMBRE	MÓDULO	HU - ID	HISTORIAS DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB-000	Ambiente de Desarrollo	-----	-----	-----	<ul style="list-style-type: none">• Analizar tanto los requerimientos funcionales como los no funcionales de nuestra aplicación.• Configuración de las herramientas a usar tanto en el <i>backend</i> como en el <i>frontend</i>.• Diseño de la base de datos.• Configuración del	20 Horas

					ambiente de desarrollo para el <i>backend</i> como para el <i>frontend</i> .	
Diseño e implementación de los módulos para usuario administrador.		Módulo Pagina Inicio	HU-001	Visualizar página Inicio	<ul style="list-style-type: none"> Definir tanto las páginas públicas como las privadas. métodos para la visualización de inicio. 	15 horas
		Módulo Login	HU-002	Realizar login y logout.	<ul style="list-style-type: none"> Diseño y consumo de <i>endpoints</i> para autentificación. Diseño de la interfaz <i>login</i>. Diseño de la interfaz para <i>logout</i>. Validación de datos. Validación de formularios. Pruebas de funcionalidad. 	20 Horas

SB-001	Módulo Admin	HU-003	Gestionar perfil Usuario	<ul style="list-style-type: none"> • Diseño y consumo de <i>endpoints</i> para gestión de perfil administrador • Diseño de interfaz perfil administrador. • Implementación de interfaz perfil administrador. • Diseño de la interfaz para cambio de contraseña. • Diseño e implementación formularios, CRUD del perfil administrador. • Validación de datos. • Validación de formularios. • Pruebas de funcionalidad. 	20 Horas
	Módulo			<ul style="list-style-type: none"> • Diseño y consumo de <i>endpoints</i>. • Diseño de interfaz de roles • Implementación de interfaz Historias para roles. 	15 horas

		Historia Clínicas	HU-004	Gestionar roles	<ul style="list-style-type: none"> Diseño e implementación formularios, CRUD Validación de datos. Validación de formularios. Pruebas de funcionalidad. 	
	Diseño e implementación de los módulos para el usuario médico	Módulo servicios	HU-005	Gestionar servicio s y precios.	<ul style="list-style-type: none"> Diseño y consumo de endpoints Diseño de interfaz para servicios Implementación de interfaz. Diseño e implementación formularios, CRUD del perfil administrador. Validación de datos. Validación de formularios. Pruebas de funcionalidad. 	15 Horas
		Módulo Login	HU-006	Visualizar perfil y cambiar contraseña	<ul style="list-style-type: none"> Diseño y consumo de endpoints. Diseño de la interfaz para 	

				ñia.	<p>visualizar información de usuario.</p> <ul style="list-style-type: none"> • Diseño de la interfaz para cambio de contraseña. • Validación de datos. • Validación de formularios. • Pruebas de funcionalidad. 	15 Horas
	Módulo Pacientes	HU-007	Gestionar perfil del paciente		<ul style="list-style-type: none"> • Diseño y consumo de endpoints. • Diseño de interfaz para servicios. • Implementación de interfaz. • Diseño e implementación formularios, CRUD del perfil administrador. • Validación de datos. • Validación de formularios. • Pruebas de funcionalidad. 	15 Horas

SB-002		Módulo Historias clínicas	HU-008	Gestionar historias clínicas	<ul style="list-style-type: none"> • Diseño y consumo de endpoints. • Diseño de interfaz para gestión de Historias clínicas. • Implementación de interfaz. • Diseño e implementación formularios, CRUD del perfil administrador. • Validación de datos. • Validación de formularios. • Pruebas de funcionalidad. 	15 Horas
--------	--	---------------------------	--------	------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------

		Módulo Citas	HU-009	Gestionar citas médicas	<ul style="list-style-type: none"> • Diseño y consumo de endpoints • Diseño de interfaz para para agendar citas. • Diseño e implementación formularios, <i>create</i>, <i>edit</i>, <i>delete</i> y <i>shows</i> del perfil administrador. • Validación de datos. • Validación de formularios. • Pruebas de funcionalidad. 	20 horas
SB-004	Pruebas de la aplicación web			<ul style="list-style-type: none"> • Pruebas unitarias. • Prueba de compatibilidad. • Prueba de aceptación. 		20 horas
SB-005	Despliegue de la aplicación web			<ul style="list-style-type: none"> • Publicación de la aplicación web en la consola de HostGator 		10 horas
	Documentación			<ul style="list-style-type: none"> • Trabajo de Integración Curricular. • Anexos. 		40 horas
TOTAL						240 H

Diseño de interfaces

A continuación, se visualiza los diferentes *mockups* de cada uno de los módulos de la aplicación *web*, donde se representa cada uno de los diseños para cada interfaz. Los cuales van desde la **Fig. 30** a la **Fig. 36**

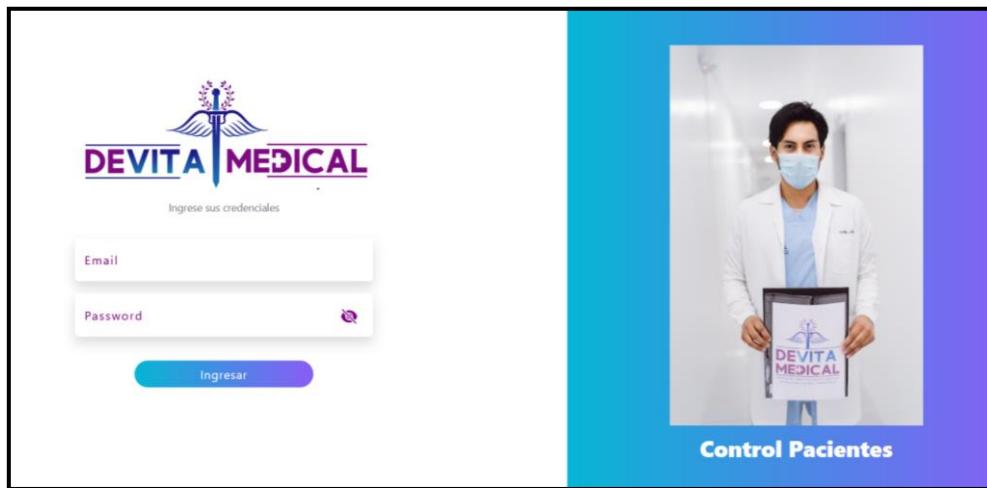


Fig. 30: Iniciar sesión.

A screenshot of the "Medico" registration module. At the top, the title "Medico" is displayed. Below the title, there is a descriptive text: "En este módulo es necesario asignar un servicio y un rol de usuario, en caso de crearse una nueva cuenta tendrá como contraseña: secretDev123. Por este motivo es obligatorio que el nuevo usuario cambie la contraseña." The form itself contains several input fields: "Nombres" and "Apellidos" (both with placeholder text "Apellido"), "Fecha y Hora" (with placeholder "username"), "Cedula (CI)" (with placeholder "Email"), "username" (with placeholder "Email"), "Email" (with placeholder "Email"), "Especialidad" (a dropdown menu with placeholder "--"), "Rol" (a dropdown menu with placeholder "--"), "Número Celular" (with placeholder "Número Celular"), "Número de Casa" (with placeholder "Número de Casa"), and "Dirección" (with placeholder "Dirección"). A large blue "GUARDAR" button is located at the bottom of the form.

Fig. 31: Registro usuario.

Usuario

En este módulo encontrarás los datos del Usuario y podrás cambiar la contraseña

[Volver Atrás](#)

* NOMBRES Luis
* APELLIDOS: Minaya
* EMAIL: gottlieb.beaulah@example.net
* NÚMERO CELULAR:
* STATE: Inactive
* NÚMERO DE CASA: N/A

Actualizar o Cambiar la contraseña

Password

Confirmar Password

ACTUALIZAR

Fig. 32: Módulo perfil.

Servicios

CREAR SERVICIO

Especialidad

Descripción

Precio (10) Sin decimales

GUARDAR

Fig. 33: Gestionar servicios.

Paciente

CREAR PACIENTE

Nombre Completo _____

Fecha y Hora _____ Cedula (CI) _____

Celular (0911112222) _____

Sexo: Alergias (Ninguna) _____

Dirección _____

Email address _____

GUARDAR

Fig. 34: Gestionar pacientes.

Cita

CREAR CITA

Paciente:

Especialidad: Medico: Seleccione una especialidad

Fecha y Hora _____

Descripción _____

GUARDAR

CREAR HISTORIA MEDICA

Signos y Síntomas
Escriba los Signos y Síntomas...

Diagnóstico
Escriba el Diagnóstico

Fig. 35: Gestionar citas.

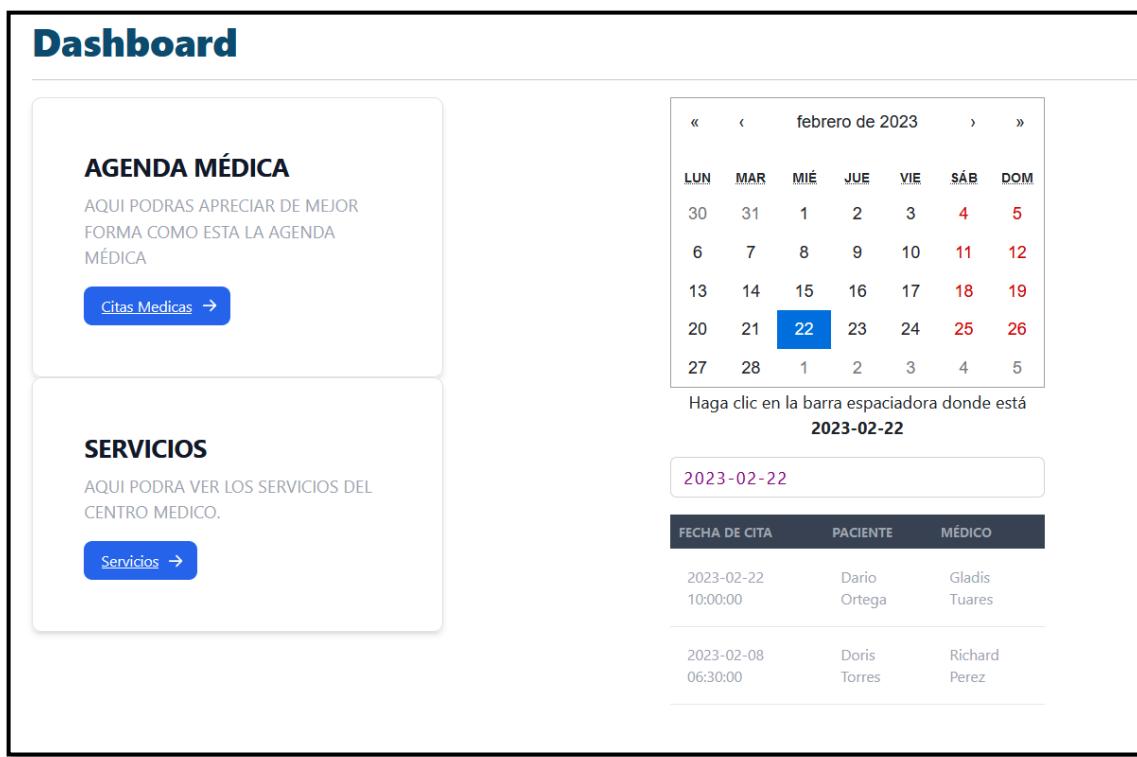


Fig. 36: Visualizar dashboard.

Diseño de la Base de datos

En la **Fig. 37** se puede visualizar la base de datos relacional la cual ha sido utilizada para el almacenamiento de los datos, conjuntamente con sus respectivas relaciones.

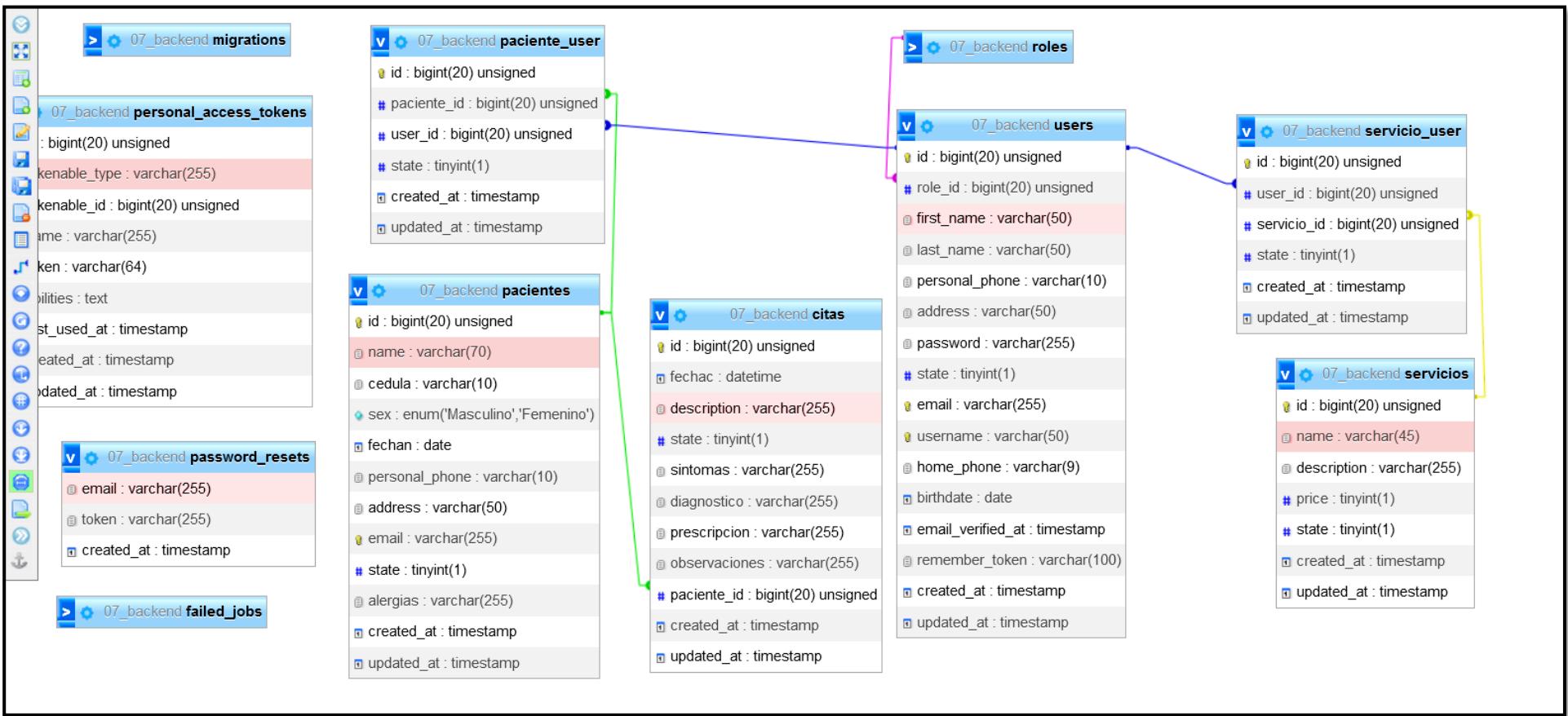


Fig. 37: Estructura de la base de datos relacional.

Pruebas

Al finalizar la codificación de la aplicación web, se procede a realizar las diferentes pruebas descritas a continuación.

Ejecución de pruebas unitarias y resultados

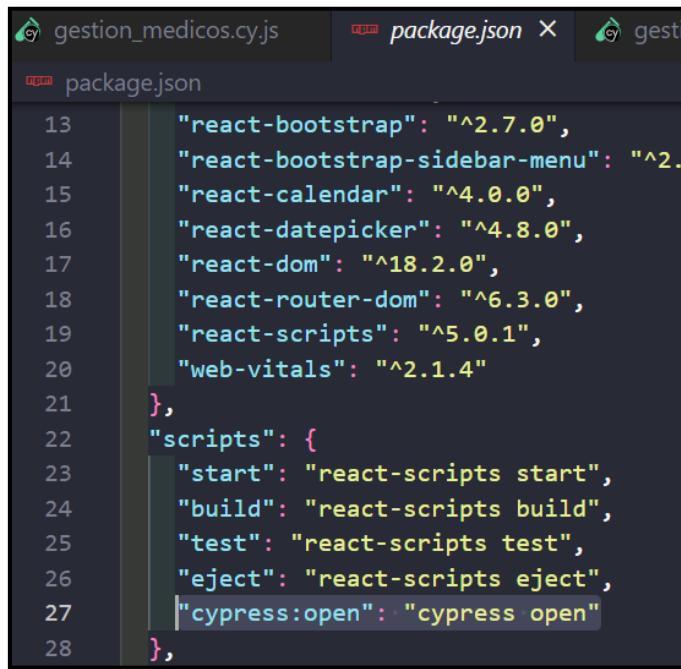
Como se observa en las siguientes figuras que van desde la **Fig. 38** hasta la **Fig. 62**, se observa las pruebas unitarias de los principales módulos de la aplicación web, para ello se ha utilizado la herramienta propia de Laravel que ayuda a realizar los *test* correspondientes.

Comando de instalación

```
npm install cypress --save-dev
```

Comando para iniciar

```
npm run cypress:open
```



The screenshot shows a terminal window with three tabs: 'gestion_medicos.cy.js', 'package.json', and 'gest'. The 'package.json' tab is active, displaying the following JSON code:

```
13   "react-bootstrap": "^2.7.0",
14   "react-bootstrap-sidebar-menu": "^2.0.0",
15   "react-calendar": "^4.0.0",
16   "react-datepicker": "^4.8.0",
17   "react-dom": "^18.2.0",
18   "react-router-dom": "^6.3.0",
19   "react-scripts": "^5.0.1",
20   "web-vitals": "^2.1.4"
21 },
22 "scripts": {
23   "start": "react-scripts start",
24   "build": "react-scripts build",
25   "test": "react-scripts test",
26   "eject": "react-scripts eject",
27   "cypress:open": "cypress open"
28 },
```

Fig. 38: Configuración archivo package.json.

ADMINISTRADOR

```
1 describe('template spec', () => {
2   it('verificado', () => {
3     cy.visit('http://localhost:3000/')
4     cy.get('input[id="email"]').type("gottlieb.beaulah@example.net")
5     cy.get('input[id="password"]').type("secret")
6     cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-base text-white tracking-wider bg-gradient-to-r from-cyan-500 to-violet-500 hover:from-cyan-600 hover:to-violet-600 w-3/8"]')
7   })
8 })
```

Fig. 39: Código de iniciar o cerrar sesión.

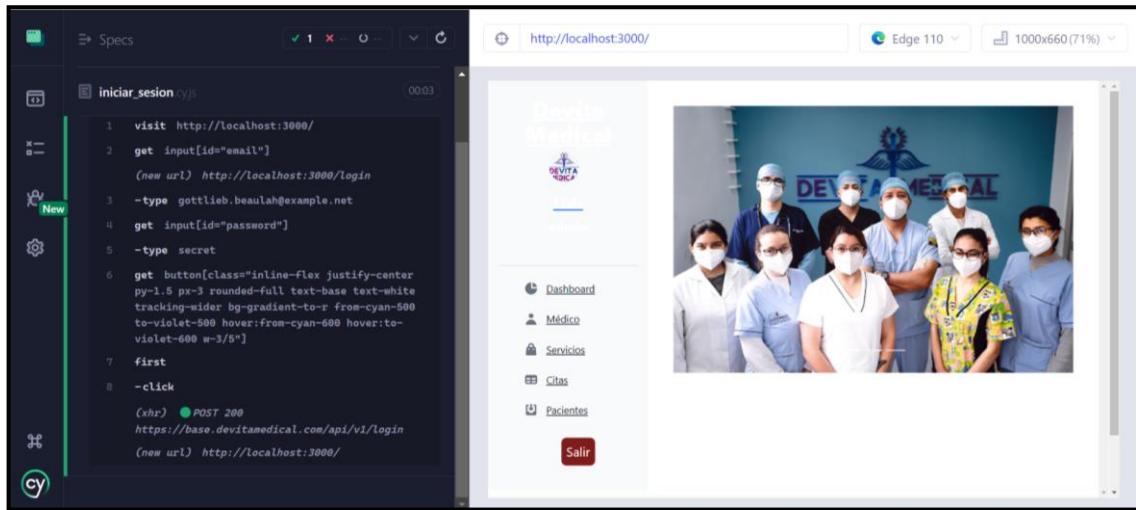


Fig. 40: Resultado de Iniciar o cerrar sesión.

The screenshot shows the Cypress Test Runner interface. The current file is 'actualización_perfil.cy.js'. The code within this file describes an 'Actualizar_perfil' test. It includes steps to visit the application, enter email and password, click a button, and type into several input fields. The browser window on the right shows the application's profile update page with a large text input field and a password confirmation field.

```
cypress > e2e > actualizar_perfil.cy.js > DIR_gestion.cyjs
cypress > e2e > actualizar_perfil.cy.js > describe('Actualizar_perfil ',()=>{
  it('verificado',()=>{
    cy.visit('http://localhost:3000/')
    cy.get('input[id="email"]').type("lamar52@example.com")
    cy.get('input[id="password"]').type("secret")
    cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-base text-white tracking-wider bg-gradient-to-r from-cyan-500 to-violet-500 hover:from-cyan-600 hover:to-violet-600 w-3/8"]')
    cy.get('h4[class="text-2xl font-black text-center text-white"]').first().click()
    cy.get('input[id="password"]').type("Secret1*")
    cy.get('input[id="password_confirmation"]').type("Secret1*")
    cy.get('input[class="bg-sky-800 w-full p-3 text-white uppercase font-bold rounded-lg hover:bg-sky-700 text-center text-white"]').click()
  })
})
```

Fig. 41: Código actualización de perfil.

The screenshot shows the results of updating a user's profile. On the left, the Cypress SpecRunner interface displays a successful POST request to update the password. On the right, the 'Usuario' (User) profile page in the Devita Medical application shows the updated password information.

Fig. 42: Resultado actualización de perfil.

The screenshot shows the terminal window running Cypress CLI commands for the 'gestion_medicos' module. The terminal output includes several Cypress test descriptions and their execution results.

```

cypress > e2e > gestion_medicos.cy.js ...
cypress > e2e > gestion_servicios.cy.js ...
cypress > e2e > gestion_citas.cy.js ...
cypress > e2e > gestion_pacientes.cy.js ...

cypress > describe ('gestion_medicos ',()=>{
  1   it('verificado',()=>{
  2     cy.visit('http://localhost:3000')
  3     cy.get('input[id="email"]').type("lamar52@example.com")
  4     cy.get('input[id="password"]').type("Secret1*")
  5     cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-base text-white tracking-wider bg-gradient-to-r from-cyan to-blue text-sm font-medium"]')
  6     cy.get('a[id="medico"]').first().click()
  7     cy.get('input[id="first_name"]').type("Elnaa")
  8     cy.get('input[class="bg-sky-800 w-full p-3 text-white uppercase font-bold rounded-lg hover:bg-sky-900 cursor-pointer transition-all"]')
  9   })
 10  })
 11  })
 12  })
 13  })
 14  })

```

Fig. 43: Código módulo médico.

The screenshot shows the results of managing doctors. On the left, the Cypress SpecRunner interface displays a successful POST request to update a director. On the right, the 'Gestión de Médicos' (Management of Doctors) page in the Devita Medical application lists three medical professionals with their names, phone numbers, specialties, and status.

NOMBRE	TELÉFONO	ESPECIALIDAD	ACCIÓN
Elna Leuschke	0918337876	Fisioterapia	VER EDITAR ACTIVO
Luis Minaya	0944052418	Fisioterapia	VER EDITAR ACTIVO
Santana Riebles	0920213608	Fisioterapia	VER EDITAR

Fig. 44: Resultado módulo médico.

```

cypress > e2e > gestion_servicios.cy.js > describe('GestionMSer') callback > it('verificado') callback
  1 describe('GestionMSer', () => {
  2   it('verificado', () => {
  3     cy.visit('http://localhost:3000')
  4     cy.get('input[id="email"]').type("lamar52@example.com")
  5     cy.get('input[id="password"]').type("Secret1**")
  6     cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-white tracking-wider bg-gradient-to-r from-cyan-500 to-violet-500 hover:from-cyan-500 hover:to-violet-500 w-3/5"]').click()
  7     cy.get('a[id="servicios"]').first().click()
  8     cy.get('button[class="bg-sky-800 block w-full text-white p-2 uppercase font-bold text-xs rounded-xl"]').first().click()
  9     cy.get('button["class="bg-green-800 block w-full text-white p-2 uppercase font-bold text-xs rounded-xl"]'.first().click()
 10    })
 11  })
 12})
 13})
 14})

```

Fig. 45: Código módulo servicios.

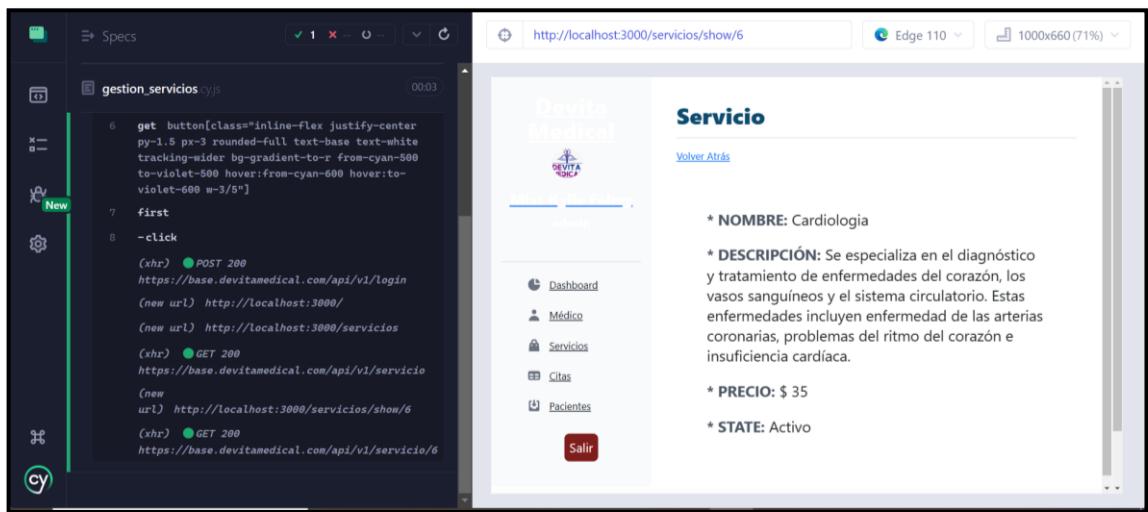


Fig. 46: Resultado de módulo servicios.

```

cypress > e2e > gestion_citas.cy.js > describe('GestionMCit') callback > it('verificado') callback
  1 describe('GestionMCit', () => {
  2   it('verificado', () => {
  3     cy.visit('http://localhost:3000')
  4     cy.get('input[id="email"]').type("lamar52@example.com")
  5     cy.get('input[id="password"]').type("Secret1**")
  6     cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-white tracking-wider bg-gradient-to-r from-cyan-500 to-red-500 hover:from-cyan-500 hover:to-red-500 w-3/5"]').click()
  7     cy.get('a[id="citas"]').first().click()
  8     cy.get('button["class="bg-sky-800 block w-full text-white p-2 uppercase font-bold text-xs rounded-xl"]'.first().click()
  9     cy.get('button["class="bg-red-800 block w-full text-white p-2 uppercase font-bold text-xs rounded-xl"]'.first().click()
 10    })
 11  })
 12})
 13})
 14})

```

Fig. 47: Código módulo citas.

Fig. 48: Resultado módulo citas.

```

Explorer (Ctrl+Shift+F) | js | gestion_servicios.cy.js | gestion_citas.cy.js | gestion_pacientes.cy.js X

cypress > e2e > gestion_pacientes.cy.js > ...
1  describe('GestionMPac', () => {
2    it('verificado', () => {
3      cy.visit('http://localhost:3000/')
4      cy.get('input[id="email"]').type("lamar52@example.com")
5      cy.get('input[id="password"]').type("Secret1**")
6      cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-white tracking-wider bg-gradient-to-r from-c")
7
8      cy.get('a[id="pacientes"]').first().click()
9      cy.get('button[class="bg-sky-800 block w-full text-white p-2 uppercase font-bold text-xs rounded-xl"]').first().click()
10     cy.get('button[class="bg-sky-800 block w-full text-white p-2 uppercase font-bold text-xs rounded-xl"]').first().click()
11   })
12 })
13
14

```

Fig. 49: Código módulo pacientes.

Fig. 50: Resultado módulo pacientes.

MÉDICO

```
1  describe('template spec', () => {
2    it('verificado', () => {
3      cy.visit('http://localhost:3000/')
4      cy.get('input[id="email"]').type("stark.abdul@example.org")
5      cy.get('input[id="password"]').type("secret")
6      cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-base text-white tracking-wider bg-gradient-to-r from-cya
7    })
8  })
```

Fig. 51: Código iniciar o cerrar sesión.

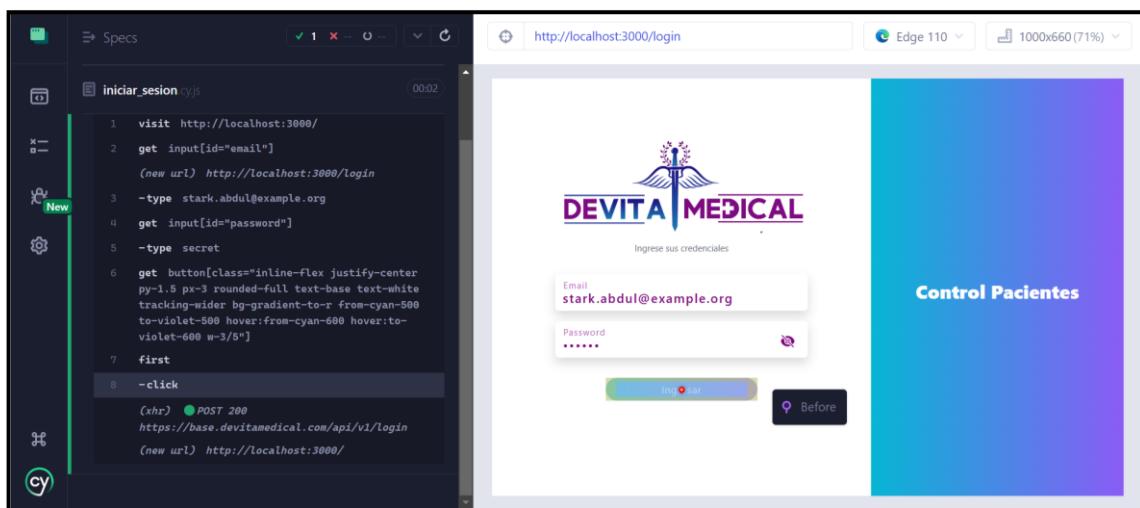


Fig. 52: Resultado iniciar o cerrar sesión.

The screenshot shows a terminal window with the following command and output:

```
cypress > e2e > DIR_actualización_perfil.cy.js
```

The terminal output shows the execution of the test file 'DIR_actualización_perfil.cy.js' with the following code:

```
1  describe ('Actualizar_perfil ',()=>{
2    it('verificado',()=>{
3      cy.visit('http://localhost:3000/')
4      cy.get('input[id="email"]').type("stark.abdul@example.org")
5      cy.get('input[id="password"]').type("secret")
6      cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-base text-white tracking-wider bg-gradient-to-r from-cyan-500
7
8      cy.get('h4[class="text-2xl font-black text-center text-white"]').first().click()
9      cy.get('input[id="password"]').type("Secret1*")
10     cy.get('input[id="password_confirmation"]').type("Secret1*")
11
12     cy.get('input[class="bg-sky-800 w-full p-3 text-white uppercase font-bold rounded-lg hover:bg-sky-600"]').click()
13     cy.get('button[class="m-auto text-white text-xl block mt-4 hover:text-red-300 text-center bg-red-900 py-1.5 px-3 rounded-full text-base tracking-wider"]').click()
14
15   })
```

Fig. 53: Código actualización de perfil.

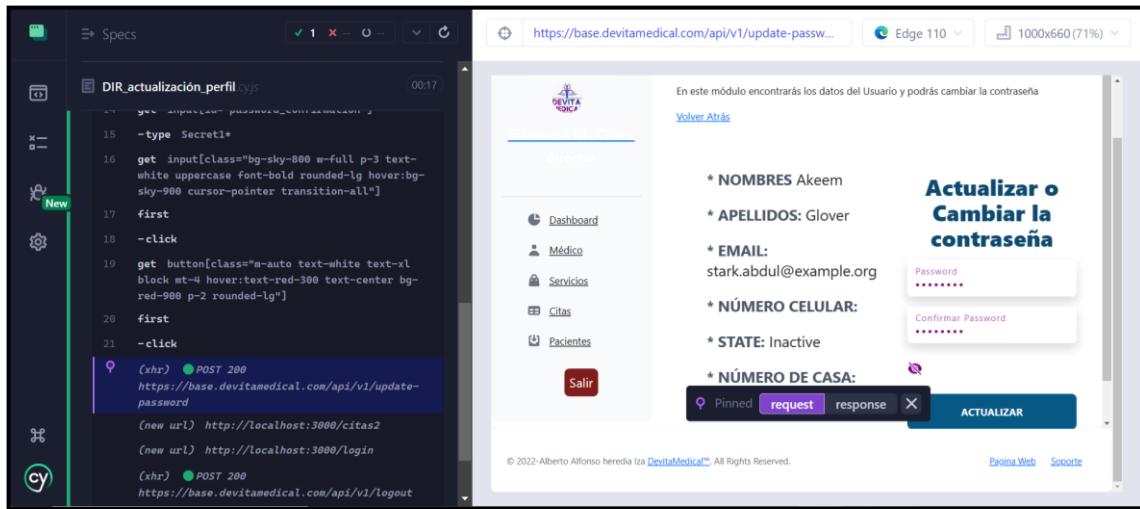


Fig. 54: Resultado de actualización de perfil.

```

cypress e2e > DIR_gestion.cy.js
describe('DIR_GestionMed', () => {
  it('verificado', () => {
    cy.visit('http://localhost:3000')
    cy.get('input[id="email"]').type("josefa.hoppe@example.net")
    cy.get('input[id="password"]').type("secret")
    cy.get('button[class="flex items-center p-2 text-base font-normal text-gray-900 rounded-lg dark:text-white hover:bg-gray-100 dark:hover:bg-gray-900 w-3/5"]').click()
    cy.wait(3000)
    cy.get('li[id="medico"]').first().click()
  })
})

```

Fig. 55: Código módulo médico.

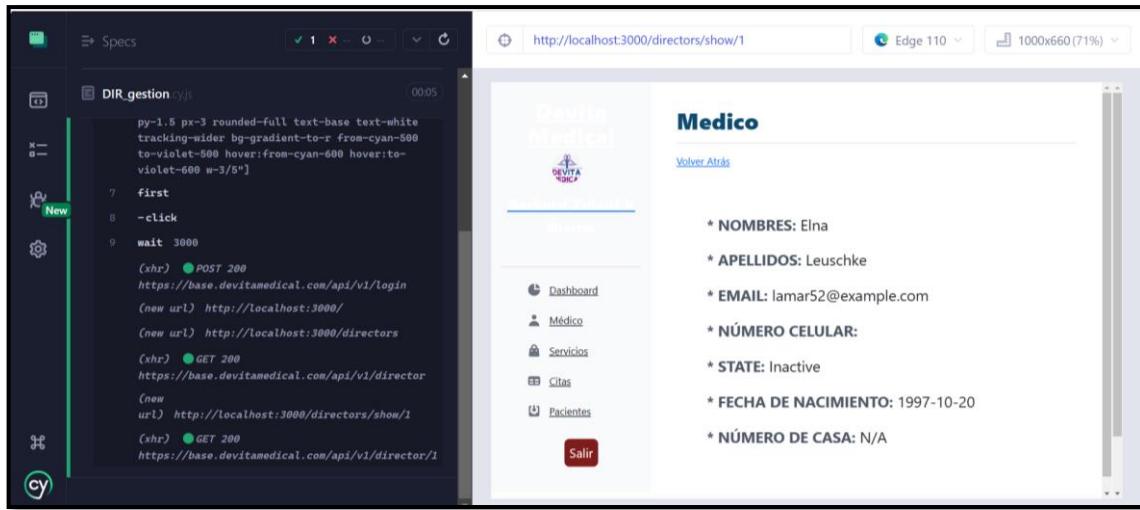


Fig. 56: Resultado módulo médico.

```

cypress > e2e > DIR_gestionSer.cy.js > describe('DIR_GestionMSer') callback > it('verificado') callback
  1 describe('DIR_GestionMSer', () => {
  2   it('verificado', () => {
  3     cy.visit('http://localhost:3000')
  4     cy.get('input[id="email"]').type("josefa.hoppe@example.net")
  5     cy.get('input[id="password"]').type("secret")
  6     cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-white tracking-wider bg-gradient-to-r from-cyan-500 to-violet-500 hover:from-cyan-500 hover:to-violet-600 w-3/5"]')
  7     cy.get('a[id="servicios"]').first().click()
  8     cy.get('button[class="bg-sky-800 block w-full text-white p-2 uppercase font-bold text-xs rounded-xl"]').first().click()
  9   })
 10 })
 11 })
 12 })
 13 })
 14 })
 15 })

```

Fig. 57: Código módulo servicios.

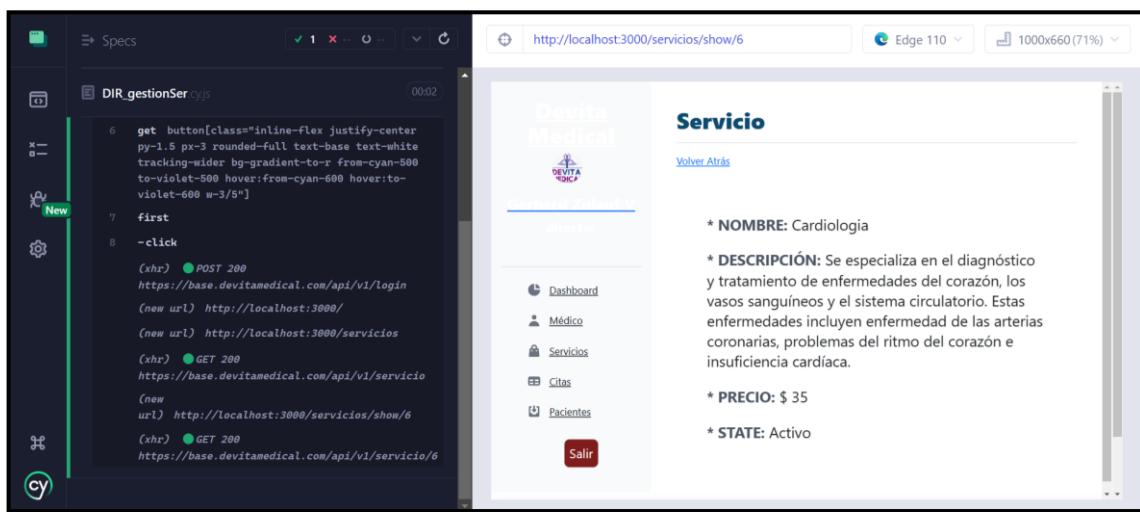


Fig. 58: Resultado módulo servicios.

```

cypress > e2e > DIR_gestionCit.cy.js > describe('DIR_GestionCit') callback > it('verificado') callback
  1 describe('DIR_GestionCit', () => {
  2   it('verificado', () => {
  3     cy.visit('http://localhost:3000')
  4     cy.get('input[id="email"]').type("roderick.batz@example.net")
  5     cy.get('input[id="password"]').type("secret")
  6     cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-white tracking-wider bg-gradient-to-r from-cyan-500 to-violet-500 hover:from-cyan-500 hover:to-violet-600 w-3/5"]')
  7     cy.get('a[id="citas"]').first().click()
  8     cy.get('button[class="bg-red-800 block w-full text-white p-2 uppercase font-bold text-xs rounded-xl"]').first().click()
  9   })
 10 })
 11 })
 12 })
 13 })

```

Fig. 59: Código ver de la “Historia clínica”.

The left pane shows the Cypress Test Runner interface with a spec file named 'DIR_gestionCita.js'. The code within the spec file is as follows:

```

5   -type secret
6   get button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-white tracking-wider bg-gradient-to-r from-cyan-500 to-violet-500 hover:from-cyan-600 hover:to-violet-600 w-3/5"]
7   first
8   -click
9
10  (xhr) POST 200
11  https://base.devitamedical.com/api/v1/login
12  (new url) http://localhost:3000/
13  (new url) http://localhost:3000/citas
14  > (xhr) GET 200
15  https://base.devitamedical.com/api/v1/cita
16  (new url) http://localhost:3000/citas/show/56
17  (xhr) GET 200
18  https://base.devitamedical.com/api/v1/cita/56

```

The right pane shows a screenshot of a web browser displaying the 'Historia Clínica' page. The page includes a sidebar with links to Dashboard, Médico, Servicios, Citas, and Pacientes, and a red 'Salir' button. The main content area displays patient information:

- * NOMBRE DEL PACIENTE: Andrea Salvador
- * FECHA DE NACIMIENTO: 2005-06-15
- * SEXO: Femenino
- * ALERGIAS: Penicilina
- * F/H DE CITA: 2023-02-09 07:30:00
- * DESCR: Response general
- * SINTOMAS: Dolor de cabeza. Fiebre. Malestar general. Dolores musculares y articulares generalizados.

Fig. 60: Resultado ver de la “Historia clínica”.

```

cypress > e2e > DIR_gestionPac.js > describe('DIR_GestionPac') callback > it('verificado') callback
1  describe('DIR_GestionPac', () => {
2    it('verificado', () => [
3      cy.visit('http://localhost:3000')
4      cy.get('input[id="email"]').type("roderick.batz@example.net")
5      cy.get('input[id="password"]').type("secret")
6      cy.get('button[class="inline-flex justify-center py-1.5 px-3 rounded-full text-white tracking-wider bg-gradient-to-r from-cyan-500 to-violet-500 hover:from-cyan-600 hover:to-violet-600 w-3/5"]')
7      cy.get('a[id="pacientes"]').first().click()
8      cy.get('button["class="bg-sky-800 block w-full text-white p-2 uppercase font-bold text-xs rounded-xl"]').first().click()
9    ])
10  })
11})
12

```

Fig. 61: Código módulo pacientes.

The left pane shows the Cypress Test Runner interface with a spec file named 'DIR_gestionPac.js'. The code within the spec file is as follows:

```

5   tracking-wider bg-gradient-to-r from-cyan-500 to-violet-500 hover:from-cyan-600 hover:to-violet-600 w-3/5"]
6   first
7   -click
8
9  (xhr) POST 200
10 https://base.devitamedical.com/api/v1/login
11 (new url) http://localhost:3000/
12 (new url) http://localhost:3000/pacientes
13 (xhr) GET 200
14 https://base.devitamedical.com/api/v1/paciente
15 (confirm) Are you sure?
16 (xhr) GET 200
17 https://base.devitamedical.com/api/v1/paciente/3
18 (destroy)
19 (xhr) GET 200
20 https://base.devitamedical.com/api/v1/paciente/3

```

The right pane shows a screenshot of a web browser displaying the 'Pacientes' page. The page includes a sidebar with links to Dashboard, Médico, Servicios, Citas, and Pacientes, and a red 'Salir' button. The main content area displays patient data:

NOMBRE	CÉDULA DE IDENTIDAD	FECHA DE NACIMIENTO	TELÉFONO	ACCIÓN
Andrea Salvador	0399503770	2005-06-15	0941250933	VER EDITAR ACTIVO
Dario Ortega	1975-07-22	0938990589		VER EDITAR ACTIVO

Fig. 62: Resultado módulo pacientes.

Ejecución de prueba de compatibilidad y resultados

Para poder realizar las pruebas de compatibilidad y verificar los resultados que dan como respuesta dichas pruebas, se ha probado con los navegadores web más utilizados por los usuarios comunes como se puede apreciar desde la **Fig. 63** hasta la **Fig. 65**

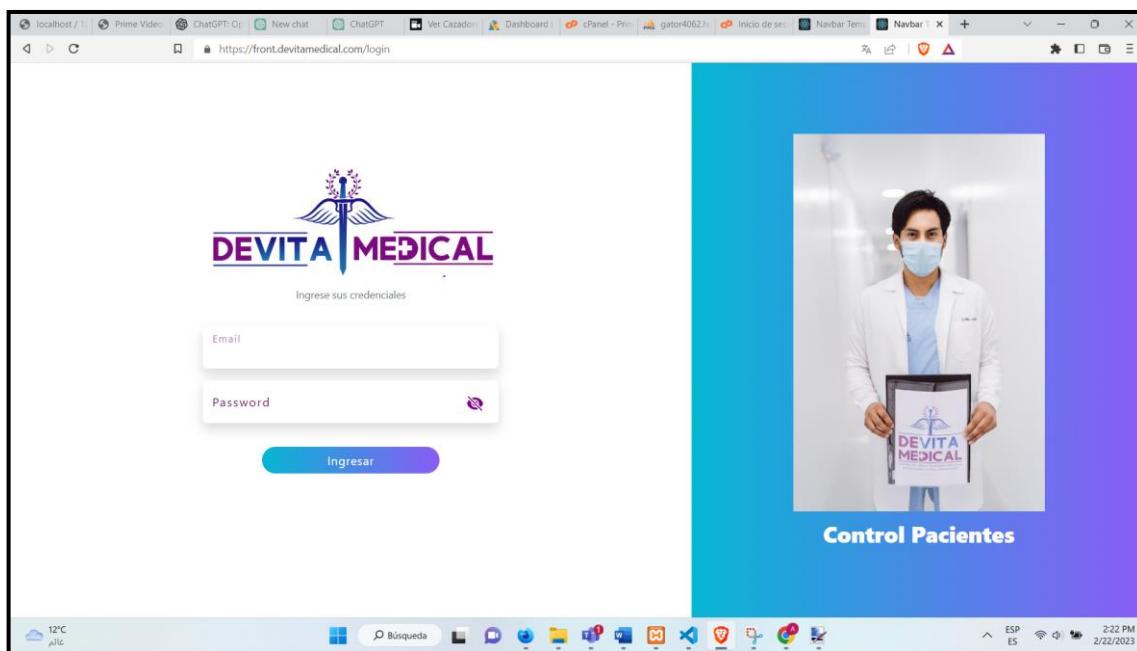


Fig. 63: Resultado de la prueba en el navegador *Brave*.

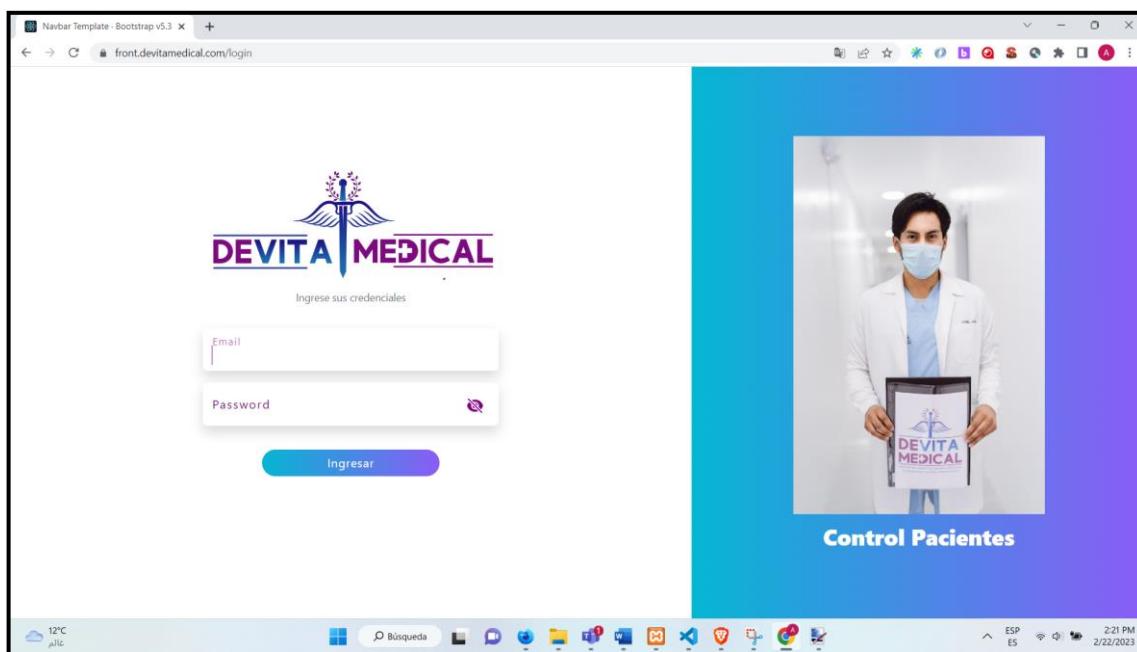


Fig. 64: Resultado de la prueba en el navegador *Chrome*.

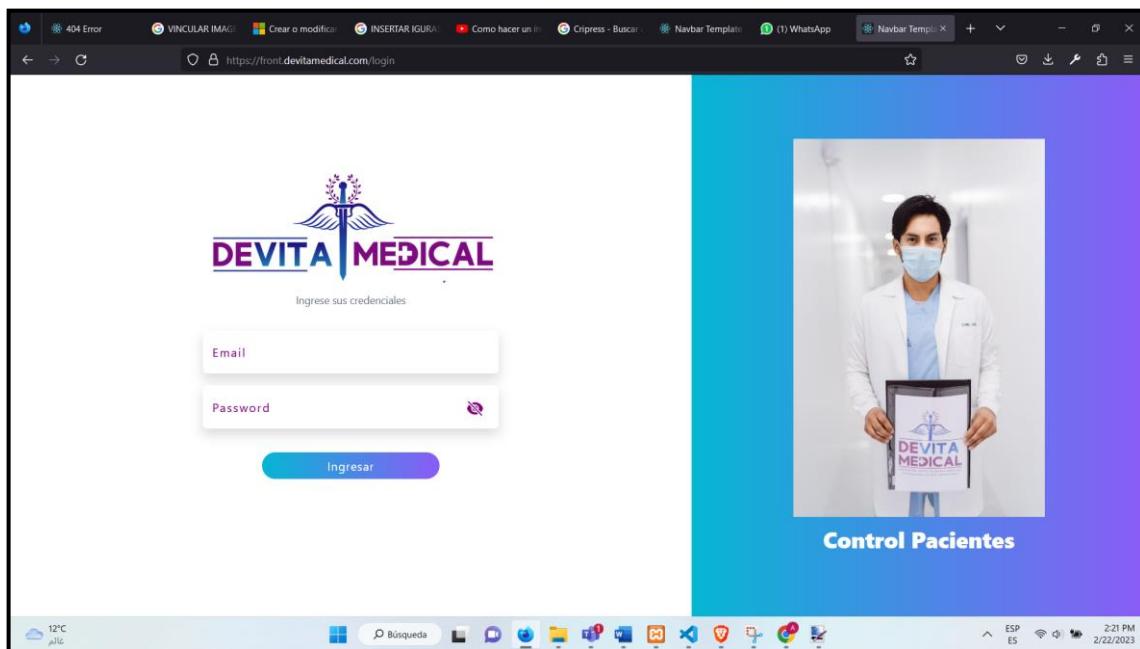


Fig. 65: Resultado de la prueba en el navegador *Firefox*.

Ejecución de pruebas de aceptación y resultados

En el presente apartado se muestra las pruebas de aceptación, donde se detallan los diferentes procedimientos que se han realizado para así validar cada uno de los resultados entregados por los módulos respectivos. En ese sentido el detalle de las pruebas se las puede apreciar de mejor manera en las tablas que van desde la **TABLA XXII** hasta la **TABLA XXX**.

TABLA XXII: Prueba de aceptación 1 – visualizar página de inicio de sesión.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA-001	Identificador historia de Usuario: HU-001
Nombre: visualización de página de inicio de sesión	
Descripción: El usuario que ingrese a la aplicación web no puede visualizar este sin primero ingresar sus credenciales	
Pasos de ejecución: <ul style="list-style-type: none">• Ingresar la URL de la aplicación web.• Visualización de formulario para ingresar credenciales	

Resultado deseado:
La aplicación web restringe el acceso de a cualquier persona que ingrese a esta, solicitando las respectivas credenciales.
Evaluación de la prueba
Se han cumplido los requisitos solicitados.

TABLA XXIII: Prueba de aceptación 2 – Ingreso a la aplicación web.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA-002	Identificador historia de Usuario: HU-002
Nombre: Ingreso a la aplicación web	
Descripción: El usuario administrador al ingresar sus credenciales (Correo electrónico, clave de seguridad) tiene acceso a la aplicación web.	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar la URL de la aplicación web. • Visualización de formulario para ingresar credenciales • Ingreso de Credenciales • Dar clic en ingresar 	
Resultado deseado:	
La aplicación web permite el correcto ingreso del usuario administrador sin presentar ningún inconveniente, si las credenciales son las correctas.	
Evaluación de la prueba	
Se han cumplido los requisitos solicitados.	

TABLA XXIV: Prueba de aceptación 3 – Gestionar perfil médico.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA-003	Identificador historia de Usuario: HU-003
Nombre: Gestionar el perfil usuario	

Descripción:
El usuario administrador tiene la posibilidad de gestionar el perfil médico para poder añadir o editar datos de los médicos
Pasos de ejecución:
<ul style="list-style-type: none"> • Ingresar la URL de la aplicación web. • Visualización de formulario para ingresar credenciales • Ingreso de Credenciales • Dar clic en ingresar • Ingresar al módulo médico. • Dar clic en editar para editar si así lo amerita el caso.
Resultado deseado:
Con las credenciales administrador el usuario puede modificar fácilmente el perfil médico.
Evaluación de la prueba
Se han cumplido los requisitos solicitados.

TABLA XXV: Prueba de aceptación 4 – Gestión de roles.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA-004	Identificador historia de Usuario: HU-004
Nombre: Gestión de Roles	
Descripción:	
El usuario administrador tiene la posibilidad de gestionar los roles de los médicos para así darles permisos de administrador a estos.	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar la URL de la aplicación web. • Visualización de formulario para ingresar credenciales • Ingreso de Credenciales • Dar clic en ingresar • Ingresar al módulo médico. • Dar clic en editar para editar 	

<ul style="list-style-type: none"> • En el formulario dar clic en rol • Seleccionar rol correspondiente.
<p>Resultado deseado:</p> <p>El usuario administrador tiene la facultad de asignar roles de administrador a los diferentes usuarios.</p>
<p>Evaluación de la prueba</p> <p>Se han cumplido los requisitos solicitados.</p>

TABLA XXVI: Prueba de aceptación 5 – Gestionar servicios.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA-005	Identificador historia de Usuario: HU-005
Nombre: Gestionar servicios	
Descripción:	
El usuario administrador tiene la posibilidad de gestionar los diferentes servicios que proporciona el centro médico.	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar la URL de la aplicación web. • Visualización de formulario para ingresar credenciales • Ingreso de Credenciales • Dar clic en ingresar • Ingresar al módulo servicios. • Dar clic en editar o añadir según amerite el caso. 	
Resultado deseado:	
El usuario administrador tiene la posibilidad de gestionar los diferentes servicios que oferta el centro médico.	
Evaluación de la prueba	
Se han cumplido los requisitos solicitados.	

TABLA XXVII: Prueba de aceptación 6 – Ingreso usuario médico.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA-006	Identificador historia de Usuario: HU-006
Nombre: Ingreso del usuario médico	
Descripción: El usuario médico al ingresar sus credenciales (Correo electrónico, clave de seguridad) tiene acceso a la aplicación web.	
Pasos de ejecución: <ul style="list-style-type: none">• Ingresar la URL de la aplicación Web.• Visualización de formulario para ingresar credenciales• Ingreso de Credenciales• Dar clic en ingresar	
Resultado deseado: <p>La aplicación web permite el correcto ingreso del usuario médico sin presentar ningún inconveniente, si las credenciales son las correctas.</p>	
Evaluación de la prueba <p>Se han cumplido los requisitos solicitados.</p>	

TABLA XXVIII: Prueba de aceptación 7 – Gestionar perfil de paciente.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA-007	Identificador historia de Usuario: HU-007
Nombre: Gestionar perfil del paciente	
Descripción: <p>El usuario médico tiene la posibilidad de gestionar el perfil paciente del centro médico.</p>	
Pasos de ejecución: <ul style="list-style-type: none">• Ingresar la URL de la aplicación web.• Visualización de formulario para ingresar credenciales• Ingreso de Credenciales• Dar clic en ingresar	

- | |
|------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • Ingresar al módulo paciente. • Dar clic en editar o añadir según amerite el caso. |
|------------------------------------------------------------------------------------------------------------------------------------------------|

Resultado deseado:

El usuario médico tiene la posibilidad de gestionar el perfil paciente

Evaluación de la prueba

Se han cumplido los requisitos solicitados.

TABLA XXIX: Prueba de aceptación 8– Gestionar historia médica del paciente.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA-008	Identificador historia de Usuario: HU-008
Nombre: Gestionar historia médica.	
Descripción:	
El usuario médico tiene la posibilidad de gestionar las historias médicas de los pacientes del centro médico.	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar la URL de la aplicación web. • Visualización de formulario para ingresar credenciales • Ingreso de Credenciales • Dar clic en ingresar • Ingresar al módulo citas. • Editar el formulario Historia clínica que se encuentra dentro de citas 	
Resultado deseado:	
Al ingresar a la aplicación web el usuario médico tiene la posibilidad de gestionar el historial médico del paciente de forma sencilla.	
Evaluación de la prueba	
Se han cumplido los requisitos solicitados.	

TABLA XXX: Prueba de aceptación 9 – Agendar Citas.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA-009	Identificador historia de Usuario: HU-009
Nombre: Agendar citas	
Descripción: El usuario médico tiene la posibilidad de gestionar el módulo citas y agendarlas en este mismo.	
Pasos de ejecución: <ul style="list-style-type: none">• Ingresar la URL de la aplicación Web.• Visualización de formulario para ingresar credenciales• Ingreso de Credenciales• Dar clic en ingresar• Ingresar al módulo citas.• Editar el formulario citas.	
Resultado deseado: La aplicación <i>web</i> permite al médico el gestionar el módulo citas del centro médico.	
Evaluación de la prueba Se han cumplido los requisitos solicitados.	

ANEXO III

A continuación, para visualizar el Manual de Usuario de la aplicación *web* se debe digitar la siguiente URL:

<https://youtu.be/FGQ-EoBnzPg>

En donde se explica de forma clara y sencilla las diversas funcionalidades de la aplicación *web*, así como cada uno de los perfiles que forman parte de este componente.

ANEXO IV

A continuación, se presenta las credenciales de acceso de la aplicación *web*, además del repositorio de GitHub, donde se encuentra todo el código y los pasos a seguir para su instalación en el apartado del README.

<https://github.com/Aheredia05/DevitaMedical.git>

Credenciales para el acceso de la aplicación *web*

Para ingresar a la aplicación *web* ya en producción, se ingresa mediante la URL:

<https://front.devitamedical.com>

Credenciales del perfil administrador:

- Correo del administrador: david.iza@gmail.com
- Contraseña: **\$Ecret123***

Credenciales del perfil médico:

- Correo del médico: belen.anchaguano@gmail.com
- Contraseña: **\$Eret123***