



*Universidad Nacional del Este*  
*Facultad Politécnica*

---

## **Ingeniería de Software III**

### **SADRM**

#### **Realizado por:**

- |                                     |                    |
|-------------------------------------|--------------------|
| • José Mario Barreto Jara           | <b>CI: 5931713</b> |
| • Mackey Ryuto Sugawara Mochinaga   | <b>CI: 4791872</b> |
| • Diego Samuel Velastiqui Santacruz | <b>CI: 5240614</b> |

**Docente: Osvaldo González**

**Semestre: 7º**

**Carrera: Ingeniería de Sistemas**

**Ciudad del Este – 2023**

# Índice

<b>Capítulo I – Etapa de Ingeniería de Requisitos .....</b>	<b>4</b>
<b>1.1 Introducción.....</b>	<b>4</b>
<b>1.2 Objetivo del capítulo .....</b>	<b>4</b>
<b>1.3 Ámbito del Sistema.....</b>	<b>4</b>
<b>1.4 Objetivos del Sistema a ser desarrollado .....</b>	<b>5</b>
<b>1.5 Conceptos generales y acrónimos. ....</b>	<b>5</b>
<b>1.6 Herramientas a ser utilizadas.....</b>	<b>6</b>
<b>1.7 Perfiles y funciones más importantes en el sistema.....</b>	<b>7</b>
<b>1.7.1 Diagrama de roles.....</b>	<b>8</b>
<b>1.8 Técnicas de levantamiento de requisitos aplicadas. ....</b>	<b>9</b>
<b>1.9 Análisis de requerimientos de software (requisitos) .....</b>	<b>10</b>
<b>1.9.1. Descripción general .....</b>	<b>10</b>
<b>1.9.2 Perspectiva de la aplicación.....</b>	<b>10</b>
<b>1.9.3 Restricciones de la aplicación .....</b>	<b>10</b>
<b>1.9.4 Necesidades básicas de la aplicación.....</b>	<b>11</b>
<b>1.9.5 Requerimientos Específicos .....</b>	<b>12</b>
<b>1.9.5.1 Requerimientos Funcionales .....</b>	<b>12</b>
<b>1.9.5.2 Requerimientos NO Funcionales .....</b>	<b>15</b>
<b>Capítulo 2 – Diseño y Estimación de Costos .....</b>	<b>18</b>
<b>2.1 Introducción.....</b>	<b>18</b>
<b>2.2 Diseño de Datos .....</b>	<b>18</b>
<b>2.2.1 Modelo Entidad-Relación .....</b>	<b>18</b>
<b>2.2.2 Modelo Relacional .....</b>	<b>19</b>
<b>2.2.3 Diccionario de Datos .....</b>	<b>19</b>
<b>2.3 Diseño arquitectónico.....</b>	<b>22</b>
<b>2.4 Diseño de interfaz .....</b>	<b>24</b>
<b>2.4 Diseño procedural .....</b>	<b>30</b>
<b>2.4.1 Diagrama de Caso de Uso.....</b>	<b>30</b>
<b>2.4.2 Diagramas de Clases .....</b>	<b>43</b>
<b>2.4.3 Diagrama de Estado .....</b>	<b>43</b>
<b>2.5 Estimación de Costos a través de Puntos de Función y COCOMO II. ....</b>	<b>43</b>

2.5.1 Por Punto de Función de IFPUG (International Function Points Users Groups) ...	43
<b>Capítulo 3 – Desarrollo</b> .....	48
3.1 Introducción.....	48
3.2 Metodología de Desarrollo Ágil .....	48
3.2.2 Product Backlog (requisitos) .....	49
3.2.3 Descripción de cada Sprint y entregables .....	50
3.3 Herramientas colaborativas de desarrollo .....	52
3.4 Herramientas colaborativas de desarrollo .....	55
3.5 Herramienta de organización en equipo .....	56
3.6 Estándares de código limpio que serán utilizados en toda la codificación del proyecto.....	57
<b>Capítulo 4 – Pruebas y Calidad del Software</b> .....	58
4.1 Introducción.....	58
4.2 El objetivo de las pruebas .....	58
4.3 ¿Cómo se llevarán a cabo las pruebas? .....	58
4.4 Aplicación de herramientas de pruebas .....	69
<b>Capítulo 5 – Pruebas y Calidad del Software</b> .....	77
5.1 Introducción.....	77
5.2 Mantenimiento del software .....	77
5.3 Control de Versiones.....	79
5.4 Encuesta de satisfacción del cliente .....	81
5.4.1 Resultados de la encuesta .....	84
5.5 Evidencia de logros de todos los Requisitos Funcionales.....	90
5.6 Evidencia de logros de todos los Requisitos No Funcionales.....	93
5.7 Conclusiones .....	96
<b>Anexo</b> .....	97
<b>Referencias bibliográficas</b> .....	98

# Capítulo I – Etapa de Ingeniería de Requisitos

## 1.1 Introducción

En este capítulo se desarrolla la Especificación de Requisitos Software (ERS) para el aplicativo a ser desarrollado **SADRM**. Este capítulo se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software IEEE 830.

## 1.2 Objetivo del capítulo

Definir y presentar de forma ordenada los requisitos y especificaciones que deberá cumplir el software a construir, el cual permitirá

1. Mejora en la productividad: El software permite una mejor gestión del flujo de trabajo, lo que puede aumentar la productividad del personal y reducir el tiempo necesario para realizar las tareas.
2. Mejor Eficiencia: El software permite a los pacientes programar sus citas en línea, lo que reduce la cantidad de tiempo que los empleados tienen que pasar en el teléfono. Además, la automatización del proceso de agendamiento puede reducir el tiempo que los empleados tienen que dedicar a esta tarea.
3. Mayor satisfacción del paciente: El software puede mejorar la satisfacción del paciente al proporcionar un mejor servicio y reducir el tiempo de espera para los estudios.
4. Reducción de errores: La automatización del proceso de gestión de información puede reducir la posibilidad de errores humanos, como errores de transcripción o de interpretación.

## 1.3 Ámbito del Sistema

**IMAP S.A.** Centro de diagnóstico por imágenes y especiales médicas. Cuenta con equipamiento digital de última generación y con software especializados de alta precisión, permitiéndole al especialista realizar un diagnóstico más efectivo.

En la empresa se realizan los siguientes estudios:

RESONANCIA MAGNÉTICA, TOMOGRAFÍA MULTISLICE, RADIOGRAFÍA DIGITAL, ECOGRAFÍA, DENSITOMETRÍA ÓSEA, MAMOGRAFÍA, ANATOMÍA PATOLÓGICA, ELECTROENCEFALOGRAMA PEDIÁTRICO, LABORATORIO ANÁLISIS CLÍNICOS, VIDEONISTAGMOGRAFÍA, ESPECIALIDADES MÉDICAS.

Misión, Visión, Valores de la empresa.

**MISIÓN:** Ofrecer un servicio eficiente con garantía de resultado en la especialidad de diagnóstico por imágenes, dando respuesta a las necesidades de nuestra sociedad.

**VISIÓN:** Pretendemos ser un Centro de Diagnóstico por Imágenes de referencia en la región, en las especialidades de Radiología, Mamografía, Densitometría, Ecografía, Tomografía y Resonancia Magnética. Queremos destacarnos con una tecnología superior, profesionalismo y excelencia en la atención para cubrir las necesidades de pacientes, clínicas y otras instituciones.

**VALORES:**

Diagnóstico por Imágenes adopta la Calidad, Ética y Honestidad para regir su conducta porque entiende que son los valores que reflejan la forma de trabajo de la Institución y el accionar de quienes forman parte del mismo, poniendo el mayor esfuerzo y responsabilidad en la realización de sus tareas.

Actualmente la empresa no cuenta con un sistema para el agendamiento de consultas, se realizan por llamada y los datos se guardan en una planilla Excel.

## **1.4 Objetivos del Sistema a ser desarrollado**

- Agilizar el proceso de agendamiento para los estudios.
- Contar con niveles de acceso de acuerdo a las funciones.
- Permitir conexión desde cualquier dispositivo conectado a internet.

## **1.5 Conceptos generales y acrónimos.**

- **Front-end:** Es todo lo que se puede observar del lado del cliente, es decir, el apartado visual de la aplicación. Por lo general se ocupan tecnologías como: HTML5, CSS3, JavaScript entre otros para la construcción de las interfaces. En este proyecto se utiliza PHP, CSS como herramienta de desarrollo para el Front-end.

- **Back-end:** El trabajo del desarrollador en la parte de back-end es todo lo que va anclado al servidor de la aplicación, al core del negocio, sin este la aplicación desarrollada en el front-end no tendría una funcionalidad.
- **Estudio de resonancia magnética (MRI):** Un examen no invasivo que utiliza campos magnéticos y ondas de radio para producir imágenes detalladas de los órganos y tejidos del cuerpo.
- **Paciente:** La persona que busca programar un estudio de resonancia magnética.
- **Interfaz de usuario (UI):** La forma en que los usuarios interactúan con el software, a través de elementos visuales como botones, menús y campos de entrada de datos.
- **Interfaz de programación de aplicaciones (API):** Un conjunto de reglas y protocolos utilizados para definir cómo los diferentes sistemas de software interactúan entre sí.

## 1.6 Herramientas a ser utilizadas

- **Visual Studio Code:** es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.
- **XAMPP:** es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.
- **GitHub:** es un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Éste permite a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso.
- **MySQLWorkBench:** es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL, que estaremos utilizando para poder crear los modelos de entidad relación.
- **Star UML:** es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture).
- **brModelo:** herramienta open source para la creación de modelos relacionales de una Base de Datos.

- **Figma:** es una aplicación basada en navegador para diseñar UI y UX, para diseñar interfaces que cuenta con excelentes herramientas de diseño, creación de prototipos y generación de código.
- **Http basic authentication headers:** RFC 7235 define el marco de autenticación HTTP que puede ser usado por un servidor para revisar la solicitud de un cliente y proveer credenciales en la forma de usuario y contraseña.

## 1.7 Perfiles y funciones más importantes en el sistema.

Hay cuatro perfiles importantes con diferentes funciones y niveles de acceso: paciente, recepción, encargado y administrador.

1. **Paciente:** Este perfil es utilizado por los pacientes que desean programar un estudio de resonancia magnética. Las funciones principales de este perfil son:
  - Ver la disponibilidad de horarios para la programación de estudios de resonancia magnética.
  - Programar una cita para un estudio en un horario disponible.
2. **Recepción:** Este perfil es utilizado por el personal de recepción en la clínica o hospital que se encarga de la programación de citas de los pacientes. Las funciones principales de este perfil son:
  - Acceder a la información de programación de citas de los pacientes.
  - Agendar, modificar o cancelar citas para estudios.
  - Ver la disponibilidad de horarios para programar nuevas citas.
  - Ver y generar informes y estadísticas sobre la programación de citas y otros datos relevantes.
  - Ver y generar informes y estadísticas sobre la programación de citas y otros datos relevantes.
3. **Encargado Principal:** Este perfil es utilizado por el personal administrativo o gerencial en la clínica o hospital. Las funciones principales de este perfil son:
  - Acceder a la información de programación de citas de todos los pacientes.

- Ver y generar informes y estadísticas sobre la programación de citas y otros datos relevantes.
  - Configurar la disponibilidad de horarios para la programación de citas.
4. Administrador: Este perfil es el de más alto rango en el sistema. Las funciones principales de este perfil son:
- Gestionar las cuentas de usuario, perfiles y permisos de acceso al sistema.
  - Debe tener acceso a todas las funcionalidades del sistema. Acceder a la información de programación de citas de todos los pacientes.
  - Ver y generar informes y estadísticas sobre la programación de citas y otros datos relevantes.
  - Configurar la disponibilidad de horarios para la programación de citas.
  - Realizar tareas de mantenimiento y actualización del sistema.

### 1.7.1 Diagrama de roles

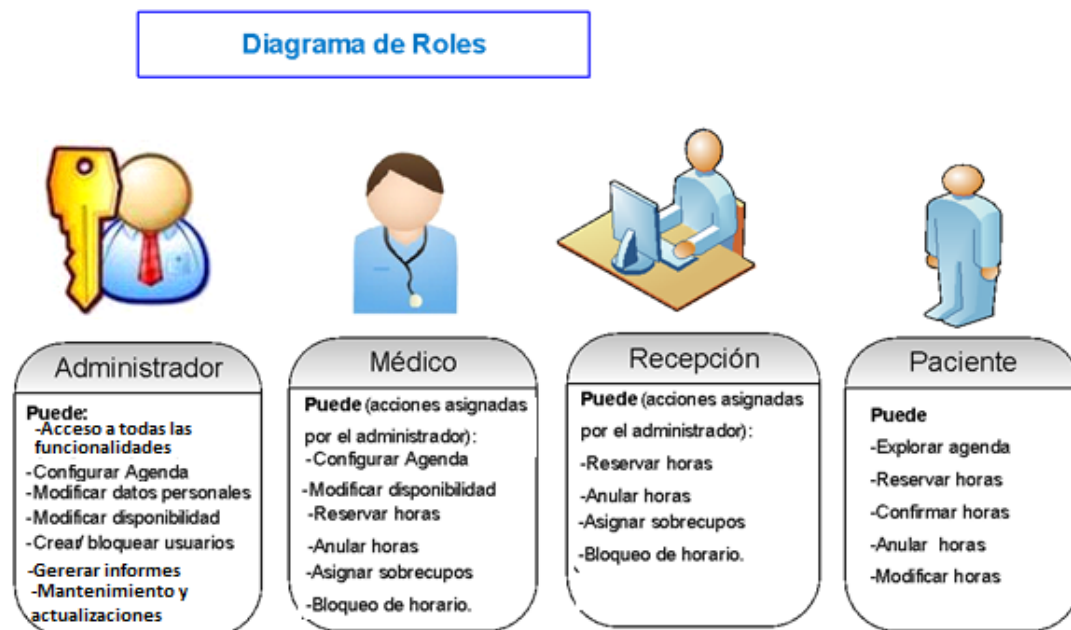


Figura 1 - Diagrama de roles



## 1.8 Técnicas de levantamiento de requisitos aplicadas.

En primer lugar, nos pusimos en contacto con el cliente para explicarle en qué consiste el proyecto y ofrecerle la oportunidad de participar en su desarrollo. Posteriormente, acordamos una entrevista presencial para llevar a cabo el levantamiento de requisitos, a la cual asistimos los miembros del equipo. Durante la reunión, el cliente nos explicó el proceso actual de agendamiento de citas en la empresa y cómo se maneja actualmente utilizando una hoja de cálculo de Excel.

El cliente expuso las funcionalidades que le gustaría que el sistema de agendamiento de citas pudiera realizar y los requisitos necesarios para su implementación. Utilizamos la nota de entrevista como principal instrumento para registrar las expectativas del cliente y el proceso de trabajo que se busca agilizar mediante la implementación y utilización del sistema.



*Ilustración 1. Reunión para el levantamiento de requisitos*

## **1.9 Análisis de requerimientos de software (requisitos)**

Descripción general de las funciones más importantes identificadas.

### **1.9.1. Descripción general**

Se tiene estimado que **SADRM** sea un aplicativo web construido con tecnologías de vanguardia, lo que garantice un rendimiento más que bueno en los navegadores actuales, además de responder de manera responsiva en los dispositivos móviles. Siendo importante también, la implementación de interfaces claras y amigables e intuitivas con el usuario final.

### **1.9.2 Perspectiva de la aplicación**

SADRM está planeado como un software independiente cuyo funcionamiento no está condicionado por ningún aplicativo. Al desarrollar la aplicación de agendamiento de citas en web, se logrará una solución flexible, accesible que facilitará el proceso de programación de citas tanto para el personal de la empresa como para los pacientes. Además, permitirá una gestión más eficiente. El funcionamiento e interacción del software será descrito posteriormente en la subsección de arquitectura.

### **1.9.3 Restricciones de la aplicación**

1. Conexión a Internet: El uso del software requerirá una conexión estable a Internet para acceder y utilizar todas sus funcionalidades.
2. Interfaces de usuario intuitivas: La aplicación debe contar con interfaces de usuario intuitivas y fáciles de usar. Esto garantiza que el personal administrativo pueda utilizarla sin dificultades y realizar las operaciones necesarias de manera eficiente.
3. Compatibilidad con navegadores comunes: La aplicación debe ser funcional en los navegadores más comunes. Esto asegura que los usuarios puedan acceder a la aplicación sin importar su elección de navegador.

4. **Arquitectura cliente-servidor:** La aplicación debe ser construida siguiendo el modelo cliente-servidor. Esto implica que existirá un servidor centralizado que gestionará la lógica de negocio y almacenará los datos, mientras que los clientes (los usuarios) se conectarán al servidor para acceder a la aplicación y realizar sus operaciones.
5. **Protocolo HTTP:** La comunicación entre el cliente y el servidor debe establecerse a través del protocolo HTTP (Hypertext Transfer Protocol). Esto es fundamental para que los clientes puedan enviar solicitudes al servidor y recibir respuestas con los datos correspondientes.
6. **Sistema de validación de sesión:** La aplicación debe contar con un sistema de validación de sesión para garantizar la seguridad y privacidad de los datos. Esto implica que los usuarios deberán autenticarse con credenciales válidas (como un nombre de usuario y una contraseña) antes de acceder a la funcionalidad de la aplicación.
7. **Escalabilidad:** La aplicación debe ser capaz de manejar un crecimiento en el número de usuarios y citas sin perder rendimiento.
8. **Mantenimiento y actualizaciones:** La aplicación debe contar con un plan de mantenimiento adecuado, que incluya actualizaciones regulares para corregir errores, mejorar la funcionalidad y agregar nuevas características.

#### **1.9.4 Necesidades básicas de la aplicación.**

1. **Contar con un Servidor de instalación:** La aplicación requerirá un servidor donde pueda ser instalada y alojada. Este servidor será responsable de gestionar la lógica de negocio y almacenar los datos del sistema.
2. **Conexión a Internet:** Para utilizar la aplicación, las computadoras y dispositivos de los usuarios deben tener una conexión a Internet estable. Esto permitirá acceder al servidor donde se encuentra instalada la aplicación y realizar las operaciones de agendamiento de citas.

3. Conocimiento básico de manejo de computadoras: Los usuarios del sistema deben tener un conocimiento básico de cómo utilizar computadoras y navegar por aplicaciones web. Esto incluye habilidades como el uso de teclado y ratón, navegación por Internet, y comprensión básica de interfaces de usuario.
4. Conocimiento de los tipos de información manejada: Para garantizar el correcto ingreso e interpretación de los datos en el sistema, los usuarios deben tener conocimiento de los tipos de información que se manejan en la empresa.

## 1.9.5 Requerimientos Específicos

### 1.9.5.1 Requerimientos Funcionales

Código del Requerimiento	RF01
Nombre	Registro de Pacientes
Propósito	El paciente debe poder registrarse en el sistema para poder programar una cita.
Descripción	El paciente se registra de manera automática rellenando el formulario de registro. El formulario de registro debe incluir información de contacto del paciente, como nombre completo, correo electrónico, número de teléfono, etc.
Entrada	Formulario con los siguientes datos del nuevo paciente: nombre completo, correo electrónico, número de teléfono, genero, ci, seguro médico. Archivo pdf o jpg de orden.
Salida	Un mensaje de confirmación de que el formulario ha sido enviado correctamente.
Prioridad	Alta

Código del Requerimiento	RF02
Nombre	Agendamiento de cita
Propósito	El paciente debe poder programar una cita en un horario disponible en la clínica o hospital correspondiente.
Descripción	Una vez registrado el paciente puede visualizar la disponibilidad de horarios para programar citas.
Entrada	Horarios y estudios seleccionados
Salida	Visualizar un calendario con la disponibilidad según estudio a realizar. Visualización de que el registro fue exitoso.
Prioridad	Alta.

<b>Código del Requerimiento</b>	<b>RF03</b>
<b>Nombre</b>	Acceso de cuenta recepción
<b>Propósito</b>	El personal de recepción debe poder acceder a su cuenta.
<b>Descripción</b>	El recepcionista, una vez registrado deberá ingresar al sistema a través de una cuenta única utilizando un formulario donde coloca su nombre de usuario y contraseña.
<b>Entrada</b>	Nombre de usuario y contraseña
<b>Salida</b>	Mensaje de inicio exitoso o fallido.
<b>Prioridad</b>	Alta

<b>Código del Requerimiento</b>	<b>RF04</b>
<b>Nombre</b>	Carga de pacientes
<b>Propósito</b>	El personal de recepción debe poder ingresar la información del paciente en el sistema, como datos personales, información de seguro médico, etc.
<b>Descripción</b>	El recepcionista debe registrar al paciente relleno el formulario de registro. El formulario de registro debe incluir información de contacto del paciente, como nombre completo, correo electrónico, número de teléfono, etc.
<b>Entrada</b>	Formulario con los siguientes datos del nuevo paciente: nombre completo, correo electrónico, número de teléfono, género, ci, seguro médico.
<b>Salida</b>	Un mensaje de confirmación de que el formulario ha sido enviado correctamente.
<b>Prioridad</b>	Alta

<b>Código del Requerimiento</b>	<b>RF05</b>
<b>Nombre</b>	Programación de Citas
<b>Propósito</b>	El personal de recepción debe poder programar una cita para el paciente en un horario disponible en la clínica o hospital correspondiente.
<b>Descripción</b>	El recepcionista, una vez registrado al paciente debe poder visualizar la disponibilidad de horarios para programar citas. Ser capaz de reprogramar una cita previamente programada.
<b>Entrada</b>	Horarios y estudios requeridos por el paciente
<b>Salida</b>	Visualizar un calendario con la disponibilidad según estudio a realizar. Mensaje de confirmación de que el registro fue exitoso.
<b>Prioridad</b>	Alta

Código del Requerimiento	RF06
Nombre	Configuración de horarios
Propósito	El encargado principal debe poder configurar la disponibilidad de horarios para programación de citas.
Descripción	El encargado principal podrá definir, crear y la disponibilidad de horarios para programación de citas. Cargando un formulario con sus respectivos campos como tipo de estudio, días y horarios.
Entrada	Formulario con datos del estudio y los días correspondientes.
Salida	Mensaje de cargado y creación correcta horario. Calendario de disponibilidad.
Prioridad	Alta.

Código del Requerimiento	RF07
Nombre	Generación de Informes
Propósito	El encargado principal debe poder generar informes y estadísticas sobre la programación de citas y otros datos relevantes para la toma de decisiones.
Descripción	El encargado principal debe poder generar informes y estadísticas a través de un documento de texto previamente definido.
Entrada	Formulario de datos estadísticas
Salida	Informe finalizado
Prioridad	Media

Código del Requerimiento	RF08
Nombre	seguros y estudios
Propósito	El sistema debe permitir hacer modificaciones de los datos de un seguro o estudio seleccionado
Descripción	El encargado principal debe poder ver y gestionar la lista de seguros y/o estudios previamente definida.
Entrada	Formulario de datos sobre seguros o estudios
Salida	Datos actualizados
Prioridad	Media

<b>Código del Requerimiento</b>	<b>RF9</b>
<b>Nombre</b>	Gestión de Usuarios
<b>Propósito</b>	El administrador del sistema debe poder crear y gestionar las cuentas de usuario del personal de la clínica o hospital, con restricciones de acceso de acuerdo a las funciones que realiza.
<b>Descripción</b>	El administrador del sistema podrá crear un nuevo usuario, dirigirse al hipervínculo que indica la creación de una nueva cuenta, diligenciar el formulario con sus respectivos campos obligatorios y proceder a finalizar el proceso pulsando en el botón indicado. El administrador debe poder asignar permisos de acceso para diferentes funciones del sistema según el cargo del usuario.
<b>Entrada</b>	Formulario con datos del nuevo usuario: datos personales y del cargo. Recepcionista o Encargado: nombre completo, correo electrónico, número de teléfono, género, ci, etc.
<b>Salida</b>	Mensaje de cargado y creación correcta del nuevo usuario Nombre del nuevo usuario y su contraseña estándar.
<b>Prioridad</b>	Alta.

### 1.9.5.2 Requerimientos NO Funcionales

<b>Código del Requerimiento</b>	<b>RNF01</b>
<b>Nombre</b>	Rendimiento
<b>Descripción</b>	El sistema debe ser capaz de procesar la información de forma rápida y eficiente para garantizar una experiencia de usuario satisfactoria.
<b>Prioridad</b>	Alta

<b>Código del Requerimiento</b>	<b>RNF02</b>
<b>Nombre</b>	Fiabilidad
<b>Descripción</b>	El sistema debe ser capaz de funcionar de manera confiable y sin errores durante largos períodos de tiempo para garantizar la satisfacción del usuario.
<b>Prioridad</b>	Media

<b>Código del Requerimiento</b>	<b>RNF03</b>
<b>Nombre</b>	Compatibilidad
<b>Descripción</b>	El sistema debe ser compatible con los navegadores web más populares y sistemas operativos para garantizar que los usuarios puedan acceder a la aplicación desde una variedad de dispositivos.
<b>Prioridad</b>	Media

<b>Código del Requerimiento</b>	<b>RNF04</b>
<b>Nombre</b>	Facilidad de mantenimiento
<b>Descripción</b>	El sistema debe ser fácil de mantener y actualizar para garantizar que el software pueda adaptarse a los cambios en los requisitos del negocio.
<b>Prioridad</b>	Alta

<b>Código del Requerimiento</b>	<b>RNF05</b>
<b>Nombre</b>	Usabilidad
<b>Descripción</b>	El sistema debe ser fácil de usar e intuitivo para el usuario, lo que significa que la navegación y la interacción con el software deben ser claras y sencillas para el usuario.
<b>Prioridad</b>	Alta

<b>Código del Requerimiento</b>	<b>RNF06</b>
<b>Nombre</b>	Seguridad
<b>Descripción</b>	El sistema debe ser seguro y proteger la información del usuario y la privacidad, lo que significa que debe cumplir con los estándares de seguridad y privacidad de datos y utilizar técnicas de cifrado para proteger la información confidencial del usuario.
<b>Prioridad</b>	Alta

<b>Código del Requerimiento</b>	<b>RNF07</b>
<b>Nombre</b>	Escalabilidad
<b>Descripción</b>	El sistema debe ser capaz de manejar un aumento en la cantidad de usuarios y datos sin afectar el rendimiento del sistema, lo que significa que debe ser capaz de escalar de manera eficiente y efectiva.
<b>Prioridad</b>	Media

<b>Código del Requerimiento</b>	<b>RNF08</b>
<b>Nombre</b>	Disponibilidad
<b>Descripción</b>	El sistema debe estar disponible y en funcionamiento la mayor parte del tiempo posible, lo que significa que debe minimizar el tiempo de inactividad y tener planes de contingencia en caso de fallas del sistema.
<b>Prioridad</b>	Alta



## Evidencias del trabajo de levantamiento de requisitos realizados en la empresa.

### Hoja de cálculo utilizada para el registro de pacientes

**HORARIO DE TURNOS IMAP**

Fecha Inicial: 08/05/2023 Fecha final: 08/05/2023 ...

HORA	5:00:00	PACIENTE		Telefono:		edad:	
ESTUDIO							
IMPORTE							
SEGURO							
HORA	6:30:00	PACIENTE		Telefono:		edad:	
ESTUDIO							
IMPORTE							
SEGURO							
HORA	7:00:00	PACIENTE		Telefono:		edad:	
ESTUDIO							
IMPORTE							
SEGURO							
HORA	7:30:00	PACIENTE		Telefono:		edad:	
ESTUDIO							

Registro: 14 1 1 de 8330

Vista Formulario

Activar Windows  
Vaya a Sistema en el Panel de control para activar Windows.

NUM

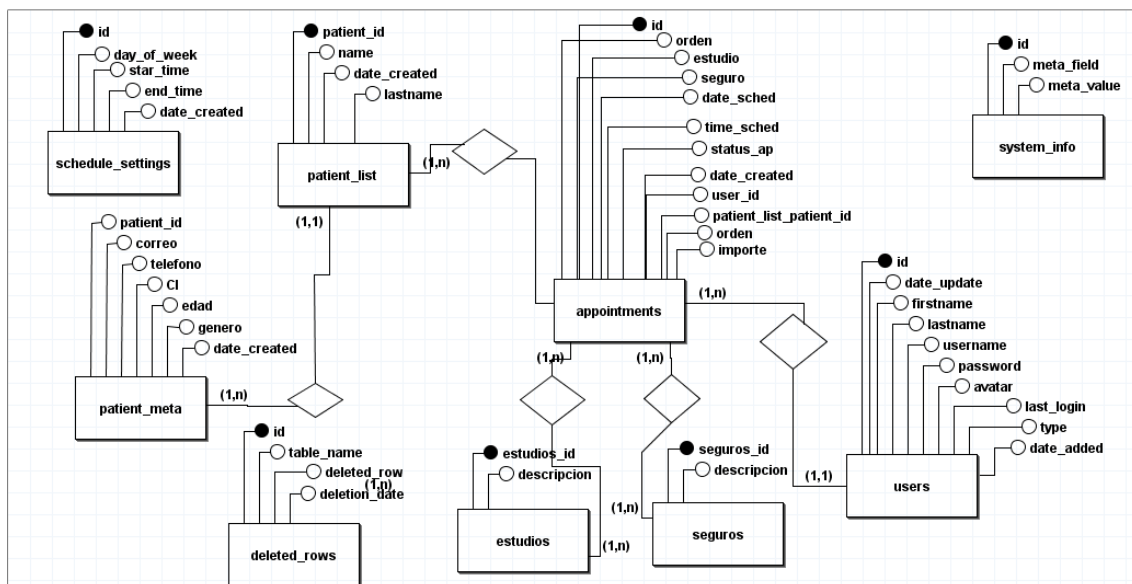
## Capítulo 2 – Diseño y Estimación de Costos

### 2.1 Introducción

En base a los requisitos adquiridos en la primera etapa, se procede a la etapa de diseño del aplicativo. El diseño de los distintos elementos del software base de datos, from-end, back-end y arquitectura y desde diferentes perspectivas proceso de análisis en diseños datos, estructurales, interfaz y procedural.

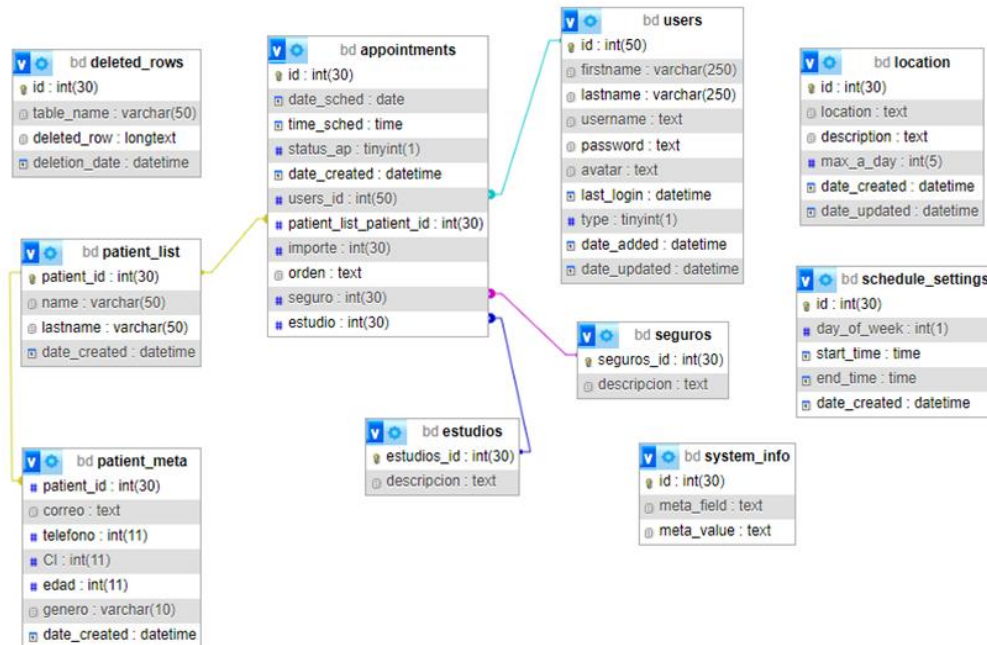
### 2.2 Diseño de Datos

#### 2.2.1 Modelo Entidad-Relación



## 2.2.2 Modelo Relacional

En base al modelo entidad-relación realizado en el apartado anterior, construir el modelo relacional de datos del sistema a ser desarrollado.



## 2.2.3 Diccionario de Datos

En este apartado cada una de las tablas del modelo relacional deben de ser descritas

**Nombre de la Tabla:** schedule\_settings **Fecha de Creación:** 10/03/23

**Descripción:** esta tabla conservara horario de agendamientos disponibles.

Clave	Campo	Tamaño	Tipo de Datos	Descripción
PK	id	30	SERIAL	Código único de la configuración
	day_of_week	1	INT	Día de la semana (0-6)
	start_time	-	TIME	Hora de inicio del horario
	end_time	-	TIME	Hora de finalización del horario
	date_created	10	DATETIME	Fecha de creación de la configuración

**Nombre de la Tabla:** patient\_list **Fecha de Creación:** 10/03/23

**Descripción:** esta tabla almacena para identificar la fecha en la que se agendo el paciente

Clave	Campo	Tamaño	Tipo de Datos	Descripción
PK	patient_id	30	SERIAL	Código único del Paciente
	name	50	VARCHAR	Nombre del paciente
	lastname	50	VARCHAR	Apellido del paciente
	date_created	10	DATETIME	Fecha de creación del paciente

**Nombre de la Tabla:** patient\_meta **Fecha de Creación:** 10/03/23

**Descripción:** esta tabla almacena las informaciones del paciente que se agendo

Clave	Campo	Tamaño	Tipo de Datos	Descripción
FK	patient_id	30	INT	ID del paciente asociado
	correo	-	TEXT	Correo electrónico del paciente
	teléfono	-	INT	Número de teléfono del paciente
	CI	-	INT	Cédula de identidad del paciente
	edad	-	INT	Edad del paciente
	genero	10	VARCHAR	Género del paciente
	date_created	10	DATETIME	Fecha de creación de los metadatos

**Nombre de la Tabla:** appointments **Fecha de Creación:** 10/03/23

**Descripción:** esta tabla almacena las informaciones referente a la cita del paciente

Clave	Campo	Tamaño	Tipo de Datos	Descripción
PK	id	30	SERIAL	Código único de la cita
	date_sched	-	DATETIME	Fecha programada de la cita
	status_ap	-	TINYINT	Estado de la cita: 1 activa, 0 inactiva
	date_created	-	DATETIME	Fecha de creación de la cita
FK	users_id	50	INT	ID del usuario relacionado con la cita
FK	patient_list_id	30	INT	ID del paciente relacionado con la cita
	importe	30	INT	Importe de la cita
	orden	-	TEXT	Orden médica asociada a la cita
FK	seguro	30	INT	ID del seguro asociado a la cita
FK	estudio	30	INT	ID del estudio asociado a la cita

**Nombre de la tabla:** users **Fecha de Creación:** 10/03/2023

Descripción: Esta tabla guarda todos los datos importantes sobre los usuarios

Clave	Campo	Tamaño	Tipo de Datos	Descripción
PK	id	50	SERIAL	Código único del Usuario
	firstname	250	VARCHAR	Nombre del usuario
	lastname	250	VARCHAR	Apellido del usuario
	username	10	TEXT	Usuario con el que se logeará en el sistema
	password	40	TEXT	Contraseña de inicio de sesión del usuario
	avatar	45	TEXT	Imagen del usuario
	last_login	10	DATETIME	Fecha de último acceso
	type	10	TINYINT	Estado del usuario: 1 activo, 0 inactivo
	date_added	10	DATETIME	Fecha en que se registró el usuario
	date_updated	10	DATETIME	Fecha en que se actualizaron los datos del usuario

**Nombre de la Tabla:** system\_info **Fecha de Creación:** 10/03/23

**Descripción:** esta tabla conservara informaciones del sistema.

Clave	Campo	Tamaño	Tipo de Datos	Descripción
PK	id	30	SERIAL	Código único del sistema
	meta_field	45	TEXT	Atributos para configuración del sistema
	meta_value	45	TEXT	Información de cada atributo del sistema

**Nombre de la tabla:** deleted\_rows **Fecha de Creación:** 10/03/2023

**Descripción:** Esta tabla guarda todos los datos eliminados

Clave	Campo	Tamaño	Tipo de Datos	Descripción
PK	id	30	SERIAL	Código único de la fila eliminada
	table_name	50	VARCHAR	Nombre de la tabla
	deleted_row	-	JSON	Fila eliminada en formato JSON
	deletion_date	10	DATETIME	Fecha de eliminación

**Nombre de la tabla:** seguros **Fecha de Creación:** 10/03/2023

**Descripción:** Esta tabla guarda todos los datos de seguros

Clave	Campo	Tamaño	Tipo de Datos	Descripción
PK	seguros_id	30	SERIAL	Código único del Seguro
	descripción	-	TEXT	Descripción del seguro

**Nombre de la tabla:** estudios **Fecha de Creación:** 10/03/2023

**Descripción:** Esta tabla guarda todos los datos de estudios

Clave	Campo	Tamaño	Tipo de Datos	Descripción
PK	estudios_id	30	SERIAL	Código único del Estudio
	descripción	-	TEXT	Descripción del estudio

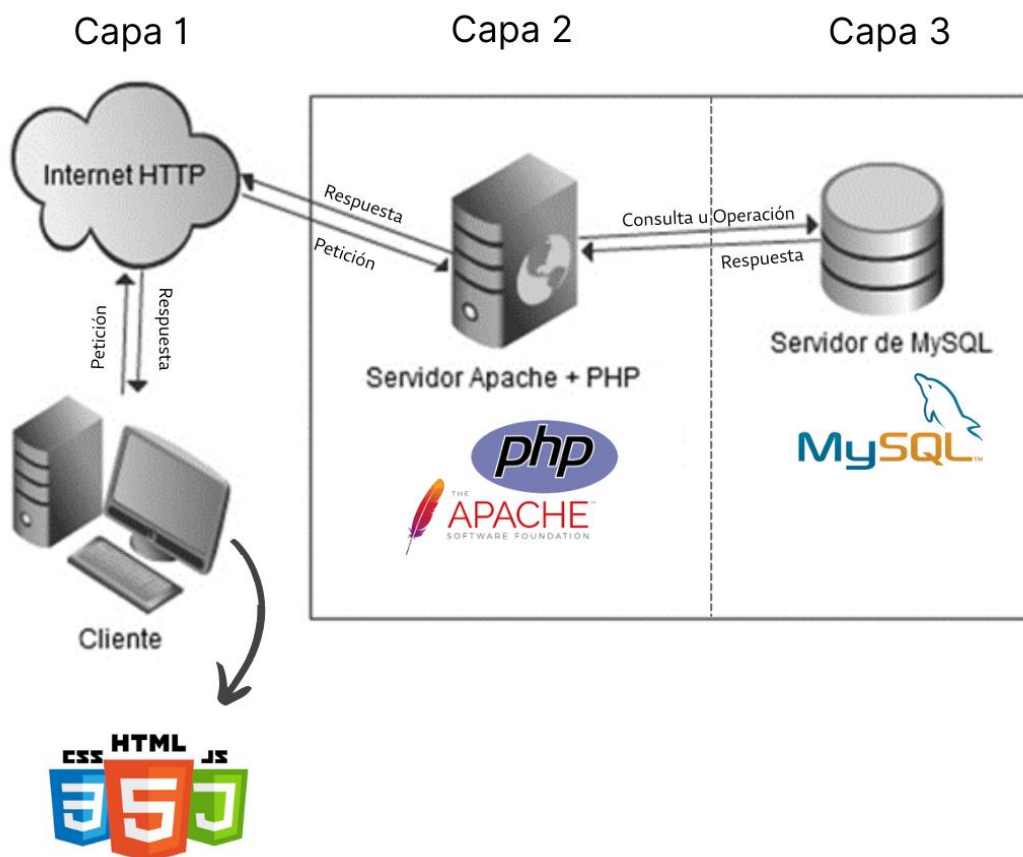
**Nombre de la tabla:** location **Fecha de Creación:** 10/03/2023

**Descripción:** Esta tabla guarda todos los datos de ubicación para futuras sucursales

Clave	Campo	Tamaño	Tipo de Datos	Descripción
PK	id	30	SERIAL	Código único de la Ubicación
	location	-	TEXT	Ubicación física
	description	-	TEXT	Descripción de la ubicación
	max_a_day	5	INT	Máximo de citas al día
	date_created	10	DATETIME	Fecha de creación de la ubicación
	date_updated	10	DATETIME	Fecha de actualización de la ubicación

## 2.3 Diseño arquitectónico

Se basa en la arquitectura Cliente-Servidor por un lado está el cliente y por otro lado el servidor que a su vez se divide en la lógica de los datos y los datos con un servidor de base de datos específico. Cabe destacar que la comunicación entre capas se da de manera bidireccional.



### **1) Capa de presentación – Cliente**

Se trata del equipo que solicita los recursos.

Genera requerimientos de bases de datos.

Recibe resultados del servidor.

Posee una interfaz de usuario para la presentación.

Para la interfaz se utilizará HTML5, CSS y JavaScript.

### **2) Capa de proceso – Servidor web + lenguaje de aplicaciones**

Se encarga de proporcionar (recibir y administrar) los recursos solicitados por el cliente que a su vez requiere de un servidor de Base de Datos específico para hacerlo.

Procesa los requerimientos de la base de datos.

Para esta capa se utilizará como servidor web Apache y el lenguaje PHP para el desarrollo web.

La comunicación entre ambas capas se realiza a través de internet utilizando el protocolo HTTP.

### **3) Capa de datos – Servidor de Base de Datos**

Proporciona al servidor de aplicaciones los datos necesarios para poder procesar y generar el recurso que solicitó el cliente en un principio.

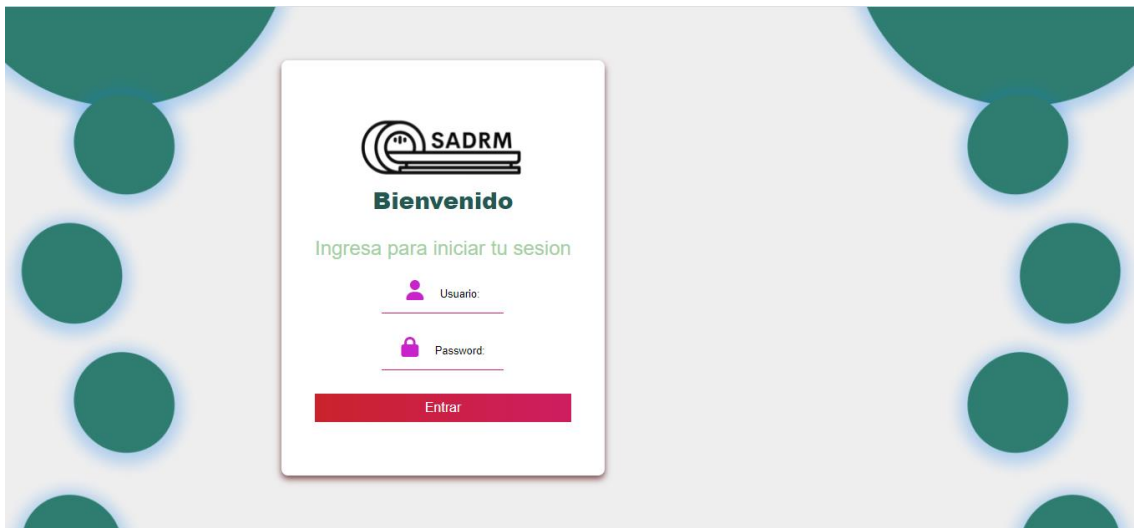
Se encarga de:

Gestionar y almacenar información.

Recuperar datos.

El sistema de gestión de bases de datos utilizado es MySQL.

## 2.4 Diseño de interfaz



Permite al usuario ingresar al sistema de acuerdo con el rol asignado, para ello debe introducir su nombre de usuario y contraseña

**Iniciar:** Luego de que el usuario ingrese sus datos en los campos se verifica en la Base Datos si posee acceso al sistema.

Agendar Cita

Es la vista que tendrá el paciente al entrar al sistema con un botón para iniciar el proceso de agendamiento.



Calendar

Home / Calendar

Estados

Confirmados

Prendientes

Cancelados

August 2023

Semana

Mes

Día

<

>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26

Es la vista que tendrá el Paciente al entrar al accionar el botón para agendar una cita, permite al mismo dar click en la fecha que desea solicitar el estudio.

\*Título del modal

Nombre del paciente:

Ingrese su nombre

Fecha de la cita:

01/08/2023

Apellido del paciente:

Ingrese su apellido

Hora de la cita:

Horario: 08:00:00 - 10:00:00

Número de cedula:

Ingrese su nro de documento

Orden de Estudio:

Seleccionar archivo

Ninguno archivo selec.

Correo electrónico:

Ingrese su correo electrónico

Ingrese su seguro:

Ingrese Particular o Busque un seguro...

Teléfono de contacto:

Ingrese su número de teléfono

Estudio a realizar:

Busque y seleccione el estudio...

Género

Masculino

Importe:

Ingrese importe

Fecha de Nacimiento

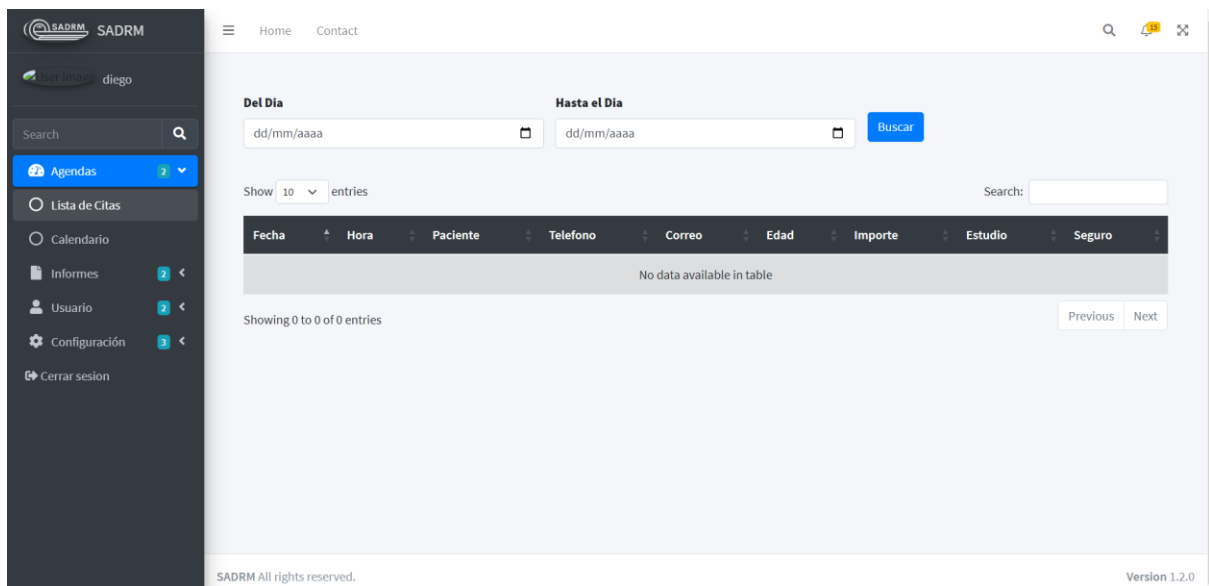
dd/mm/aaaa

Estado Cita

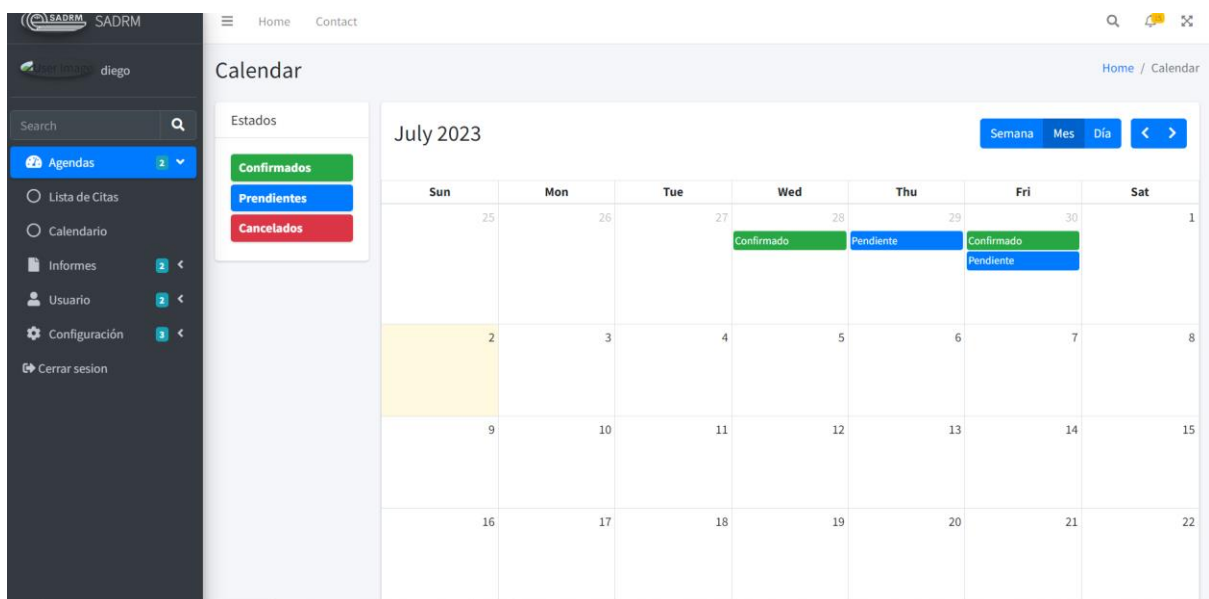
Pendiente

Programar cita

Cancelar



**Pantalla lista de citas:** su objetivo es permitir observar el listado de pacientes agendados en una fecha o en un intervalo de fechas, y ordenarlos por algún campo seleccionado



**Pantalla de calendario:** Desde el punto de vista del administrador puede ver los agendamientos por fecha

## Informe de Estudios

Seleccione el rango de tiempo

Desde: dd/mm/aaaa Hasta: dd/mm/aaaa

Filtrar

PDF

Resultados :

Show 10 entries

Search:

Fecha	Hora	Importe	Estudio	Seguro	Usuario
No se encontraron resultados					

Showing 1 to 1 of 1 entries

Previous 1 Next

SADRM All rights reserved.

Version 1.2.0

**Informe de estudios:** Genera un reporte de los estudios que fueron realizados en un intervalo de fechas, también tiene la posibilidad de generar un pdf con todos los datos del mismo

## Informe de Usuarios

Seleccione el rango de tiempo

Desde: dd/mm/aaaa Hasta: dd/mm/aaaa

Filtrar

PDF

Resultados

Show 10 entries

Search:

Usuario	Fecha	Hora	Nombre	Apellido	Importe	Estudio	Seguro
No se encontraron resultados							

Total: 0


Showing 1 to 1 of 1 entries

Previous 1 Next

**Informe de usuarios:** Genera un reporte para los usuarios, también tiene la posibilidad de hacer un cálculo del total de importes, que fueron aprobados

### Perfil Usuario

Mi foto de perfil:



**Nombre:**

**Apellido:**

**Username:**



[Modificar](#)

**Perfil de usuario:** su objetivo es poder modificar los datos de usuario logueado

### Lista de usuarios

[+ Crear Usuario](#)

Show  entries Search:

Avatar	Nombre	Apellido	Usuario	Rol	Acciones
	diego	velastiqui	svs	Administrador	<a href="#">Editar</a>   <a href="#">Eliminar</a>
	Diego	1	abc	Administrador	<a href="#">Editar</a>   <a href="#">Eliminar</a>

Showing 1 to 2 of 2 entries

[Previous](#) [1](#) [Next](#)

**Listado de usuarios:** Su objetivos es listar los usuarios que tienen el acceso al sistema, como también poder editar y eliminar a los usuarios.

### Configuracion de Horarios

Editar Horarios

Días	Hora de inicio	Hora de finalizacion
Sábado	07:00:00	15:00:00
Miércoles, Jueves, Viernes	07:00:00	17:00:00
Lunes, Martes	08:00:00	10:00:00
Domingo	08:00:00	12:00:00

SADRM All rights reserved. Version 1.2.0

**Configuración de horarios:** su objetivo es permitir modificar los horarios ya predefinidos por la empresa

### LISTA DE SEGUROS

+ Agregar Seguro

Show 10 entries Search:

ID	Seguros	Acciones
1	MEP	Editar   Eliminar
2	TESAI	Editar   Eliminar
3	PARTICULAR IMAP	Editar   Eliminar
4	PRECIOS ESPECIALES	Editar   Eliminar
5	PRECIO HOSPITALARIO	Editar   Eliminar
6	FUNDACIÓN STA. LIBRADA	Editar   Eliminar
7	MEP	Editar   Eliminar
8	MIGONE IMAP	Editar   Eliminar

**Lista de seguros:** su objetivo es agregar, modificar y eliminar los seguros.

## Configuracion de Estudios

+ Agregar Estudio

Show 10 entries

Search:

ID	Estudios	Acciones
1	resonancia	<a href="#">Editar</a>   <a href="#">Eliminar</a>
2	diagnostico	<a href="#">Editar</a>   <a href="#">Eliminar</a>
3	prueba	<a href="#">Editar</a>   <a href="#">Eliminar</a>
4	Nuevo	<a href="#">Editar</a>   <a href="#">Eliminar</a>

Showing 1 to 4 of 4 entries

Previous

1

Next

SADRM All rights reserved.

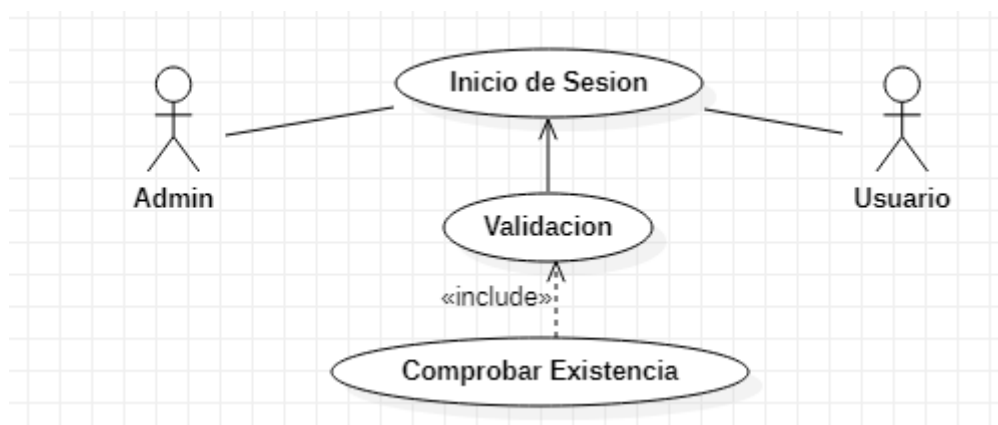
Version 1.2.0

**Configuración de estudios:** su objetivo es agregar, modificar y eliminar los estudios

## 2.4 Diseño procedural

### 2.4.1 Diagrama de Caso de Uso

#### Inicio de Sesión



<b>CU-001</b>	<b>Inicio de sesión</b>
<b>Fecha</b>	Marzo-2023
<b>Descripción</b>	Ingresar al sistema con un usuario existente con determinados permisos
<b>Actores</b>	Admin-Usuario
<b>Precondiciones</b>	El usuario debe conocer sus credenciales de acceso.

	Pas o	Acción
<b>Secuencia Normal</b>	1	El usuario accede a la página de la aplicación web.
	2	El usuario coloca sus credenciales en los campos correspondientes y presiona iniciar sesión.
	3	El sistema valida los datos que fueron ingresados.
	4	El sistema abre la página que le corresponde según el usuario que haya ingresado.
<b>Excepciones</b>	1.1	El usuario no accede a la url correcta.
	1.2	El usuario coloca mal su nombre de usuario y/o contraseña.
	3.1	El sistema no encuentra coincidencias con los datos proveídos.

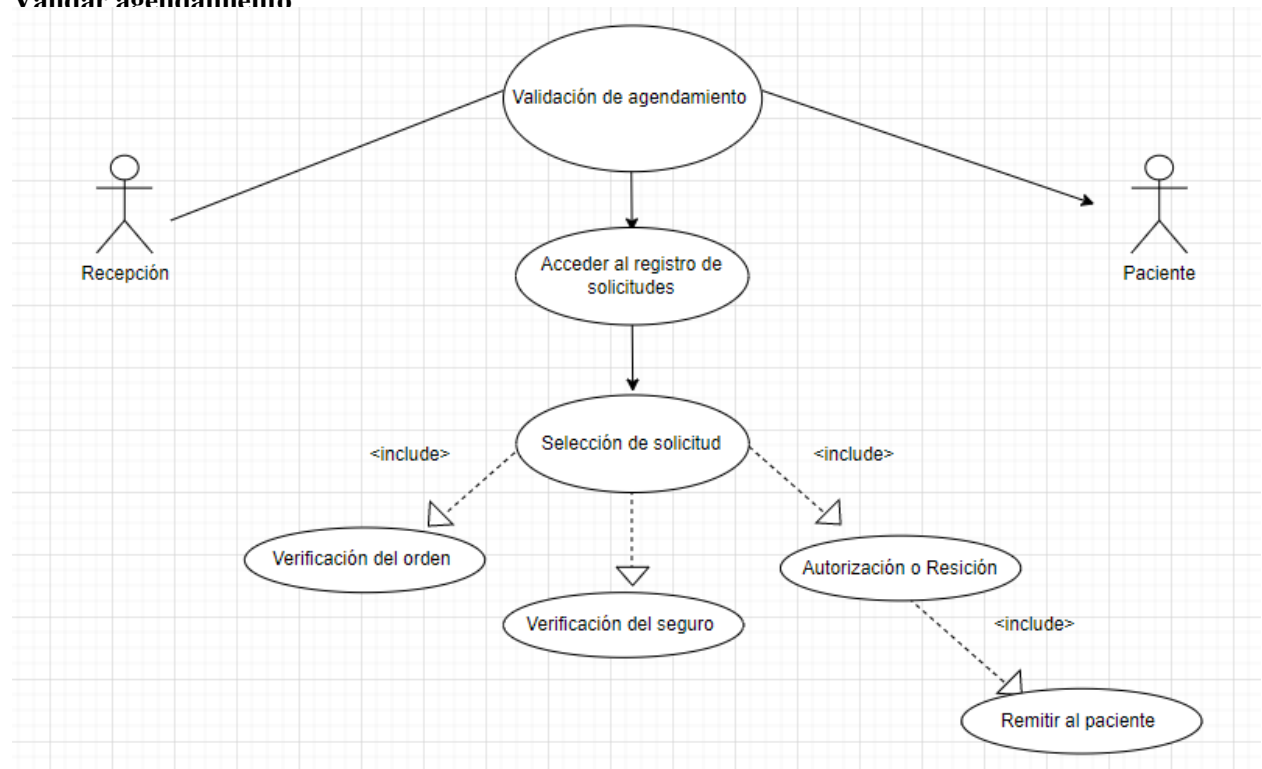
### Proceso de agendamiento



<b>CU-002</b>	<b>Proceso de agendamiento</b>	
<b>Fecha</b>	Marzo-2023	
<b>Descripción</b>	Proceso de agendamiento del paciente	
<b>Actores</b>	Paciente – recepción	
<b>Precondiciones</b>	El Paciente debe tener acceso a la Red de Internet, debe poseer un dispositivo por el cual conectarse a la aplicación WEB, debe poseer un escaneado de orden médica.	
	Paso	Acción
<b>Secuencia Normal</b>	1	El paciente inicia la aplicación, solicita hacer un agendamiento.
	2	El paciente accede al calendario de fechas disponibles para el estudio, elige una en la cual desea agendarse.
	3	Inicio de proceso de agendamiento.
	4	El paciente carga sus datos personales en el sistema.
	5	El paciente indica si posee un seguro, carga los datos en caso de afirmación en el sistema.
	6	El paciente adjunta el escaneado de su orden médica en el sistema.
	7	El paciente elige un horario disponible para su cita.
	8	El paciente envía su solicitud de agendamiento para que la recepción valide.
<b>Excepciones</b>	2.1	El paciente elige una fecha incorrecta.
	4.1	El paciente no llena todos los campos obligatorios.
	5.1	El paciente si posee seguro, pero no se especificó.
	6.1	No adjunto su orden médica correctamente.
	7.1	El paciente coloco un horario invalido.



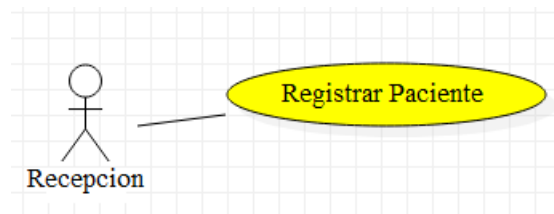
## Validar agendamiento



<b>CU-003</b>	<b>Validación de agendamiento</b>	
<b>Fecha</b>	Marzo-2023	
<b>Descripción</b>	Proceso de validación del paciente	
<b>Actores</b>	Recepción	
<b>Precondiciones</b>	El Recepcionista tiene que iniciar sesión con una cuenta predefinida, debe tener acceso a la Red de Internet, deben existir solicitudes pendientes.	
	<b>Paso</b>	<b>Acción</b>
<b>Secuencia Normal</b>	1	El recepcionista inicia la aplicación, ingresa sus respectivas credenciales.
	2	El recepcionista accede al listado de solicitudes de estudio, elige una en la cual desea verificar.
	3	El recepcionista intenta ver el escaneado de la orden médica del paciente.

	4	El Recepcionista autoriza la solicitud del paciente.
<b>Excepciones</b>	3.1	No adjunto su orden médica correctamente.

### Agendamiento de Paciente hecho por Recepción



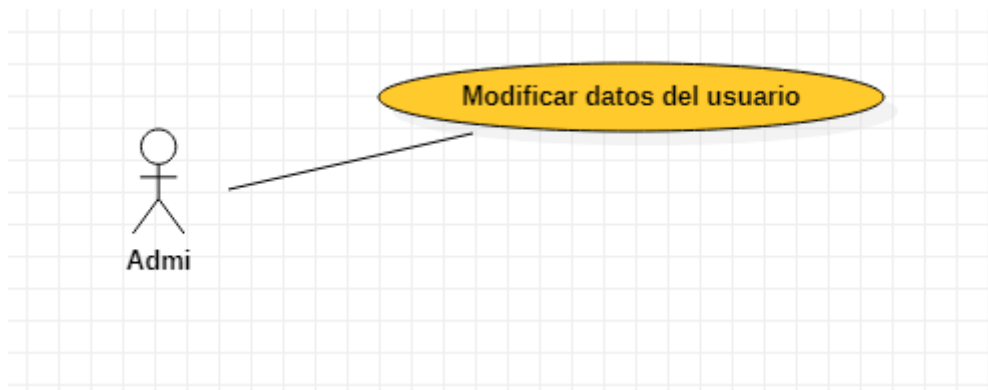
<b>CU-004</b>	<b>Agendamiento de Paciente hecho por Recepción</b>	
<b>Fecha</b>	Marzo-2023	
<b>Descripción</b>	Permite agendar a un nuevo paciente al sistema.	
<b>Actores</b>	Recepción	
<b>Precondiciones</b>	El usuario tiene que estar identificado como Recepcionista en el sistema	
	Paso	Acción
<b>Secuencia Normal</b>	1	El recepcionista accede al calendario de fechas disponibles para el estudio, elige una en la cual desea agendar.
	2	El sistema mostrará un formulario en el cual se solicitan los datos del paciente.
	3	El Recepcionista completa los datos solicitados y da click en el botón Programar cita.
<b>Excepciones</b>	3	- Si se deja en blanco algún campo obligatorio tampoco permitirá guardar.

### Registro usuarios



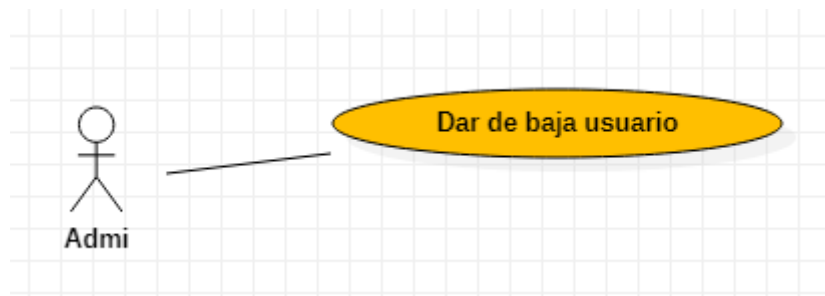
<b>CU-005</b>	<b>Registrar usuario</b>	
<b>Fecha</b>	Marzo-2023	
<b>Descripción</b>	Registrar un nuevo usuario en el sistema.	
<b>Actores</b>	Administrador	
<b>Precondiciones</b>	Se debe iniciar sesión como administrador.	
	Paso	Acción
<b>Secuencia Normal</b>	1	En el menú principal el administrador selecciona el ítem de lista de usuario, da click en el botón crear usuario.
	2	El sistema mostrará un formulario en el cual solicita: nombre, usuario, contraseña y tipo de usuario.
	3	El administrador completa los datos solicitados y da click en el botón Crear Usuario.
	4	El nuevo usuario se muestra en la lista de usuarios.
<b>Excepciones</b>	3	<ul style="list-style-type: none"> <li>- El administrar intenta cargar un username repetido.</li> <li>- Si se deja en blanco algún campo obligatorio tampoco permitirá guardar.</li> </ul>

## Modificar usuarios



<b>CU-006</b>	<b>Modificar usuarios</b>	
<b>Fecha</b>	Marzo-2023	
<b>Descripción</b>	Permite realizar modificaciones de los datos del usuario.	
<b>Actores</b>	Admin	
<b>Precondiciones</b>	El usuario tiene que estar identificado como administrador del sistema.	
	Paso	Acción
<b>Secuencia Normal</b>	1	En el menú principal el admin selecciona el ítem de lista de usuario, da click en el botón editar.
	2	El sistema mostrará una ventana con los datos del usuario seleccionado, como campos editables.
	3	El administrador edita los datos del usuario que requiera y da click en el botón actualizar.
	4	El sistema almacena los datos.
<b>Excepciones</b>	3	<ul style="list-style-type: none"> <li>- El administrar intenta cargar un username repetido.</li> <li>- Si se deja en blanco algún campo obligatorio tampoco permitirá actualizar.</li> </ul>

## Dar de baja usuarios

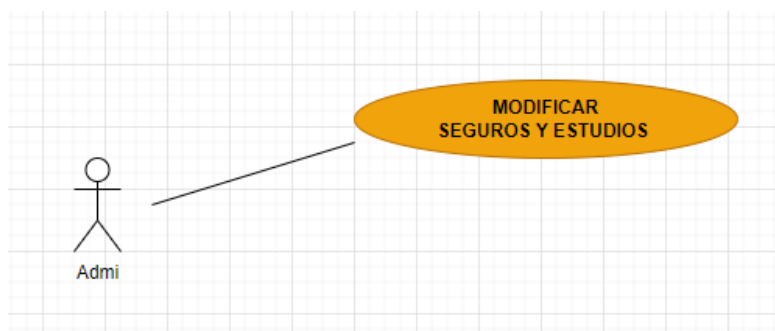


<b>CU-007</b>	<b>Dar de baja usuario</b>	
<b>Fecha</b>	Marzo-2023	
<b>Descripción</b>	Permite al administrador dar de baja un usuario	
<b>Actores</b>	Admin	
<b>Precondiciones</b>	El usuario tiene que estar identificado como administrador del sistema.	
	Paso	Acción
<b>Secuencia Normal</b>	1	En el menú principal el administrador selecciona el ítem de lista de usuario, da click en el botón eliminar usuarios.
	2	El sistema mostrará una ventana emergente para confirmar la acción.
	3	El administrador confirma la acción.
<b>Excepciones</b>	3	Si el administrador no confirma la acción regresa al paso 1.



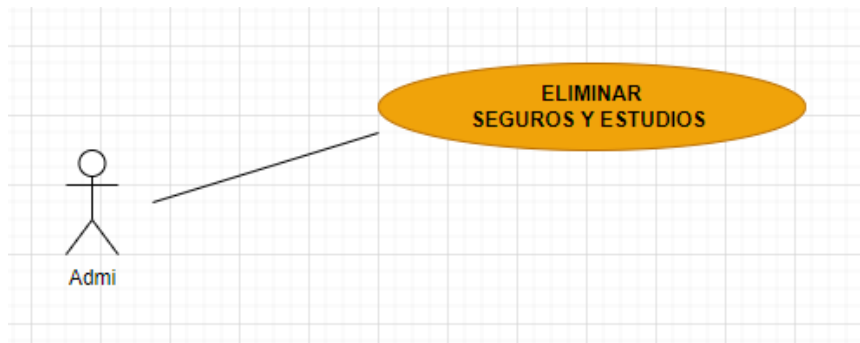
<b>CU-008</b>	<b>Agregar estudios y seguros</b>	
<b>Fecha</b>	Marzo-2023	
<b>Descripción</b>	Registrar un nuevo seguros y estudios.	
<b>Actores</b>	Administrador	
<b>Precondiciones</b>	Se debe iniciar sesión como administrador.	
	Paso	Acción
<b>Secuencia Normal</b>	1	En el menú principal el administrador selecciona el ítem de configuración, da click en el botón conf de seguros o conf de estudios y luego da click en uno de ellos y agregar.
	2	El sistema mostrará un formulario en el cual solicita: descripción de seguros o estudios.
	3	El administrador completa los datos solicitados y da click en el botón agregar seguro o estudio.
	4	El sistema redirecciona a la lista de estudios o seguros.
<b>Excepciones</b>	3	- Si se deja en blanco algún campo obligatorio tampoco permitirá guardar.

## Modificar seguros y estudios



<b>CU-009</b>	<b>Modificar seguros y estudios</b>	
<b>Fecha</b>	Marzo-2023	
<b>Descripción</b>	Permite realizar modificaciones de los datos de seguros y estudios.	
<b>Actores</b>	Admin	
<b>Precondiciones</b>	El usuario tiene que estar identificado como administrador del sistema.	
	<b>Paso</b>	<b>Acción</b>
<b>Secuencia Normal</b>	1	En el menú principal el admin selecciona el ítem de lista de seguros o estudios, da click en el botón editar.
	2	El sistema mostrará una ventana con los datos del seguros o estudios seleccionado, como campos editables.
	3	El administrador edita los datos del seguro o estudio que requiera y da click en el botón actualizar.
	4	El sistema almacena los datos.
<b>Excepciones</b>	3	- Si se deja en blanco algún campo obligatorio tampoco permitirá actualizar.

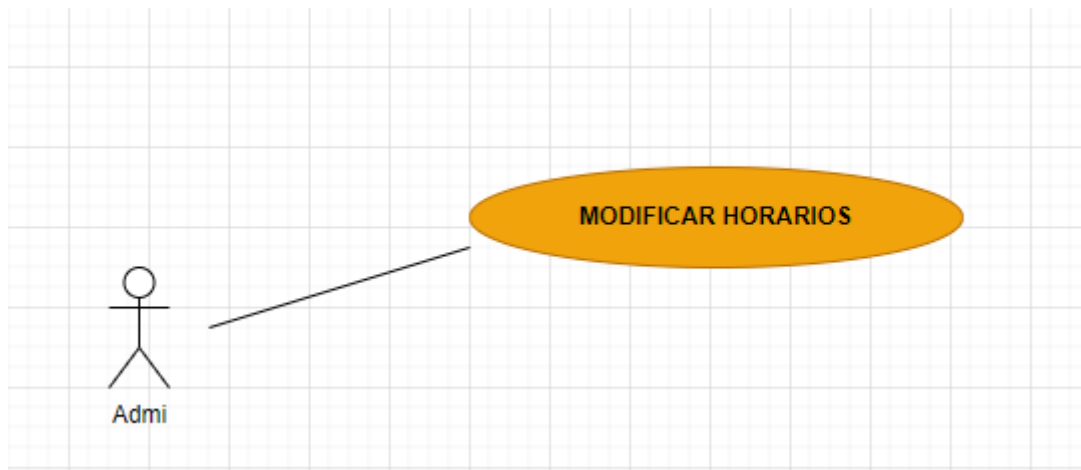
## Eliminar seguros y estudios



<b>CU-010</b>	<b>Eliminar seguros y estudios</b>	
<b>Fecha</b>	Marzo-2023	
<b>Descripción</b>	Permite al administrador eliminar seguros y estudios.	
<b>Actores</b>	Admin	
<b>Precondiciones</b>	El usuario tiene que estar identificado como administrador del sistema.	
	Paso	Acción
<b>Secuencia Normal</b>	1	En el menú principal el admin selecciona el ítem de lista de seguros o estudios, da click en el botón eliminar.
	2	El administrador accederá a una vista de eliminación.
	3	El administrador confirma la acción.
<b>Excepciones</b>	3	Si el administrador no confirma la acción

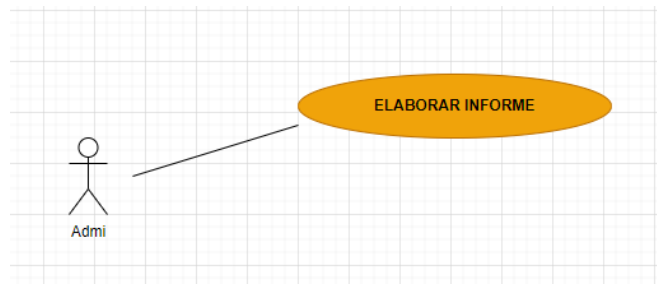


## Modificar horarios



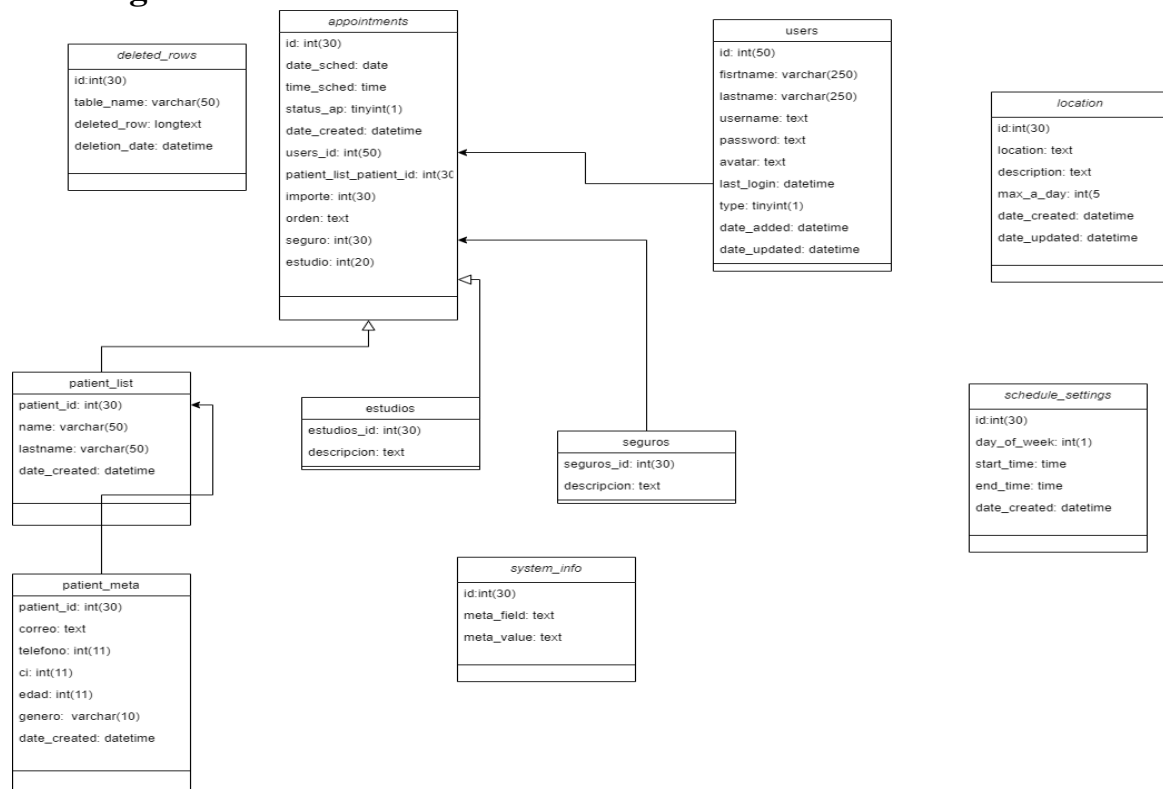
<b>CU-011</b>	<b>Modificar horarios</b>	
<b>Fecha</b>	Marzo-2023	
<b>Descripción</b>	Permite realizar modificaciones de los datos de horarios.	
<b>Actores</b>	Admin	
<b>Precondiciones</b>	El usuario tiene que estar identificado como administrador del sistema.	
	Paso	Acción
<b>Secuencia Normal</b>	1	En el menú principal el admin selecciona el ítem de conf de horarios, da click en el botón editar horarios.
	2	El sistema mostrará una ventana donde se puede modificar los datos de horario, como campo editable.
	3	El administrador configura el horario que desea y selecciona a que día aplicar.
	4	El administrador hace click en el botón guardar.
<b>Excepciones</b>	3	- Si se deja en blanco algún campo da mensaje de error.

## Elaborar informes

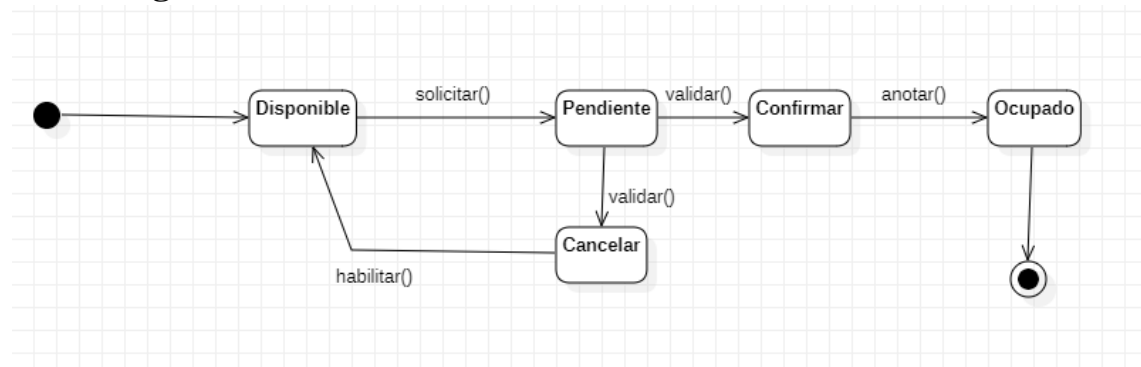


<b>CU-012</b>	<b>Elaborar informes</b>	
<b>Fecha</b>	Marzo-2023	
<b>Descripción</b>	Permite elaborar informes de ingresos o gastos registrados.	
<b>Actores</b>	Admin	
<b>Precondiciones</b>	El usuario tiene que estar identificado como administrador del sistema.	
	Paso	Acción
<b>Secuencia Normal</b>	1	El administrador ingresa a la sección de informes.
	2	El administrador selecciona el tipo de informe que desea y el rango tiempo a tomar en cuenta.
	3	El sistema verifica que los campos obligatorios estén completos.
	4	El sistema genera la información correspondiente presentada en formato PDF.
<b>Excepciones</b>	3	Si un campo obligatorio está vacío se muestra un mensaje de aviso al usuario solicitando que verifique los datos y regrese al paso 1 de la Secuencia normal.

## 2.4.2 Diagramas de Clases



## 2.4.3 Diagrama de Estado



## 2.5 Estimación de Costos a través de Puntos de Función y COCOMO II.

### 2.5.1 Por Punto de Función de IFPUG (International Function Points Users Groups)

Componentes funcionales básicos:

#### - Interacción Función de transacción

- Entrada externa (EI → External input).
- Salida externa (EO → External output).
- Consulta externa (EQ → External query).

### - Almacenamiento función de datos

Archivo lógico interno (**ILF** > **Internal Logical File**).

Archivo de interfaz externa (**EIF** > **External Interface File**)

### Nivel de complejidad

Tipo/nivel	Simple	Promedio	Complejo
EI	3 PF	4 PF	6 PF
EO	4 PF	5 PF	7 PF
EQ	3 PF	4 PF	6 PF
ILF	7 PF	10 PF	15 PF
EIF	5 PF	7 PF	10 PF

### Funciones identificadas del sistema

Registro de Pacientes	EI <b>3 PF</b>
Agendamiento de cita	EI <b>4 PF</b>
Visualización de info estudio	ILF <b>7 PF</b>
Acceso de cuenta recepción	EI <b>4 PF</b>
Carga de Pacientes	EI <b>3 PF</b>
Programación de citas	EI <b>4 PF</b>
Visualización de información	EO <b>4 PF</b>
Configuración de horarios	ILF <b>3 PF</b>
Gestión de usuarios	EI <b>4 PF</b>
Filtro de fechas	EQ <b>4 PF</b>
Búsqueda de paciente	EQ <b>3 PF</b>
Búsqueda por estado	EQ <b>4 PF</b>
6 tablas de la base de datos.	ILF <b>48 PF</b>

### Punto de función sin ajustar(conteo)

Tipo/nivel	Simple	Promedio	Complejo	Total
(EI)Entrada externa	<b>2</b> x 3 PF	4 x 4 PF	0 x 6 PF	<b>22</b>
(EO)Salida externa	<b>0</b> x 4 PF	1 x 5 PF	0 x 7 PF	<b>5</b>
(EQ)Consulta externa	1 x 3 PF	2 x 4 PF	0 x 6 PF	<b>11</b>
(ILF)Archivo lógico interno	5 x 7 PF	<b>2</b> x 10 PF	0 x 15 PF	<b>55</b>

(EIF)Archivo de interfaz externo	0 x 5 PF	0 x 7 PF	0 x 10 PF	0
----- -	-----	-----	<b>PFSA</b>	<b>93</b>

### Factor de Ajuste

FACTOR DE AJUSTE	Puntaje
Comunicación de datos	3
Procesamiento distribuido	1
Objetivos de rendimiento	3
Configuración de equipamientos	2
Tasa de transacciones	4
Entrada de datos en línea	5
Interface con el usuario	4
Actualizaciones en línea	1
Procesamiento complejo	0
Reusabilidad del código	3
Facilidad de implementación	2
Facilidad de operación	2
Instalaciones múltiples	0
Facilidad de Cambios	4
FACTOR DE AJUSTE	<b>34</b>

### Cálculo del Punto de función Ajustado

$$PFA = PFSA * [0.65 + (0.01 * FACTOR DE AJUSTE)]$$

$$PFA = 93 * [0.65 + (0.01 * 34)]$$

$$PFA = 92,07$$

Dónde:

PFA = Puntos de función ajustado

PFSA = Puntos de función sin ajustar

### Tabla de Productividad para estimación de esfuerzo

Lenguaje	Horas por PF (promedio)	Líneas de código por PF
Ensamblador	25	300
COBOL	15	200
<b>Lenguaje 4ta Generación</b>	<b>8</b>	<b>20</b>

### Cálculo de líneas de código

$$\text{Líneas de código} = PFA * (\text{Líneas de código por PF})$$

$$\text{Líneas de código} = 92,07 * 20$$

$$= 1841.4 \approx 1842 \text{ líneas de código}$$

### Cálculo de esfuerzo (hs/persona)

Esfuerzo = PFA \* horas PF promedio (En nuestro caso el lenguaje de programación es de cuarta generación)

$$\text{Esfuerzo} = 92,07 * 8$$

$$\text{Esfuerzo} = 736,56 \approx 737 \text{ hs/persona}$$

### Duración del proyecto en horas

$$\frac{\text{Esfuerzo}}{\text{Cant.de Desarrolladores}}$$

$$\text{Desarrolladores} = 3$$

$$737/3 = 245,666667 \text{ hs por miembro}$$

### Cálculo de días de trabajo

Suponiendo que se trabaja 5 horas (activas) diarias a la semana  $\approx 20$  días en un mes

En días:

$$\frac{\text{Duracion del proyecto en horas}}{5 \text{ horas diarias a la semana de trabajo}}$$

$$= 246/5 = 49 \text{ días}$$

$$50 \text{ días aprox. de trabajo}$$

En meses:

$$\frac{\text{Días de trabajo}}{20 \text{ días en un mes}}$$

$$50/20 = 2.5 \text{ meses de trabajo}$$

### Estimación de Duración del Proyecto

Se necesitan aproximadamente dos meses y medio para desarrollar el software trabajando de lunes a viernes cinco horas diarias con tres desarrolladores.

### Cálculo del presupuesto del proyecto

Sueldo mensual: 350 \$

Duración en meses del proyecto = 2,5 meses

Desarrolladores = 3

Costos adicionales durante la realización del proyecto = 2.750.000 Gs 400 \$ aprox.

COSTO = Desarrolladores \* Sueldo \* Duración en meses del proyecto + Costos adicionales durante la realización del proyecto

$$\text{COSTO} = (3 * 350 * 2,5) + 400$$

$$\text{COSTO} = 3025 \$$$

$$20.000.000 \text{ Gs aprox.}$$

## 2.5.2 Por COCOMO II

### Punto de función

SLOC Input Dialog - SADRM

Sizing Method

☐ SLOC

☒ Function Points

☐ Adaptation and Reuse

Breakage

% of code thrown away due to requirements evolution and volatility

REVL

Module Size in Function Points

Language   20

Ratio Type : ☒ Jones ☐ David

Calculation Method : ☒ Using Table ☐ Input Calculated Function Point

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	<input type="text" value="2"/>	<input type="text" value="4"/>	<input type="text" value="0"/>	22
Outputs	<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="0"/>	5
Files	<input type="text" value="5"/>	<input type="text" value="2"/>	<input type="text" value="0"/>	55
Interfaces	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Queries	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="0"/>	11
Total Unadjusted Function Points				93
Equivalent Total in SLOC				1860

### Scale Factors

Scale Factors

	base	Incr%
Precedentedness .....	<input type="text" value="NOM"/>	<input type="text" value="0%"/>
Development Flexibility .....	<input type="text" value="LO"/>	<input type="text" value="0%"/>
Architecture / risk resolution	<input type="text" value="NOM"/>	<input type="text" value="0%"/>
Team cohesion .....	<input type="text" value="NOM"/>	<input type="text" value="0%"/>
Process maturity .....	<input type="text" value="LO"/>	<input type="text" value="0%"/>

Scale Factor :

## Captura del software COCOMO II con los resultados completos

Project Name:  Scale Factor: 18.97 Schedule

Project Notes  Development Model:

X	Module Name	Module Size	LABOR Rate (\$/month)	EEAF	Language	NCM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	SADRM	F:1860	350.00	1.00	Fourth Genera	5.8	5.8	319.7	2036.10	1.1	0.9	0.0

	Estimated	Effort Sched	PROD	COST	INST	Staff	RISK
Total Lines of Code: 1860	Optimistic	3.9	5.7	477.2	1364.19	0.7	0.7
Hours/PM: 152.00	Most Likely	5.8	6.4	319.7	2036.10	1.1	0.9
	Pessimistic	8.7	7.3	213.2	3054.15	1.6	1.2

## Capítulo 3 – Desarrollo

### 3.1 Introducción

Desde el primer momento estamos aplicando metodología ágil para el desarrollo del software, por las características propias que se solicita del proyecto, cantidad de integrantes, tiempo corto de desarrollo. Como ya sabrán existen diferentes tipos de metodologías ágiles todos estos fueron elaborados en base al manifiesto ágil, también son llamados de “conjuntos de buenas prácticas” porque más que un método que se debe seguir paso a paso estas dan sugerencias y técnicas que facilitan, mejoran y optimizan el proceso del desarrollo de software. Las metodologías ágiles que vamos a estar utilizando es una combinación entre las metodologías ágiles SCRUM y XP. En este capítulo se describirá más a detalle la metodología de desarrollo ágil, las herramientas, las responsabilidades de los integrantes del equipo, también en este capítulo se tiene en cuenta las buenas prácticas para una codificación limpia.

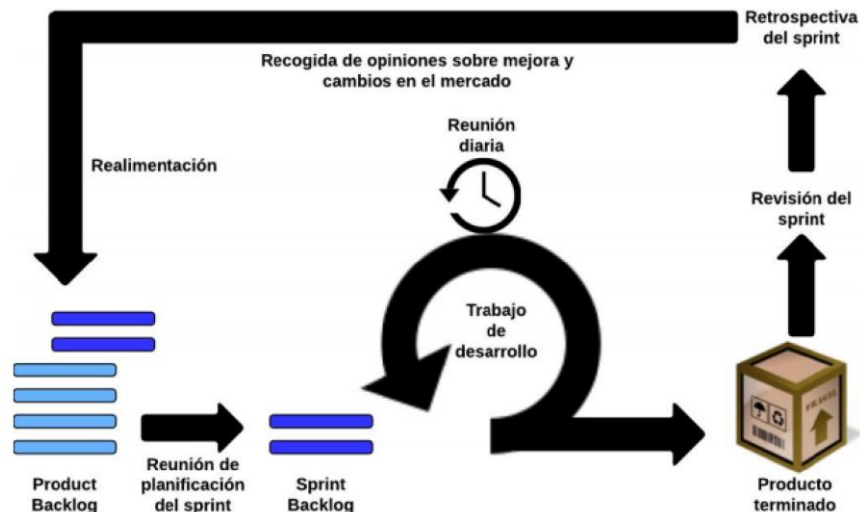
### 3.2 Metodología de Desarrollo Ágil

Se escogió la metodología de Desarrollo Ágil Scrum que se basa en el desarrollo de software de forma iterativa e incremental. Cada iteración, denominada Sprint, tiene una duración preestablecida de entre 2 y 4 semanas, obteniendo como resultado una versión del software con nuevas prestaciones listas para ser usadas. En cada nuevo Sprint, se va ajustando la funcionalidad ya construida y se añaden nuevas prestaciones priorizándose siempre aquellas que aporten mayor valor de negocio.



El principal motivo de elección de esta metodología fue porque ayuda a desarrollar el software en periodos cortos, permitiendo obtener de forma rápida feedback por parte del cliente, adaptaciones y una mejora continuada.

Además, porque ayuda a gestionar los flujos de trabajo y mejora la productividad el trabajo en equipo y la cooperación.



### 3.2.1 Roles de los integrantes

Nombre del Integrante	Rol del Integrante	Descripción de actividades a realizar
José Mario Barreto Jara	Líder del proyecto, analista, desarrollador, implementador, Testing.	Es el encargado de coordinar las tareas generales del proyecto, de la programación, implementación, pruebas y controlar que todo se realice en tiempo y forma.
Mackey Ryuto Sugawara Mochinaga	Analista, desarrollador, diseñador, documentador.	Encargado del desarrollo del Sistema y de la documentación.
Diego Samuel Velastiqui Santacruz	Analista, desarrollador, implementador, documentador, diseñadora grafica	Encargado del diseño de los prototipos del sistema, visitar al cliente y de la redacción de los requisitos de la programación.

### 3.2.2 Product Backlog (requisitos)

Sprint	Requisitos	Prioridades	Duración
0	1. Instalación y configuración de las herramientas	Alta	03/04/2023 al 08/04/2023
1	2. Registro de Pacientes	Alta	09/04/2023 al 20/04/2023
	3. Agendamiento de cita		

	4. Visualización de info estudio		
2	5. Gestión de Usuarios	Alta	21/04/2023 al 05/05/2023
	6. Acceso de cuentas		
3	7. Carga de pacientes	Alta	06/05/2023 al 13/05/2023
	8. Programación de citas		
	9. Visualización de Información		
4	10. Configuración de horarios	Media	14/05/2023 al 21/05/2023
	11. Generación de Informes		
5	12. Generación de la vista del paciente	Media	22/05/2023 al 30/05/2023
6	Revisión y prueba general del funcionamiento del sistema.	Alta	31/05/2023 al 30/06/2023

### 3.2.3 Descripción de cada Sprint y entregables

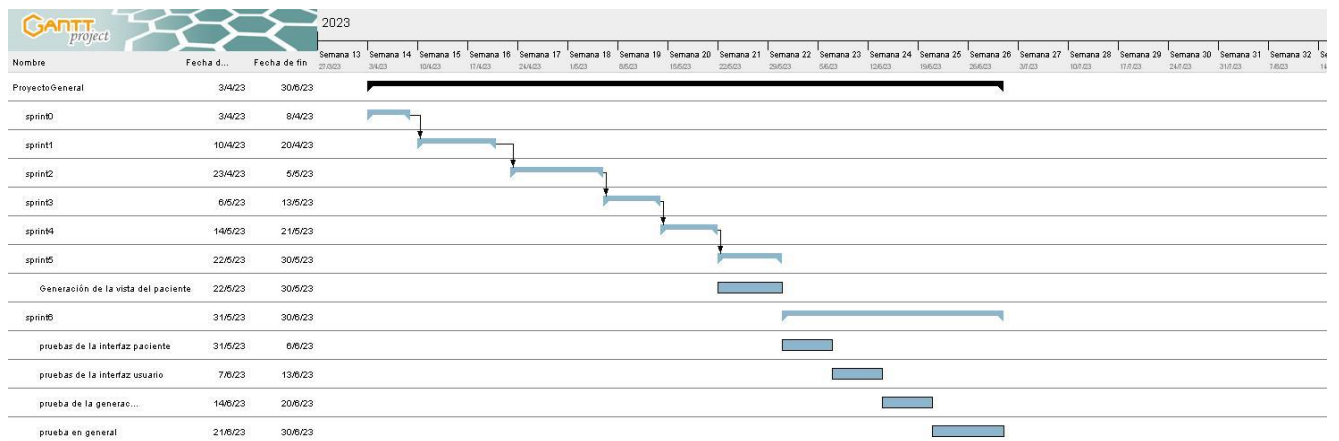
En este apartado vamos a describir cada Sprint y el objetivo de cada uno de ellos hasta desarrollar todo el proyecto, cuáles son los entregables al final de cada Sprint, es importante incluir un diagrama de Gantt para explicar la secuencia de cada Sprint.

Como recomienda la metodología ágil debe existir un producto/entregable al final de cada Sprint. En este apartado vamos a estar detallando que módulos o funcionalidades se estará entregando al final de cada Sprint hasta finalizar la versión 1 del software.

Los requisitos fueron seleccionados por orden de prioridad.

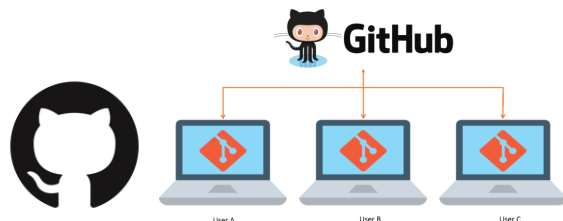
Sprint	Descripción	Entregables
0	Instalación y configuración de las herramientas a ser utilizadas, estudio del lenguaje para la codificación, las mismas son todas libres Creación de BD y carga de datos para la prueba	Se espera tener todas las herramientas instaladas y funcionando correctamente para el desarrollo.
1	Realización de los abm de las tablas paciente con sus respectivas interfaces graficas.	El módulo funcional de la interfaz de registro para pacientes.
2	Creación de la interfaz de Login. Funcionalidades: autenticación y restricción de contenido para los usuarios.	La interfaz de Login para los respectivos perfiles.

3	Realización de la interfaz de perfiles con la funcionalidad de programar y agendar citas de pacientes.	El módulo funcional de perfiles.
4	Creación de plantillas para los informes y generación de los mismos.	Informes en PDF.
5	Creación de la vista “home principal” y página de inicio del admin.	Página de inicio.
6	Revisión y prueba general del funcionamiento del sistema.	Página de inicio, login, interfaz usuario, generaciones de informes, etc.



### 3.3 Herramientas colaborativas de desarrollo

Para el desarrollo en software en equipo es indispensable utilizar herramientas colaborativas de desarrollo distribuido. Antes de tipear la primera línea de código, toda la infraestructura debe estar instalada en todas las máquinas de cada miembro del equipo, configurada y probar. Las herramientas utilizadas para el desarrollo son:



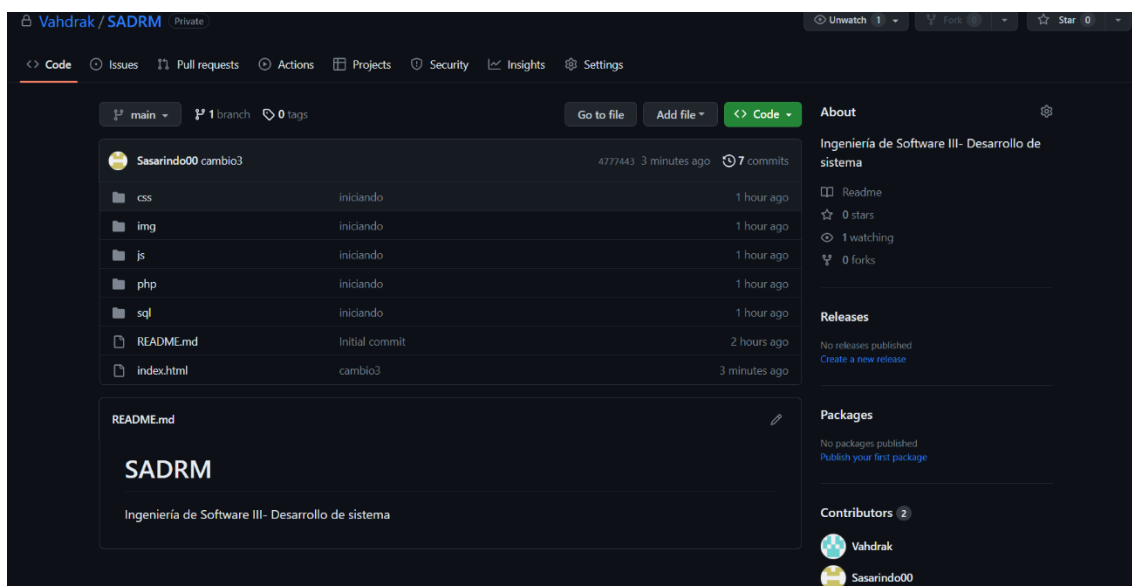
GitHub es un servicio de alojamiento de repositorios de código fuente que permite a los desarrolladores almacenar, compartir y colaborar en proyectos de software.

Para crear un repositorio en Github: Se inicia sesión en la cuenta de Github en <https://github.com/>. Se hace clic en el botón "Nuevo" en la página principal de Github para crear un nuevo repositorio.

En la página "Crear un nuevo repositorio", se ingresa el nombre del repositorio como "nombre". Opcionalmente, se puede agregar una descripción y una licencia para el repositorio. Se selecciona la opción "Crear repositorio" para crear el repositorio.

Para agregar colaboradores en Github: Se hace clic en la pestaña "Configuración" en la parte superior de la página del repositorio.

Se selecciona "Colaboradores" en el menú de la izquierda. Se ingresa el nombre de usuario o la dirección de correo electrónico de los colaboradores que se desea agregar. Se hace clic en "Agregar colaborador" y se selecciona el nivel de acceso que se desea otorgarles.



Repositorio de github

La integración de GitHub en Visual Studio Code permite a los desarrolladores trabajar de manera más eficiente en proyectos de software almacenados en repositorios de GitHub. Esto hace que la colaboración en equipo sea más fácil y productiva.

### Editor:

Visual Studio Code



Visual Studio Code

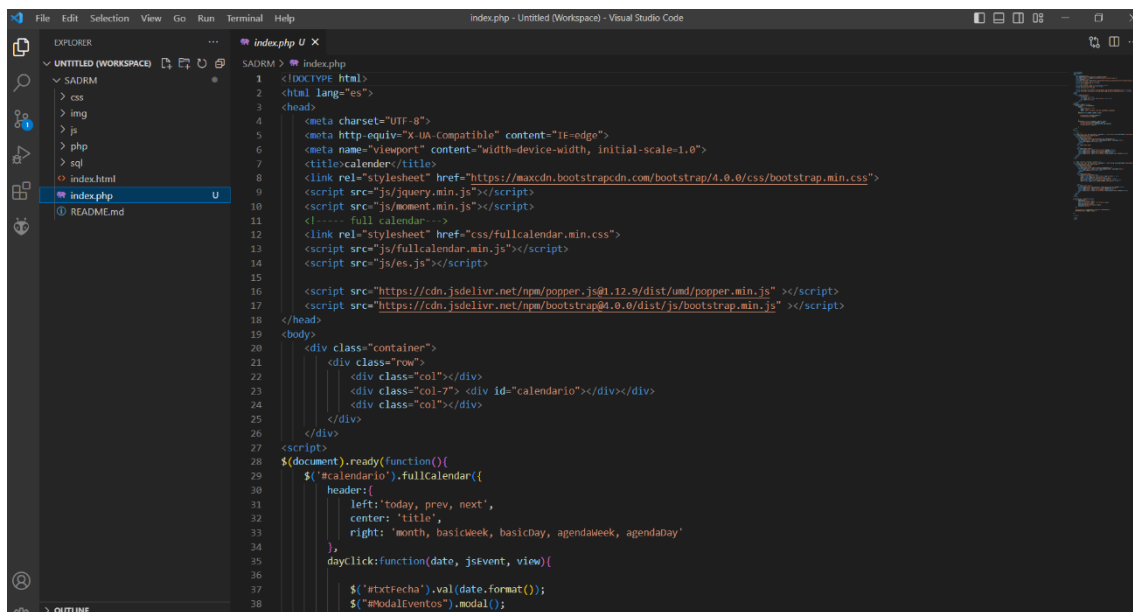
Es un editor de código fuente gratuito que ofrece una amplia gama de herramientas y extensiones para el desarrollo de software.

Para vincular el repositorio con Visual Studio Code:

Se abre Visual Studio Code, se selecciona "Clonar repositorio" en la pantalla de inicio. En el cuadro de diálogo "Clonar repositorio", se ingresa la URL del repositorio que se acaba de crear y se hace clic en "Clonar".

Una vez que se haya vinculado el repositorio con Visual Studio Code, se puede comenzar a trabajar en el proyecto. Cuando se esté listo para guardar los cambios, se realiza un "commit" y un "push" al repositorio en Github.

El dueño del repositorio revisará los cambios y los aprobará si están correctos. Los colaboradores podrán trabajar en el proyecto y hacer mejoras continuas para ayudar a que el repositorio crezca y prospere.



Repositorio vinculado con visual studio code

## Lenguaje De Programación: PHP



Lenguaje de programación interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.

Abrir Visual Studio Code y acceder a la pestaña de Extensiones. Buscar las extensiones "PHP IntelliSense" y "PHP Server", instalarlas. Luego de eso el programa debería detectar automáticamente que se está trabajando en un archivo PHP y cambiará el modo de resaltado de sintaxis de acuerdo. Comenzar a escribir el código PHP, y Visual Studio Code proporcionará sugerencias y autocompletará código PHP según la extensión que se instaló.

Para utilizar la extensión "PHP Server", acceder a la paleta de comandos de Visual Studio Code.

Seleccionar la carpeta raíz del proyecto PHP y hacer clic en "OK". El servidor PHP ahora está ejecutándose y se puede acceder al proyecto PHP desde un navegador web.

Seguido de esto, se puede escribir y trabajar en el código PHP en Visual Studio Code.

## Gestor de Base De Datos: MySQL



Sistema de base de datos relacional o RDBMS (Relational Database Management System) que emplea un modelo cliente-servidor.

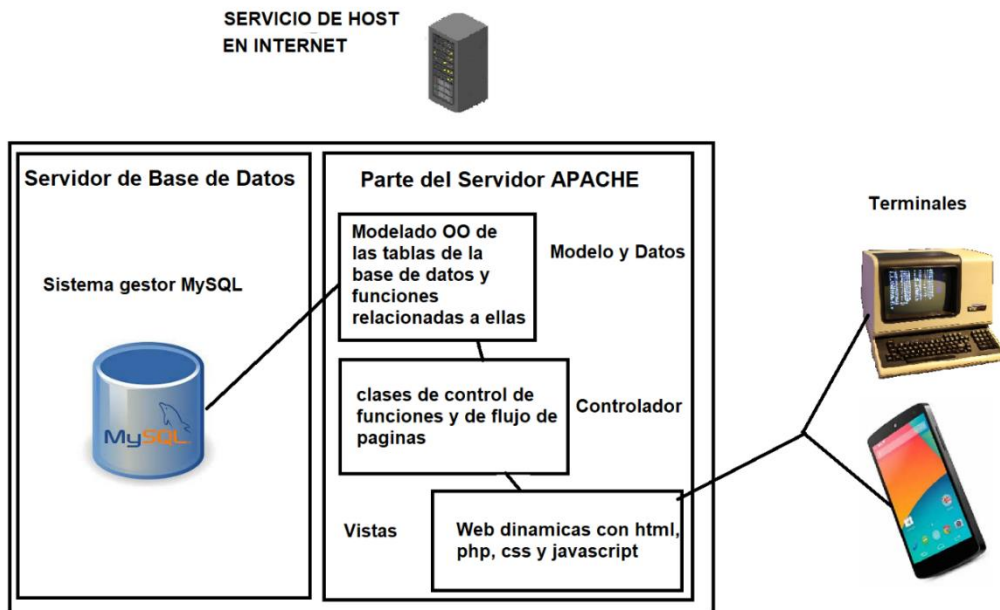
Versión: 5.7

## XAMPP



Es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. Es el servidor que estaremos usando para el desarrollo del proyecto.

Versión 3.3.0

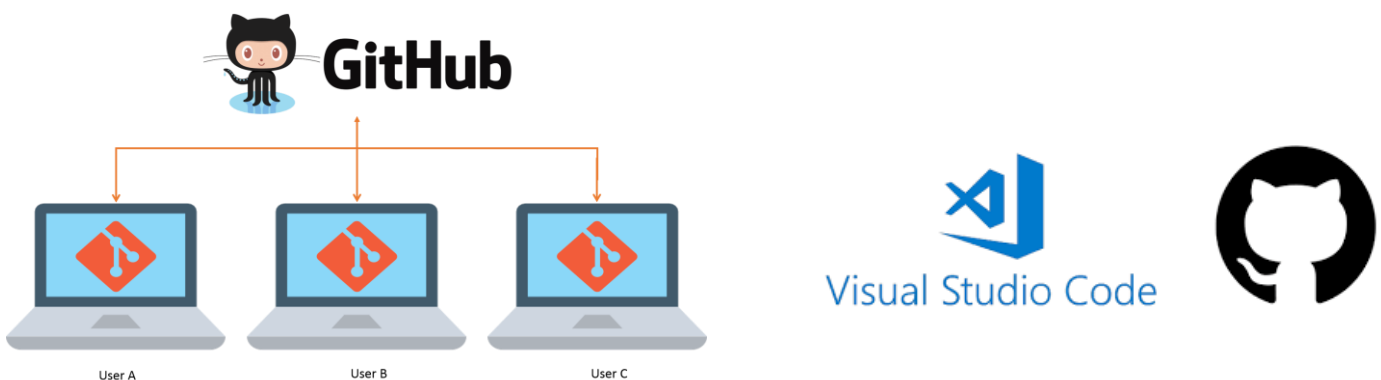


Arquitectura de la aplicación

Las terminales se conectan con la capa de vistas que está en el servicio de hosting en la parte del servidor apache, esta a su vez se conecta con la capa de controlador que hace las llamadas a los métodos y controla el flujo de las vistas, para hacer las llamadas la capa del controlador se conecta a la capa de modelo y datos en la cual esta codificado el mapeo a la base de datos y las funciones pertinentes para la aplicación, y esta capa está conectada al gestor de base de datos que está en el servidor de base de datos del hosting.

### 3.4 Herramientas colaborativas de desarrollo

Para el desarrollo en software en equipo es indispensable utilizar herramientas colaborativas de desarrollo distribuido. Antes de tipear la primera línea de código, toda la infraestructura debe estar instalada en todas las máquinas de cada miembro del equipo, configurada y probar. Utilizaremos GitHub en conjunto con Visual Studio Code.



### 3.5 Herramienta de organización en equipo

Para no retrasar el desarrollo del trabajo todos deben de saber exactamente lo que deben hacer, para esto es importante organizarse, para esta actividad existen diversas herramientas en líneas. Pueden usar la herramienta que mejor les parezca. Se utilizará la herramienta **Todoist**, que es un software de productividad en la nube que gestiona las tareas diarias de equipos de un modo sencillo. Sincroniza los dispositivos de todos los miembros y se integra con otras aplicaciones para centralizar la gestión de las tareas.

Aquí debe de mencionar la tarea que debe realizar cada integrante del equipo por cada Sprint.

The screenshot displays the Todoist interface for a project named "SADRM". The interface is organized into a list of tasks, each with a circular checkbox, a title, and a due date. The tasks are grouped into sprints, indicated by a downward arrow and a sprint name (e.g., "Sprint0 1", "Sprint1 3", "Sprint2 2", "Sprint3 3").

**SADRM** (Project Name)

Top navigation: Compartir, Vista, Comentarios, and a menu icon.

**Task List:**

- ☐ Proyecto (Jun 30)
- Sprint0 1**
  - ☐ Instalación y configuración e herramientas (Mackey,Jose,Diego)
- Sprint1 3**
  - ☐ Registro de Pacientes(Mackey,Jose) (Hoy)
  - ☐ Agendamiento de cita(Mackey,Diego) (Hoy)
  - ☐ Visualización de información de estudios(Jose,Diego) (Hoy)
- Sprint2 2**
  - ☐ Gestión de usuarios(Diego,Jose,Mackey) (May 5)
  - ☐ Acceso de cuentas (Jose,Mackey,Diego) (May 5)
- Sprint3 3**
  - ☐ Carga de Pacientes(Mackey,Jose) (May 13)
  - ☐ Programación de citas(Diego,Jose) (May 13)
  - ☐ Visualización de información (Mackey,Diego) (May 13)

Bottom navigation: Compartir, Vista, and a menu icon.

**Task Board (Kanban View):**

The task board shows the same tasks organized into columns representing sprints. Each task card includes the task title, assignees, and due date.

- Sin sección 1**: Proyecto (Jun 30)
- Sprint0 1**: Instalación y configuración e herramientas (Mackey,Jose,Diego)
- Sprint1 3**: Registro de Pacientes(Mackey,Jose) (Hoy), Agendamiento de cita(Mackey,Diego) (Hoy), Visualización de información de estudios(Jose,Diego) (Hoy)
- Sprint2 2**: Gestión de usuarios(Diego,Jose,Mackey) (May 5), Acceso de cuentas (Jose,Mackey,Diego) (May 5)
- Sprint3 3**: Carga de Pacientes(Mackey,Jose) (May 13), Programación de citas(Diego,Jose) (May 13), Visualización de información (Mackey,Diego) (May 13)
- Sprint4 2**: Configuración de horarios(Jose,Mackey) (May 21), Generación de informes(Diego) (May 21)
- Sprint5 1**: Generación de Vista paciente(Jose,Mackey,Diego) (May 31)



### 3.6 Estándares de código limpio que serán utilizados en toda la codificación del proyecto.

1. Mantener el código simple: La simplicidad es la clave para un código limpio. Se debe evitar la complejidad innecesaria y mantener el código lo más simple posible.
2. Utilizar nombres de variables y funciones descriptivas: Los nombres de variables y funciones deben ser descriptivos y representar con precisión su función en el código.
3. Documentar el código: La documentación es importante para un código limpio. Se debe incluir comentarios en el código para explicar su función y proporcionar detalles adicionales.

```
citac > classes > DBConnection.php > DBConnection
1  <?php
2  if(!defined('DB_SERVER')){
3      |   require_once("../initialize.php");
4  }
5  class DBConnection{
6
7      private $host = DB_SERVER;
8      private $username = DB_USERNAME;
9      private $password = DB_PASSWORD;
10     private $database = DB_NAME;
11     private $port = DB_PORT;
12
13     public $conn;
14
15     public function __construct(){
16
17         if (!isset($this->conn)) {
18
19             $this->conn = new mysqli($this->host, $this->username, $this->password, $this->database, $this->port);
20
21             if (!$this->conn) {
22                 |   echo 'Cannot connect to database server';
23                 |   exit;
24             }
25         }
26     }
27
28     public function __destruct(){
29         |   $this->conn->close();
30     }
31
32     ?>
```

Fragmento de código

## **Capítulo 4 – Pruebas y Calidad del Software**

### **4.1 Introducción**

En este capítulo se trata el proceso de pruebas de sistemas, incluyendo su relación con el proceso de obtención de requerimientos y resto del proceso de desarrollo de software.

La prueba intenta demostrar que un programa hace lo que se pretende hacer y descubre los defectos del programa antes de su puesta en uso. Puede revelar la presencia de errores, no sus ausencias.

### **4.2 El objetivo de las pruebas**

Para el software personalizado, esto significa que debe haber por lo menos una prueba por cada requisito funcional o no funcional. El objetivo principal de las pruebas es aportar calidad al producto que se está desarrollando. La ISO 25000 es un conjunto de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software, dice: “la calidad es el grado en que el producto de software satisface las necesidades expresadas o implícitas, cuando es usado bajo condiciones determinadas”.

### **4.3 ¿Cómo se llevarán a cabo las pruebas?**

Para llevar a cabo las pruebas verificaremos el comportamiento del programa sobre un conjunto de casos de pruebas. Estos casos de pruebas se generarán mediante técnicas y estrategias específicas de pruebas que nos ayudarán a conseguir la búsqueda de los errores de un programa.

Para la prueba de tipo caja negra hemos seleccionado las pruebas de caso de uso que realizamos en la etapa de diseño, para llevar a cabo esta prueba se analiza los casos de uso uno a uno y se verifica si se cumplen las acciones que previamente definimos.

Los casos de uso describen las interacciones entre actores (que pueden ser usuarios o sistemas) que producen un resultado que agrega algún valor. A partir de estos se pueden derivar casos de prueba.

Tienen precondiciones que deben cumplirse para que estos funcionen de forma exitosa.

Los casos de uso terminan con post-condiciones, que son resultados observables y estado del sistema después de la ejecución. Son útiles para definir las pruebas de aceptación, en las que participa el usuario o cliente.

Los casos de uso fueron de gran utilidad para llevar a cabo las pruebas correspondientes.

### **Prueba de tipo caja negra: pruebas de caso de uso**

<b>Título del caso de prueba</b>		CU-001: Inicio de sesión	
<b>Resumen de la prueba</b>		El objetivo de la prueba es verificar que el sistema valida correctamente las credenciales ingresadas por el usuario y le abre según el tipo de usuario.	
<b>Precondiciones</b>		El sistema debe estar conectado a la red. El usuario debe haber sido registrado previamente.	
<b>Pasos</b>		<b>Resultados</b>	
1	El usuario accede a la página de la aplicación web.	1	El usuario puede acceder a la página de la aplicación web.
2	El usuario coloca sus credenciales en los campos correspondientes y presiona iniciar sesión.	2	El usuario puede colocar sus datos en el formulario.
3	El sistema valida los datos que fueron ingresados.	3	El sistema valida correctamente los datos del usuario.
4	El sistema abre la página que le corresponde según el usuario que haya ingresado.	4	El sistema abre la página correctamente de acuerdo con el tipo de usuario que haya ingresado, en este caso el usuario "admin".
<b>Excepciones</b>		<b>Resultados</b>	
1.1	El usuario no accede a la url correcta.	1.1	El usuario no puede acceder al sistema.
1.2	El usuario coloca mal su nombre de usuario y/o contraseña.	1.2	El sistema no le permite acceder al usuario.
3.1	El sistema no encuentra coincidencias con los datos proveídos.	3.1	El usuario no puede acceder al sistema.

<b>Título del caso de prueba</b>		CU-002: Proceso de agendamiento	
<b>Resumen de la prueba</b>		El objetivo de esta prueba es verificar si el proceso de agendamiento se registra correctamente, también si respeta las restricciones del horario.	
<b>Precondiciones</b>		El Paciente debe tener acceso a la Red de Internet, debe poseer un dispositivo por el cual conectarse a la aplicación WEB, debe poseer un escaneado de orden médica.	
<b>Pasos</b>		<b>Resultados</b>	
1	El paciente inicia la aplicación, solicita hacer un agendamiento.	1	El paciente puede visualizar para solicitar su agendamiento.
2	El paciente accede al calendario de fechas disponibles para el estudio, elige una en la cual desea agendarse.	2	El paciente puede acceder al calendario y ver las fechas disponibles para el estudio y elegir en la cual desea agendarse.
3	Inicio de proceso de agendamiento.	3	El paciente puede iniciar su proceso de agendamiento.
4	El paciente carga sus datos personales en el sistema.	4	El paciente puede cargar sus datos personales en el sistema.
5	El paciente indica si posee un seguro, carga los datos en caso de afirmación en el sistema.	5	El paciente puede seleccionar un seguro y puede cargar los datos en caso de afirmación en el sistema.
6	El paciente adjunta el escaneado de su orden médica en el sistema.	6	El paciente puede adjuntar el escaneado de su orden médica en el sistema.
7	El paciente elige un horario disponible para su cita.	7	El paciente puede elegir un horario disponible para su cita.
8	El paciente envía su solicitud de agendamiento para que la recepción valide.		El paciente ve su solicitud de agendamiento enviado.
<b>Excepciones</b>		<b>Resultados</b>	
2.1	El paciente elige una fecha incorrecta.	2.1	El formulario muestra la fecha seleccionada.
4.1	El paciente no llena todos los campos obligatorios.	4.1	El formulario muestra una alerta en el campo vacío.
5.1	El paciente si posee seguro, pero no se especificó.	5.1	El sistema envía 0 particular, será revisado manualmente el día de la cita.
6.1	No adjunto su orden médica.	6.1	Su cita no se podrá confirmar de manera asíncrona.
7.1	El paciente coloco un horario invalido.	7.1	El formulario muestra un mensaje que esta fuera del rango del horario.

<b>Título del caso de prueba</b>		CU-003: Validación de agendamiento	
<b>Resumen de la prueba</b>		El objetivo de la prueba es comprobar que el sistema permite verificar las agendas por parte de la recepción.	
<b>Precondiciones</b>		El Recepcionista tiene que iniciar sesión con una cuenta predefinida, debe tener acceso a la Red de Internet, deben existir solicitudes pendientes.	
<b>Pasos</b>		<b>Resultados</b>	
1	El recepcionista inicia la aplicación, ingresa sus respectivas credenciales.	1	El recepcionista ingresa al sistema correctamente.
2	El recepcionista accede al listado de solicitudes de estudio, elige una en la cual desea verificar.	2	El sistema muestra el formulario con los datos cargados del paciente.
3	El recepcionista intenta ver el escaneado de la orden médica del paciente.	3	Se abre una sección con la imagen de la orden médica.
4	El Recepcionista autoriza la solicitud del paciente.	4	Se visualiza el nuevo estado de la cita.
<b>Excepciones</b>		<b>Resultados</b>	
3. 1	No adjunto su orden médica correctamente.	3	La sección que contiene la imagen está en blanco.

<b>Título del caso de prueba</b>		CU-004: Agendamiento de Paciente hecho por Recepción	
<b>Resumen de la prueba</b>		El objetivo de la prueba es comprobar que el sistema permite agendar al paciente desde la recepción.	
<b>Precondiciones</b>		El usuario tiene que estar identificado como Recepcionista en el sistema	
<b>Pasos</b>		<b>Resultados</b>	
1	El recepcionista accede al calendario de fechas disponibles para el estudio, elige una en la cual desea agendar.	1	El recepcionista es capaz de visualizar los turnos existentes y crear uno nuevo.

2	El sistema mostrará un formulario en el cual se solicitan los datos del paciente.	2	El sistema visualiza el formulario con los campos necesarios para la agenda.
3	El Recepcionista completa los datos solicitados y da click en el botón Programar cita.	3	El Recepcionista envía el formulario y agenda una cita.
<b>Excepciones</b>		<b>Resultados</b>	
3	El recepcionista deja algún campo en blanco o sin completar	3	No se permitirá enviar el formulario.

<b>Título del caso de prueba</b>		CU-005: Registrar usuario (Administrador)	
<b>Resumen de la prueba</b>		El objetivo de la prueba es comprobar que el sistema permite dar de baja un artículo seleccionado.	
<b>Precondiciones</b>		Se debe iniciar sesión como administrador.	
<b>Pasos</b>		<b>Resultados</b>	
1	En el menú principal el administrador selecciona el ítem de lista de usuario, da click en el botón crear usuario.	1	El administrador visualiza la lista de los usuarios actuales, es redireccionado a un formulario de inserción.
2	El sistema mostrará un formulario en el cual solicita: nombre, usuario, contraseña y tipo de usuario.	2	El administrador visualizara los datos necesarios para crear un usuario nuevo.
3	El administrador completa los datos solicitados y da click en el botón Crear Usuario.	3	El sistema almacena los datos y genera un identificador único para el usuario creado de forma automática.
4	El nuevo usuario se muestra en la lista de usuarios.	4	El administrador visualiza la lista de usuario actualizada.
<b>Excepciones</b>		<b>Resultados</b>	
3	- El administrar intenta cargar un username repetido. - Si se deja en blanco algún campo obligatorio tampoco permitirá guardar.	3	Si el sistema verifica que el usuario(username) existe no permitirá guardar si no se corrige. El formulario no se enviará si no se completa los campos obligatorios.

<b>Título del caso de prueba</b>		CU-006: Modificar usuarios	
<b>Resumen de la prueba</b>		El objetivo de la prueba es comprobar si el sistema puede registrar el pago de los clientes.	
<b>Precondiciones</b>		El usuario tiene que estar identificado como administrador del sistema.	
<b>Pasos</b>		<b>Resultados</b>	
1	En el menú principal el admin selecciona el ítem de lista de usuario, da click en el botón editar.	1	El administrador visualiza la lista de los usuarios actuales, es redireccionado a un formulario de edición.
2	El sistema mostrará una ventana con los datos del usuario seleccionado, como campos editables.	2	El administrador visualizara los datos necesarios para modificar un usuario existente.
3	El administrador edita los datos del usuario que requiera y da click en el botón actualizar.	3	El sistema actualiza y almacena los datos para el usuario seleccionado de forma automática.
4	El sistema almacena los datos.	4	El administrador visualiza la lista de usuario actualizada.
		<b>Resultados</b>	
3	<ul style="list-style-type: none"> <li>- El administrar intenta cargar un username repetido.</li> <li>- Si se deja en blanco algún campo obligatorio tampoco permitirá actualizar.</li> </ul>	3.1	Si el sistema verifica que el usuario(username) existe no permitirá guardar si no se corrige. El formulario no se enviará si no se completa los campos obligatorios.

<b>Título del caso de prueba</b>		CU-007: Dar de baja usuario	
<b>Resumen de la prueba</b>		El objetivo de la prueba es verificar si el usuario se le dio baja correctamente.	
<b>Precondiciones</b>		El usuario tiene que estar identificado como administrador del sistema.	
<b>Pasos</b>		<b>Resultados</b>	

1	En el menú principal el administrador selecciona el ítem de lista de usuario, da click en el botón eliminar usuarios.	1	El administrador visualiza la lista de los usuarios actuales, es redireccionado a una ventana de confirmación.
2	El administrador accederá a una vista de eliminación.	2	El sistema mostrará una ventana emergente con los datos de usuario seleccionado para confirmar la acción.
3	El administrador confirma la acción.	3	El sistema da de baja el usuario seleccionado.
<b>Excepciones</b>		<b>Resultados</b>	
3	Si el administrador no confirma la acción	3	El sistema regresa a la lista de usuarios.

<b>Título del caso de prueba</b>		CU-008: Agregar estudios y seguros	
<b>Resumen de la prueba</b>		El objetivo de la prueba es agregar los seguros y estudios en el sistema.	
<b>Precondiciones</b>		Se debe iniciar sesión como administrador.	
<b>Pasos</b>		<b>Resultados</b>	
1	En el menú principal el administrador selecciona el ítem de configuración, da click en el botón conf de seguros o conf de estudios y luego da click en uno de ellos y agregar.	1	El administrador visualiza la lista de los seguros o estudios actuales, es redireccionado a un formulario de inserción.
2	El sistema mostrará un formulario en el cual solicita: descripción de seguros o estudios.	2	El sistema muestra un formulario de nuevo registro.
3	El administrador completa los datos solicitados y da click en el botón agregar seguro o estudio.	3	El administrador puede registrar un nuevo estudio o seguro.
4	El sistema redirecciona a la lista de estudios o seguros.	4	El administrado visualiza la lista de los seguros o estudios actualizados con registros nuevos.
<b>Excepciones</b>		<b>Resultados</b>	
3	- Si se deja en blanco algún campo obligatorio tampoco permitirá guardar.	3	El sistema no permite guardar y sale un mensaje de advertencia.



<b>Título del caso de prueba</b>		CU-009: Modificar seguros y estudios	
<b>Resumen de la prueba</b>		El objetivo de la prueba es comprobar que el sistema permite hacer modificaciones de los datos de un seguro o estudio seleccionado.	
<b>Precondiciones</b>		El usuario tiene que estar identificado como administrador del sistema.	
<b>Pasos</b>		<b>Resultados</b>	
1	En el menú principal el admin selecciona el ítem de lista de seguros o estudios, da click en el botón editar.	1	El administrador visualiza la lista de los estudios y seguros actuales, es redireccionado a un formulario de edición.
2	El sistema mostrará una ventana con los datos del seguros o estudios seleccionado, como campos editables.	2	El administrador visualizara los datos necesarios para modificar un seguro o estudio existente.
3	El administrador edita los datos del seguro o estudio que requiera y da click en el botón actualizar.	3	El sistema actualiza y almacena los datos para el seguro o estudio seleccionado de forma automática.
4	El sistema almacena los datos.	4	El administrador visualiza la lista de seguros o estudios actualizada.
<b>Excepciones</b>		<b>Resultados</b>	
3	- Si se deja en blanco algún campo obligatorio tampoco permitirá actualizar.	3	El sistema no permite guardar y sale un mensaje de advertencia.

<b>Título del caso de prueba</b>		CU-010: Eliminar seguros y estudios.	
<b>Resumen de la prueba</b>		El objetivo de la prueba es verificar si el estudio o seguro se eliminó correctamente.	
<b>Precondiciones</b>		El usuario tiene que estar identificado como administrador del sistema.	
<b>Pasos</b>		<b>Resultados</b>	
1	En el menú principal el admin selecciona el ítem de lista de seguros o estudios, da click en el botón eliminar.	1	El administrador visualiza la lista de los seguros y estudios actuales, es redireccionado a una ventana de confirmación.

2	El administrador accederá a una vista de eliminación.	2	El sistema mostrará una ventana emergente con los datos de estudio o seguro seleccionado para confirmar la acción.
3	El administrador confirma la acción.	3	El sistema elimina el seguro o estudio seleccionado.
<b>Excepciones</b>		<b>Resultados</b>	
3.1	Si el administrador no confirma la acción	3.1	El sistema regresa a la lista de seguros o estudios.

<b>Título del caso de prueba</b>		CU-011: Modificar horarios	
<b>Resumen de la prueba</b>		El objetivo de la prueba es comprobar que el sistema permite hacer modificaciones de los datos de horarios.	
<b>Precondiciones</b>		El usuario tiene que estar identificado como administrador del sistema.	
<b>Pasos</b>		<b>Resultados</b>	
1	En el menú principal el admin selecciona el ítem de conf de horarios, da click en el botón editar horarios.	1	El administrador visualiza la lista de los horarios, es redireccionado a un formulario de edición.
2	El sistema mostrará una ventana donde se puede modificar los datos de horario, como campo editable.	2	El administrador visualizara los datos necesarios para modificar los horarios.
3	El administrador configura el horario que desea y selecciona a que día aplicar.	3	El sistema actualiza y almacena los datos para el horario seleccionado de forma automática.
4	El administrador hace click en el botón guardar.	4	El administrador visualiza la lista de horarios actualizada.
<b>Excepciones</b>		<b>Resultados</b>	
3	- Si se deja en blanco algún campo da mensaje de error.	3	El sistema no permite guardar y sale un mensaje de advertencia.

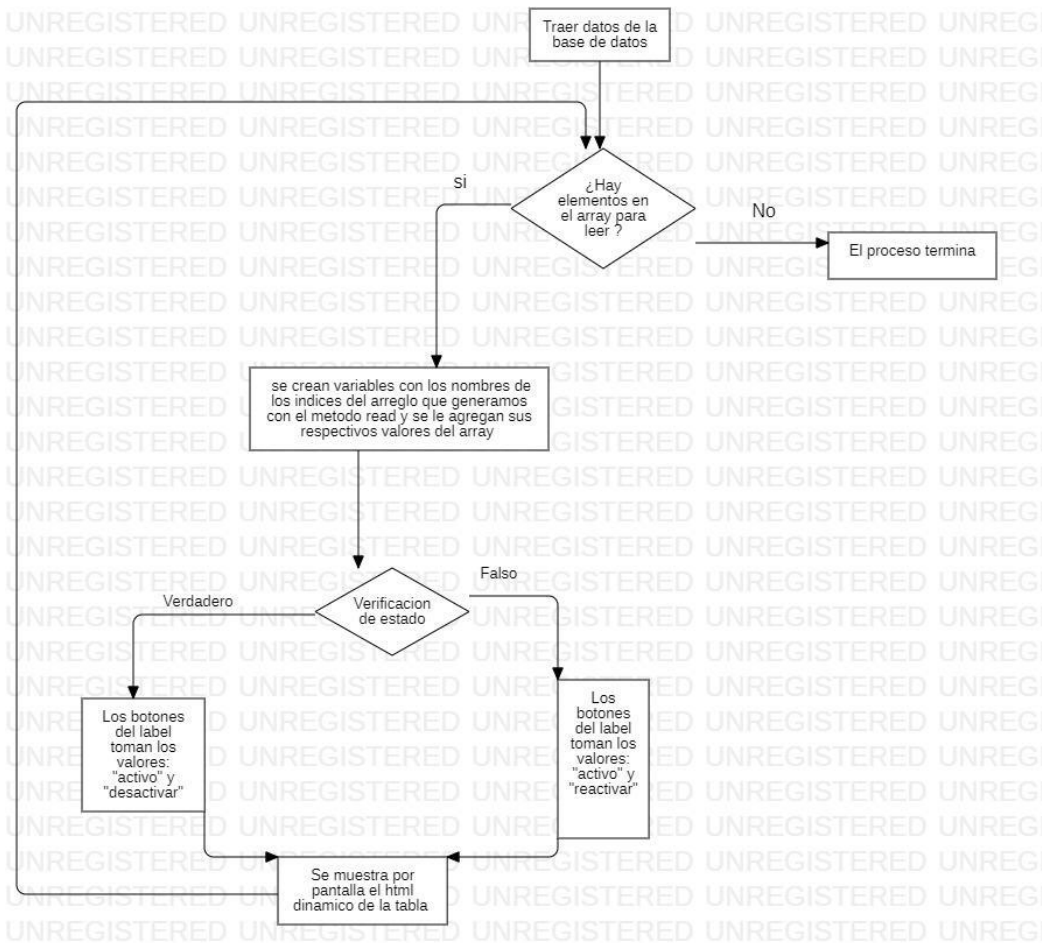
<b>Título del caso de prueba</b>		CU-012: Elaborar informes	
<b>Resumen de la prueba</b>		El objetivo de la prueba es comprobar que el sistema permite generar informes.	

<b>Precondiciones</b>		El usuario tiene que estar identificado como administrador del sistema.	
<b>Pasos</b>		<b>Resultados</b>	
1	El administrador ingresa a la sección de informes.	1	El administrador puede acceder a la vista de elaboración de informes.
2	El administrador selecciona el tipo de informe que desea y el rango tiempo a tomar en cuenta.	2	El administrador puede determinar los parámetros necesarios para generar el informe.
3	El sistema verifica que los campos obligatorios estén completos.	3	El sistema puede comprobar que los datos ingresados están completos.
4	El sistema genera la información correspondiente presentada en formato PDF.	4	El sistema puede generar los datos correspondientes al informe y presentarlos al usuario.
<b>Excepciones</b>		<b>Resultados</b>	
3	Si un campo obligatorio está vacío se muestra un mensaje de aviso al usuario solicitando que verifique los datos y regrese al paso 1 de la Secuencia normal.	3	Regresa al paso 1 de la Secuencia normal.

Para el tipo de caja blanca optamos por la técnica de estructuras de control de flujo (Decisión/Condición) que se trata de un tipo de prueba de software que utiliza el flujo de control del programa como modelo. La prueba de flujo de control es una estrategia de prueba estructural. Para el tipo de prueba de flujo de control, el equipo de prueba debe conocer toda la estructura, el diseño, el código y la implementación del software.

Prerrequisito: conocer el diseño, el código y la implementación.

## Prueba de tipo caja blanca: técnica de estructuras de control de flujo



## Prueba de tipo caja blanca: prueba de mutación.

En este tipo de pruebas los desarrolladores prueban una parte concreta del sistema después de haber ejecutado un cambio determinado en él.

Prueba en el archivo editar horario

Prueba 1:

- Mutación: Cambiar el valor de la condición en la línea 5 para que siempre se cumpla.
- Prueba: Verificar si se redirige a la página "../admin/index.php".
- Resultado esperado: La prueba debe fallar, ya que la condición siempre se cumple y debería redirigir a la página de administrador.

```
SADRM > admin > editarhorario.php
1  <?php
2  //session_start();
3  include_once '../login/database.php';
4  include_once '../login/log.php';
5  if ($privilegio == 3) {
6  |   header('location: ../admin/index.php');
7  }
8
```

#### Prueba 2:

- Mutación: Comentar la línea 3 que incluye el archivo "database.php".
- Prueba: Verificar si se obtienen los datos del horario correctamente desde la base de datos.
- Resultado esperado: La prueba debe fallar, ya que no se incluye el archivo "database.php" y no se pueden obtener los datos del horario correctamente.

```
8
9 // Obtener los datos actuales del horario desde la base de datos
10 $db = new Database();
11 $getQuery = $db->connect()->prepare("SELECT * FROM schedule_settings");
12 $getQuery->execute();
13 $data = $getQuery->fetchAll(PDO::FETCH_ASSOC);
14
```

#### Prueba 3:

- Mutación: Cambiar la asignación de la variable \$start\_time en la línea 34 para que siempre sea nula.
- Prueba: Verificar si se actualiza el tiempo de inicio correctamente en la base de datos.
- Resultado esperado: La prueba debe fallar, ya que se asigna un valor nulo a la variable \$start\_time y no se actualiza el tiempo de inicio correctamente en la base de datos.

```
31 if (!empty($existingData)) {
32     $existingData = array_values($existingData); // Reindexar el array
33
34     $existingStartTime = $existingData[0]['start_time'] ?? null;
35     $existingEndTime = $existingData[0]['end_time'] ?? null;
36
37     $updateQuery = $db->connect()->prepare("UPDATE schedule_settings SET start_time = :start_time, end_time = :
38
```

#### Prueba 4:

- Mutación: Cambiar la asignación de la variable \$end\_time en la línea 26 para que siempre sea nula.
- Prueba: Verificar si se actualiza el tiempo de finalización correctamente en la base de datos.
- Resultado esperado: La prueba debería pasar si la variable \$end\_time no se utiliza más adelante en el código o si su valor nulo no afecta el funcionamiento esperado.

## 4.4 Aplicación de herramientas de pruebas

Todos los proyectos, por muy pequeños que sean, pueden llegar a tener una cantidad de casos de pruebas muy elevado, sin contar que los testing se repetirán varias veces debido a las pruebas de regresión. Estos proyectos, necesitan de una administración, planificación y ejecución, así como de herramientas que permitan realizar pruebas automáticas.

## **PHPUnit**

PHPUnit es un framework para PHP que facilita la creación de clases de pruebas sobre aplicaciones basadas en PHP.

PHPUnit utiliza aserciones para verificar el comportamiento de la funcionalidad que hay que probar. El objetivo de las pruebas unitarias es demostrar que el funcionamiento de cada funcionalidad es correcto y encontrar problemas en las fases iniciales del desarrollo de software.

El funcionamiento de PHPUnit se basa en la creación de casos de prueba, que son clases que contienen métodos de prueba que verifican el comportamiento esperado de la unidad de código bajo prueba. Estos métodos se escriben utilizando las aserciones proporcionadas por PHPUnit, que permiten verificar si un valor cumple con una condición específica.

Al ejecutar PHPUnit, se recopilan y se ejecutan todos los casos de prueba definidos en un directorio o archivo específico. Durante la ejecución, se informa sobre el éxito o fracaso de cada prueba, junto con detalles sobre los errores o fallos encontrados. Además, se generan informes detallados que ayudan a identificar qué pruebas pasaron, cuáles fallaron y proporcionan información útil para depurar y corregir errores.

```
Símbolo del sistema
c:\xampp\htdocs\SADRM>php ./vendor/bin/phpunit tests/operationTest.php
PHPUnit 9.6.0 by Sebastian Bergmann and contributors.

Cannot open file "tests/operationTest.php".

c:\xampp\htdocs\SADRM>php ./vendor/bin/phpunit tests/operationsTest.php
PHPUnit 9.6.0 by Sebastian Bergmann and contributors.

E                                                                    1 / 1 (100%)
Error connection: SQLSTATE[HY000] [2002] No se puede establecer una conexión ya
que el equipo de destino denegó expresamente dicha conexión

Time: 00:02.099, Memory: 4.00 MB

There was 1 error:

1) OperationsTest::testGetUsers
Error: Call to a member function prepare() on null

C:\xampp\htdocs\SADRM\operaciones\operations.php:46
C:\xampp\htdocs\SADRM\tests\operationsTest.php:12

ERRORS!
Tests: 1, Assertions: 0, Errors: 1.

c:\xampp\htdocs\SADRM>php ./vendor/bin/phpunit tests/operationsTest.php
PHPUnit 9.6.0 by Sebastian Bergmann and contributors.

.                                                                    1 / 1 (100%)

Time: 00:00.121, Memory: 4.00 MB

OK (1 test, 1 assertion)

c:\xampp\htdocs\SADRM>
```

El comando ejecutado comienza a ejecutar las pruebas y muestra el siguiente mensaje de error: "Error connection: SQLSTATE[HY000] [2002] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión". Esto indica que hay un problema con la conexión a la base de datos. Dentro del archivo operations.php, en el método connect() de la clase Database, se intenta establecer una conexión a la base de datos utilizando los valores de conexión proporcionados. Sin embargo, ocurre un error al intentar establecer la conexión, lo que resulta en el mensaje de error mostrado.

Se verifica el archivo de conexión y se vuelve a ejecutar el comando. Esta vez, las pruebas se ejecutan correctamente y se muestra el siguiente mensaje: "OK (1 test, 1 assertion)". Esto indica que la prueba definida en el archivo operationsTest.php se ejecuta

correctamente y se realizó al menos una aserción exitosa. Dentro del archivo `operations.php`, el método `getUsers()` de la clase `Operations` se ejecuta correctamente y devuelve los resultados esperados de la consulta a la base de datos.

Con las pruebas anteriores pudieron ser identificados errores en casos específicos, en la última imagen podemos ver la salida cuando los errores ya fueron solucionados. El proceso consiste en hacer una llamada a las funciones y/o módulos a ser examinados y proveerlos de alguna aserción que pueda validar la salida en dicho método o función.

## **JMeter**

Es una herramienta de código abierto desarrollada por Apache Software Foundation que se utiliza para realizar pruebas de carga y rendimiento en aplicaciones web. Proporciona una plataforma para simular escenarios de carga y medir el rendimiento de diversos componentes de una aplicación, como servidores web, bases de datos y servicios web.

El funcionamiento de JMeter se basa en el concepto de "plan de pruebas", que es un conjunto de pasos que describen las acciones que se deben llevar a cabo durante la prueba. Estos pasos se llaman "elementos" y pueden incluir solicitudes HTTP, extracción y procesamiento de datos, y lógica de control, entre otros.

Para ejecutar una prueba con JMeter, se configuran los elementos necesarios en el plan de pruebas, como la configuración del servidor objetivo, las solicitudes HTTP que se enviarán, los datos de entrada, etc. Una vez configurado, se inicia la prueba y JMeter simula múltiples usuarios enviando solicitudes al servidor objetivo, registrando los tiempos de respuesta y otros datos relevantes.

Después de la ejecución de la prueba, JMeter proporciona resultados detallados en forma de gráficos y tablas, que permiten analizar el rendimiento de la aplicación bajo diferentes cargas de trabajo. Estos resultados pueden ayudar a identificar cuellos de botella, problemas de rendimiento y áreas que requieren optimización.



## 4.5 Calidad del software.



Ilustración 44 - Dimensiones de Calidad

**1. Adecuación funcional:** La capacidad del software para cumplir con las necesidades del usuario y satisfacer los requisitos establecidos es una característica fundamental.

Esta capacidad se puede desglosar en las siguientes subcaracterísticas:

**Adecuación:** se refiere a la capacidad del software para ofrecer las funciones y características requeridas por el usuario.

**Precisión:** se refiere a la capacidad del software para proporcionar resultados correctos y precisos.

**Interoperabilidad:** se refiere a la capacidad del software para interactuar y comunicarse de manera efectiva con otros sistemas.

**Seguridad:** se refiere a la capacidad del software para proteger los datos y garantizar la seguridad de la información.

El sistema debe poder cumplir con las siguientes funcionalidades correctamente:

- Registro y modificación de usuarios
- Registro y modificación de estudios o seguros
- Agendamiento de citas
- Elaboración de informes

**2. Eficiencia de desempeño:** La capacidad del software para cumplir con las tareas necesarias dentro de un límite de tiempo y recursos determinados es una característica esencial. Esta capacidad se desglosa en las siguientes subcaracterísticas:

- Rendimiento en tiempo de ejecución: se refiere a la habilidad del software para completar las tareas dentro de un tiempo establecido.
- Utilización de recursos: hace referencia a la cantidad de recursos que el software requiere para llevar a cabo las tareas.

Es fundamental que el sistema pueda llevar a cabo las tareas requeridas en un tiempo razonable, evitando prolongados períodos de ejecución. Además, es importante mantener un equilibrio en el consumo de energía y CPU de la aplicación mientras está en funcionamiento. Esto se logra optimizando las consultas a la base de datos y asegurándose de que no haya procesos en segundo plano que no sean realmente útiles.

**3. Compatibilidad:** Se refiere a la capacidad del software para establecer una comunicación eficaz con otros sistemas.

En el contexto de nuestro proyecto, en principio no es necesario que el software se comunique con otros sistemas. No obstante, resulta crucial asegurarse de que la ejecución de la aplicación no cause ningún problema en el funcionamiento adecuado del dispositivo en el cual está instalado.

**4. Usabilidad:** Se refiere a la capacidad del software para ofrecer una experiencia de uso efectiva, eficiente y satisfactoria para el usuario.

Esta característica se puede desglosar en las siguientes subcaracterísticas:

- Comprensibilidad: se refiere a la capacidad del software para ser comprendido y entendido por el usuario.
- Aprendizaje: se refiere a la facilidad con la que el usuario puede aprender a utilizar el software.
- Operabilidad: se refiere a la facilidad con la que el usuario puede operar y manejar el software.
- Atractivo visual: se refiere a la capacidad del software para atraer y mantener la atención visual del usuario.

Es esencial que el sistema pueda ser utilizado de manera eficiente y satisfactoria, permitiendo agilizar el trabajo del usuario. El software desarrollado debe contar con una interfaz cómoda, visualmente atractiva, fácil de comprender e interpretar. Además, es importante utilizar un lenguaje y terminología familiar al usuario en su entorno de trabajo. También se deben distribuir las opciones y funciones disponibles de manera

organizada y de fácil acceso, para que el usuario pueda encontrar rápidamente lo que necesita en cada contexto de uso.

**5.Fiabilidad:** Se refiere a la capacidad del software para mantener un nivel de funcionamiento constante a lo largo del tiempo y en diversas condiciones de uso.

Esta característica se puede dividir en las siguientes subcaracterísticas:

- **Madurez:** se refiere a la capacidad del software para evitar errores y fallas.
- **Tolerancia a fallos:** se refiere a la capacidad del software para mantener su nivel de funcionamiento incluso en caso de fallos o errores.
- **Capacidad de recuperación:** se refiere a la capacidad del software para recuperarse y restaurar su funcionamiento normal después de una falla o interrupción.

Es esencial que el software cumpla de manera satisfactoria con sus funciones en cualquier momento. Además, no debe colapsar por completo en caso de errores de software, problemas de hardware o acciones incorrectas del usuario. Para abordar esto, se puede implementar la eliminación lógica, lo que significa que los archivos que el usuario desea eliminar no se eliminan definitivamente de la base de datos, lo que permite recuperar datos en caso de eliminaciones accidentales.

**6. Seguridad:** Se refiere a la capacidad del software para salvaguardar los datos y asegurar la seguridad de la información.

Esta característica se puede desglosar en las siguientes subcaracterísticas:

- **Confidencialidad:** se refiere a la capacidad del software para proteger la información confidencial y mantenerla privada.
- **Integridad:** se refiere a la capacidad del software para garantizar que los datos se mantengan intactos y no se vean comprometidos por modificaciones no autorizadas.
- **Autenticidad:** se refiere a la capacidad del software para asegurar que los usuarios sean auténticos y sean quienes dicen ser.

Los usuarios deberán autenticarse en la aplicación a través de un proceso de inicio de sesión para acceder a sus funciones. El sistema debe garantizar que los datos de los usuarios y los registros de los pacientes estén protegidos de manera adecuada, utilizando técnicas de encriptación para resguardar la información de posibles accesos no autorizados.

## **7. Mantenibilidad:**

Se refiere a la capacidad del software para ser modificado y corregido de manera eficiente.

Para lograr esto, es fundamental utilizar un sistema de control de versiones que permita rastrear y gestionar los avances realizados durante el proceso de desarrollo. Esto nos facilitará comprender y analizar el código en el futuro, lo que nos permitirá implementar mejoras y agregar nuevas funcionalidades de manera más efectiva. Además, es recomendable incluir comentarios en el código para facilitar su comprensión y agilizar el proceso de interpretación.

**8. Portabilidad:** Se refiere a la capacidad del software para funcionar en diversos entornos de hardware y software.

Para lograr esto, es importante que la instalación del software en diferentes dispositivos con sistemas operativos compatibles sea sencilla y no requiera pasos complicados. Se busca proporcionar un archivo de instalación único que permita poner en marcha el software de manera fácil y sin complicaciones. El sistema debe ser capaz de ser utilizado correctamente en distintos entornos y dispositivos, sin importar las características específicas de cada uno de ellos.

# Capítulo 5 – Pruebas y Calidad del Software

## 5.1 Introducción

En este capítulo se presentan planes sobre el mantenimiento del software, se presentan requisitos funcionales y no funcionales que serán incluidos en la siguiente versión. También aplicamos una encuesta de satisfacción de todos los usuarios. Como conclusión presentamos los logros funcionales del sistema.

## 5.2 Mantenimiento del software

Lo ideal siempre es realizar mantenimiento periódicos programados preventivos al software para evitar que generen problemas mayores, como el mantenimiento de un auto, por eso es sumamente importante planificar el mantenimiento y sus procesos.

Tareas a realizar de acuerdo con la clasificación de las actividades de mantenimiento:

**1. Adaptable:** adaptar el software a nuevos entornos, tecnologías o requisitos cambiantes.

- a) Análisis de requerimientos: Se identifican y analizan los nuevos requerimientos y cambios en el entorno del software.
- b) Diseño de cambios: Se definen los cambios necesarios en la arquitectura y el diseño del software para adaptarlo a los nuevos requerimientos.
- c) Implementación de cambios: Se realizan las modificaciones en el código fuente y se integran en el sistema existente.
- d) Pruebas de aceptación: Se verifican los cambios implementados para asegurar que cumplen con los requerimientos y no introducen errores adicionales.
- e) Despliegue y distribución: Se lleva a cabo la instalación y puesta en producción de la versión adaptada del software.

**2. Perfectivo:** El mantenimiento perfectivo tiene como objetivo mejorar la calidad y el rendimiento del software sin cambiar su funcionalidad.

Las actividades principales son:

- a) Análisis de calidad: Se evalúa la calidad del software existente y se identifican áreas de mejora.
- b) Identificación de mejoras: Se definen las mejoras a realizar, como optimizaciones de rendimiento, mejoras de usabilidad o incremento de la eficiencia.

- c) Implementación de mejoras: Se llevan a cabo los cambios necesarios para implementar las mejoras identificadas.
- d) Pruebas de rendimiento: Se realizan pruebas exhaustivas para medir el impacto de las mejoras en el rendimiento del software.
- e) Despliegue de mejoras: Se realiza la instalación y puesta en producción de la versión mejorada del software.

**3. Correctivo:** El mantenimiento correctivo se centra en la corrección de errores y fallas en el software. Las actividades principales son:

- a) Registro y análisis de errores: Se registran y analizan los errores reportados por los usuarios o detectados internamente.
- b) Reproducción de errores: Se intenta reproducir los errores para comprender y aislar las causas subyacentes.
- c) Depuración y corrección: Se realizan las modificaciones necesarias en el código fuente para solucionar los errores identificados.
- d) Pruebas de regresión: Se llevan a cabo pruebas exhaustivas para asegurarse de que las correcciones no hayan introducido nuevos errores.
- e) Despliegue de correcciones: Se instala y pone en producción la versión corregida del software.

**4. Preventivo:** El mantenimiento preventivo tiene como objetivo evitar futuros problemas y mejorar la mantenibilidad del software. Las actividades principales son:

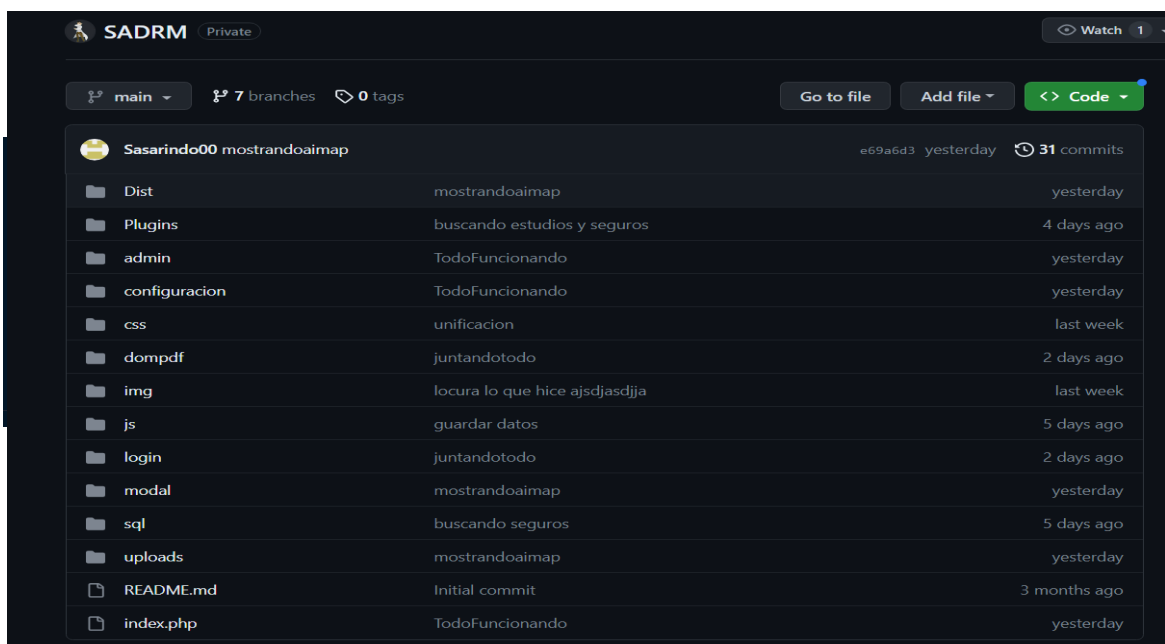
- a) Análisis de riesgos: Se identifican posibles riesgos y problemas potenciales en el software.
- b) Mantenimiento preventivo planificado: Se establecen actividades periódicas de revisión y mejora del software para prevenir problemas futuros.
- c) Optimización de rendimiento: Se realizan ajustes y optimizaciones en el software para mejorar su rendimiento y eficiencia.
- d) Actualizaciones de tecnología: Se actualizan las tecnologías subyacentes utilizadas por el software para mantenerlo compatible y seguro.
- e) Mejoras en la documentación: Se actualiza la documentación técnica y de usuario para reflejar los cambios realizados y facilitar el mantenimiento futuro.

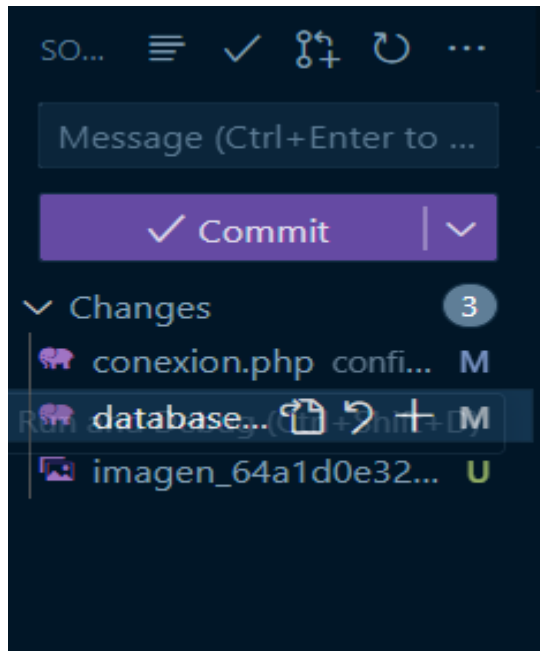
La fecha aproximada para realizar estos mantenimientos es entre agosto y septiembre.

## 5.3 Control de Versiones

Es fundamental que cualquier software, sin importar su tamaño, cuente con un sistema de control de versiones para garantizar la confianza y el seguimiento de los cambios realizados. Esto resulta especialmente útil para recuperarnos de posibles errores irreversibles que puedan surgir en el software.

En este proyecto, hemos optado por utilizar GitHub como nuestra herramienta de control de versiones. GitHub nos permite gestionar los repositorios de Git, donde almacenamos y controlamos el código fuente del proyecto. Mediante el uso de ramas, podemos trabajar en diferentes líneas de desarrollo de manera simultánea. Al realizar commits, registramos cambios específicos, y a través de los pull requests, fusionamos los cambios realizados en ramas secundarias con la rama principal del proyecto. GitHub mantiene un historial completo de todos los commits realizados, lo que facilita la colaboración y el seguimiento de versiones anteriores del proyecto.





Default branch			
main	Updated yesterday by Sasarindo00	Default	
Your branches			
DV2	Updated 3 days ago by Sasarindo00	9   8	New pull request
DV	Updated 2 weeks ago by diegosvs01	23   40	New pull request
Active branches			
DV2	Updated 3 days ago by Sasarindo00	9   8	New pull request
MS2	Updated 3 days ago by Sasarindo00	14   11	New pull request
MS	Updated last week by Sasarindo00	24   21	New pull request
JB2	Updated last week by Vahdrak	23   63	New pull request
DV	Updated 2 weeks ago by diegosvs01	23   40	New pull request
<a href="#">View more active branches &gt;</a>			



## 5.4 Encuesta de satisfacción del cliente

### Encuesta de satisfacción del cliente

Formulario para que el cliente exprese su satisfacción o descontento con el sistema.

Nombre \*

Tu respuesta

1-Es nuestro sistema fácil de utilizar?

- ☐ Si
- ☐ Un Poco
- ☐ No

2-Es fácil de recordar como usar el sistema?

- ☐ Si
- ☐ Un poco
- ☐ No

3-El sistema cumple con las funcionalidades solicitadas?

- ☐ Si
- ☐ Un poco
- ☐ No

4-Esta satisfecho con la apariencia del sistema?

- ☐ Si
- ☐ Un poco
- ☐ No

5-Esta usted conforme con el sistema?

- ☐ Si
- ☐ Un poco
- ☐ No

6-El sistema le ayudo a agilizar sus actividades?

- ☐ Si
- ☐ Un poco
- ☐ No

7-Se traba el sistema al utilizarlo?

- ☐ Si
- ☐ Un poco
- ☐ No

8-Que calificación le pondría al sistema?

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5

9-Fue agradable trabajar con nuestro equipo?

☐ Si

☐ No

10-Que otras mejoras le gustaría ver en su sistema para las versiones futuras?



Tu respuesta

Otros aportes o comentarios

Tu respuesta

### 5.4.1 Resultados de la encuesta

3 respuestas

 [Vínculo a Hojas de cálculo](#) 

Se aceptan respuestas ☒

Resumen      Pregunta      Individual

Nombre

3 respuestas

Ing. Elena Duarte Cáceres

Esteban Tadeo Barreto Jara

Arturo

1-Es nuestro sistema fácil de utilizar?

 Copiar

3 respuestas

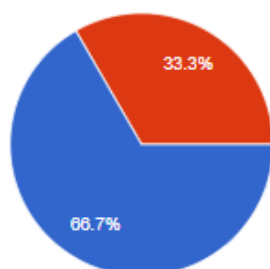


● Si  
● Un Poco  
● No

2-Es fácil de recordar como usar el sistema?

 Copiar

3 respuestas

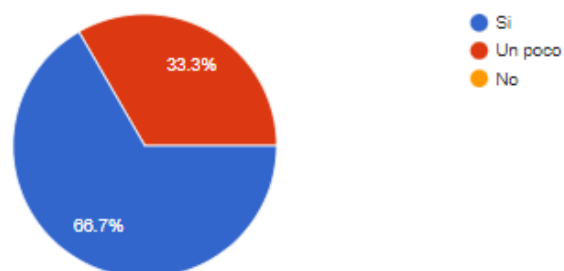


● Si  
● Un poco  
● No

3-El sistema cumple con las funcionalidades solicitadas?

 Copiar

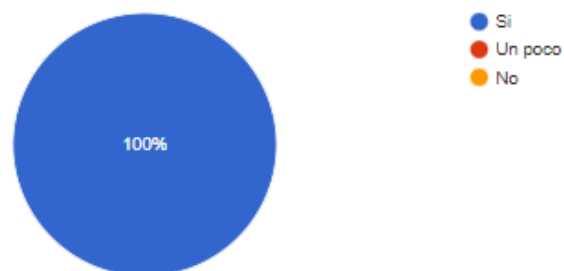
3 respuestas



4-Esta satisfecho con la apariencia del sistema?

 Copiar

3 respuestas



5-Esta usted conforme con el sistema?

 Copiar

3 respuestas



- Si
- Un poco
- No

6-El sistema le ayudo a agilizar sus actividades?

 Copiar

3 respuestas

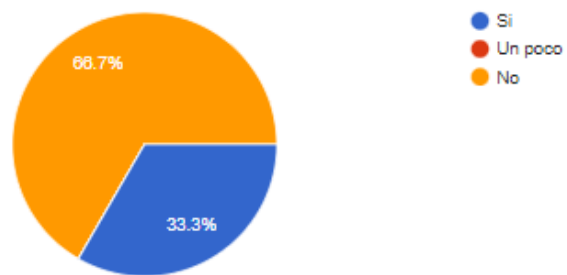


- Si
- Un poco
- No

7-Se traba el sistema al utilizarlo?

 Copiar

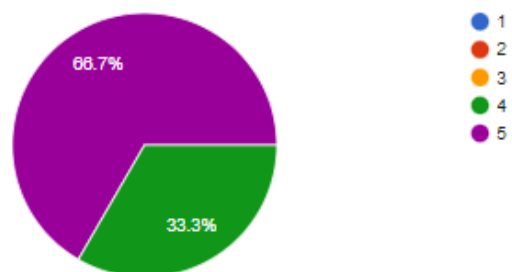
3 respuestas



8-Que calificación le pondría al sistema?

 Copiar

3 respuestas





9-Fue agradable trabajar con nuestro equipo?

 Copiar

3 respuestas



● Si  
● No

10-Que otras mejoras le gustaría ver en su sistema para las versiones futuras?

2 respuestas

Extender para tener agendas separadas para otros servicios

Que pueda ser aplicable a todas las sucursales y a todos los estudios

Otros aportes o comentarios

2 respuestas

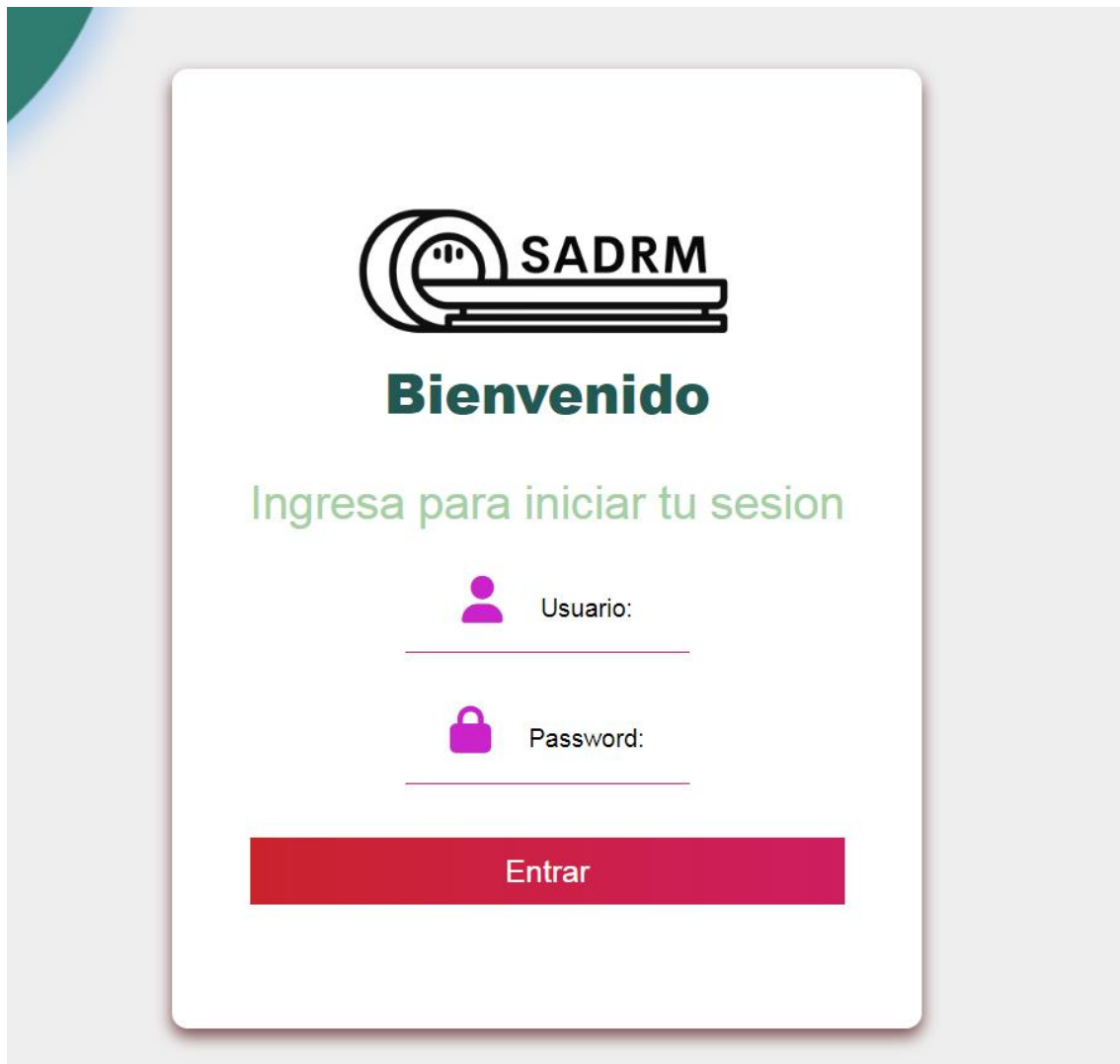
Mejorar comunicación con el cliente

Incluir tutorial, para las personas ajenas a la informática

## 5.5 Evidencia de logros de todos los Requisitos Funcionales.

### 5.5.1 Autenticación e inicio de sesión por Gestión de Roles:

Implementamos un sistema de inicio de sesión seguro que permite a los usuarios acceder al aplicativo según sus roles. Se asignaron roles específicos para gestionar la seguridad y la autenticación de los usuarios, asegurando un acceso adecuado a las funcionalidades del sistema.





### 5.5.2 Registro de Usuarios:

Implementamos una interfaz de usuario intuitiva que permite al administrador del sistema completar un formulario con los datos necesarios para registrar un nuevo usuario en el sistema. Validamos los datos ingresados para garantizar su exactitud, almacenamos de forma segura la información en la base de datos. Realizamos pruebas para garantizar la funcionalidad y confiabilidad del sistema.

## Lista de usuarios

[+ Crear Usuario](#)

Show  entries Search:

Avatar	Nombre	Apellido	Usuario	Rol	Acciones
	diego	velastiqui	svs	Administrador	<a href="#">Editar</a>   <a href="#">Eliminar</a>
	Diego	1	abc	Administrador	<a href="#">Editar</a>   <a href="#">Eliminar</a>

Showing 1 to 2 of 2 entries Previous **1** Next

### 5.5.4 Registro de estudio o seguro

Implementamos una interfaz que permite registrar los estudios a realizar a los pacientes registrados en el sistema.

## Configuracion de Estudios

[+ Agregar Estudio](#)

Show  entries Search:

ID	Estudios	Acciones
1	resonancia	<a href="#">Editar</a>   <a href="#">Eliminar</a>
2	diagnostico	<a href="#">Editar</a>   <a href="#">Eliminar</a>
3	prueba	<a href="#">Editar</a>   <a href="#">Eliminar</a>
4	Nuevo	<a href="#">Editar</a>   <a href="#">Eliminar</a>

Showing 1 to 4 of 4 entries Previous **1** Next

5.5.5

LISTA DE SEGUROS

+ Agregar Seguro

Search:

Show 10 entries

ID	Seguros	Acciones
1	MEP	<a href="#">Editar</a>   <a href="#">Eliminar</a>
2	TESAI	<a href="#">Editar</a>   <a href="#">Eliminar</a>
3	PARTICULAR IMAP	<a href="#">Editar</a>   <a href="#">Eliminar</a>
4	PRECIOS ESPECIALES	<a href="#">Editar</a>   <a href="#">Eliminar</a>
5	PRECIO HOSPITALARIO	<a href="#">Editar</a>   <a href="#">Eliminar</a>
6	FUNDACIÓN STA. LIBRADA	<a href="#">Editar</a>   <a href="#">Eliminar</a>
7	MEP	<a href="#">Editar</a>   <a href="#">Eliminar</a>

Agendar Citas

July 2023

Semana

Mes

Día

<

>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28 Confirmado	29 Pendiente	30 Confirmado Pendiente	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

### 5.5.6 Generación de informes

#### Informe Estudio Desde: 28-05-2023 | Hasta: 28-07-2023

Fecha	Hora	Importe	Estudio	Seguro	Usuario
29-06-2023	07:30:00	5000000	diagnostico	TESAI	svs
30-06-2023	14:00:00	400	Nuevo	MEP	svs
28-06-2023	15:47:00	8000	prueba	FUNDACIÓN STA. LIBRADA	svs
30-06-2023	11:30:00	50000	resonancia	PRECIO HOSPITALARIO	svs

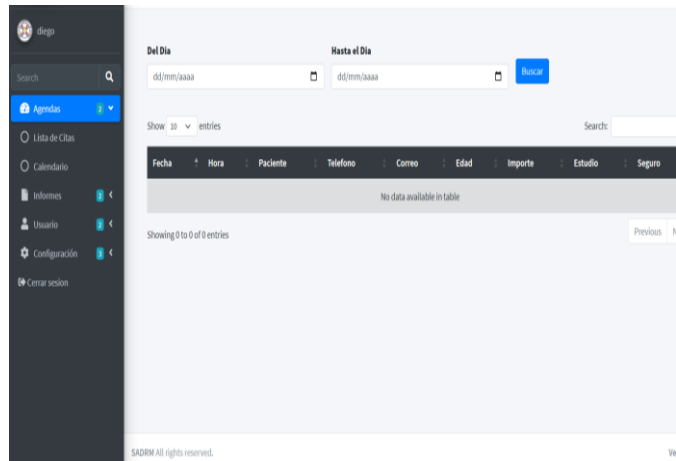
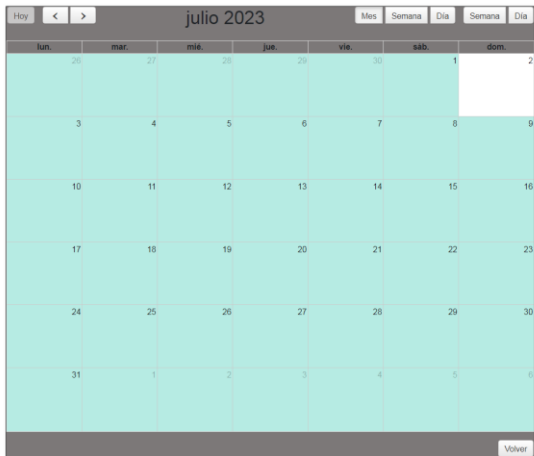
## 5.6 Evidencia de logros de todos los Requisitos No Funcionales.

### 5.6.1 Restricción de contenido

Para cumplir con los requisitos de seguridad y privacidad, hemos implementado un sistema de gestión de roles y permisos en el sistema. Esto nos ha permitido definir diferentes roles de usuario, como administrador, encargado y recepcionista.

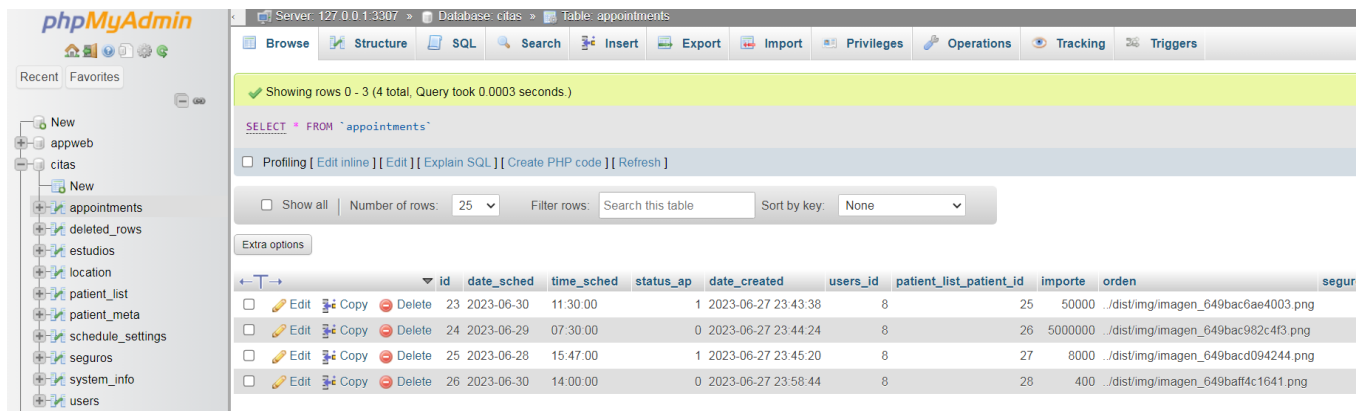
Los usuarios con roles de administrador y encargado tienen acceso completo a todas las funcionalidades y tablas del sistema, incluyendo usuarios, pacientes, estudios e informes. Por otro lado, el rol de recepcionista tiene acceso limitado y solo puede realizar tareas específicas, como agendar citas y ver su propio perfil. Asimismo, los pacientes solo tienen la capacidad de agendar sus citas.

Hemos logrado establecer esta estructura mediante la asignación adecuada de permisos y restricciones de acceso tanto en la base de datos como en el sistema. Esto garantiza que cada usuario tenga acceso únicamente a la información relevante según su rol y nivel de autorización, asegurando así la seguridad y confidencialidad de los datos.

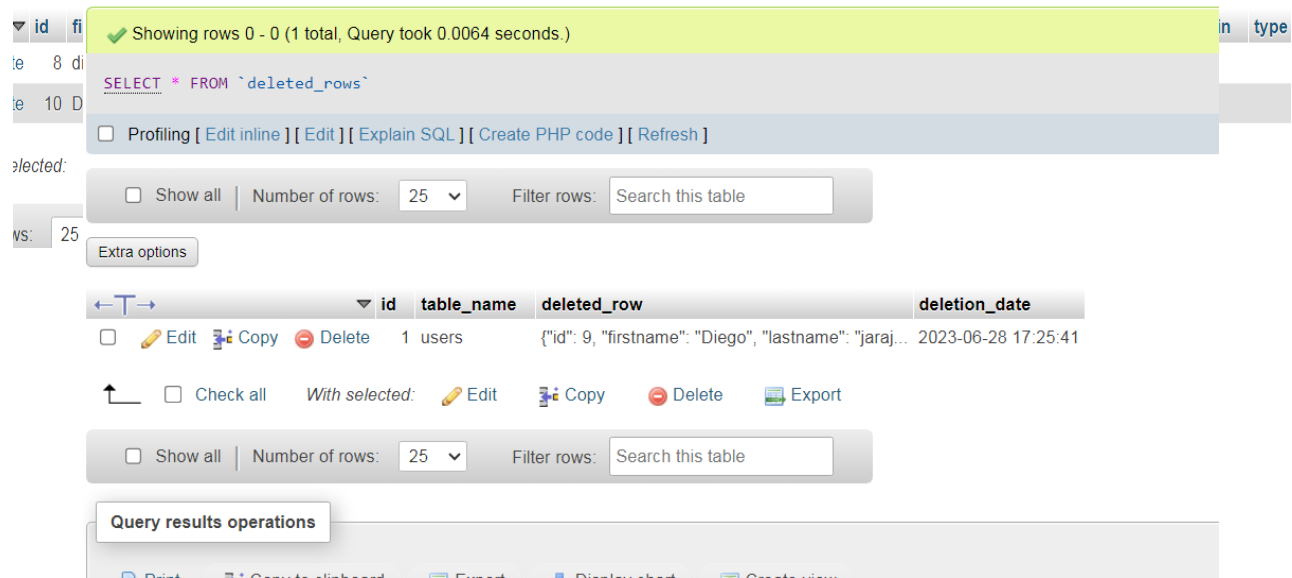


## 5.6.2 Sincronización con la Base de Datos

Hemos logrado implementar una sincronización eficiente con la base de datos MySQL. Para ello, diseñamos una estructura de base de datos que se ajusta a los requisitos del sistema y establecimos una conexión sólida entre el sistema y la base de datos. Llevamos a cabo pruebas exhaustivas para asegurarnos de que la comunicación entre ambos sea correcta, y optimizamos las consultas para garantizar un rendimiento óptimo.

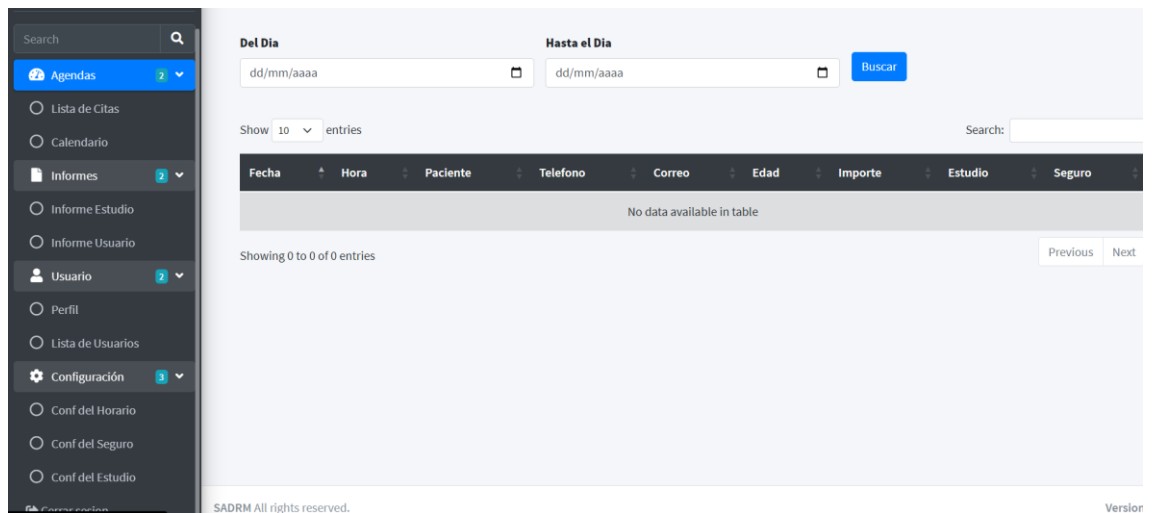


**5.6.3 Seguridad:** Se implementaron medidas de seguridad como la encriptación de datos y la gestión adecuada de permisos. Así también un backup de los archivos que se eliminan



**5.6.4 Usabilidad**

Con el fin de alcanzar un nivel apropiado de usabilidad en el sistema, se llevaron a cabo pruebas de usabilidad y se recopiló retroalimentación de los usuarios para mejorar la interfaz y hacerla más intuitiva y amigable. Como resultado, el sistema ha logrado ser intuitivo y de fácil manejo, lo que permite a los usuarios navegar y utilizar sus funcionalidades de manera eficiente y sin dificultades.



## 5.7 Conclusiones

Nuestro proyecto de sistema de agendamiento para resonancia magnética ha cumplido con todos los requisitos funcionales y no funcionales establecidos, dando una solución completa y eficiente para la empresa encargada de la resonancia magnética. El sistema mejora y optimiza el agendamiento de pacientes para los distintos estudios que disponen, así también mejora la gestión de usuarios, estudios, informes, seguridad y usabilidad. Esta solución jugará un papel fundamental en la optimización de los procedimientos administrativos, al mismo tiempo que mejorará la atención y la experiencia del paciente. Además, permitirá un seguimiento más detallado de los estudios realizados



## Anexo

**HORARIO DE TURNOS IMAP**

Fecha Inicial: 08/05/2023 Fecha final: 08/05/2023

HORA	5:00:00	PACIENTE		Telefono:		edad:	
ESTUDIO		IMPORTE		SEGURO			
HORA	6:30:00	PACIENTE		Telefono:		edad:	
ESTUDIO		IMPORTE		SEGURO			
HORA	7:00:00	PACIENTE		Telefono:		edad:	
ESTUDIO		IMPORTE		SEGURO			
HORA	7:30:00	PACIENTE		Telefono:		edad:	
ESTUDIO		IMPORTE		SEGURO			

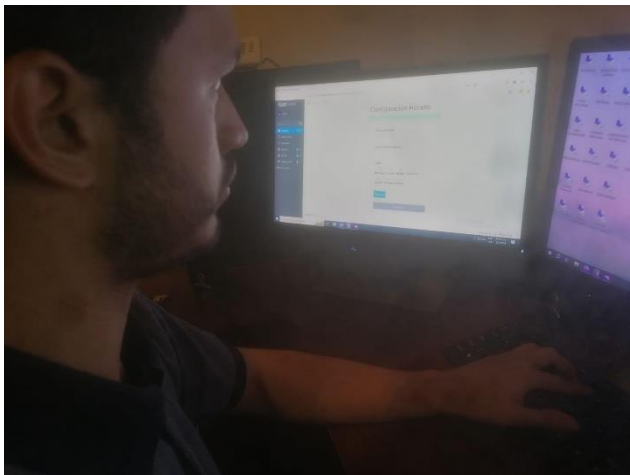
Registro: 1 de 8330

Vista Formulario

Activar Windows  
Vaya a Sistema en el Panel de control para activar Windows.

NUM

Planilla utilizada para agendar anteriormente



Probando funcionalidades con el cliente

## Referencias bibliográficas

- PRESSMAN, Roger S. (2006). “Ingeniería Del Software: Un Enfoque Práctico.” 6ta Edición. Ed. McGraw-Hill.
- Guía SWEBOK”. Alfaomega –Garceta.
- RUMBAUGH, J. Y Otros (1996). “Modelado y Diseño Orientado a Objetos”. Ed. Prentice Hall.
- BOOCH, Grady. (1996). “Análisis y Diseño Orientado a Objetos Con Aplicaciones”. Ed. Addison Wesley.
- CHALEÁT, P., CHAMAY, D. and CHALMOND, M., 2002. *PHP*. Paris: Eyrolles.
- SÁNCHEZ, Salvador; SICILIA, Miguel Angel. “Ingeniería del Software un Enfoque Desde la
- Apache JMeter - Apache JMeterTM. (2019, 22 abril). Apache JMeter - Apache JMeter. <https://jmeter.apache.org/>
- «proyectosagiles,» [En línea]. Available: <https://proyectosagiles.org/que-es-scrum/>. [Último acceso: 2023 Mayo 20].
- G. Lopez, «Blog Gonzalo Lopez,» [En línea]. Available: <https://blogprog.gonzalolopez.es/articulos/pruebas-unitarias-con-phpunit.html>. [Último acceso: 2023 mayo 21].
- pmoinformatica, «pmoinformatica,» 20 febrero 2017. [En línea]. Available: <http://www.pmoinformatica.com/2017/02/pruebas-de-caja-negra-ejemplos.html>. [Último acceso: 2023 20 mayo].
- Y. FERNÁNDEZ, «Xataka,» Xataka, 30 octubre 2019. [En línea]. Available: <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>. [Último acceso: 2023 mayo 17].
- coremedia, «coremedia,» [En línea]. Available: <https://documentation.coremedia.com/cmcc-10/artifacts/2104/webhelp/frontend-en/content/documentationOverview.html>. [Último acceso: 2023 Mayo 18].
- pmoinformatica, «pmoinformatica,» 6 abril 2015. [En línea]. Available: <http://www.pmoinformatica.com/2015/04/estimacion-puntos-funcion-introduccion.html>. [Último acceso: 2023 Mayo 15].
- juntadeandalucia, «juntadeandalucia,» [En línea]. Available: <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/273>. [Último acceso: 2023 mayo 21].