

REPÚBLICA BOLIVARIANA DE VENEZUELA
MINISTERIO DEL PODER POPULAR PARA LA EDUCACIÓN
UNIVERSITARIA
UNIVERSIDAD POLITECNICA TERRITORIAL DE FALCON “ALONSO
GAMERO”
PROGRAMA NACIONAL DE FORMACIÓN EN INFORMÁTICA

*Manual de Sistema; del Sistema Digital de
Registro y Control de Citas Médicas San
Judas Tadeo*

Autores:

Aldama Reyes Jesús David C.I: 30017065
Díaz Primera Ángel Guillermo C.I: 31422335
Castillo Pulgar Leomer José C.I: 31150625
Bracho Lermont Yosmer Emmanuel C.I: 24787868

Santa Ana De Coro, 2025

CONTROL DE VERSIONES (Git)

Versión	Fecha	Descripción
v1.0	1/10/2024	Implementación inicial del sistema con módulos básicos: Inicio de Sesión, Usuarios, Pacientes
v1.1	5/10/2024	Adición del Módulo de secretarías y Consultorios. Mejoras en la gestión de usuarios.
v1.2	20/10/2024	Desarrollo del Módulo de Doctores y Horarios. Integración de validaciones automáticas.
v1.3	23/10/2024	Implementación del Módulo de Reservas. Inclusión de calendario interactivo para citas médicas.
v1.4	14/11/2024	Adición del Módulo de Historial Clínico. Mejoras en la generación de reportes (Excel, PDF)
v1.5	27/11/2024	Optimización de Dashboard para diferentes roles (Administrador, secretaria)
v2.0	11/12/2024	Versión estable del sistema con todos los módulos implementados. Corrección de errores menores.
v2.1	05/01/2025	Mejoras en la interfaz de usuario. Ajustes en la paginación y personalización de columnas.
v2.2	12/01/2025	Actualización del Módulo de Reservas: inclusión de filtros avanzados y reportes personalizados.

DERECHOS DE AUTOR

Esta obra está licenciada bajo CC BY 4.0 

Basado En AdminLTE, Modificado Para Este Proyecto.

El sistema **Digital de Registro y Control de Citas Médicas para el Centro de Salud San Judas Tadeo**, está registrado bajo la licencia Creative Commons, permitiendo el derecho de redistribuir y adaptar la obra, incluso con fines comerciales siempre y cuando se del crédito adecuado al creador original.

Fue desarrollado por Aldama Reyes Jesús David C.I: 30017065 , Díaz Primera Ángel Guillermo C.I: 31422335, Castillo Pulgar Leomer José C.I: 31150625, Bracho Lermont Yosmer Emmanuel C.I: 24787868.

Evaluated por los docentes del Área de Programación del PNFI de la UPTAG – FALCÓN (**pueden colocar los nombres de los docentes evaluadores para darle rigor de validación al mismo**), en el mes (FECHA) del 2025

PRESENTACIÓN

1. DESCRIPCIÓN DEL SISTEMA DE INFORMACIÓN DESARROLLADO
2. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES
 - 2.1. REQUERIMIENTOS FUNCIONALES
 - 2.2. REQUERIMIENTOS NO FUNCIONALES
3. DISEÑO TÉCNICO DEL SISTEMA DE INFORMACIÓN
 - 3.1. ESQUEMA O MODELO DE REQUERIMIENTOS
 - 3.2. SOFTWARE BASE DEL SISTEMA Y PRE-REQUISITOS
 - 3.3. MODELO DE DATOS
 - 3.4. INSTALACIÓN
 - 3.5. CONFIGURACIÓN E INSTALACIÓN

PRESENTACIÓN

Este manual técnico ha sido desarrollado con el objetivo de proporcionar una guía completa, detallada y accesible sobre la definición, diseño, organización, estructura del sistema y solución de problemas del Sistema Digital de Registro y Control de Citas Médicas San Judas Tadeo . Este sistema representa una herramienta innovadora diseñada para optimizar los procesos administrativos y operativos dentro del ámbito de la gestión de citas médicas, asegurando eficiencia, precisión y facilidad de uso.

El contenido de este manual está estructurado de manera que sirva como un recurso valioso tanto para usuarios finales como para administradores de sistemas y personal de soporte técnico. Para los usuarios finales , se incluyen instrucciones claras y paso a paso que les permitirán interactuar con el sistema de manera intuitiva, realizar tareas esenciales como agendar citas, consultar disponibilidad de médicos y gestionar su historial médico. Para los administradores de sistemas , se proporcionan detalles técnicos sobre la arquitectura del sistema, configuración inicial, mantenimiento preventivo y procedimientos de actualización. Finalmente, para el personal de soporte técnico , se ofrece una sección dedicada a la resolución de problemas comunes, diagnóstico de fallos y estrategias para garantizar la continuidad operativa del sistema.

Además de cubrir los aspectos funcionales y técnicos del sistema, este manual también aborda temas relacionados con la seguridad de la información, cumplimiento normativo y buenas prácticas en la gestión de datos médicos. Estos aspectos son fundamentales para garantizar la confidencialidad, integridad y disponibilidad de la información sensible manejada por el sistema.

1. DESCRIPCIÓN DEL SISTEMA DE INFORMACIÓN DESARROLLADO

Propósito del Sistema:

El Sistema Digital de Registro y Control de Citas Médicas tiene como objetivo principal **optimizar la gestión de citas médicas** mediante el uso de tecnología en el **Centro de Salud San Judas Tadeo**, garantizando una administración más eficiente y reduciendo errores administrativos.

El sistema está diseñado para ser utilizado por:

1. **Secretarias:** Encargadas de gestionar módulos relacionados con la programación de citas, manejo de pacientes, consultorios y horarios.
2. **Administradores:** Responsables de la supervisión general del sistema, incluyendo la gestión de usuarios.

El sistema consta de los siguientes nueve módulos principales:

1. **Usuarios:**
 - Permite visualizar un listado completo de los usuarios registrados en el sistema.
2. **Secretarias:**
 - Función para la creación de perfiles de secretarias.
 - Listado de todas las secretarias registradas.
3. **Pacientes:**
 - Permite añadir pacientes al sistema (sin registrarlos como usuarios).
 - Visualización y gestión de un listado de pacientes registrados.
4. **Consultorios:**
 - Registro de nuevos consultorios del centro de salud.
 - Acceso a un listado de consultorios existentes.
5. **Doctores:**
 - Permite registrar doctores (sin incluirlos como usuarios).
 - Listado general de los doctores registrados.
 - Generación de reportes de personal en formato de documento.
6. **Horarios:**
 - Configuración de horarios habituales para doctores y consultorios.
 - Listado para administrar los horarios creados.
7. **Reservas:**
 - Gestión de la creación de citas médicas.
 - Provisión de un listado organizado de citas existentes.
8. **Historial Clínico:**
 - Almacenamiento de información sensible de los pacientes.
 - Posibilidad de listar pacientes y generar documentos detallados con toda la información clínica.

El sistema **no incluye** funcionalidades para la facturación ni el manejo de inventarios médicos. Su uso está enfocado únicamente en la gestión de citas médicas.

2. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

2.1. REQUERIMIENTO FUNCIONAL

ID	Requerimiento Funcional
RF1	El sistema debe permitir a los usuarios registrarse
RF2	El sistema debe permitir a los Usuarios registrar nuevos pacientes en la base de datos.
RF3	El sistema debe permitir a los Usuarios buscar información de pacientes por nombre, identificación u otro Dato otorgado.
RF4	El sistema debe enviar notificaciones por correo electrónico sobre fechas de vencimiento.
RF5	El sistema debe permitir a los Usuarios agendar citas médicas para los pacientes.
RF6	El sistema debe permitir a los Administradores y secretarias reprogramar citas en caso de cambios.
RF7	El sistema debe permitir a los Administradores y secretarias agregar, editar y eliminar información de médicos, horarios y disponibilidad.

2.2. Requerimiento No Funcional

ID	Requerimiento No Funcional
RNF1	El sistema debe ser accesible para los Usuarios las 24 horas del día, los 7 días de la semana, para gestionar citas y pacientes en cualquier momento necesario.
RNF2	El tiempo de respuesta para la búsqueda de pacientes, horarios disponibles o registros históricos debe ser inferior a 2 segundos.
RNF3	El sistema debe soportar al menos 10 usuarios concurrentes, considerando el tamaño del equipo administrativo y posibles futuras expansiones.
RNF4	El sistema debe ser compatible con los navegadores más comunes (Chrome, Firefox, Safari, Edge) para asegurar accesibilidad desde diferentes dispositivos del personal administrativo.
RNF5	Los datos de los pacientes (como información personal, historial médico y citas) deben ser almacenados de forma segura y cifrada, para proteger la privacidad y cumplir con regulaciones de confidencialidad.
RNF6	El sistema debe garantizar una disponibilidad del 99.9%, para evitar interrupciones en el manejo de citas y servicios médicos.
RNF7	La interfaz del sistema debe ser intuitiva y fácil de usar para el personal administrativo, con menús claros y un flujo de trabajo optimizado.

ID	Requerimiento No Funcional
RNF8	El sistema debe ser escalable para soportar el crecimiento futuro, como agregar nuevos médicos, especialidades o ampliaciones de horarios.

3. DISEÑO TÉCNICO DEL SISTEMA DE INFORMACIÓN

3.1 ESQUEMA O MODELO DE REQUERIMIENTOS

Los requerimientos conforman la definición formal del sistema o solución esperada, la especificación de requerimientos:

- Carta Estructurada

Requerimiento 1: El sistema debe permitir a las secretarias y a la administradora iniciar sesión para gestionar y administrar las citas y la información de los pacientes.

Historia de Usuario:

- “Como secretaria/administradora, quiero poder iniciar sesión en el sistema para gestionar y administrar las citas y la información del paciente de forma segura”

Caso de Uso:

- **Nombre:** Inicio de Sesión

- **Actor Principal:** secretaria/Administradora

- **Precondiciones:** La secretaria/administradora debe estar registrada en el sistema.

- **Secuencia de Acciones:**

1. La secretaria/administradora navega a la página de inicio de sesión.

2. Introduce su nombre de usuario/Correo y contraseña.

3. El sistema verifica las credenciales.

4. Si las credenciales son correctas, el sistema redirige a la secretaria/administradora a su panel de control.

5. Si las credenciales son incorrectas, el sistema muestra un mensaje de error.

- **Postcondiciones:** La secretaria/administradora tiene acceso a las funcionalidades del sistema según su rol.

Requerimiento 2: El sistema debe permitir a la secretaria registrar y guardar la información de un paciente.

Historia de Usuario:

- “Como secretaria, quiero poder guardar la información de un paciente en el sistema para que sus datos estén disponibles y actualizados para futuras citas.”

Caso de Uso:

- **Nombre:** Guardar Información del Paciente

- **Actor Principal:** secretaria

- **Precondiciones:** La secretaria debe estar autenticada en el sistema.

- **Secuencia de Acciones:**

1. La secretaria ingresa al sistema.

2. Selecciona la opción de "Registrar Nuevo Paciente".

3. Introduce los datos personales del paciente (nombre, apellido, cedula, dirección, teléfono, fecha de nacimiento, contacto de emergencia, Genero, Correo.).

4. Introduce los datos médicos relevantes (historial médico, alergias, Grupo Sanguíneo)

5. Revisa la información introducida para asegurarse de su exactitud.

6. Guarda la información del paciente en el sistema.

7. El sistema confirma que los datos han sido guardados correctamente y muestra un mensaje de éxito.

- **Postcondiciones:** La información del paciente queda registrada en el sistema y está disponible para futuras consultas y citas.

Requerimiento 3: El sistema debe permitir a la secretaria crear y guardar citas médicas para los pacientes.

Historia de Usuario:

- “Como secretaria, quiero poder crear y guardar una cita para un paciente en el sistema para gestionar eficientemente las citas médicas.”

Caso de Uso:

- **Nombre:** Crear y Guardar Cita Médica

- **Actor Principal:** secretaria.

- **Precondiciones:** * La secretaria debe estar autenticada en el sistema y el paciente debe estar registrado.

- **Secuencia de Acciones:**

- 1. La secretaria/administradora ingresa al sistema.**
 - 2. Selecciona la opción de "Crear Cita".**
 - 3. Introduce el nombre del paciente y verifica su registro en el sistema.**
 - 4. Selecciona el médico y la especialidad requerida.**
 - 5. Selecciona la fecha y hora disponibles para la cita.**
 - 6. Introduce cualquier observación o nota adicional relevante para la cita.**
 - 7. Guarda la cita en el sistema.**
 - 8. El sistema confirma que la cita ha sido guardada correctamente**
- Postcondiciones: La cita queda registrada en el sistema y está disponible para su consulta por parte del paciente y del médico.**

• Casos de uso del sistema:

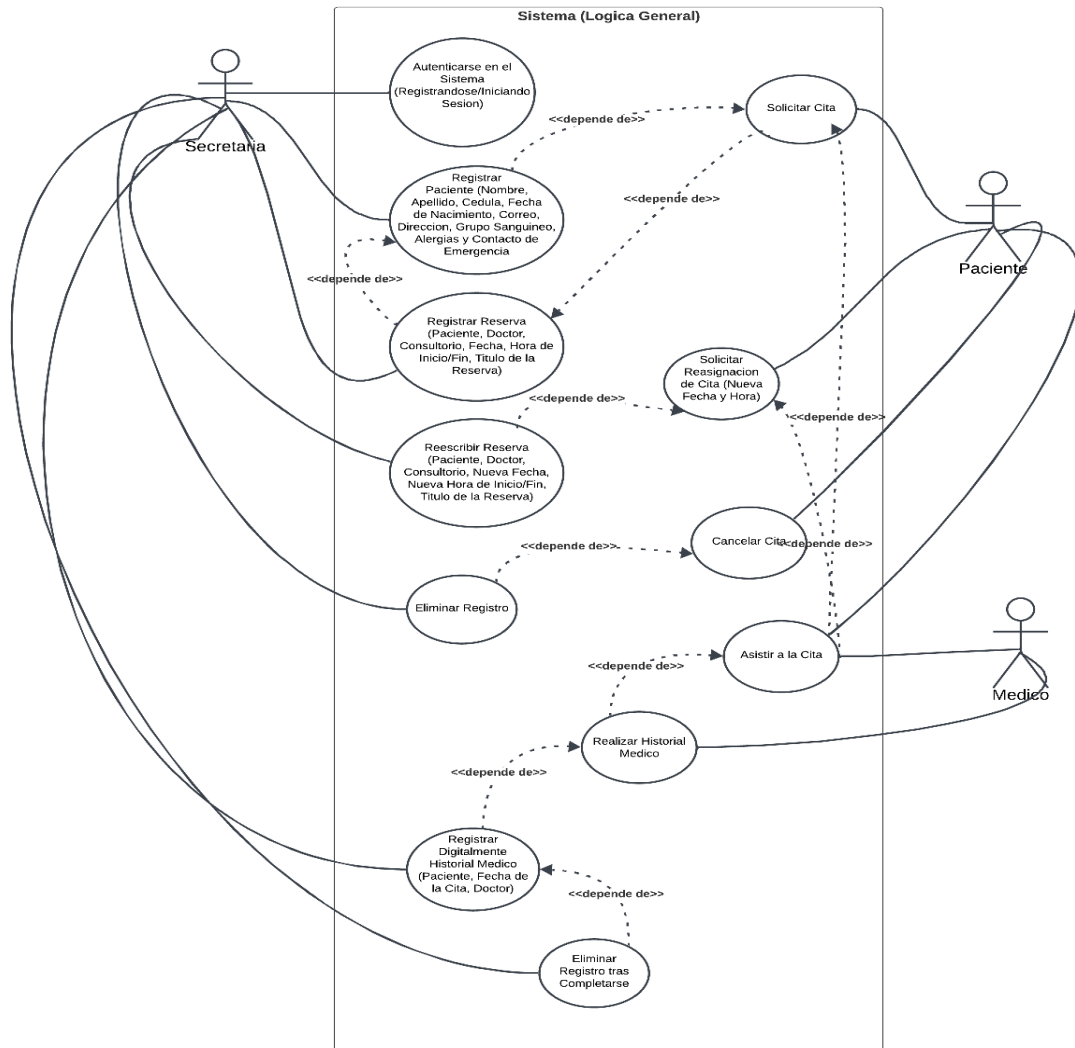


Figura 1. Caso de Uso – Función general del Sistema

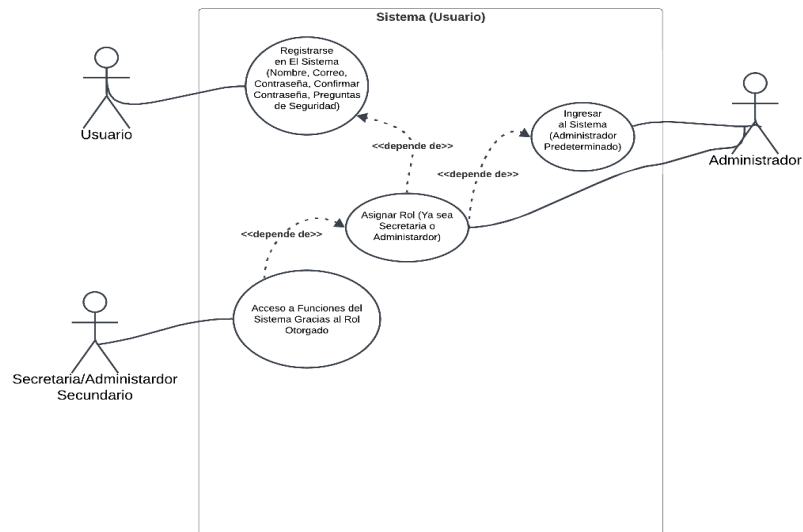


Figura 2. Caso de Uso – Modulo de Usuario

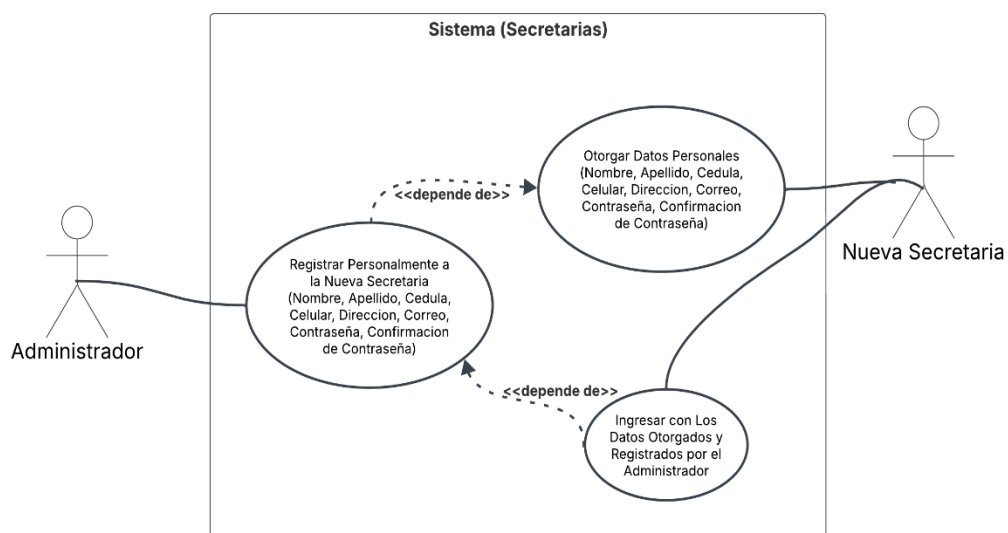


Figura 3. Caso de Uso – Modulo de secretarias

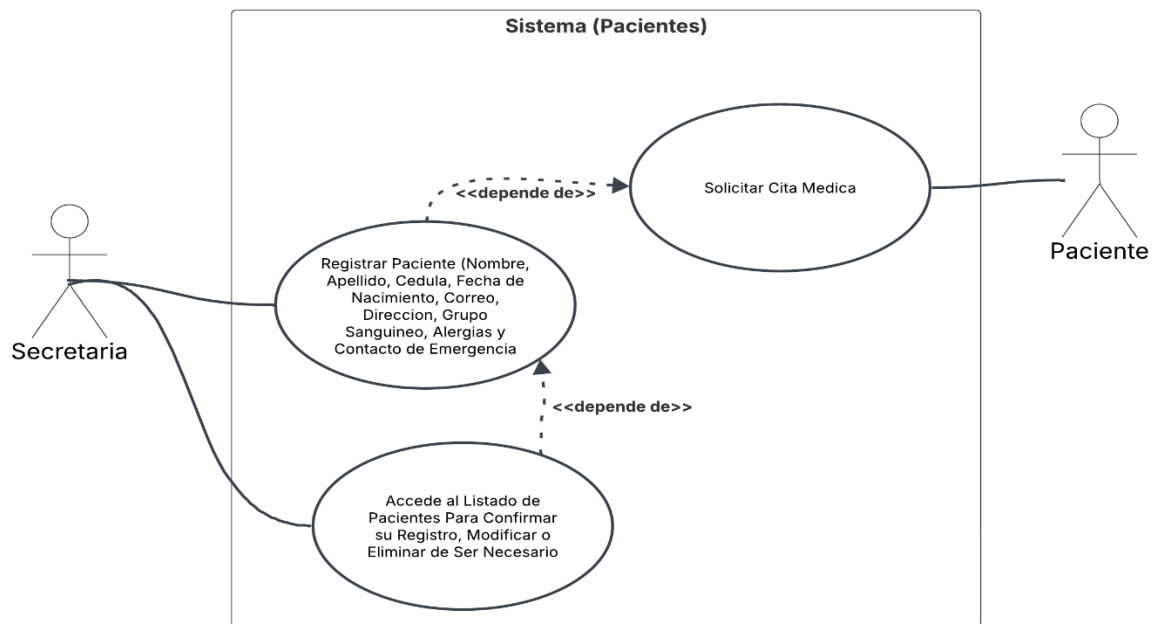


Figura 4. Caso de Uso – Modulo de Pacientes

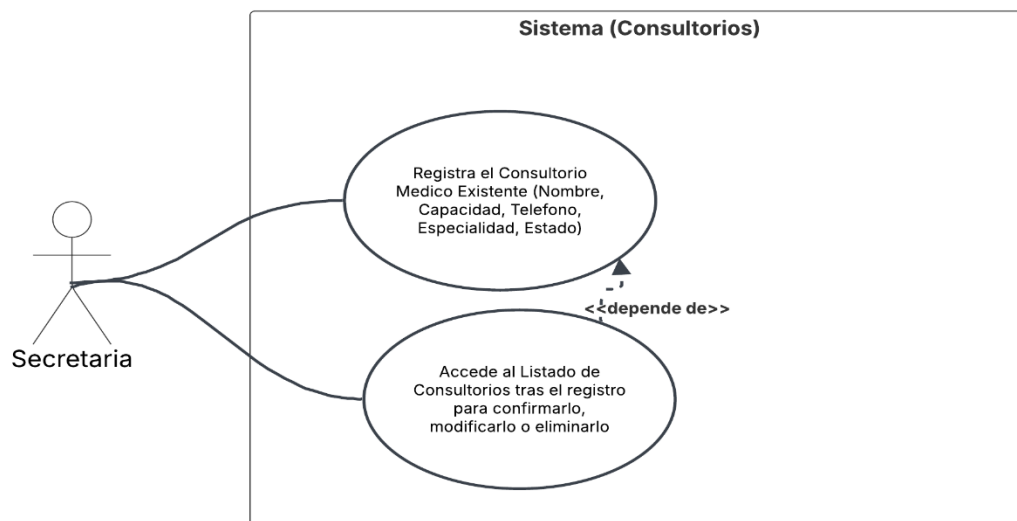


Figura 5. Caso de Uso – Modulo de Consultorios

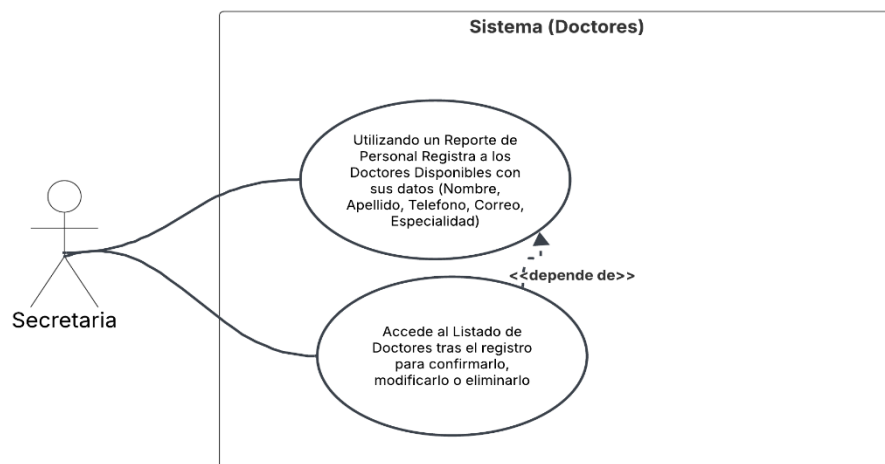


Figura 6. Caso de Uso – Modulo de Doctores

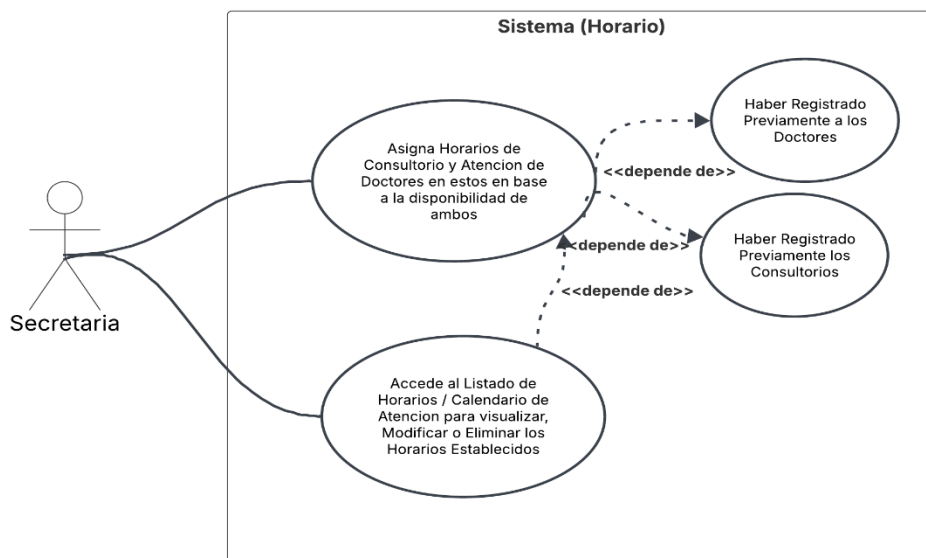


Figura 7. Caso de Uso – Modulo de Horarios

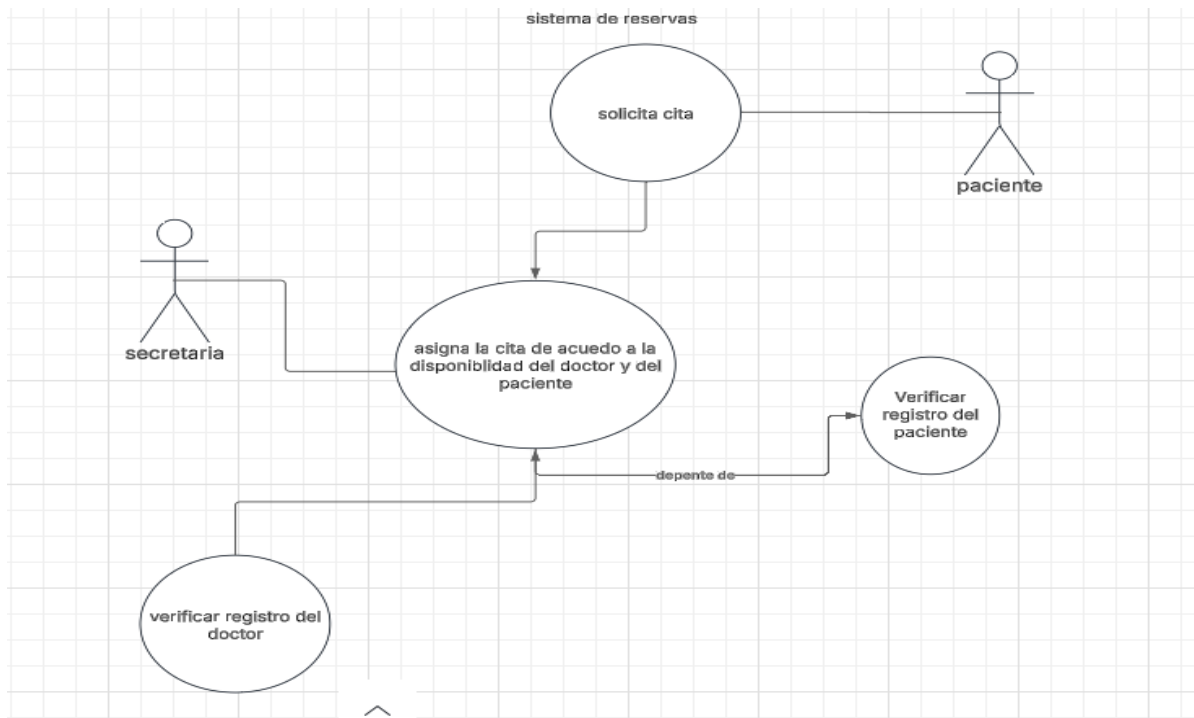


Figura 8. Caso de Uso – Modulo de Reservas

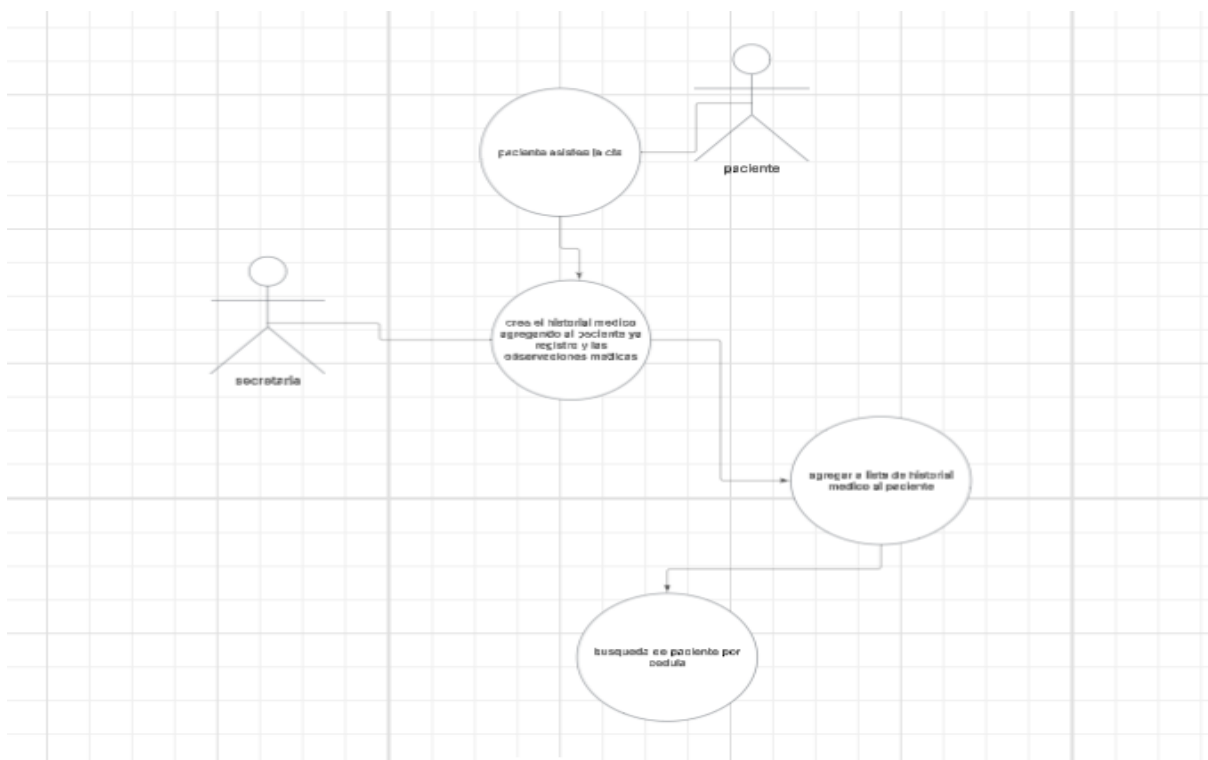


Figura 9. Caso de Uso – Modulo de Historial Medico

3.2 SOFTWARE BASE DEL SISTEMA Y PRE-REQUISITOS

Esta sección describe el software base y los pre-requisitos necesarios para la correcta implementación y operación del sistema Digital de Registro y Control de Citas Médicas San Judas Tadeo . A continuación, se detallan los requerimientos mínimos y recomendados, así como las herramientas y tecnologías utilizadas en su desarrollo.

1. Requerimientos Mínimos y Recomendados de Hardware

Para garantizar un rendimiento óptimo del sistema, se recomienda cumplir con las siguientes especificaciones de hardware:

COMPONENTE	REQUERIMIENTO MÍNIMO	REQUERIMIENTO RECOMENDADO
Disco Duro	10 GB (espacio libre)	20 GB o más
Memoria RAM	2 GB	4 GB o más
CPU	Procesador dual-core 1.8 GHz	Procesador quad-core 2.4 GHz
Tarjeta Gráfica	No es crítico	No es critico

2. Requerimientos Mínimos de Software

A continuación, se enumeran los componentes de software necesarios para ejecutar el sistema:

COMPONENTE	VERSIÓN MÍNIMA	VERSIÓN RECOMENDADA
PHP	7.4	8.0 o superior
MySQL/MariaDB	5.7	8.0 o superior
Composer	1.10	2.x
Node.js (opcional)	14.x	16.x o superior

3. Sistema Operativo de los Servidores

El sistema está diseñado para ser compatible con los siguientes sistemas operativos:

- Servidor de Aplicaciones :
 - Linux (Ubuntu 20.04 LTS o superior, CentOS 7 o superior).
 - Windows Server 2016 o superior.
 - Servidor de Base de Datos :
 - Linux (Ubuntu 20.04 LTS o superior, CentOS 7 o superior).
 - Windows Server 2016 o superior.
-

4. Servidores de Bases de Datos Admitidos

El sistema utiliza bases de datos relacionales para la persistencia de datos. Las opciones admitidas son:

- MySQL : Versión 5.7 o superior.
- MariaDB : Versión 10.3 o superior.

5. Servidores de Aplicación

El sistema requiere un servidor de aplicaciones que ejecute el backend desarrollado en Laravel. Las opciones admitidas son:

- Apache : Versión 2.4 o superior.
- Nginx : Versión 1.18 o superior.

Ambos servidores deben estar configurados para apuntar al directorio **public/** del proyecto Laravel.

6. Navegadores Compatibles y su Versión

El sistema está diseñado para ser compatible con los siguientes navegadores web:

Navegador	Versión Mínima
Google Chrome	90
Mozilla Firefox	88
Microsoft Edge	90
Safari	14

8. Lenguajes de Programación Utilizados en el Desarrollo

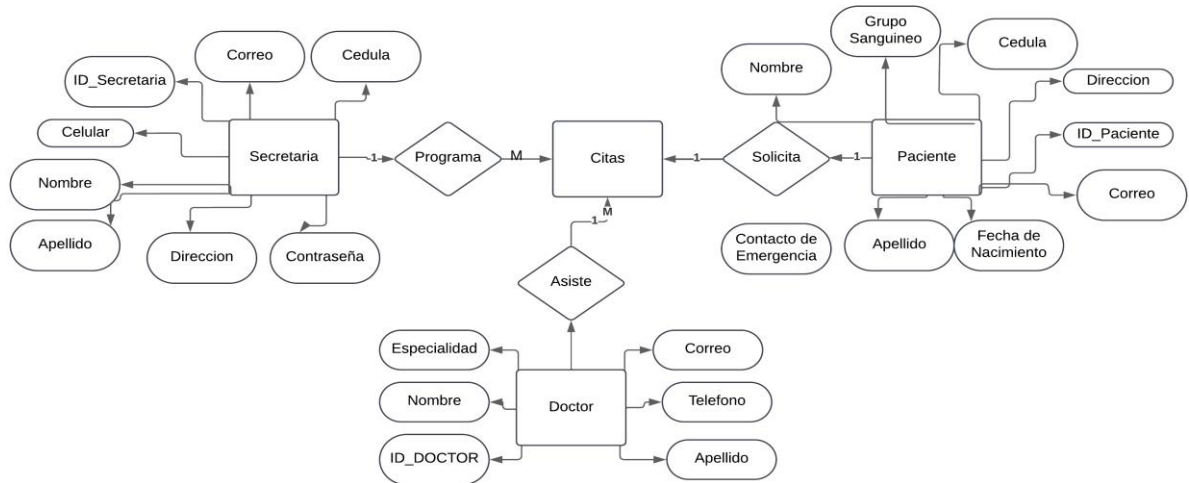
El sistema fue desarrollado utilizando los siguientes lenguajes de programación y tecnologías:

- PHP : Versión 8.0 (Framework Laravel 9.x).
- JavaScript : Bibliotecas como jQuery para interactividad en el frontend.
- HTML/CSS : Bootstrap 5 para el diseño responsivo de la interfaz de usuario.
- SQL : Para consultas y gestión de la base de datos MySQL/MariaDB.

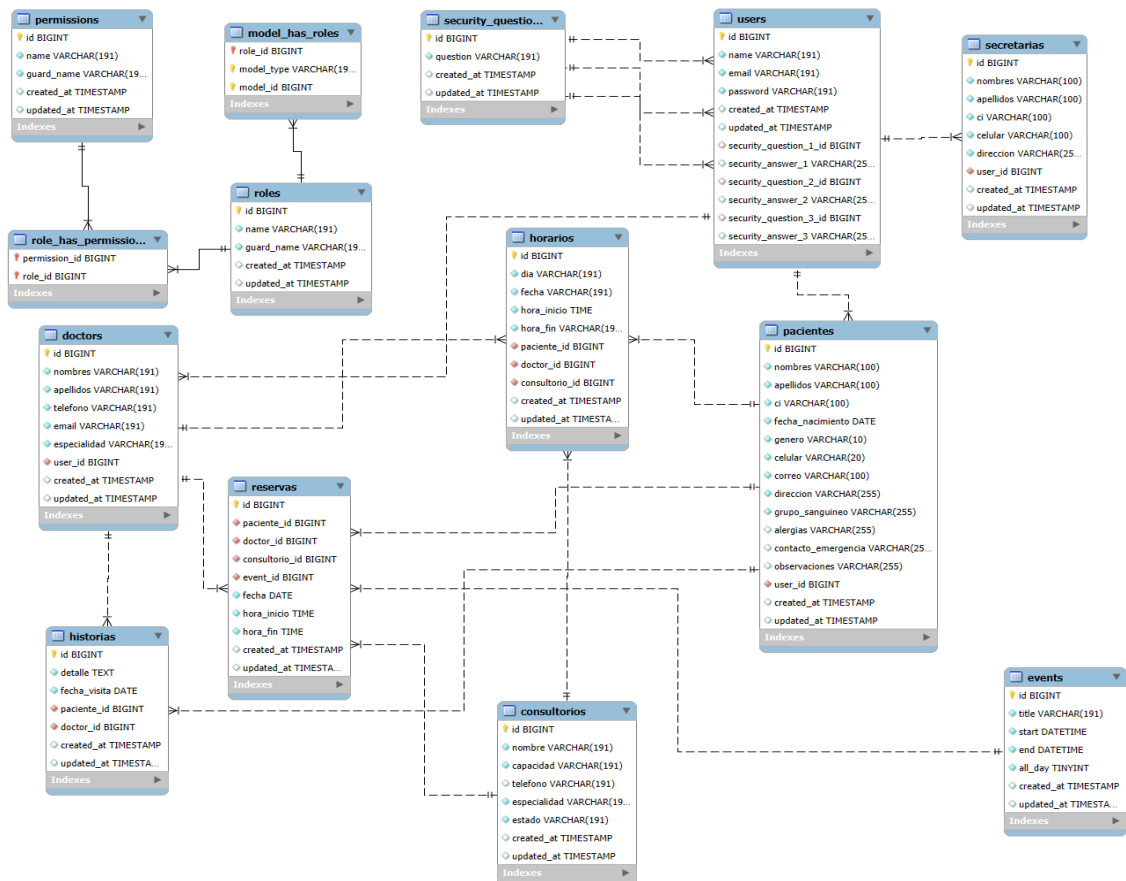
3.3. MODELO DE DATOS

Esta sección debe incluir el modelo de datos tanto lógico como físico

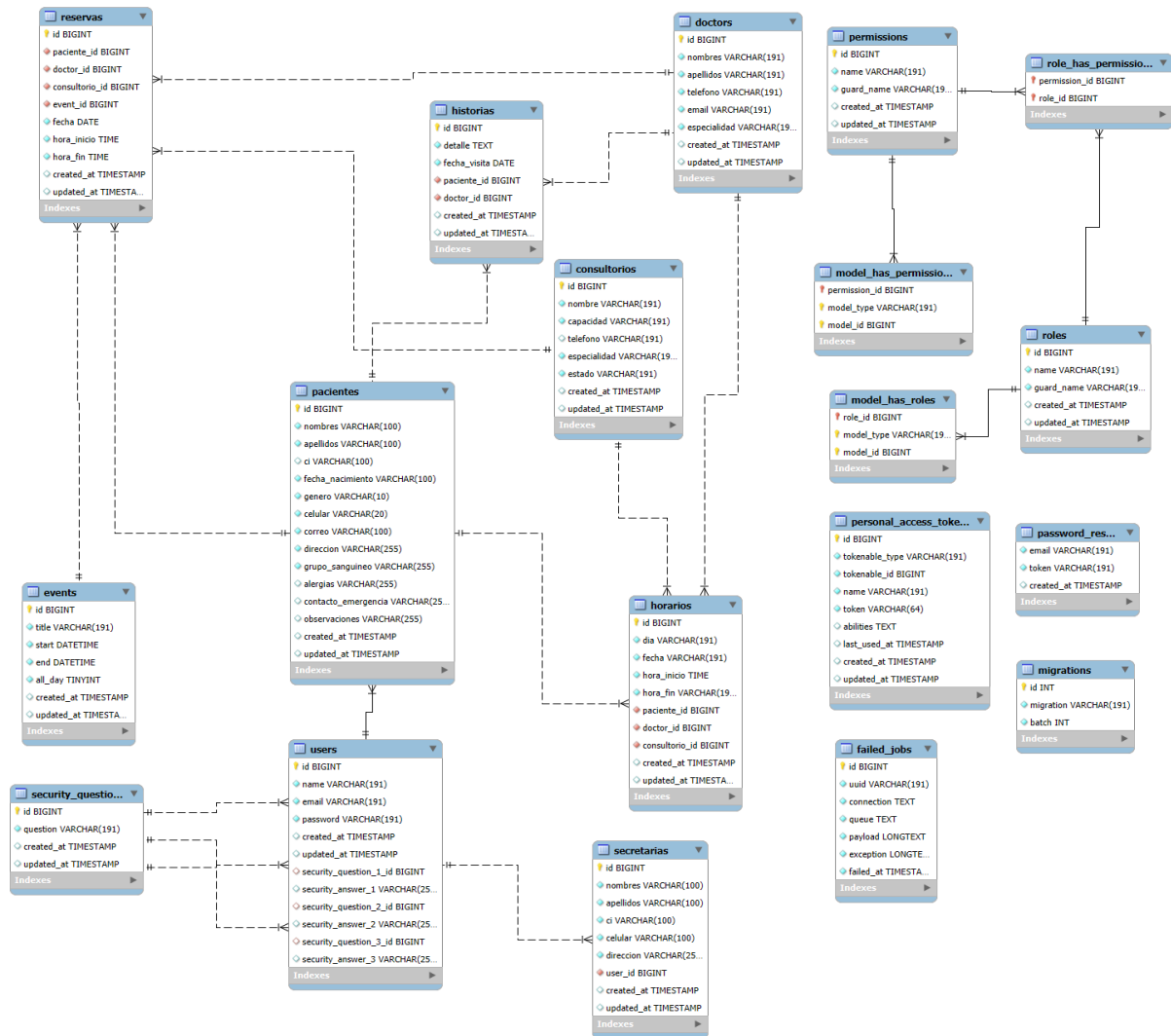
- **Modelo entidad-relación:**



- **Modelo De Clase:**



• Modelo Relacional:



3.4 INSTALACIÓN

Este numeral describe los pasos y requisitos necesarios para la instalación del sistema Digital de Registro y Control de Citas Médicas San Judas Tadeo , asegurando su correcta implementación en un entorno de producción. A continuación, se detallan los aspectos clave relacionados con la instalación:

Pre-requisitos de instalación del sistema

Para instalar y ejecutar el sistema, es necesario cumplir con los siguientes pre-requisitos:

1. Entorno de desarrollo:
 - Servidor web compatible (Apache).
 - PHP versión 8.0 o superior.
 - MySQL versión 5.7 o superior (u otro motor de base de datos compatible).
 - Composer instalado para gestionar dependencias de PHP.
2. Herramientas adicionales:
 - Git para clonar el repositorio del proyecto.
 - Node.js (opcional, si se requiere compilar assets como CSS/JS).
 - Acceso a la línea de comandos para ejecutar comandos de Laravel.
3. Configuración del entorno:
 - Archivo .env correctamente configurado con las variables de conexión a la base de datos y otros parámetros específicos del entorno.

Script de instalación de la Base de Datos

El sistema utiliza migraciones de Laravel para crear y gestionar la estructura de la base de datos. Los scripts de instalación incluyen:

1. Migraciones:

- Las migraciones definen la estructura de las tablas necesarias para el sistema (usuarios, pacientes, doctores, consultorios, horarios, citas, historiales clínicos, etc.).
- Comando para ejecutar las migraciones:

```
php artisan migrate
```

Seeders:

- Se utilizan seeders para cargar datos iniciales, como preguntas de seguridad y roles de usuario.
- Comandos para ejecutar los seeders:

```
php artisan db:seed
```

```
php artisan db:seed --class=SecurityQuestionsSeeder
```

Frameworks y estándares

El sistema está construido utilizando los siguientes frameworks y estándares:

1. Framework principal:

- Laravel : Versión 9.x o superior. Proporciona herramientas robustas para el desarrollo de aplicaciones web, como autenticación, rutas, migraciones y validaciones.

2. Bibliotecas adicionales:

- Bootstrap : Framework CSS utilizado para el diseño de la interfaz de usuario.
- jQuery : Biblioteca JavaScript para mejorar la interactividad del sistema.

3. Estándares seguidos:

- PSR-4 : Estándar de autoloading de clases para PHP.
- MVC (Modelo-Vista-Controlador) : Arquitectura utilizada para organizar el código del sistema.

Fuentes de instalación de los componentes de software

Los componentes de software necesarios para el sistema se pueden obtener de las siguientes fuentes:

1. Repositorio del proyecto:

- El código fuente del sistema está alojado en GitHub. Para clonar el repositorio, ejecutar:

```
git clone https://github.com/LeomerNeroz/Sistema-Reserva-Citas-Medicas-Laravel.git
```

```
cd Sistema-Reserva-Citas-Medicas-Laravel
```

2. Dependencias de Composer:

- Instalar las dependencias del proyecto mediante Composer:

```
composer install
```

Paso a paso para la instalación de los servidores y configuración de los aplicativos

A continuación, se detalla el proceso completo para instalar y configurar el sistema en un entorno de producción:

1. Clonar el repositorio:

```
git clone https://github.com/LeomerNeroz/Sistema-Reserva-Citas-Medicas-Laravel.git
```

```
cd Sistema-Reserva-Citas-Medicas-Laravel
```

2. Instalar dependencias:

```
composer install
```

3. Configurar el archivo **.env**:

- Copiar el archivo **.env.example** a **.env**:

```
cp .env.example .env
```

- Editar el archivo **.env** y configurar las variables de conexión a la base de datos:

```
DB_CONNECTION=mysql
```

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=nombre_de_tu_base_de_datos

DB_USERNAME=tu_usuario_mysql

DB_PASSWORD=tu_contraseña_mysql

4. Generar la clave de aplicación:

```
php artisan key:generate
```

5. Ejecutar migraciones y seeders:

```
php artisan migrate --seed
```

```
php artisan db:seed --class=SecurityQuestionsSeeder
```

6. Configurar el servidor web:

- Configurar un servidor Apache para servir el sistema.
- Asegurarse de que el directorio **public/** del proyecto sea el punto de entrada.

7. Acceder al sistema:

- Abrir el navegador y acceder a la URL del sistema (ejemplo: **http://localhost**).
- Credenciales predeterminadas:
 - Administrador:
 - Email: **admin@admin.com**
 - Contraseña: **12345678**
 - Secretaria:
 - Email: **secretaria@admin.com**
 - Contraseña: **12345678**

8. Configuración adicional (opcional):

- Configurar el envío de correos electrónicos en el archivo. env si se requiere notificaciones por correo.
- Habilitar HTTPS para mayor seguridad.

3.5 CONFIGURACIÓN E INSTALACIÓN

Este numeral describe la configuración e instalación del sistema Digital de Registro y Control de Citas Médicas San Judas Tadeo, detallando los servidores utilizados, su configuración de puertos y distribución para asegurar una correcta puesta en marcha del sistema.

1. Arquitectura del Sistema

El sistema sigue una arquitectura cliente-servidor típica, donde se distinguen los siguientes componentes:

- Servidor de Aplicaciones : Ejecuta la lógica del sistema (backend) y gestiona las solicitudes HTTP.
- Servidor de Base de Datos : Almacena y gestiona toda la información del sistema.
- Cliente (Frontend) : Interfaz web accesible a través de un navegador, que interactúa con el servidor de aplicaciones.

2. Servidores Utilizados

A continuación, se describen los servidores necesarios para la operación del sistema:

2.1 Servidor de Aplicaciones

- Software utilizado : Apache.
- Función : Procesar las solicitudes HTTP y ejecutar el código PHP del sistema.
- Configuración de puertos :
 - Puerto estándar para HTTP: **80**.
 - Puerto estándar para HTTPS: **443** (recomendado para producción).
- Requisitos adicionales :

- Habilitar el módulo **mod_rewrite** en Apache para manejar rutas limpias.
- Configurar el directorio raíz del servidor para apuntar al directorio **public/** del proyecto Laravel.

2.2 Servidor de Base de Datos

- Software utilizado : MySQL.
- Función : Almacenar y gestionar los datos del sistema (usuarios, pacientes, doctores, consultorios, horarios, citas, etc.).
- Configuración de puertos :
 - Puerto estándar para MySQL: **3306**.
- Requisitos adicionales :
 - Crear una base de datos específica para el sistema.
 - Configurar un usuario con permisos de lectura, escritura y modificación sobre la base de datos.

2.3 Servidor de Negocio (Opcional)

- En este caso, el sistema no utiliza un servidor de negocio independiente, ya que toda la lógica de negocio está implementada en el backend de Laravel.

3. Distribución de Componentes

La distribución de los componentes del sistema es la siguiente:

1. Frontend :
 - Desarrollado utilizando HTML, CSS (Bootstrap) y JavaScript (jQuery).
 - Los archivos estáticos (CSS, JS, imágenes) se encuentran en el directorio **public/**.
2. Backend :
 - Implementado en Laravel, siguiendo el patrón MVC (Modelo-Vista-Controlador).
 - El código fuente del backend se encuentra en los directorios **app/**, **routes/** y **database/**.

3. Base de Datos :

- Las tablas y relaciones están definidas mediante migraciones de Laravel.
- Los datos iniciales se cargan mediante seeders.

4. Configuración de Puertos

Para garantizar una comunicación adecuada entre los componentes del sistema, se deben configurar los siguientes puertos:

COMPONENTE	PUERTO	DESCRIPCIÓN
Servidor Web (HTTP)	80	Puerto estándar para acceso sin cifrado.
Servidor Web (HTTPS)	443	Puerto seguro recomendado para producción.
Base de Datos (MySQL)	3306	Puerto estándar para conexiones a MySQL.

5. Pasos para la Configuración e Instalación

A continuación, se detalla el proceso completo para configurar e instalar el sistema:

5.1 Configuración del Servidor Web

1. Instalar Apache en el servidor.
2. Configurar el servidor para que apunte al directorio **public/** del proyecto Laravel.

5.2 Configuración del Servidor de Base de Datos

1. Instalar MySQL en el servidor.
2. Crear una base de datos para el sistema:

3. `CREATE DATABASE nombre_de_tu_base_de_datos;`

4. Crear un usuario con permisos sobre la base de datos:

```
GRANT ALL PRIVILEGES ON nombre_de_tu_base_de_datos.* TO  
'tu_usuario'@'localhost' IDENTIFIED BY 'tu_contraseña';
```

```
FLUSH PRIVILEGES;
```

5.3 Configuración del Proyecto Laravel

1. Clonar el repositorio del proyecto:

```
git clone https://github.com/LeomerNeroz/Sistema-Reserva-Citas-Medicas-Laravel.git
```

```
cd Sistema-Reserva-Citas-Medicas-Laravel
```

2. Instalar dependencias:

```
composer install
```

3. Configurar el archivo **.env**:

```
cp .env.example .env
```

Editar el archivo **.env** y configurar las variables de conexión a la base de datos:

```
DB_CONNECTION=mysql
```

```
DB_HOST=127.0.0.1
```

```
DB_PORT=3306
```

```
DB_DATABASE=nombre_de_tu_base_de_datos
```

```
DB_USERNAME=tu_usuario_mysql
```

```
DB_PASSWORD=tu_contraseña_mysql
```

4. Generar la clave de aplicación:

```
php artisan key:generate
```

5. Ejecutar migraciones y seeders:

```
php artisan migrate --seed
```

```
php artisan db:seed --class=SecurityQuestionsSeeder
```

6. Verificación Final

Una vez completados los pasos anteriores, verificar que el sistema funcione correctamente:

1. Acceder al sistema desde un navegador:
 - URL: **http://localhost**
2. Iniciar sesión con las credenciales predeterminadas:
 - Administrador:
 - Email: **admin@admin.com**
 - Contraseña: **12345678**
 - Secretaria:
 - Email: **secretaria@admin.com**
 - Contraseña: **12345678**
3. Realizar pruebas básicas:
 - Agendar una cita médica.
 - Registrar un nuevo paciente.
 - Generar un reporte en formato Excel o PDF.