

SERVICIO NACIONAL DE APRENDIZAJE - SENA

Regional Sucre

Programa de Formación: Análisis y Desarrollo de Software

INFORME TECNICO

Título del Informe

Desarrollo de una plataforma web eficiente y segura para la gestión de citas médicas y documentación clínica.

Presentado por

Carlos Eduardo Correa Baloco

Ficha No: 2827621

Instructor

Eliseo Vega

Ciudad, Sincelejo-Sucre

Tabla de Contenido

Introducción	3
Resumen	4
Objetivo General.....	5
Objetivos Específicos	5
Alcance del Proyecto	6
Incluye:	6
No incluye:	6
Análisis del Problema.....	7
Planteamiento del Problema	7
Justificación	8
Usuarios Involucrados	8
Diseño del Aplicativo Web.....	9
Arquitectura del Sistema	9
Flujo de Navegación.....	9
Desarrollo e Implementación	10
Tecnologías Utilizadas	10
Estructura del Proyecto	10
Seguridad y Autenticación	11
Implementación de Funcionalidades	11
Pruebas y Validación.....	12
Pruebas de Funcionalidad.....	12
Pruebas de Interfaz	13
Pruebas de Seguridad	13
Despliegue del Aplicativo	14
Instalación y Ejecución	14
Conclusiones y Recomendaciones	15
Resultados Obtenidos	15
Limitaciones	15
Mejoras Futuras	16
Bibliografía y Referencias	17

Introducción

La digitalización en el sector sanitario ha impulsado la necesidad de sistemas eficientes que optimicen la gestión de citas médicas, reduzcan los tiempos de espera y faciliten el acceso a la información clínica. MediCitas es una plataforma web diseñada para automatizar la programación de citas, la gestión de historiales médicos y la facturación electrónica, ofreciendo una solución integral para centros de salud, médicos y pacientes. Su interfaz intuitiva y accesible permite a los usuarios registrar y administrar citas, visualizar tratamientos, recibir notificaciones y descargar documentos médicos en formato PDF.

Para garantizar un sistema robusto y escalable, MediCitas está desarrollado con **Python (Flask)** en el backend, **MySQL** para el almacenamiento de datos estructurados y **Bootstrap 5.3** en el frontend, asegurando una experiencia moderna y adaptable a cualquier dispositivo. Además, la plataforma integra **ReportLab** para la generación de facturas en PDF y **SQLAlchemy** como ORM para la gestión eficiente de la base de datos. Con un enfoque en la seguridad, MediCitas implementa autenticación basada en roles con **Flask-Login**, control de acceso en cada módulo y protección contra inyecciones SQL y ataques CSRF. Este informe detalla la arquitectura del sistema, sus funcionalidades clave, los módulos de seguridad, las pruebas realizadas y las mejoras futuras para seguir optimizando su rendimiento y alcance.

Resumen

MediCitas es una plataforma web diseñada para la gestión eficiente de citas médicas. Permite a pacientes, médicos y administradores programar, modificar y consultar citas en tiempo real. Su interfaz intuitiva facilita el acceso a la información y mejora la experiencia del usuario, reduciendo tiempos de espera y errores en la asignación.

El sistema optimiza la comunicación entre usuarios y agiliza los procesos administrativos en centros de salud. Además, ofrece herramientas para la generación de reportes, control de pagos y gestión de historiales médicos. Su diseño escalable y seguro permite su implementación en diversas instituciones médicas, mejorando la organización y eficiencia del servicio. MediCitas es una plataforma web para la gestión de citas médicas. Facilita la programación y consulta de citas, optimizando la comunicación y organización en centros de salud.

Problema que resuelve: Automatiza la asignación y seguimiento de citas, evitando problemas de organización y optimizando el tiempo.

Usuarios principales: Administradores, médicos y pacientes.

Funcionalidades principales:

- Registro y autenticación de usuarios con roles.
- Programación y gestión de citas.
- Generación de facturas en formato PDF.
- Notificaciones y control de pagos.

Objetivo General

Desarrollar una plataforma web eficiente y segura para la gestión de citas médicas y documentación clínica.

Objetivos Específicos

- Diseñar los instrumentos de recolección de datos
- Analizar los requerimientos del software
- Desarrollar un sistema de autenticación por roles.
- Desarrollar una interfaz responsiva.
- Desarrollar las funcionalidades para la gestión de Pacientes, Médicos, Citas y Tratamientos
- Desarrollar funcionalidades para la generación de reportes y facturas en PDF.
- Garantizar la protección de datos mediante sesiones protegidas y validaciones de seguridad.
- Desarrollar funciones de notificación para mejorar la interacción con los usuarios.
- Facilitar la exportación de historiales médicos en PDF.
- Automatizar la facturación médica y permitir su descarga por parte del paciente.

Alcance del Proyecto

El sistema "MediCitas Plus" es una plataforma web de gestión médica diseñada para mejorar la administración de citas, pacientes, facturación y tratamiento médico.

Incluye:

1. Registro de Usuarios y Autenticación
 - Registro y autenticación de administradores, médicos y pacientes.
 - Gestión de permisos y roles con control de acceso seguro.
2. Gestión de Citas y Pacientes
 - Creación, edición y eliminación de citas médicas.
 - Visualización del historial de citas.
 - Administración de disponibilidad de los médicos.
 - Confirmaciones y cambios de estado de citas.
3. Historial Clínico y Tratamiento Médico
 - Registro y consulta de historias clínicas.
 - Registro de diagnósticos, tratamientos, recetas médicas y notas adicionales.
 - Exportación de historias clínicas en formato PDF.
4. Panel de Administración
 - Panel exclusivo para administradores y médicos.
 - Gestión de usuarios (pacientes y médicos).
 - Supervisión de citas, facturación y tratamientos.
 - Control y monitoreo de actividades médicas y administrativas.
5. Creación de Facturas y Reportes en PDF
 - Generación de facturas médicas detalladas.
 - Exportación de facturas en formato PDF.
 - Historial de pagos y estado de facturación.
6. Sistema de Notificaciones
 - Notificaciones sobre citas agendadas, confirmaciones y recordatorios.
 - Alertas de actualización de tratamientos y cambios en la facturación.

No incluye:

1. Pasarelas de pago: No se implementará pago en línea.
2. Aplicación móvil: Solo versión web.

Análisis del Problema

Planteamiento del Problema

Los centros médicos en Colombia y a nivel internacional enfrentan dificultades en la gestión de citas y facturación debido a la falta de automatización. Esto genera demoras, errores en la asignación de turnos y problemas en la comunicación entre pacientes y médicos, afectando la calidad del servicio.

Muchos centros aún dependen de procesos manuales, lo que provoca sobrecarga en el personal administrativo y dificulta el acceso rápido a los historiales clínicos. Además, la falta de un sistema eficiente para la facturación genera retrasos en los pagos y posibles errores en los cobros.

También existe el riesgo de vulnerabilidades en la seguridad de la información, exponiendo los datos de los pacientes a accesos no autorizados.

Ante esta problemática, surge la necesidad de desarrollar una plataforma web que optimice la gestión de citas, historiales clínicos y facturación, garantizando eficiencia, seguridad y una mejor experiencia para pacientes y médicos.

Justificación

Los centros médicos en Colombia y a nivel internacional enfrentan dificultades en la gestión de citas y facturación debido a la falta de automatización. Esto genera demoras, errores manuales y una experiencia insatisfactoria tanto para los pacientes como para el personal administrativo. En el contexto colombiano, esta problemática afecta la eficiencia operativa de clínicas y hospitales, lo que puede traducirse en pérdidas económicas y una menor calidad en la atención médica.

MediCitas surge como una solución tecnológica para optimizar la programación de citas, mejorar la gestión documental y fortalecer la seguridad de la información en el sector salud. La plataforma automatiza procesos administrativos, reduciendo la carga de trabajo y minimizando errores. Además, permite la digitalización de historiales clínicos y facturación electrónica, contribuyendo a una atención más ágil y eficiente.

Con tecnologías como Python (Flask) para el backend, MySQL para la base de datos y Bootstrap 5.3 en la interfaz de usuario, MediCitas ofrece una experiencia fluida y segura. Su sistema de autenticación por roles y medidas de seguridad protege la información sensible de los pacientes.

MediCitas no solo representa un avance tecnológico en la atención médica, sino también una mejora en la calidad del servicio y optimización de recursos en los centros de salud, permitiendo a hospitales y clínicas operar con mayor eficiencia y confiabilidad.

Usuarios Involucrados

- **Administradores:** Controlan la plataforma y gestionan los usuarios.
- **Médicos:** Visualizan y gestionan sus citas y tratamientos.
- **Pacientes:** Solicitan citas, consultan su historial médico y descargan facturas.

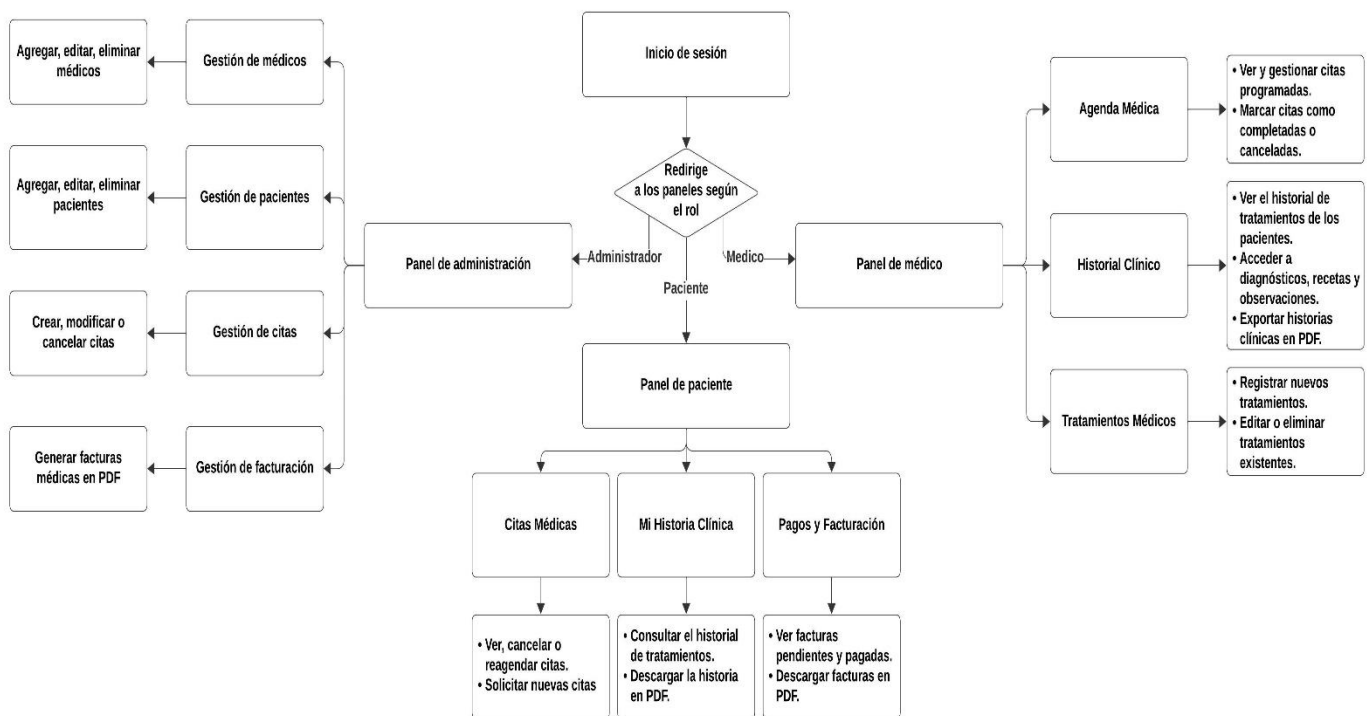
Diseño del Aplicativo Web

Arquitectura del Sistema

Basado en MVC (Modelo-Vista-Controlador):

- **Modelo:** Lógica de negocio e interacción con la base de datos (SQLAlchemy, clases de modelos).
- **Vista:** Interfaz web desarrollada con HTML, CSS, JS y Bootstrap (plantillas).
- **Controlador:** Maneja las solicitudes del usuario y la lógica de aplicación (routes de Flask).

Flujo de Navegación



Desarrollo e Implementación

Tecnologías Utilizadas

1. **Backend:** Python (Flask)
2. **Frontend:** HTML, CSS, JavaScript, Bootstrap 5.3
3. **Base de Datos:** MySQL
4. **Control de Versiones:** Git/GitHub

Estructura del Proyecto

```
/Proyecto_Empresa: (Directorio raíz)
/static: (Archivos estáticos: CSS, JS, imágenes)
/templates: (Plantillas HTML)
    /pagina_principal: (Plantillas para la página principal)
    /templates_login: (Plantillas de login/recuperación)
    /templates_dashboard: (Plantillas del dashboard)
    /templates_admin: (Plantillas del Admin)
        /inicio_admin.html (Dashboard principal del Admin)
        /medico: (Plantillas de gestión de médicos)
        /paciente: (Plantillas de gestión de pacientes)
        /cita: (Plantillas de gestión de citas)
        /factura: (Plantillas de gestión de facturas)
        /perfil_admin.html (Perfil del Administrador)
    /templates_medicos: (Plantillas para médicos)
        /inicio_medico.html (Dashboard principal del Médico)
        /agenda_medica: (Plantillas de la agenda)
        /historia_medica.html (Gestión de historial médico)
        /tratamiento: (Plantillas de tratamientos)
        /perfil_medico.html (Perfil del Médico)
    /templates_paciente: (Plantillas para pacientes)
        /inicio_paciente.html (Dashboard principal del Paciente)
        /mis_citas.html (Listado de citas del Paciente)
        /solicitar_cita.html (Solicitud de cita)
        /mi_historia.html (Historial del Paciente)
        /mis_pagos.html (Pagos del Paciente)
        /mi_perfil_paciente.html (Perfil del Paciente)
model.py: (Definición de la base de datos/Modelos)
app.py: (Archivo principal de la aplicación Flask)
```

Seguridad y Autenticación

- Usar Flask-Login para gestionar la autenticación de usuarios (registro, inicio de sesión, cierre de sesión, protección de rutas).
- Implementar hashing seguro de contraseñas (bcrypt o similar). *No* almacenar contraseñas en texto plano.
- Sanear las entradas de los usuarios para prevenir inyecciones SQL (usar las funciones de SQLAlchemy para la base de datos).
- Activar la protección CSRF (Cross-Site Request Forgery) usando Flask-WTF al usar formularios.

Implementación de Funcionalidades

Registro e inicio de sesión:

- Formularios de registro/inicio de sesión (Flask-WTF).
- Validación de datos de entrada (Flask-WTF).
- Creación/verificación de usuarios en la base de datos (SQLAlchemy).
- Gestión de sesiones de usuario (Flask-Login).

Gestión de clientes, empleados y productos:

- Modelos de base de datos para clientes, empleados y productos (SQLAlchemy).
- Formularios para crear, editar y eliminar registros.
- Rutas para listar, mostrar detalles, crear, actualizar y eliminar registros.

Generación de reportes:

- Consultas a la base de datos para obtener los datos necesarios.
- Uso de librerías como ReportLab para generar PDFs.
- Opciones para filtrar y ordenar los datos en los reportes.

Pruebas y Validación

Pruebas de Funcionalidad

Se realizaron pruebas unitarias exhaustivas en cada función y módulo para verificar su correcto funcionamiento.

- **Inicio de sesión:** Se confirmó que solo los usuarios registrados con credenciales válidas pueden acceder al sistema, y que el intento de inicio de sesión con credenciales inválidas genera el error esperado. **Resultado: Correcto**
- **Creación de médicos:** Se validó que los datos de los médicos se guardan correctamente en la base de datos, y que la validación de los campos obligatorios y formatos se realiza correctamente. **Resultado: Correcto**
- **Creación de pacientes:** Se validó que los datos de los pacientes se guardan correctamente en la base de datos, y que la validación de los campos obligatorios y formatos se realiza correctamente. **Resultado: Correcto**
- **Gestión de citas:** Se verificó la correcta creación, edición y cancelación de citas. **Resultado: Correcto**
- **Gestión de facturas:** Se comprobó que las facturas se generan correctamente con los datos asociados. **Resultado: Correcto**

Pruebas de Interfaz

- Se evaluó la usabilidad de la interfaz, confirmando que es intuitiva y fácil de navegar. **Resultado: Satisfactorio**
- Se probó la responsividad de la aplicación en diferentes dispositivos, asegurando que se visualice correctamente en escritorios, tablets y teléfonos. **Resultado: Satisfactorio**
- Se verificó que todos los elementos de la interfaz se visualizan correctamente. **Resultado: Satisfactorio.**

Pruebas de Seguridad

- Se realizaron pruebas para prevenir inyecciones SQL y se confirmó la protección contra este tipo de ataques. **Resultado: Seguro**
- Se verificó el control de accesos, confirmando que solo los usuarios autorizados pueden acceder a las áreas restringidas. **Resultado: Seguro**
- Se implementaron y probaron medidas contra ataques CSRF. **Resultado: Implementado**
- Se validó que las contraseñas no se almacenan en texto plano. **Resultado: Seguro**
- Se realizaron pruebas de autorización y los roles de usuario funcionan correctamente impidiendo el acceso a recursos no autorizados. **Resultado: Seguro**

Despliegue del Aplicativo

Instalación y Ejecución

Los pasos para instalar las dependencias y ejecutar el sistema en entorno local fueron:

1. Clonar el repositorio Git

- Se clonó el repositorio de la aplicación desde GitHub al equipo local utilizando el comando: `git clone`
https://github.com/Carlos545225/Actividad-2-FLASK/tree/main/Proyecto_Empresa

2. Activar el entorno virtual

- Se activó el entorno virtual para aislar las dependencias del proyecto:

```
python -m venv venv  
source venv/bin/activate # En Linux/macOS  
venv\Scripts\activate # En Windows
```

3. Instalar las dependencias

- Se instalaron las dependencias necesarias utilizando el archivo `requirements.txt`:

```
pip install -r requirements.txt
```

4. Configurar las variables de entorno

- Se configuraron las variables necesarias para la conexión a la base de datos y otras configuraciones del sistema:

```
set FLASK_APP=run.py  
set FLASK_ENV=development  
set  
DATABASE_URL=mysql+pymysql://root:@localhost/db_empresa
```

5. Migrar la base de datos

- Se ejecutaron las migraciones para crear las tablas necesarias:

```
flask db upgrade
```

6. Ejecutar el servidor

- Se inició la aplicación utilizando Flask en Visual Studio Code:

```
flask run
```


La aplicación estará disponible en <http://127.0.0.1:5000/>.

Este proceso asegura que el sistema "MediCitas Plus" esté correctamente instalado y en funcionamiento en un entorno local.

Conclusiones y Recomendaciones

Resultados Obtenidos

MediCitas Plus cumple con los objetivos planteados de gestionar citas médicas, pacientes, médicos y facturas. El sistema proporciona una interfaz web intuitiva para administradores, médicos y pacientes, permitiendo la gestión eficiente de la información y la programación de citas. Las funcionalidades de generación de facturas y reportes contribuyen a una mejor organización y control de los procesos administrativos. La implementación de seguridad y autenticación protege la información sensible de los usuarios.

Limitaciones

Durante el desarrollo, se encontraron las siguientes limitaciones:

- **Gestión de archivos PDF:** La generación de archivos PDF consume recursos del servidor. Si la demanda de generación de facturas aumenta significativamente, se podría considerar el uso de un servicio de generación de PDF en la nube para aliviar la carga del servidor.
- **Falta de pruebas automatizadas exhaustivas:** Debido a limitaciones de tiempo, no se implementaron pruebas automatizadas para todas las funcionalidades. Se recomienda invertir en la creación de pruebas automatizadas para garantizar la calidad y estabilidad del sistema a largo plazo.
- **Integración de pasarelas de pago:** Actualmente, el sistema no integra pasarelas de pago en línea.

Mejoras Futuras

Para futuras versiones del sistema, se proponen las siguientes expansiones:

- **Integración de pagos en línea:** Implementar la integración con pasarelas de pago en línea (Stripe, PayPal) para permitir a los pacientes pagar sus facturas directamente desde el sistema.
- **Desarrollo de una aplicación móvil:** Desarrollar una aplicación móvil para iOS y Android que permita a los pacientes programar citas, ver su historial médico y recibir notificaciones.
- **Implementación de un sistema de recordatorios de citas:** Implementar un sistema de recordatorios de citas por SMS o correo electrónico para reducir la tasa de ausentismo.
- **Mejoras en la generación de reportes:** Implementar más opciones de filtrado y personalización en la generación de reportes.
- **Integración con sistemas de historias clínicas electrónicas (HCE):** Explorar la posibilidad de integrar MediCitas Plus con sistemas de HCE existentes para facilitar el intercambio de información médica.
- **Internacionalización:** Adaptar el sistema para soportar diferentes idiomas y monedas.

Bibliografía y Referencias

Flask Documentation. (n.d.). *Flask (2.3.x)*. Retrieved from <https://flask.palletsprojects.com/>

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python* (2nd ed.). O'Reilly Media.

Real Python. (n.d.). *Flask Tutorials*. Retrieved from <https://realpython.com/>

ReportLab. (n.d.). *ReportLab User Guide*. Retrieved from <https://www.reportlab.com/docs/reportlab-userguide.pdf>

SQLAlchemy Documentation. (n.d.). *SQLAlchemy (2.0.x)*. Retrieved from <https://www.sqlalchemy.org/>

Stack Overflow. (n.d.). Retrieved from <https://stackoverflow.com/>

Bootstrap Documentation. (n.d.). *Bootstrap (v5.3)*. Retrieved from <https://getbootstrap.com/docs/5.3/>