

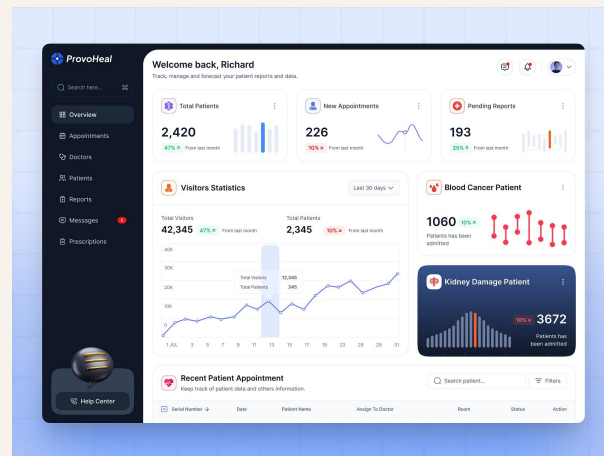
# Sistema de Gestión de Datos Médicos

Rafael Martín

# Sistema de Gestión de Datos Médicos

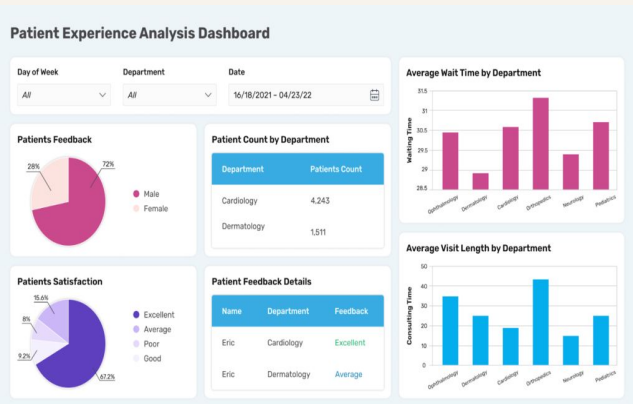
## Solución Integral para Gestión de Datos Médicos con Python & Flask

Bienvenidos a la presentación de nuestro innovador Sistema de Gestión de Datos Médicos, una aplicación web desarrollada con Flask y SQLite. Este proyecto proporciona una solución completa para el manejo de información médica, incluyendo operaciones CRUD, autenticación segura, capacidades de exportación de datos y visualizaciones interactivas. Diseñado pensando en la escalabilidad y la seguridad, nuestro sistema ofrece una interfaz intuitiva para profesionales de la salud.



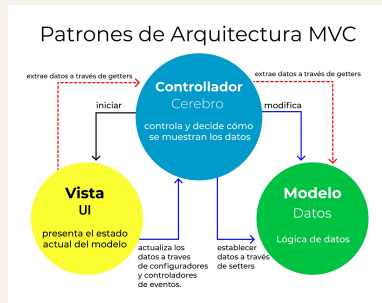
# Descripción General del Proyecto

## Sistema de gestión médica basado en tecnologías web modernas



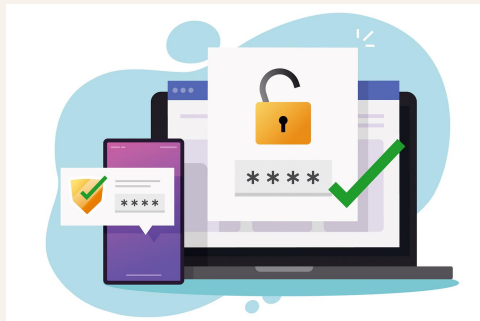
- Desarrollo de una aplicación web compacta y eficiente para gestión médica
- Sistema completo de gestión de datos de pacientes con interfaz intuitiva
- Implementación de operaciones CRUD para manejo integral de registros médicos
- Sistema robusto de autenticación para proteger información sensible de pacientes
- Capacidades avanzadas de exportación de datos en múltiples formatos
- Visualización gráfica interactiva para análisis de datos médicos
- Diseño modular que permite escalabilidad y mantenimiento sencillo

# Objetivos Técnicos



## Desarrollo Web y Base de Datos

- Implementar arquitectura MVC completa utilizando Flask framework
- Desarrollar operaciones CRUD eficientes en base de datos SQLite
- Crear interfaz responsiva moderna con Bootstrap 5



## Seguridad y Autenticación

- Implementar sistema robusto de hash para credenciales
- Establecer autenticación segura de usuarios médicos
- Garantizar protección completa de datos sensibles



## Funcionalidades Avanzadas

- Desarrollar sistema de búsqueda y paginación eficiente
- Implementar exportación de datos en múltiples formatos
- Crear visualizaciones interactivas de datos médicos

# Objetivos

## Desarrollo de Pensamiento

- Promover pensamiento algorítmico aplicado a casos médicos reales
- Fomentar análisis y diseño modular en aplicaciones web
- Desarrollar habilidades de resolución de problemas estructurados

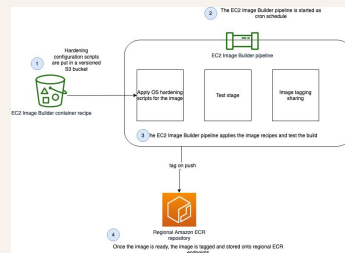
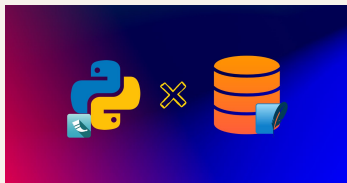
## Gestión y Seguridad

- Implementar protocolos de seguridad para datos médicos sensibles
- Comprender la importancia de la privacidad del paciente
- Aplicar mejores prácticas en el manejo de información clínica

## Habilidades Técnicas

- Dominar el despliegue de aplicaciones en entornos cloud
- Integrar múltiples tecnologías en un sistema unificado
- Desarrollar competencias en arquitectura de sistemas médicos

# Stack Tecnológico



## Backend y Framework

- Python 3 como lenguaje principal de desarrollo
- Flask Framework para estructura MVC y rutas
- SQLite como base de datos relacional ligera

## Frontend y Visualización

- HTML5, CSS3 y Bootstrap para diseño responsivo
- Chart.js para visualización dinámica de datos médicos
- Pandas y Openpyxl para manejo y exportación

## Despliegue y Servidor

- AWS EC2 para hosting en la nube
- Docker para contenerización del ambiente
- Gunicorn como servidor WSGI de producción

## Seguridad y Autenticación

- Hashlib para encriptación segura de contraseñas
- Implementación de autenticación de usuarios médicos
- Protección de datos sensibles de pacientes

# Estructura del Desarrollo - Parte 1

## Diseño del Modelo de Datos

Creación del esquema de base de datos SQLite con tablas para pacientes, historiales médicos, usuarios y relaciones entre entidades siguiendo mejores prácticas de diseño.

## Implementación CRUD en Flask

Desarrollo de rutas y controladores en Flask para operaciones Create, Read, Update y Delete, implementando la lógica de negocio y validaciones necesarias para gestionar registros médicos.

## Vistas y Formularios

Construcción de interfaces de usuario con Bootstrap, incluyendo formularios para ingreso de datos médicos, tablas dinámicas para visualización y componentes interactivos para mejor experiencia de usuario.

# Estructura del Desarrollo - Parte 2

## Módulo 4: Exportación de Datos

Implementación de funcionalidades para exportar registros médicos a formatos CSV y Excel, utilizando Pandas y Openpyxl para procesamiento eficiente de datos.

## Características de Exportación

Desarrollo de filtros personalizables, selección de campos específicos para exportar, y generación de reportes estructurados con formato profesional para uso médico.

## Módulo 5: Autenticación de Usuarios

Implementación de sistema seguro de login con hashlib, gestión de sesiones de usuario, niveles de acceso diferenciados y protección de rutas sensibles.

## Módulo 6: Visualización Estadística

Integración de Chart.js para crear gráficos interactivos de tendencias médicas, distribución de pacientes y métricas clave del sistema de salud.



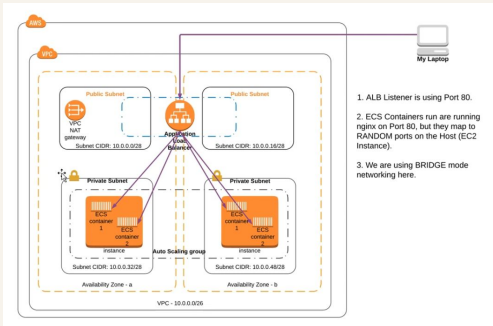
# Estructura del Desarrollo - Parte 3

- Módulo 7: Implementación de búsqueda avanzada por múltiples campos y paginación de resultados para mejor rendimiento
- Módulo 8: Configuración y despliegue del sistema en entorno local y migración a AWS EC2 con configuraciones optimizadas
- Módulo 9: Implementación opcional de contenedores Docker para facilitar el despliegue y la gestión de dependencias
- Configuración de balanceadores de carga para distribuir el tráfico eficientemente entre múltiples instancias
- Implementación de monitoreo y logging para seguimiento del rendimiento del sistema
- Desarrollo de scripts de automatización para facilitar el despliegue y las actualizaciones



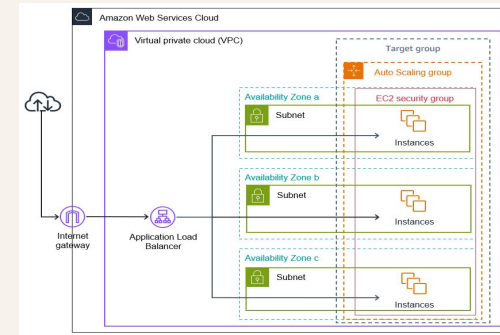
- Documentación detallada del proceso de instalación y configuración para futuros desarrolladores

# Despliegue y Escalabilidad



## Infraestructura y Despliegue

- Implementación en AWS EC2 free tier con configuración optimizada
- Contenerización mediante Docker para gestión eficiente de recursos
- Configuración de Gunicorn como servidor WSGI con Nginx proxy



## Estrategias de Escalabilidad

- Balanceo de carga automático entre múltiples instancias EC2
- Monitorización y ajuste dinámico según demanda del sistema
- Arquitectura modular preparada para expansión futura del servicio

# Evaluación y Entregables

## Entregables Principales

- Implementación completa de todas las funcionalidades del sistema médico
- Documentación detallada del código fuente con comentarios explicativos
- Manual de usuario completo con instrucciones paso a paso

## Criterios de Evaluación

- Pruebas exhaustivas con conjunto de datos médicos de ejemplo
- Calidad y claridad de la documentación técnica presentada
- Despliegue exitoso en AWS EC2 con configuración apropiada

# Consideraciones de Seguridad

## Implementación de medidas robustas para proteger datos sensibles de pacientes

- Encriptación end-to-end de datos médicos sensibles siguiendo estándares HIPAA
- Implementación de hashing seguro de contraseñas usando algoritmos como bcrypt o Argon2
- Sistema de autenticación multinivel con verificación de identidad en dos factores
- Gestión de sesiones con tokens JWT y tiempo de expiración configurable
- Implementación de HTTPS y certificados SSL para comunicación segura
- Registro detallado de accesos y auditoría de modificaciones en datos sensibles
- Sanitización de entradas y protección contra inyecciones SQL y XSS



# Próximos Pasos y Mejoras Futuras

El sistema de gestión médica tiene un amplio potencial de crecimiento y mejora. Las próximas etapas incluirán la integración con sistemas hospitalarios externos, expansión de funcionalidades clínicas avanzadas, y rediseño de la interfaz para optimizar la experiencia del usuario. También se implementarán mejoras significativas en el rendimiento del sistema y se desarrollará una arquitectura más robusta para soportar mayor escalabilidad y volumen de datos.

