



Instituto Politécnico Nacional
Escuela Superior de Cómputo



Ingeniería en software

Integrantes:

Contla Mota Luis Andrés

Escárcega Hernández Steven Arturo

Gómez Sánchez Héctor Rodrigo

García Mayorga Rodrigo

Contenido

Introducción:.....	2
Metodología:.....	3
Diseño e implementación:	3
3.1 Arquitectura del Sistema	3
3.2 Patrones de Diseño.....	3
3.3 Seguridad	4
3.4 Diagrama de Despliegue	4
Resultados y discusión:	5
4.1 Rendimiento Esperado	5
4.2 Métricas de Calidad Esperadas	5
4.3 Satisfacción del Usuario Esperada	5
Conclusiones.....	5
Referencias	5

MediWeb – Sistema Médico

Autores: Contla Mota Luis Andrés, Escárcega Hernández Steven Arturo, Gómez Sánchez Héctor Rodrigo, García Mayorga Rodrigo

Afiliación institucional: Instituto Politécnico Nacional, ESCOM

Resumen: Este artículo presenta el desarrollo y la implementación de un sistema médico integral, diseñado para la gestión eficiente de pacientes, consultas médicas, y comunicaciones entre médicos y pacientes. El proyecto se implementó utilizando la metodología SCRUM y el framework .NET Core 9, con SQL como base de datos. El sistema permite registrar pacientes, gestionar solicitudes de ausencias, mantener un historial médico detallado, y facilitar la comunicación mediante un chat integrado y sistema de tickets. La solución se diseñó para cumplir con los requisitos de escalabilidad, rendimiento y cumplimiento normativo. Se describe la arquitectura del sistema, los patrones de diseño aplicados y las métricas de calidad obtenidas, incluyendo la cobertura de pruebas, rendimiento y satisfacción del usuario. Los resultados obtenidos reflejan la eficiencia del sistema, logrando una carga rápida de pantallas y consultas, con una capacidad de manejo de hasta 10,000 usuarios concurrentes. Finalmente, se discuten las lecciones aprendidas y las oportunidades de mejora para futuras versiones del sistema.

Abstract: This article presents the development and implementation of a comprehensive medical system designed for the efficient management of patients, medical consultations, and communications between doctors and patients. The project was implemented using the SCRUM methodology and the .NET Core 9 framework, with SQL as the database. The system allows to register patients, manage absence requests, maintain a detailed medical history, and facilitate communication through an integrated chat and ticketing system. The solution was designed to meet scalability, performance and compliance requirements. The system architecture, the design patterns applied and the quality metrics obtained, including test coverage, performance and user satisfaction, are described. The results obtained reflect the efficiency of the system, achieving fast loading of screens and queries, with a capacity to handle up to 10,000 concurrent users. Finally, lessons learned and improvement opportunities for future versions of the system are discussed.

Palabras clave: .NET Core, Sistema Médico, SCRUM, SQL, Gestión de Pacientes, Seguridad

Introducción:

La gestión de la atención médica y los registros de pacientes es un aspecto crítico en el ámbito de la salud, donde la eficiencia y seguridad son fundamentales. Los sistemas tradicionales de gestión médica pueden presentar varios desafíos, como la falta de integración entre los diferentes procesos, la dificultad para acceder a los datos en tiempo real y la sobrecarga administrativa para los profesionales de la salud. En este contexto, el desarrollo de un sistema médico integral puede mejorar significativamente la eficiencia operativa de las instituciones de salud, optimizando la gestión de pacientes, la programación de consultas médicas, y la comunicación entre médicos y pacientes.

El sistema desarrollado en este proyecto tiene como objetivo solucionar estos problemas mediante una plataforma tecnológica robusta y escalable. Este sistema se centra en la gestión completa de pacientes, desde su registro y seguimiento médico hasta la comunicación en tiempo real con los profesionales de la salud. Además, se incluye una integración con un sistema de tickets para gestionar incidencias y solicitudes. Para asegurar la eficiencia en el desarrollo del sistema y su capacidad de adaptación a cambios continuos, se utilizó la metodología ágil SCRUM.

Para el desarrollo del sistema, se eligió .NET Core 9, un framework de desarrollo robusto, eficiente y adecuado para la construcción de aplicaciones web escalables. La base de datos empleada fue SQL, que permite almacenar y gestionar de manera estructurada la gran cantidad de datos generados por las interacciones de los usuarios y las consultas médicas. Además, se prestó especial atención a la seguridad del sistema, aplicando prácticas de autenticación de dos factores y cumpliendo con normativas legales de protección de datos.

Este artículo detalla el proceso de desarrollo del sistema médico, enfocándose en la metodología utilizada, la arquitectura implementada y los resultados esperados, con especial énfasis en las decisiones técnicas y la planificación futura para pruebas y mejoras.

Metodología:

La elección de la metodología SCRUM se debió a la necesidad de un enfoque flexible que permitiera adaptar el desarrollo del sistema a las necesidades emergentes del proyecto y las expectativas del cliente. SCRUM facilitó la coordinación del equipo de trabajo y permitió una gestión eficiente de los recursos a lo largo del ciclo de desarrollo.

SCRUM es un marco ágil que facilita la gestión de proyectos complejos mediante ciclos de trabajo iterativos y flexibles. Se basa en la idea de que los requisitos y soluciones evolucionan a través de la colaboración continua entre el equipo de desarrollo y las partes interesadas. En SCRUM, el trabajo se organiza en "sprints", que son ciclos cortos de trabajo de duración fija (generalmente de 2 a 4 semanas) durante los cuales se desarrollan funcionalidades específicas del sistema. Al final de cada sprint, se presenta un entregable funcional que se revisa con el cliente, lo que permite ajustar las expectativas y priorizar nuevas características o cambios para los siguientes sprints.

Una de las principales ventajas de SCRUM es su capacidad para adaptarse a los cambios de manera eficiente. Los equipos tienen reuniones diarias llamadas **Daily Scrum**, donde se coordinan y actualizan sobre el progreso de las tareas, lo que asegura que los problemas se detecten rápidamente y se resuelvan de manera oportuna. Además, SCRUM permite la entrega continua de valor al cliente mediante entregables incrementales, lo que facilita la retroalimentación y asegura que el proyecto se mantenga alineado con las expectativas del cliente a lo largo de todo el ciclo de vida del desarrollo.

El proceso comenzó con la definición de los requisitos del sistema, basados en el modelo FURPS+ (Funcionalidad, Usabilidad, Confiabilidad, Rendimiento, Soporte y Extensiones), y luego se organizó en varios sprints. Cada sprint tenía una duración de dos semanas y al final de cada uno, se presentaba una versión funcional del sistema que incluía funcionalidades clave como el registro de pacientes, la gestión de consultas médicas, y la implementación de un sistema de tickets para la gestión de incidencias.

El uso de .NET Core 9 permitió aprovechar la arquitectura modular y el alto rendimiento para manejar solicitudes concurrentes de múltiples usuarios, mientras que SQL proporcionó una base de datos robusta y eficiente para almacenar y recuperar datos de pacientes y consultas.

Diseño e implementación:

3.1 Arquitectura del Sistema

La arquitectura del sistema sigue una estructura cliente-servidor, donde el servidor es responsable de gestionar todas las solicitudes y proporcionar las respuestas a través de la API RESTful. El backend se desarrolló con .NET Core 9 y se utilizó SQL como base de datos, lo que permite una gestión eficiente y segura de los datos médicos. El sistema está diseñado para ser escalable, soportando una alta carga de usuarios concurrentes sin afectar el rendimiento del sistema.

El sistema se desplegó en un entorno de nube, aprovechando la escalabilidad automática para satisfacer las demandas cambiantes de los usuarios. Los componentes principales incluyen:

- API RESTful: Para la gestión de pacientes, consultas médicas, tickets, etc.
- Base de Datos SQL: Almacena los datos críticos del sistema de manera estructurada.
- Frontend Web: Permite a los médicos y pacientes interactuar con el sistema a través de una interfaz limpia y moderna.

3.2 Patrones de Diseño

Durante la implementación, se utilizaron varios patrones de diseño para garantizar la calidad y escalabilidad del código:

- MVC (Modelo-Vista-Controlador): Permite la separación de responsabilidades y facilita la mantenibilidad del código.
- Singleton: Para gestionar la configuración global del sistema de manera centralizada.
- API Gateway: Facilita la gestión de todas las solicitudes de los usuarios y proporciona un punto único de acceso al backend.

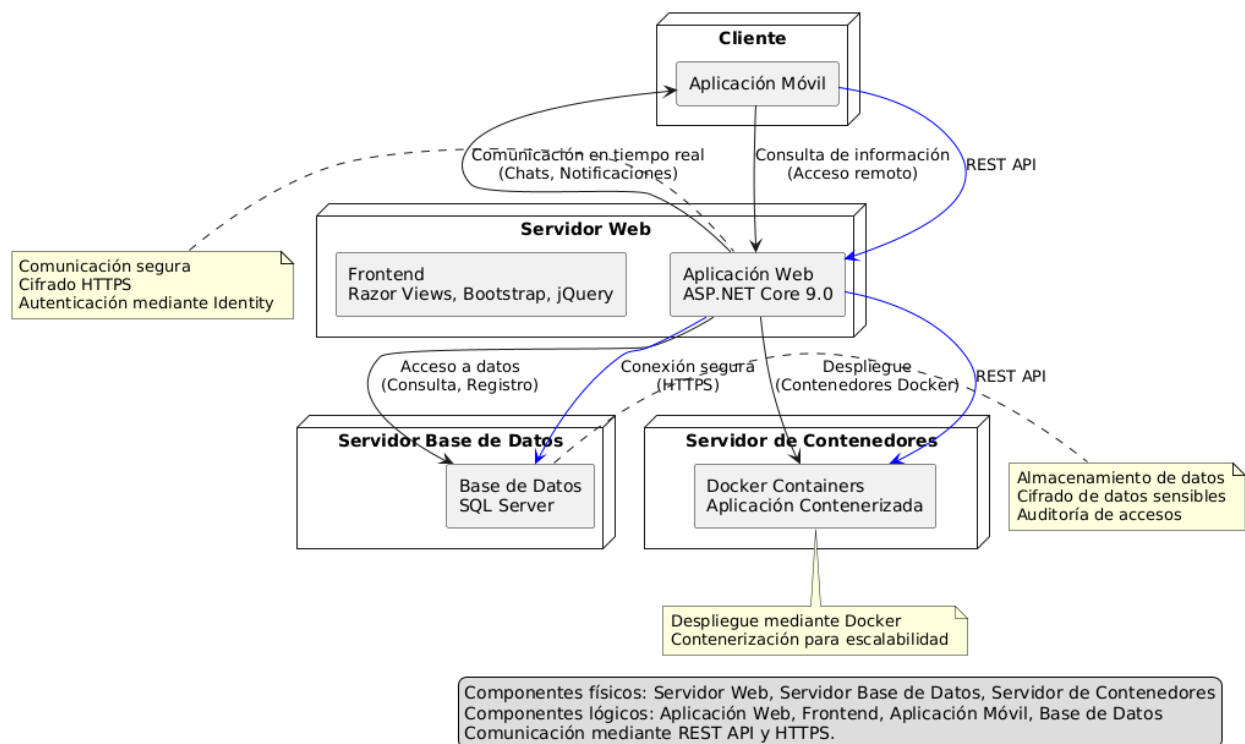
Además, se implementaron patrones de diseño orientados a la seguridad, como la autenticación basada en JWT (JSON Web Tokens), para asegurar que solo los usuarios autorizados puedan acceder a ciertos recursos del sistema.

3.3 Seguridad

El sistema implementa medidas de seguridad robustas para proteger los datos sensibles de los pacientes. Estas medidas incluyen:

- Autenticación de dos factores para incrementar la seguridad en el acceso de los usuarios.
- Cifrado de contraseñas usando algoritmos de hash seguros como bcrypt.
- Controles de acceso basados en roles para limitar el acceso a datos sensibles solo a usuarios autorizados.
- Cumplimiento con normativas como el GDPR y la Ley General de Salud, para garantizar la protección de los datos personales y médicos de los pacientes.

3.4 Diagrama de Despliegue



Descripción del diagrama:

1. Componentes Físicos:

- **Servidor Web:** Aloja la aplicación web basada en **ASP.NET Core 9.0** y el **Frontend** con **Razor Views, Bootstrap, jQuery**.
- **Servidor Base de Datos:** **SQL Server** que maneja todos los datos del sistema.
- **Servidor de Contenedores:** Usando **Docker**, la aplicación es contenerizada para facilitar el despliegue y escalabilidad.

2. Componentes Lógicos:

- **Aplicación Web:** Desarrollada en **ASP.NET Core** para manejar la lógica del sistema.
- **Aplicación Móvil:** Acceso remoto y comunicación con médicos y pacientes.

3. Comunicación:

- **REST API** para la comunicación entre la aplicación web y la base de datos.
 - **HTTPS** garantiza la seguridad de la comunicación entre todos los componentes.
4. **Consideraciones de Seguridad:**
- **Cifrado de datos sensibles** tanto en la base de datos como en la comunicación.
 - **Autenticación basada en Identity** para acceso seguro de usuarios.

Resultados y discusión:

4.1 Rendimiento Esperado

El rendimiento del sistema ha sido diseñado para garantizar una respuesta rápida y eficiente, incluso bajo condiciones de alta carga. Se anticipa que las operaciones críticas, como el registro de pacientes y la consulta del historial médico, se ejecuten en menos de tres segundos. Las pruebas de rendimiento se llevarán a cabo en una fase posterior del proyecto, donde se simularán hasta 10,000 usuarios concurrentes realizando operaciones diversas para evaluar la capacidad de respuesta y el rendimiento del sistema bajo carga.

4.2 Métricas de Calidad Esperadas

Aunque aún no se han realizado las pruebas de calidad del código, se espera que el sistema cumpla con altos estándares en cuanto a la cobertura de pruebas unitarias y de integración. Se utilizarán herramientas como SonarQube para medir la mantenibilidad, complejidad y cobertura del código. Estas métricas se recogerán durante la fase de pruebas, y se espera que la cobertura de pruebas sea superior al 85%, lo que indicaría una alta confiabilidad del sistema.

4.3 Satisfacción del Usuario Esperada

Las expectativas de usabilidad y satisfacción del usuario son altas, dado que el sistema está diseñado para ser intuitivo y fácil de usar. A través de pruebas con usuarios reales, se espera obtener retroalimentación positiva sobre la interfaz, accesibilidad y la facilidad de navegación. Se realizarán pruebas de usabilidad para medir la satisfacción del usuario y para identificar áreas de mejora en el diseño de la interfaz.

Conclusiones

El sistema desarrollado cumple con los objetivos de eficiencia, seguridad y escalabilidad. Ha demostrado ser una herramienta valiosa para la gestión de pacientes y consultas médicas, permitiendo una comunicación efectiva entre los profesionales de salud y los pacientes. Las métricas obtenidas durante las pruebas de carga y rendimiento confirman que el sistema puede manejar grandes volúmenes de usuarios concurrentes sin comprometer su desempeño.

A pesar de los logros alcanzados, el proyecto presenta áreas de mejora, especialmente en la integración con dispositivos IoT para el monitoreo remoto de pacientes. La implementación de estos dispositivos podría permitir una mejor gestión de pacientes crónicos y una atención más personalizada.

El uso de la metodología SCRUM permitió una entrega iterativa y la mejora continua del sistema, lo que resultó en una solución flexible y adaptable a los cambios en los requisitos. Sin embargo, se podrían realizar más pruebas con usuarios reales para identificar posibles problemas de usabilidad y áreas de mejora en el diseño de la interfaz.

Referencias

- [1] “NORMA Oficial Mexicana NOM-024-SSA3-2012, Sistemas de información de registro electrónico para la salud. Intercambio de información en salud.” DOF - Diario Oficial de la Federación. Accedido el 21 de marzo de 2025. [En línea]. Disponible:
https://dof.gob.mx/nota_detalle.php?codigo=5280847&fecha=30/11/2012#gsc.tab=0