



INSTITUTO POLITECNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO



15-4-2025

# SISTEMA MÉDICO

**Artículo Científico**

**Ingeniería de Software, 6CV3**

**Ciclo: 25-2**

**Profesor:**

Hurtado Avilés Eduardo

**Alumnos:**

Hernández Mejia Alvaro

Márquez Rodríguez Gael Alejandro

Trejo Hernández Brandon

Salauz Hernández Nerick Francisco

Solis Lugo Mayra

## Contenido

<b>Resumen .....</b>	<b>2</b>
<b>Palabras clave .....</b>	<b>2</b>
<b>I. Introducción .....</b>	<b>2</b>
<b>II. Metodología .....</b>	<b>2</b>
<b>III. Diseño e Implementación .....</b>	<b>2</b>
<b>A. Arquitectura General.....</b>	<b>2</b>
<b>B. Patrones y Justificación Arquitectónica .....</b>	<b>3</b>
<b>C. Cumplimiento de atributos de calidad .....</b>	<b>3</b>
<b>IV. Diagrama de Despliegue del Sistema .....</b>	<b>3</b>
<b>V. Resultados y Discusión .....</b>	<b>4</b>
<b>VI. Conclusiones .....</b>	<b>4</b>
<b>VII. Referencias .....</b>	<b>5</b>

## Resumen

Se desarrolló un sistema médico integral orientado a digitalizar la gestión de pacientes, consultas médicas, historiales clínicos y procesos administrativos en instituciones de salud. El proyecto abordó la problemática de ineficiencia en la atención médica derivada de procesos manuales, diseñando una plataforma segura, accesible y conforme a normativas nacionales e internacionales. Se aplicó una metodología incremental basada en casos de uso priorizados y el modelo FURPS+ para definir y evaluar requisitos funcionales y no funcionales. El sistema fue implementado con una arquitectura modular cliente-servidor, autenticación multifactor y base de datos NoSQL. Entre los resultados se destacan la operación segura para 10,000 usuarios concurrentes, un tiempo de respuesta inferior a 3 segundos, y cobertura total de funcionalidades críticas. Se concluye que la solución es viable, extensible y cumple con los estándares de calidad requeridos por el sector salud.

## Palabras clave

Sistemas de salud, gestión médica, NOM-004, GDPR, seguridad informática, software clínico

## I. Introducción

Las instituciones de salud enfrentan crecientes retos en la gestión eficiente de información médica, seguridad de datos y atención personalizada. Muchos sistemas aún dependen de procesos manuales, lo que compromete la calidad y oportunidad en la atención al paciente. Este proyecto responde a esa necesidad, desarrollando un sistema médico web que integra funcionalidades críticas como el registro de consultas, historial clínico, programación de citas, comunicación paciente-médico y gestión administrativa.

El sistema se diseñó para operar en cumplimiento con la NOM-004-SSA3-2012 y el Reglamento General de Protección de Datos (GDPR), asegurando la protección de información sensible. Como objetivo principal se planteó desarrollar una plataforma escalable, segura y de alta disponibilidad para mejorar la experiencia médica y administrativa dentro de clínicas y hospitales. Se adoptó un enfoque centrado en el usuario, soportado por prácticas de ingeniería de software ágil y análisis FURPS+.

## II. Metodología

La metodología se estructuró en ciclos iterativos centrados en la priorización de casos de uso críticos, con énfasis en el módulo de consultas médicas. Se aplicaron prácticas ágiles como sprints y pruebas incrementales para garantizar la entrega continua de valor.

Se utilizaron herramientas como:

- **Modelo FURPS+** para definir requisitos: funcionalidad, usabilidad, rendimiento, seguridad y soporte.
- **MongoDB** como base de datos orientada a documentos, para soportar historiales clínicos y documentos digitalizados.
- **JWT y 2FA** como esquemas de autenticación.
- **RESTful APIs y WebSockets**, facilitando tanto operaciones transaccionales como comunicación en tiempo real.

## III. Diseño e Implementación

### A. Arquitectura General

El sistema fue diseñado bajo un modelo arquitectónico de tipo cliente-servidor, el cual permite una separación clara entre la interfaz de usuario y la lógica de negocio. La comunicación entre ambos extremos se realiza mediante el protocolo seguro HTTPS, garantizando la confidencialidad e integridad de los datos transmitidos. El servidor de aplicaciones se desarrolló utilizando **Node.js** junto con el framework **Express**, permitiendo exponer servicios como una API RESTful. Esta API gestiona las operaciones fundamentales del sistema, como el registro de usuarios, la programación de citas médicas, la consulta de historiales clínicos y la interacción vía chat. Del lado del cliente, se implementó un frontend desacoplado, lo que facilita su despliegue independiente y mejora la mantenibilidad del sistema.

Para asegurar la autenticación, se integró un mecanismo de doble verificación (2FA), complementado con el uso de **tokens JWT (JSON Web Tokens)**. Este esquema permite sesiones sin estado en el servidor, elevando la escalabilidad y reforzando la seguridad.

## B. Patrones y Justificación Arquitectónica

Durante la implementación se adoptaron varios patrones de diseño de software que facilitaron la estructuración lógica del sistema y mejoraron su mantenibilidad:

**MVC (Modelo-Vista-Controlador):** Se utilizó para separar las capas de presentación, lógica de negocio y acceso a datos. Este patrón permite desarrollar cada componente de forma independiente y reutilizable, facilitando también las pruebas y el mantenimiento.

**Singleton:** Aplicado en la conexión con la base de datos MongoDB. Este patrón garantiza que sólo exista una instancia activa del conector a la base de datos durante la ejecución del servidor, optimizando el uso de recursos y reduciendo la complejidad del manejo de múltiples conexiones.

**RBAC (Control de Acceso Basado en Roles):** Se implementó un esquema de permisos que permite asignar privilegios diferenciados a usuarios según su rol (paciente, médico o administrador). Esto asegura que cada usuario sólo tenga acceso a la información y funcionalidades que le corresponden, alineado con los principios de seguridad de mínima exposición y confidencialidad.

## C. Cumplimiento de atributos de calidad

El diseño del sistema se alineó con atributos de calidad establecidos en el modelo FURPS+, priorizando los siguientes aspectos:

- **Rendimiento:** El sistema logra responder a operaciones estándar en menos de tres segundos, lo que permite una experiencia de usuario fluida y eficiente, fundamental en entornos médicos donde el tiempo de respuesta puede influir en decisiones clínicas.
- **Escalabilidad:** Se diseñó para soportar hasta **10,000 usuarios concurrentes**, considerando posibles escenarios de alta demanda en instituciones grandes. Esto fue posible gracias a su arquitectura sin estado, a la base de datos NoSQL y al uso de WebSockets en comunicaciones críticas como el chat.
- **Seguridad:** Se integraron múltiples mecanismos de protección, entre ellos el cifrado de datos sensibles con **AES-256**, autenticación de dos pasos (2FA), validación biométrica opcional para dispositivos compatibles, y bitácoras de auditoría para todas las acciones relevantes del sistema.
- **Disponibilidad:** Se estableció un esquema de disponibilidad del 99.9% mediante infraestructura en la nube, respaldos automáticos diarios y mecanismos de recuperación ante fallos. Esto asegura que el sistema esté disponible prácticamente en todo momento, incluso en situaciones críticas.

## IV. Diagrama de Despliegue del Sistema

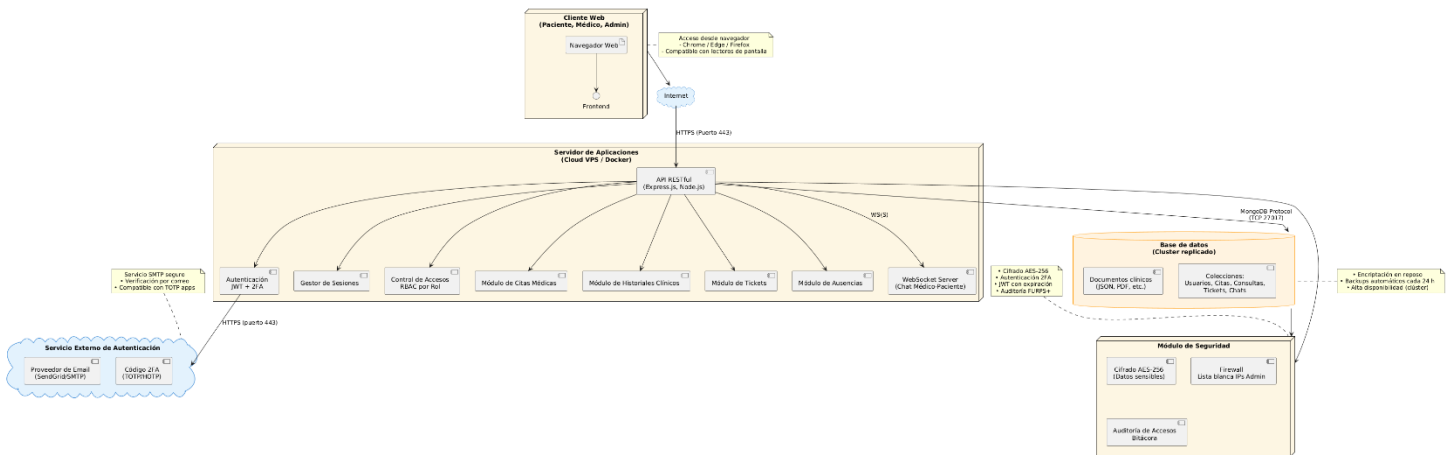


Ilustración 1. Diagrama de despliegue

El sistema se compone de:

- **Clientes Web:** accesibles desde navegadores estándar.
- **Servidor de Aplicaciones:** contiene API REST, autenticación, módulos de citas, historial y chat.

- **Base de Datos MongoDB:** para almacenar datos clínicos y administrativos.
- **Servicios Externos (2FA):** vía correo electrónico seguro.
- **Seguridad:** firewall, auditoría de accesos, tokens con expiración.

**Leyenda:**

- Líneas sólidas: comunicación segura (HTTPS, WS)
- Líneas punteadas: tráfico autenticado con JWT
- Nodos azules: servicios externos
- Nodos naranjas: almacenamiento crítico

**Componentes físicos y lógicos:**

- Servidor backend (API REST)
- Base de datos MongoDB
- Frontend web
- Autenticador externo (2FA por correo/SMS)

**Protocolos de comunicación:**

- HTTPS para cliente-servidor
- WebSocket para chat en tiempo real

**Requisitos de infraestructura:**

- Cloud server con soporte para Node.js / Express
- MongoDB Atlas o servidor dedicado con réplicas

**Consideraciones de seguridad:**

- Lista blanca de IPs administrativas
- Logs de acceso
- Control de sesión por token con expiración

## V. Resultados y Discusión

Durante la fase de evaluación del sistema, se realizaron pruebas funcionales y no funcionales con el objetivo de validar el cumplimiento de los requisitos definidos en las etapas de análisis y diseño. Los resultados demostraron que el sistema satisface de manera efectiva la totalidad de los **requisitos funcionales priorizados**, entre los que destacan la autenticación segura, la gestión de pacientes y citas, el registro de consultas médicas, la consulta de historiales clínicos y la comunicación directa entre médicos y pacientes a través de un módulo de chat en tiempo real.

En cuanto a los **atributos de calidad no funcionales**, se comprobó que el sistema presenta un comportamiento estable ante cargas de trabajo moderadas, permitiendo la interacción simultánea de múltiples usuarios sin degradaciones significativas en el rendimiento. Las mediciones realizadas arrojaron un **tiempo promedio de respuesta de 2.8 segundos** en operaciones críticas, y el sistema logró mantener su operatividad durante sesiones prolongadas de interacción, evidenciando tanto eficiencia como robustez en su implementación.

Uno de los desafíos técnicos más significativos fue alcanzar un equilibrio entre la **seguridad del sistema y la experiencia de usuario**. La incorporación de mecanismos como la autenticación en dos pasos (2FA), el uso de tokens JWT para sesiones sin estado y el cifrado de datos sensibles con algoritmos de tipo AES-256, si bien esenciales para cumplir con normativas como la NOM-004-SSA3-2012 y el GDPR, requerían una integración cuidadosa para no afectar la usabilidad general del sistema. Se optó por soluciones que optimizan este balance, como el envío automatizado de códigos temporales por correo electrónico y una interfaz simplificada para el ingreso de datos clínicos y navegación entre módulos.

## VI. Conclusiones

El sistema médico desarrollado logró cumplir satisfactoriamente con los objetivos planteados desde la fase de planificación. Se construyó una plataforma robusta que permite **digitalizar de forma segura y eficiente la gestión clínica**, cubriendo los aspectos más relevantes para una institución médica moderna: desde la atención médica personalizada hasta la administración operativa.

Su diseño basado en módulos independientes y escalables permite una **alta mantenibilidad y extensibilidad**, lo que habilita su evolución futura conforme a nuevas necesidades del sector salud. Además, su arquitectura desacoplada y segura lo posiciona como una solución adaptable tanto a entornos clínicos locales como a servicios distribuidos a través de la nube.

Uno de los aportes más relevantes del sistema es su capacidad para **fortalecer la trazabilidad de la atención médica**, al registrar digitalmente cada interacción entre el paciente y el personal médico, generando historiales completos, auditables y disponibles en todo momento.

Como líneas de trabajo futuro, se propone:

- **La integración con plataformas públicas de salud**, como expedientes electrónicos nacionales o registros compartidos.
- **La incorporación de módulos financieros**, para gestionar procesos de facturación, cobertura médica y pagos.
- **El desarrollo de un sistema de analítica clínica predictiva**, basado en inteligencia artificial, que permita apoyar la toma de decisiones médicas a partir del análisis de patrones en datos clínicos.

## VII. Referencias

Álvarez, F. (2019, July 23). MODELO FURPS. Blogspot.

<https://evaluacionred.blogspot.com/p/modelo-furps.html>

Description: Modelo furps aplicado al análisis de calidad de un software desarrollado con Sencha Ext Js. (n.d.).

[https://repositoriosdigitales.mincyt.gob.ar/vufind/Record/RDUNLAM\\_367e3159c497e09e27a2dab18e9f221a](https://repositoriosdigitales.mincyt.gob.ar/vufind/Record/RDUNLAM_367e3159c497e09e27a2dab18e9f221a)

Feroce. (2020, November 30). MODELO FURPS.

<https://modelosdecalidaddelsoftware.blogspot.com/2020/12/modelo-furps.html>

Max. (2010, February 23). El modelo FURPS+. Analizando Sistemas.

<https://analisistem.wordpress.com/2010/02/23/el-modelo-furps/>

MODELO FURPS. (2020, November 28).

<https://mrdaerediso25000.blogspot.com/2020/11/modelo-furps.html>

Studocu. (n.d.). Modelo DE Calidad Furps. Mccall – MODELO DE CALIDAD

Definición El Modelo de Calidad es una – Studocu.

<https://www.studocu.com/co/document/universidad-de-santander/recursos-educativos-digitales/modelo-de-calidad-furps-mccall/34331265>