# **Sistema de Gestión Clínica MediPlus**

# Prototipo MVC con Seguridad y Base de Datos Oracle

NOMBRE: Jhordan Burdiles, Victor Manuel Diaz

CARRERA: ingenieria en informatica

ASIGNATURA: programación orientada a objeto seguro

PROFESOR: Ricardo Elías Gutiérrez

FECHA:03-12-2025

### **1. Introducción**

La Clínica MediPlus, un centro médico de tamaño medio con diversas especialidades ambulatorias, enfrenta desafíos críticos en la gestión de su información. El uso actual de planillas Excel y formularios en papel ha resultado en pérdida de datos, dificultad en el seguimiento de historial clínico y serios problemas de seguridad debido a la falta de control de acceso y protección de la información sensible.

El presente informe detalla el desarrollo de un módulo prototipo de sistema de información diseñado para digitalizar y centralizar la gestión de pacientes, médicos, consultas, recetas e insumos. El objetivo es proporcionar una solución robusta y segura, basada en una Base deDatos Relacional Oracle XE 21c e implementada bajo la arquitectura Modelo-Vista-Controlador (MVC), garantizando la seguridad de las credenciales mediante el uso de *hashing*. Este sistema busca subsanar las deficiencias actuales y proveer una plataforma eficiente para la operación diaria de la clínica.

# objetivo general

Desarrollar un módulo prototipo de sistema de información robusto y seguro para la Clínica MediPlus, que permita la gestión centralizada y digitalizada de usuarios (pacientes, médicos, administradores), consultas médicas, insumos y recetas, mitigando los riesgos asociados al manejo de información en formatos no estructurados (Excel y papel, u otro conocido).

# Objetivos especificos

* *Implementar un sistema de registro y autenticación* de usuarios con manejo de sesión, asegurando la integridad de las credenciales mediante el uso de algoritmos de *hashing*.
* *Diseñar e implementar una Base de Datos Relacional* en Oracle XE 21c para el almacenamiento persistente de toda la información del sistema (usuarios, pacientes, médicos, consultas, recetas, insumos).
* *Desarrollar las funcionalidades CRUD (Crear, Leer, Actualizar, Eliminar)* para la gestión de entidades clave: pacientes, médicos, insumos, consultas, y recetas.
* *Integrar la carga inicial de usuarios* a la base de datos a partir de un archivo JSON preexistente.
* *Implementar la arquitectura MVC* para asegurar la mantenibilidad, escalabilidad y separación de responsabilidades del código.
* *Garantizar la seguridad de las transacciones manuales* a la base de datos mediante la prevención de ataques de inyección SQL, utilizando la **parametrización de consultas**
* *Mostrar los valores monetarios del sistema* (costos de insumos, valor de consultas, costos de recetas) en Pesos Chilenos (CLP), realizando las conversiones pertinentes.

## requerimientos detectados

Los requerimientos se clasifican en **Funcionales** y **No Funcionales** .

requerimientos funcionales (RF)

|  |  |  |
| --- | --- | --- |
| **ID** | **Requerimiento funcional** | **Descripción** |
| RF 1 | Gestión de usuarios | El sistema debe permitir el registro inicial de usuarios desde un archivo JSON y el manejo de sus roles (Paciente, Médico, Administrador). |
| RF 2 | Autenticación segura | Debe manejar el inicio y cierre de sesión de usuarios, verificando credenciales con *hash* de contraseñas. |
| RF 3 | Perfiles diferenciados | Debe ofrecer un menú y funcionalidades diferentes según el rol del usuario autenticado. |
| RF 4 | CRUD de pacientes | El sistema debe permitir crear, consultar, actualizar y eliminar la información de los pacientes. |
| RF 5 | CRUD de medicos | El sistema debe permitir crear, consultar, actualizar y eliminar la información de los médicos. |
| RF 6 | CRUD de consultas | El sistema debe permitir registrar, consultar y gestionar las consultas médicas asociadas a un paciente y médico. |
| RF 7 | CRUD de recetas | Debe permitir el registro de recetas médicas asociadas a una consulta y que incluyan medicamentos recetados. |
| RF 8 | CRUD de insumos | Debe permitir la gestión (CRUD) del inventario de insumos (nombre, tipo, stock, costo). |
| RF 9 | perfiles Diferenciados | Todo usuario debe poder editar su propia información personal, y el administrador puede editar a cualquier usuario. |
| RF 10 | Conversión de monedas | Los costos en USD (insumos) deben mostrarse al usuario en CLP (Pesos Chilenos) al consultar. |

## requerimientos no funcionales RNF

|  |  |  |
| --- | --- | --- |
| **ID** | **Requerimientos no Funcionales** | **Descripciones** |
| RNF 1 | seguridad | Las credenciales de usuario deben ser almacenadas mediante *hashing* seguro. |
| RNF 2 | integridad de datos | Se debe utilizar una Base de Datos Relacional “Oracle” para el almacenamiento persistente de los datos. |
| RNF 3 | proyeccion de inyección SQL | Todas las interacciones manuales con la BD deben utilizar la parametrización de consultas. |
| RNF 4 | arquitectura | El código debe ser estructurado bajo el patrón Modelo-Vista-Controlador (MVC). |
| RNF 5 | Documentacion | El código fuente debe ser gestionado y versionado en un repositorio de **GitHub**. |

# 

# 

# Diagrama de Clases

El Diagrama de Clases, modelado en UML, representa la estructura estática del sistema. Se establecen las relaciones de Herencia entre los tipos de usuarios y la clase base Usuario, y las relaciones de Asociación o Composición entre las demás entidades.

#### Clases del Modelo de Negocio

* *Usuario*: Contiene los atributos comunes a todos los actores del sistema.
* *Paciente, Médico, Administrador*: Heredan de Usuario y añaden atributos específicos de su rol.
* *Consultas*: Clase central que registra la atención médica. Se asocia a Paciente, Médico y Recetas.
* *Recetas*: Detalle de la prescripción asociada a una consulta.
* *Insumos*: Representa los materiales o productos utilizados en la clínica.
* *Agenda*: Gestiona la planificación de las consultas, asociando un paciente, un médico y un estado a una fecha específica.

Diagrama: