**FACULTAD DE INGENIERÍA**



**CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES**

Texto

Descripción generada automáticamente

INFORME DE PROYECTO FINAL

**TÍTULO:** Sistema de Gestión de Pacientes en un Hospital

**CURSO:** TÉCNICAS DE PROGRAMACIÓN ORIENTADA A OBJETOS

**DOCENTE DEL CURSO:** Ing. José Anicama Silva **AUTORES:**

* Gonzalez Poma Franco (100%)
* Tito Huacho Nicole (100%)
* Atanacio Jhon (100%)

CHORRILLOS – PERÚ

2025-

**Contenido**

**Historial de Versiones** ................................................................................................ 3 **Información del Proyecto** .......................................................................................... 4

**Aprobaciones**................................................................................................................ 4

**1.** **Descripción del problema** ........................................................................... 5 **2.** **Antecedentes / referencias** ......................................................................... 5

1. **Restricciones del proyecto** ........................................................................ 5
   1. **Identificación y análisis de todas las restricciones, criterios y**

**suposiciones** ............................................................................................................. 5

* 1. **Generación y análisis de múltiples soluciones** ................................ 5
  2. **Criterios para seleccionar la solución *(dentro de las restricciones económicas, ambientales, sociales, políticas, éticas, de seguridad, entre otras.)*** .... 5

1. **Alcance del producto / Software** ............................................................... 5
2. **Objetivo general** ............................................................................................ 6
3. **Objetivos específicos** .................................................................................. 6
4. **Estructuras de desglose del proyecto** .................................................... 7
5. **Herramientas y tecnología utilizada** ........................................................ 7
6. **Funcionalidades del producto** ................................................................... 9
7. **Diagrama de clases y relaciones** ............................................................ 11
8. **Diccionario de clases** ................................................................................. 12
9. **Requerimientos funcionales** .................................................................... 12
   1. (Nombre de la funcionalidad 1) ................................................................... 13
   2. (Nombre de la funcionalidad 2) ................................................................... 14
   3. (Nombre de la funcionalidad N) ................................................................... 14
10. **Reglas de negocio** ...................................................................................... 14
11. **Requerimientos de interfaces externas** ................................................ 14
    1. **Interfaces de usuario** .............................................................................. 14
12. **Requerimientos no funcionales** .............................................................. 15
13. **Glosario** ......................................................................................................... 15
14. **Bibliografía**.................................................................................................... 15

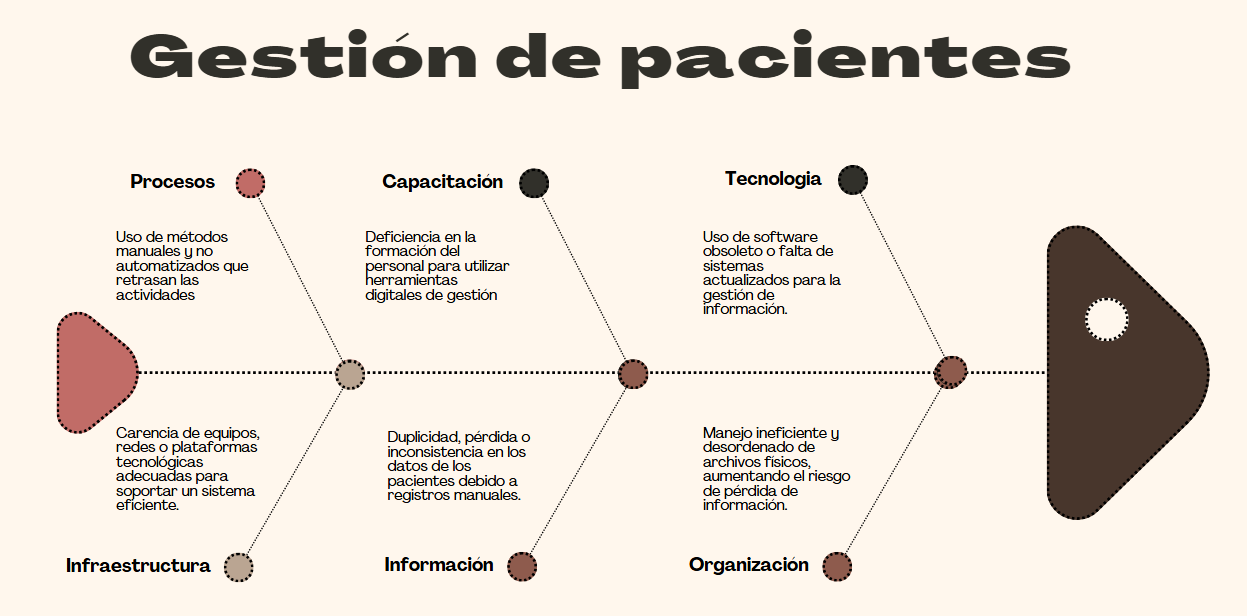
**1. Descripción del problema**

*Actualmente, muchos hospitales enfrentan serias dificultades para gestionar de manera eficiente los datos de sus pacientes. La falta de un sistema automatizado provoca retrasos en la atención médica, errores en la asignación de citas, duplicidad de registros y pérdida de información importante.*

*Esta situación no solo afecta la calidad del servicio, sino también la seguridad del paciente, incrementando los riesgos médicos y administrativos.*

*Por ello, se ha identificado la necesidad de desarrollar un Sistema de Gestión de Pacientes que permita automatizar el registro, almacenamiento y consulta de datos personales, citas, historial clínico, recetas médicas, resultados de laboratorio, emergencias y vacunaciones.*

*Figura I. Diagrama de Ishikawa*

**

*Este sistema estará diseñado para facilitar el trabajo del personal médico, administrativo y técnico del hospital, contribuyendo a una atención más eficiente, segura y centrada en el paciente.*

**2. Antecedentes / referencias**

*La implementación de sistemas de gestión hospitalaria ha sido ampliamente abordada en múltiples investigaciones y proyectos a nivel nacional e internacional.*

*Por ejemplo, desarrollos de softwares más sofisticados, como el implementado por Crispin Pari y Mayhua de la Cruz (2024) en el Hospital Departamental de Huancavelica, demuestran que mejoran significativamente el flujo de actividades administrativas, financieras y de la gestión de pacientes; los mismos que demandan una atención clínica o medica efectiva y de calidad.*

*Otro antecedente relevante es el sistema desarrollado por Ordoñez Hernán (2018) en la Escuela Politécnica Nacional del Ecuador, el cual automatiza el proceso de admisión de pacientes, permitiendo gestionar datos personales, verificar convenios médicos y controlar el acceso del personal autorizado.*

*Este tipo de soluciones muestra lo útil que resulta introducir programas específicos en hospitales, puesto que aumenta la eficacia operativa, disminuye los fallos humanos y organiza la información médica. Además, subraya la relevancia de crear sistemas que se ajusten al entorno de la institución y al flujo real del proceso de atención médica.*

**3. Restricciones del proyecto**

**3.1. Identificación y análisis de todas las restricciones, criterios y suposiciones**

* Recursos limitados (personal y tiempo de desarrollo)
* El sistema debe ejecutarse sin conexión a internet (modo local).
* El lenguaje de desarrollo es Java SE 8.
* La base de datos inicial será un archivo .txt y luego se migrará a MySQL.
* Los usuarios no tienen experiencia técnica avanzada (interfaz amigable).

**3.2. Generación y análisis de múltiples soluciones**

* Implementarlo solo con Excel (descartado por limitaciones)
* Usar un software gratuito externo (descartado por no adaptarse al flujo del hospital)
* Desarrollar el sistema personalizado en java (elegido)

**3.3. Criterios para seleccionar la solución**

* Bajo costo de implementación.
* Facilidad para adaptar a las reglas del hospital.
* Cumple restricciones de hardware del hospital.
* Desarrollo modular y escalable.

**4. Alcance del producto / Software**

El software busca facilitar la gestión de información de pacientes en un hospital.  
Permitirá registrar datos personales, citas, historial clínico, recetas, resultados de laboratorio, emergencias y vacunaciones, todo en un entorno digital accesible y seguro.

Algunos beneficios que traería consigo la implementación del software seria:

* Evita perdida de información
* Mejora la calidad de atención al paciente
* Aumenta la eficiencia administrativa
* Facilita el acceso a información médica en tiempo real

Los objetivos y metas funcionales del software están directamente alineados con las necesidades operativas del hospital:

* **Gestionar pacientes y citas**: Mejora la atención médica, reduce tiempos de espera y evita sobrecarga en la recepción.
* **Almacenar historia clínica digitalmente**: Permite cumplir con los estándares del Ministerio de Salud (MINSA) y mejora el acceso a información médica segura.
* **Gestionar recetas y medicamentos**: Optimiza el control de stock farmacéutico y previene errores en la entrega de insumos.
* **Facilitar alertas médicas y resultados críticos**: Refuerza la respuesta rápida del personal ante casos urgentes y mejora la calidad asistencial.

**5. Objetivo general**

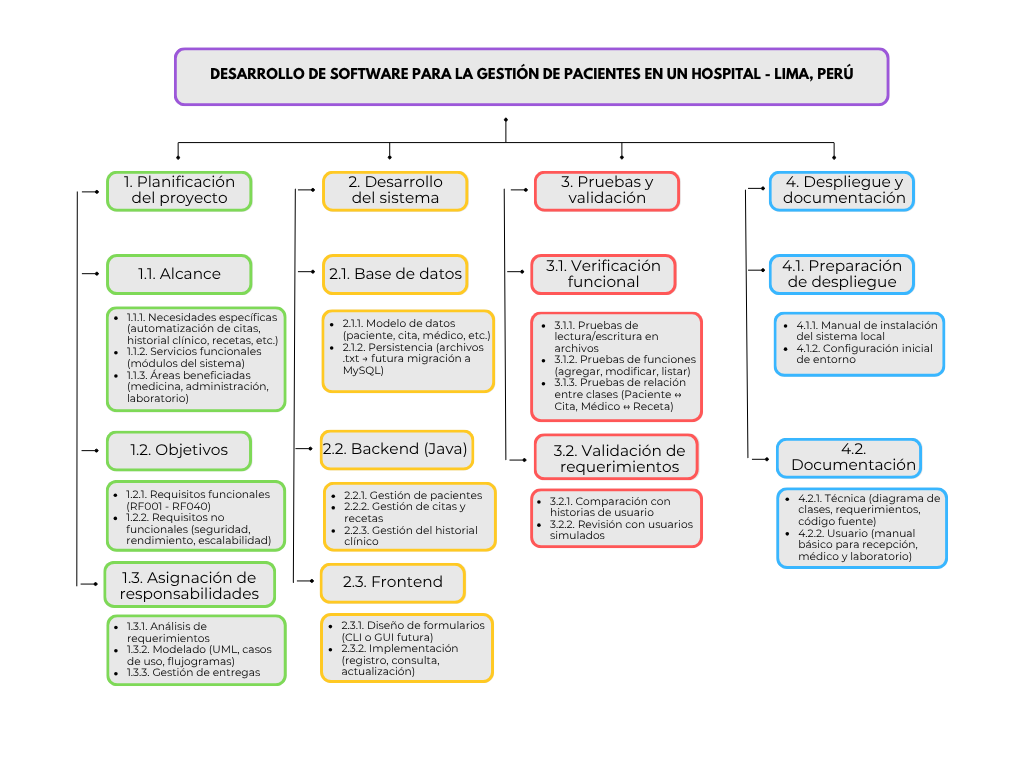
Desarrollar un sistema de gestión de pacientes para hospitales, que permitan digitalizar y automatizar la atención médica y administrativa, mejorando la calidad del servicio y la seguridad de los datos clínicos.

**6. Objetivos específicos**

* Analizar los procesos de atención médica y gestión de pacientes en el hospital para identificar oportunidades de mejora mediante automatización.
* Definir y documentar los requerimientos funcionales y no funcionales del sistema, a partir de entrevistas, historias de usuario y estándares del sector salud.
* Modelar la arquitectura del sistema utilizando diagramas de clases, casos de uso y flujogramas que representen el comportamiento y estructura del software.
* Desarrollar el sistema en lenguaje Java, aplicando principios de programación orientada a objetos y técnicas de persistencia en archivos de texto.
* Implementar funcionalidades clave como el registro de pacientes, gestión de citas, historial clínico, recetas médicas y resultados de laboratorio.
* Integrar manejo de archivos en Java para la lectura y escritura de información médica en un entorno local.
* Realizar pruebas funcionales del sistema**,** verificando el cumplimiento de los requerimientos implementados y asegurando la calidad del software.
* Documentar el proyecto de manera completa, incluyendo código fuente, manual de usuario, requerimientos y evidencia de pruebas realizadas.

**7. Estructuras de desglose del proyecto**

*Figura II. Estructura de desglose del proyecto*



*Fuente: elaboración propia*

**8. Herramientas y tecnología utilizada**

**Metodologías consideradas:**

Para el desarrollo del sistema de gestión de pacientes se evaluaron diferentes metodologías de desarrollo de software, considerando el entorno dinámico de trabajo, la evolución continua de los requerimientos, y la necesidad de entregas funcionales parciales.

* **Modelo en Cascada:** El modelo en cascada plantea una secuencia rígida de etapas (análisis, diseño, implementación, pruebas, etc.), lo que resulta útil en proyectos con requerimientos completamente definidos desde el inicio. Sin embargo, en proyectos como este, donde se requiere flexibilidad y ajustes frecuentes, su rigidez representa una limitación importante.
* **Metodología Ágil (Scrum):** Scrum se basa en ciclos iterativos e incrementales denominados sprints, lo que permite una rápida entrega de funcionalidades, revisión constante y adaptabilidad a cambios.
* **Metodología RUP:** Metodología orientada a grandes proyectos industriales, con ciclos por fases y enfoque en documentación pesada.
* **Metodología XP:** Metodología centrada en buenas prácticas técnicas y desarrollo conjunto, pero menos estructurada para el entorno educativo y documental del proyecto.

Se eligió la metodología Scrum debido a su enfoque ágil, iterativo y adaptable. Permite dividir el desarrollo en sprints cortos, facilitando la entrega progresiva de funcionalidades como el registro de pacientes, gestión de citas e historial clínico. Además, Scrum se ajusta a proyectos con requerimientos en evolución, mejora la comunicación del equipo y permite realizar pruebas constantes para asegurar la calidad del sistema.

***Herramientas Hardware:***

* *Servidor HP ProLiant ML110, con las siguientes características:* o *Procesador Intel Xeon-B 3204 6-Core (1.90GHz 8.25MB L3 Cache) o* *Memoria RAM: 32GB DDR4 RDIMM / 192GB (máximo)* o *Disco Duro: HDD 4TB 6G SATA 7.2k*

**Herramientas Software:**

* Sistema de gestión de bases de datos (SGBD):

*SQL Server Management Studio 20: Es un entorno gráfico integrado proporcionado por Microsoft para la administración de bases de datos SQL Server. Permite gestionar instancias de bases de datos, realizar consultas SQL, diseñar tablas, realizar respaldos y mantener la seguridad de los datos. Su interfaz amigable facilita el trabajo tanto a desarrolladores como a administradores de bases de datos.*

* Lenguaje de Programación:

*Java. Lenguaje robusto, multiplataforma, orientado a objetos. Se usará en combinación con el framework Spring Boot, que permite desarrollar aplicaciones web escalables con configuración mínima y estructura modular.*

* Entorno de desarrollo integrado (IDE)

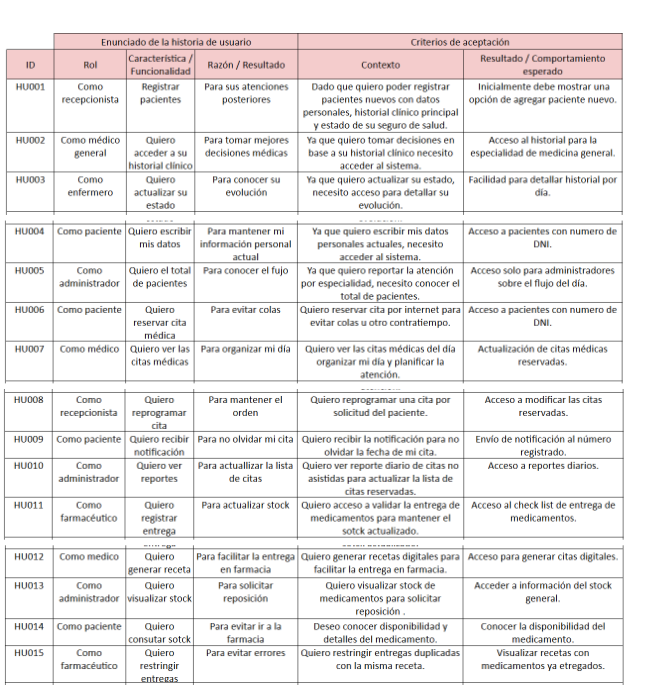
*Visual Studio Code: Ligero y extensible, adecuado para frontend y scripts ligeros.*

Se selecciona el lenguaje de programación Java, usando Visual Studio Code como entorno de desarrollo por su flexibilidad y soporte para extensiones. Para la gestión de base de datos se utiliza SQL Server Management Studio 20, que permite administrar y consultar bases de datos SQL Server de forma eficiente.

**9. Funcionalidades del producto**

Se enumeran las principales funcionalidades del sistema de gestión de pacientes en un hospital, las cuales están alineadas con los requerimientos funcionales:

1. Registro y administración de pacientes  
   Permite registrar, modificar y consultar los datos personales y de contacto de cada paciente. *(RF001, RF004, RF035)*
2. Gestión de citas médicas  
   Incluye la reserva, modificación y cancelación de citas por parte del paciente o la recepción. *(RF006, RF008, RF030, RF038, RF040)*
3. Acceso al historial clínico  
   Permite al personal médico visualizar el historial completo del paciente.  
   *(RF002, RF017, RF018, RF019, RF027)*
4. Emisión de recetas médicas  
   El médico puede generar recetas digitales, asociarlas al historial del paciente y marcarlas como usadas. *(RF012, RF015)*
5. Gestión de medicamentos y stock  
   Consulta de disponibilidad, alertas por bajo stock y actualización de inventario.  
   *(RF011, RF013, RF014)*
6. Registro de resultados de laboratorio  
   Registro y visualización de resultados de análisis clínicos, incluyendo alertas por valores críticos. *(RF020, RF021, RF036)*
7. Control de ingresos por emergencia  
   Registro de ingresos por emergencia con motivo y fecha. *(RF028)*
8. Registro de vacunación del paciente  
   Permite registrar la aplicación de vacunas y fechas correspondientes. *(RF039)*
9. Gestión de usuarios y roles  
   Control de autenticación, asignación de roles y acceso restringido según perfil. *(RF022, RF025)*

Historias de usuario:

**10. Diagrama de clases y relaciones**

*Figura III. Diagrama de clases*

Imagen que contiene Calendario

Descripción generada automáticamente

Fuente: elaboración propia

**11. Diccionario de clases**

*Se elabora un diccionario de clases y se mencionan particularidades de los atributos.*

*Tabla I. Clase Paciente*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

*Tabla II. Clase Usuario*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

*Tabla III. Clase Rol*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

*Tabla IV. Clase CitaMedica*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

*Tabla V. Clase Medico*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

*Tabla VI. Clase HistorialClinico*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

*Tabla VII. Clase EntradaHistorial*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

*Tabla VIII. Clase RecetaMedica*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

*Tabla IX. Clase Medicamento*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

*Tabla X. Clase ResultadoLaboratorio*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

*Tabla XI. Clase Emergencia*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

*Tabla XII. Clase Vacunacion*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

**12. Requerimientos funcionales**

**12.1.** Registro y administración de pacientes

Descripción***:*** *Permite registrar nuevos pacientes con sus datos personales, así como editar, consultar y descargar su información médica.*

Prioridad***:*** *Alta*

Acciones iniciadoras y comportamiento esperado:

* *El recepcionista o paciente accede al formulario de registro.*
* *El sistema valida los datos ingresados (como el DNI o correo electrónico).*
* *Si todo es válido, almacena el registro y genera una confirmación.*
* *Permite editar los datos y exportar los informes médicos del paciente.*

Requerimientos funcionales:

* REQ-1*: El sistema debe permitir registrar nuevos pacientes con nombre,* apellido, DNI, contacto y dirección. (RF001)
* REQ-2: *El sistema debe permitir modificar los datos del paciente. (RF004)*
* REQ-3: El sistema debe permitir al paciente descargar sus informes médicos. (RF035)

**12.2.**  Gestión de citas médicas

Descripción:  
Permite agendar, modificar, cancelar citas y generar reportes mensuales de atención médica.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

* El paciente o recepcionista solicita una cita desde el sistema.
* El sistema verifica disponibilidad del médico y registra la cita.
* Puede modificarse o cancelarse antes de la fecha.
* Se genera reporte de citas por médico y especialidad.

Requerimientos funcionales:

* REQ-4: El sistema debe permitir a los pacientes reservar citas médicas. *(RF006)*
* REQ-5: El sistema debe permitir modificar la fecha y hora de una cita médica. *(RF008)*
* REQ-6: El sistema debe registrar el horario y duración de cada cita. *(RF030)*
* REQ-7: El sistema debe permitir agendar controles médicos posteriores. *(RF038)*
* REQ-8: El sistema debe generar reportes mensuales de citas por médico y especialidad. *(RF040)*

**12.3. Acceso al historial clínico**

Descripción:  
Permite a los médicos visualizar el historial clínico completo del paciente, incluyendo diagnósticos, archivos adjuntos y observaciones.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

* El médico accede al perfil del paciente.
* Visualiza entradas del historial por fecha.
* Puede cargar archivos (PDF, imágenes) o agregar observaciones.

Requerimientos funcionales:

* REQ-9: El sistema debe permitir al médico acceder al historial clínico. *(RF002)*
* REQ-10: El sistema debe permitir registrar observaciones diarias. *(RF017)*
* REQ-11: El sistema debe comprobar cambios realizados en el historial. *(RF018)*
* REQ-12: El sistema debe permitir subir archivos adjuntos. *(RF019)*
* REQ-13: El sistema debe mostrar alertas sobre alergias registradas. *(RF027)*

**12.4. Emisión de recetas médicas**

Descripción:  
Permite al médico generar recetas digitales, asociarlas al historial y verificar si ya fueron utilizadas.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

* El médico prescribe medicamentos desde el sistema.
* El paciente puede visualizar la receta.
* El sistema valida si ya fue usada antes de dispensar los medicamentos.

Requerimientos funcionales:

* REQ-14: El sistema debe permitir al médico generar recetas digitales. *(RF012)*
* REQ-15: El sistema debe verificar si una receta ya fue utilizada. *(RF015)*

**12.5. Gestión de medicamentos y stock**

Descripción:  
Permite al personal consultar disponibilidad de medicamentos, recibir alertas por bajo stock y registrar entregas.

Prioridad: Media

Acciones iniciadoras y comportamiento esperado:

* El paciente consulta disponibilidad.
* El sistema muestra stock actual.
* Se notifica al administrador si hay stock crítico.

Requerimientos funcionales:

* REQ-16: El sistema debe registrar la entrega de medicamentos y actualizar el stock. *(RF011)*
* REQ-17: El sistema debe mostrar alertas de stock crítico. *(RF013)*
* REQ-18: El sistema debe permitir al paciente consultar disponibilidad. *(RF014)*

**12.6. Resultados de laboratorio**

Descripción:  
Permite registrar resultados de exámenes clínicos y alertar a médicos si los resultados son críticos.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

* El laboratorio registra resultados.
* El médico recibe notificación en caso de valores anormales.
* El paciente puede ver sus resultados desde su perfil.

Requerimientos funcionales:

* REQ-19: El sistema debe registrar resultados de laboratorio. *(RF020)*
* REQ-20: El sistema debe permitir al médico visualizar resultados. *(RF021)*
* REQ-21: El sistema debe enviar alertas sobre resultados críticos. *(RF036)*

**12.7. Emergencias y vacunación**

Descripción:  
 Permite registrar ingresos por emergencia y aplicación de vacunas a pacientes.

Prioridad: Media

Acciones iniciadoras y comportamiento esperado:

* El personal registra un ingreso por emergencia, con fecha y motivo.
* Se puede registrar una vacuna aplicada al paciente con su respectiva fecha.

Requerimientos funcionales:

* REQ-22: El sistema debe registrar ingresos por emergencia. *(RF028)*
* REQ-23: El sistema debe permitir registrar vacunas aplicadas. *(RF039)*

**12.8. Gestión de usuarios y roles**

Descripción:  
Permite crear usuarios, asignar roles y restringir el acceso según privilegios definidos.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

* El administrador crea usuarios y les asigna roles (ej. médico, recepcionista).
* El sistema restringe accesos según el rol asignado.

Requerimientos funcionales:

* REQ-24: El sistema debe restringir acceso a historiales clínicos según permisos. *(RF022)*
* REQ-25: El sistema debe permitir asignar y gestionar roles. *(RF025)*

**13. Reglas de negocio**

* 1. ***Solo los usuarios con rol de médico pueden generar recetas médicas.***
  + *Esta acción no está disponible para otros perfiles como recepcionistas o pacientes.*
  1. ***Solo el personal autorizado puede visualizar o modificar el historial* clínico de un paciente.**
  + El acceso está restringido según los privilegios definidos en el rol del usuario (médico o administrador).
  1. **Los pacientes solo pueden acceder a su propia información.**
  + Bajo ninguna circunstancia pueden visualizar historiales o recetas de otros usuarios.
  1. ***El sistema debe validar la existencia de stock disponible antes de registrar la entrega de medicamentos.***
  + *Si no hay stock suficiente, se debe bloquear la entrega y emitir una alerta.*
  1. ***Una receta solo puede ser utilizada una vez.***
  + *Una vez marcada como usada, no podrá volver a entregarse ningún medicamento relacionado con esa receta.*
  1. ***Las citas médicas no pueden agendarse en fechas pasadas ni en horarios donde el médico ya tiene otra cita asignada.***
  + *El sistema debe validar disponibilidad antes de confirmar la programación.*
  1. ***Solo los usuarios con rol administrador pueden crear, modificar o eliminar cuentas de usuario y roles.***
  + *Incluye la asignación de permisos de acceso.*
  1. ***Todas las acciones críticas deben quedar registradas en el sistema.***
  + *Esto incluye el registro, modificación o eliminación de pacientes, recetas, resultados y usuarios.*
  1. ***El registro de resultados de laboratorio debe estar vinculado a un paciente existente y validado.***
  + *No se pueden cargar resultados a pacientes inexistentes.*
  1. ***Las observaciones médicas y archivos adjuntos solo pueden ser añadidos por personal médico o de enfermería autorizado.***
  + *Se debe validar el tipo de usuario antes de permitir esta acción.*

**14. Requerimientos de interfaces externas**

**14.1. Interfaces de usuario**

El sistema de gestión de pacientes contará con interfaces diseñadas para distintos tipos de usuario, con el objetivo de facilitar el uso y garantizar la eficiencia en la interacción con el sistema. Las pantallas se diseñarán siguiendo principios de claridad, simplicidad y consistencia, aplicando estándares de diseño orientados a la experiencia del usuario (UX/UI).



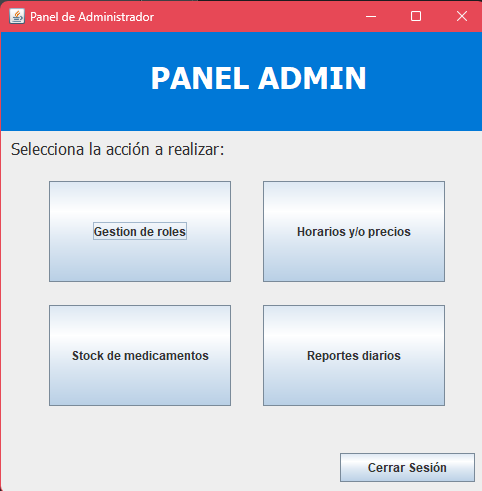
**14.2. Clasificación por tipo de usuario**

* **Administrador**

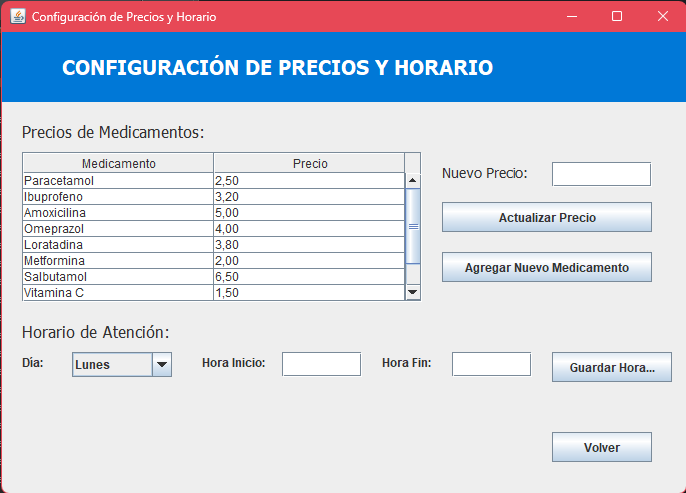
Acceso: Total.

Interfaz principal: Panel administrativo con pestañas para usuarios, roles, auditoría y reportes.

Funciones clave: Crear usuarios, asignar roles, revisar logs del sistema, etc.











* **Médico**

Acceso: Parcial (clínico).

Interfaz principal: Panel médico con acceso al historial clínico, recetas y resultados de laboratorio.

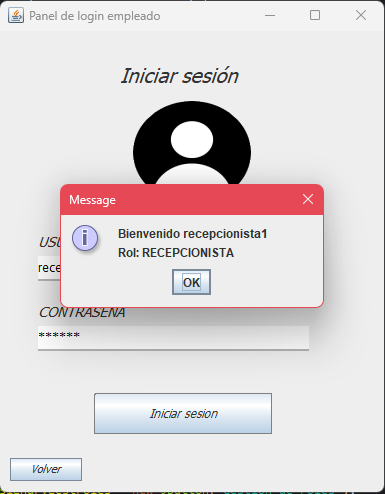
Funciones clave: Ver/editar historial, generar recetas, revisar alertas de resultados críticos.

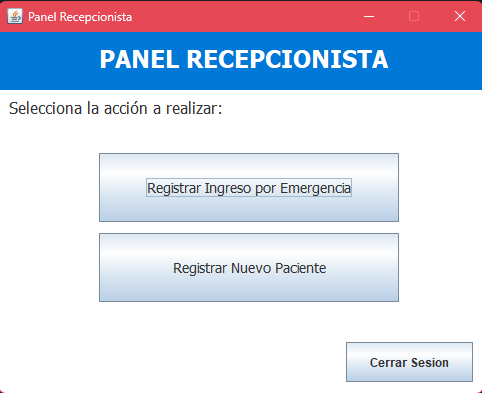
* **Recepcionista**

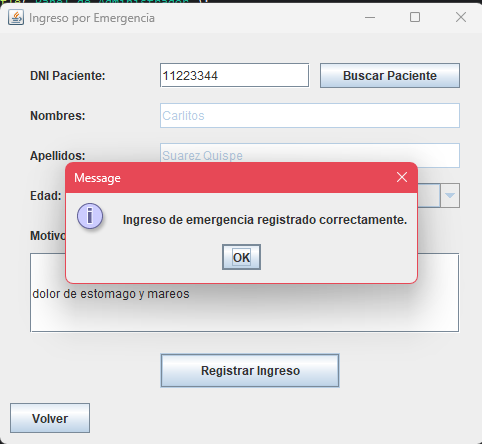
Acceso: Administrativo parcial.

Interfaz principal: Panel de agenda y pacientes.

Funciones clave: Registrar nuevos pacientes, programar o cancelar citas.







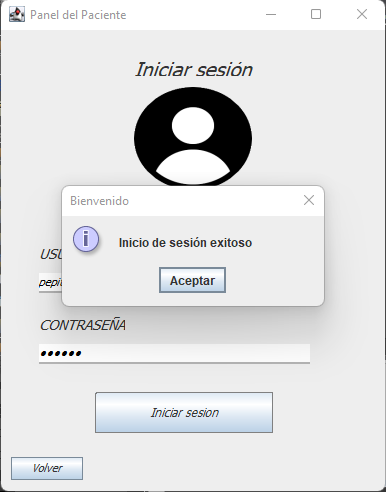


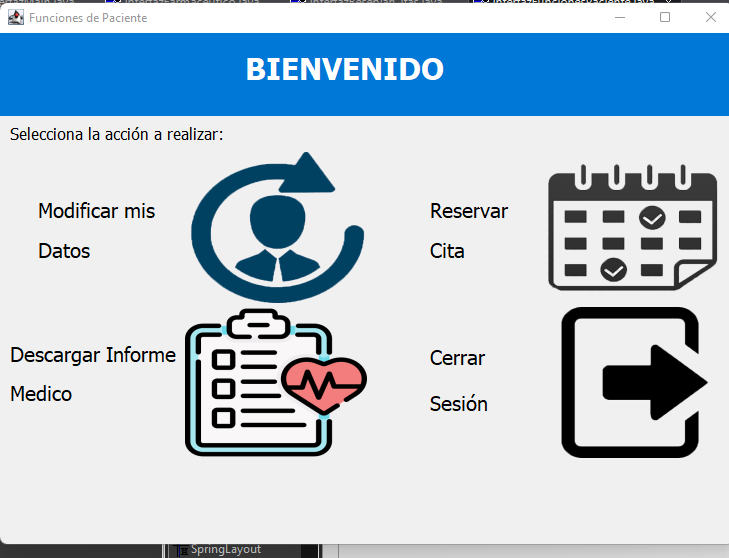
* **Paciente**

Acceso: Personal.

Interfaz principal: Portal con información personal y médica.

Funciones clave: Ver citas, recetas e historial, descargar informes.

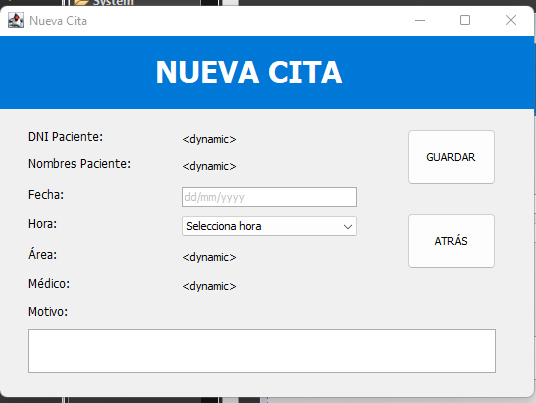
Interfaz de login paciente 

Interfaz de acciones de pacienteInterfaz modificar datos personales del paciente

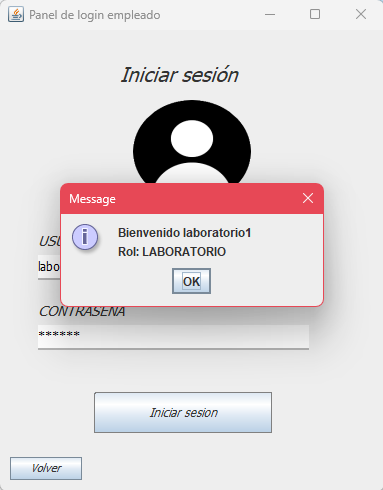


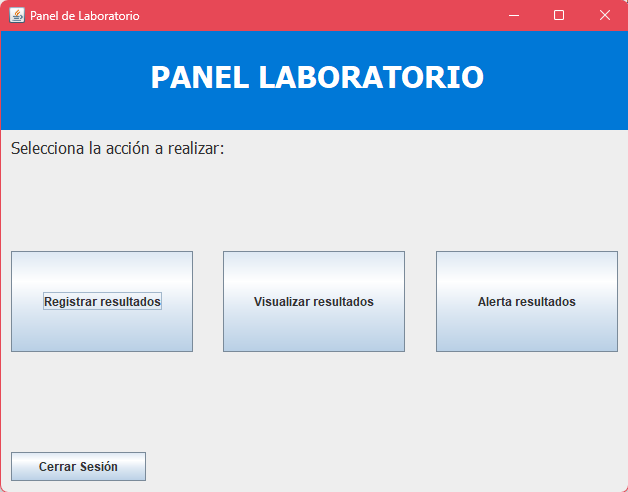
Interfaz para reservar citas



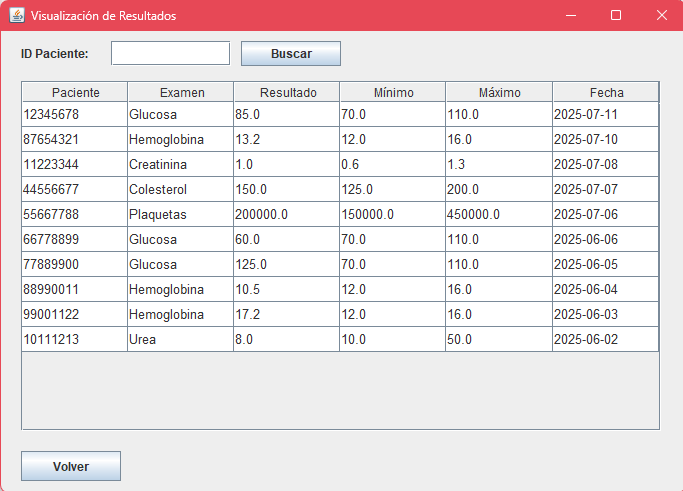


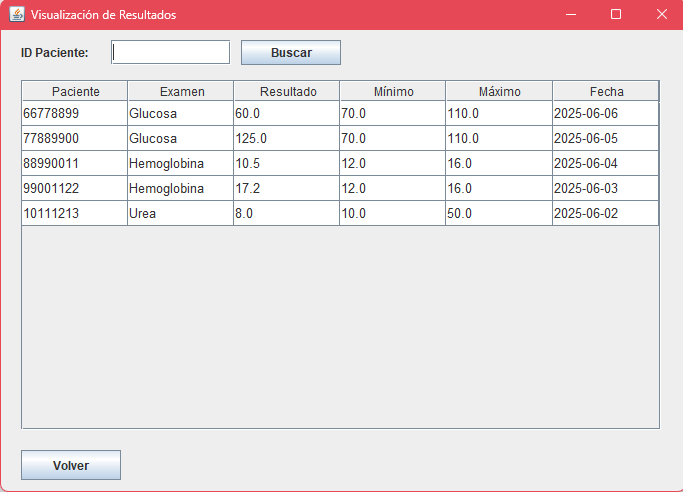
* **Laboratorio**

****

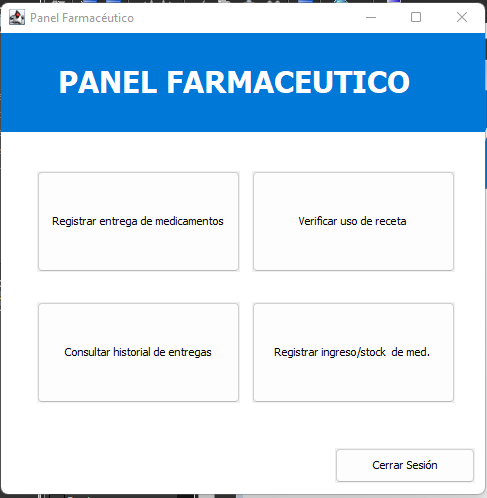
****

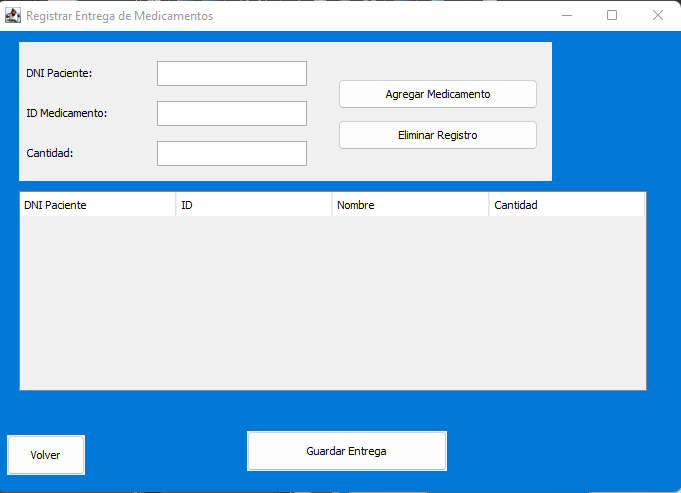
****

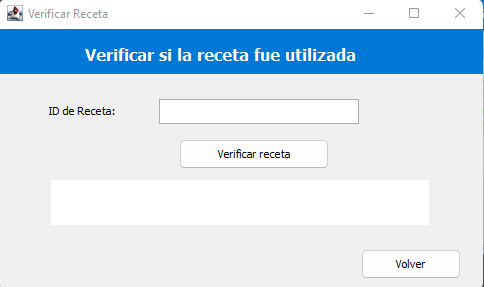
****

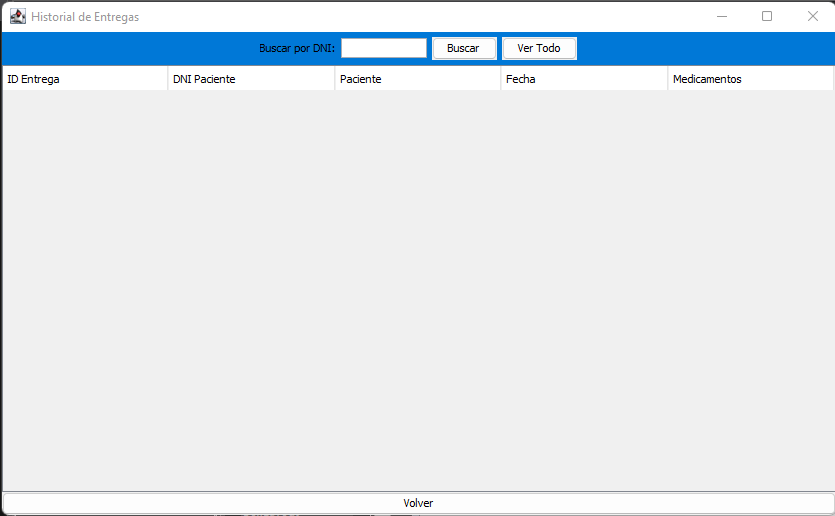
****

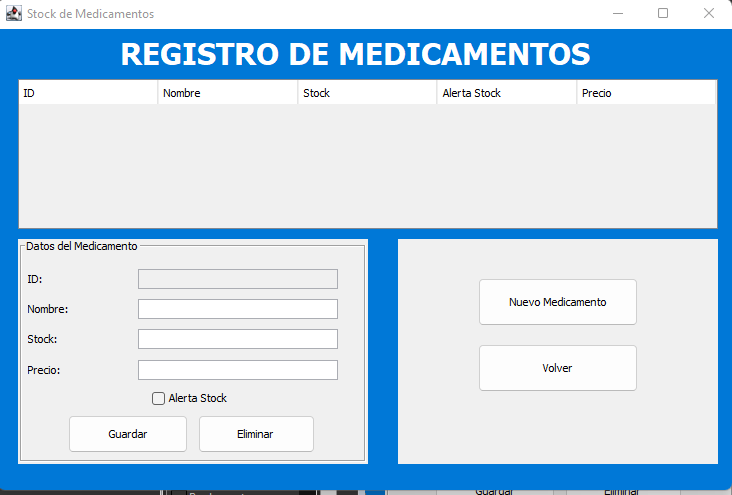
* **Farmaceutico**

****

****

****

****

****

**14.4. Estándares de interfaz gráfica (GUI)**

* Diseño responsivo**:** adaptable a distintos tamaños de pantalla (escritorio, tablet).
* Paleta de colores sobria y profesional: tonos azules y grises.
* Tipografía legible**:** uso de fuentes en tamaños accesibles (12–16 pt).
* Estructura modular por roles**:** menús o paneles visibles según tipo de usuario.

**14.5. guías de estilo (Estándares para los botones)**

*Tabla XII. Guías de estilo*Tabla

Descripción generada automáticamente

**Fuente:** elaboración propia

**15. Requerimientos no funcionales**

**15.1. Rendimiento**

* RNF-01: Las operaciones de registro y consulta no deben demorar más de 2 segundos bajo condiciones normales de uso.
* RNF-02: El sistema debe soportar al menos 20 usuarios concurrentes sin afectar la estabilidad.

**15.2. Mantenibilidad**

* RNF-03: El sistema debe estar modularizado, permitiendo incorporar nuevas funcionalidades sin alterar el núcleo existente.

**15.3. Portabilidad**

* RNF-04: El sistema debe poder ejecutarse en cualquier equipo con sistema operativo Windows 10 o superior.
* RNF-05: El sistema debe ser compatible con futuras migraciones a bases de datos MySQL o PostgreSQL.

**15.4. Usabilidad**

* RNF-06: Las interfaces deben ser intuitivas y fáciles de usar para usuarios sin experiencia técnica.
* RNF-07: Todo mensaje de error debe ser comprensible y guiar al usuario para corregirlo.

**15.5. Escalabilidad**

* RNF-08: El sistema debe permitir en el futuro la conexión con módulos externos (ejemplo: facturación electrónica).
* RNF-09: El sistema debe permitir crecimiento en el número de pacientes y registros sin comprometer el rendimiento.

**15.6. Documentación**

* RNF-09: El sistema debe contar con manual técnico y manual de usuario actualizados.
* RNF-10: Cada módulo (clase) debe incluir comentarios y documentación del código fuente.

**15.7. Seguridad**

* RNF-11: El sistema debe autenticar a los usuarios mediante nombre de usuario y contraseña antes de otorgar acceso.
* RNF-12: El historial clínico debe estar protegido para evitar acceso no autorizado.
* RNF-13: Toda modificación de datos críticos debe registrarse en un log de auditoría.
* RNF-14: El acceso a la información debe estar restringido por roles (paciente, medico, administrados, recepcionista, etc) .

**16. Glosario**

Descripción de términos y siglas necesarias para el entendimiento del documento de requerimientos de software.

| Término / Sigla | Descripción |
| --- | --- |
| Paciente | Persona registrada en el sistema que recibe atención médica. |
| Cita médica | Compromiso agendado entre un paciente y un médico para una consulta. |
| Historial clínico | Registro cronológico de antecedentes médicos y atención del paciente. |
| Receta médica | Documento emitido por el médico con la prescripción de medicamentos. |
| Medicamento | Sustancia farmacológica prescrita a un paciente. |
| Resultado de laboratorio | Documento o registro que contiene el resultado de un análisis clínico. |
| Emergencia | Ingreso no programado del paciente por condición crítica o urgente. |
| Vacunación | Registro de la aplicación de una o más vacunas a un paciente. |
| Usuario | Persona que accede al sistema mediante un nombre de usuario y contraseña. |
| Rol | Permisos o funciones asignadas a un usuario dentro del sistema. |
| Administrador | Usuario con control total sobre el sistema, configuración y seguridad. |
| Recepcionista | Usuario encargado del registro de pacientes y gestión de citas. |
| Médico | Profesional autorizado a atender pacientes, emitir recetas y registrar diagnósticos. |
| Observación médica | Comentario o anotación realizada por el personal médico o de enfermería. |
| PDF | Formato de archivo utilizado para adjuntar informes clínicos. |
| Base de datos | Conjunto estructurado de información que almacena registros del sistema. |
| Java | Lenguaje de programación utilizado para desarrollar el sistema. |
| Visual Studio Code | Entorno de desarrollo (IDE) utilizado para programar en Java. |
| SQL Server | Sistema de gestión de base de datos relacional utilizado por el sistema. |
| SSMS | SQL Server Management Studio, herramienta para administrar SQL Server. |
| Scrum | Metodología ágil usada para gestionar el desarrollo del sistema. |
| Requerimiento funcional (RF) | Descripción de una funcionalidad que el sistema debe realizar. |
| Requerimiento no funcional (RNF) | Condiciones de calidad o restricciones generales del sistema. |
| GUI (Graphical User Interface) | Interfaz gráfica de usuario que permite interactuar con el sistema. |
| Log de auditoría | Registro de las acciones realizadas por los usuarios dentro del sistema. |
| Alta médica | Registro del egreso de un paciente luego de su atención. |
| Reporte mensual | Documento generado con estadísticas del uso del sistema en un periodo. |

**17. Bibliografía**

* Añel Rodríguez, R. M., García Alfaro, I., Bravo Toledo, R., & Carballeira Rodríguez, J. D. (2021). Historia clínica y receta electrónica: riesgos y beneficios detectados desde su implementación. Vol. (*53*) <https://www.sciencedirect.com/science/article/pii/S0212656721002547?via%3Dihub>
* Barreno Cortijo, R. R., & Delgado Rojas, W. M. (2023). *Implementación de sistema integrado de gestión hospitalaria para hospitales del sector público* (Trabajo de suficiencia profesional, Universidad Peruana de Ciencias Aplicadas). Repositorio Académico UPC. <http://hdl.handle.net/10757/671478>
* Crispin Pari, N., & Mayhua de la Cruz, J. A. (2024). Implementación de un sistema de gestión hospitalaria, para mejorar la atención de pacientes en el Hospital Departamental de Huancavelica [Tesis de pregrado, Universidad Nacional de Huancavelica]. Repositorio Institucional UNH. <https://repositorio.unh.edu.pe/items/99d62648-93dd-4dea-984e-f97b96eabb9a>
* Gutarra Mejía, C. R., Quiroga Rosas, R. C. (2014). *Implementación de un sistema de historias clínicas electrónicas para el Centro de Salud Perú 3era zona (Trabajo de suficiencia profesional, Universidad San Martin de Porres). Repositorio Académico USMP.* [*https://repositorio.usmp.edu.pe/handle/20.500.12727/1463*](https://repositorio.usmp.edu.pe/handle/20.500.12727/1463)
* *Mazco Puma, S. (2012). Sistema integral de gestión hospitalaria utilizando la arquitectura orientada a servicios, para el Hospital de Baja Complejidad Vitarte [Tesis de grado, Universidad Nacional del Altiplano]. Repositorio Institucional UNAP.* [*http://repositorio.unap.edu.pe/handle/20.500.14082/2368*](http://repositorio.unap.edu.pe/handle/20.500.14082/2368)
* Ordoñez Calero, H. D. (2018). Evaluación de la calidad en uso de productos de software para gestión de historias clínicas [Tesis de maestría, Escuela Politécnica Nacional]. Repositorio institucional. <https://bibdigital.epn.edu.ec/handle/15000/19731>