

**Universidad Tecnológica del Perú**

**Escuela Profesional:**

Ingeniería de Sistemas e Informática

**Curso:**

Curso Integrador I: Sistemas Software

**Ciclo:**

2025 - Ciclo 01 marzo

**Docente:**

Carlos Alberto Chirinos Mundaca

**Integrantes:**

* Farro Zapata Jhonatan Alfredo
* Castillo Dávila Amy Rubí
* Ramos Vásquez Harold Jeanpierre
* Santisteban Coico Estefano Rodrigo

1. **Descripción del Sistema**

El Sistema de Gestión de Salud es una aplicación diseñada para facilitar la administración de citas médicas, historial clínico de pacientes, gestión de personal médico y la generación de reportes clínicos. Su objetivo es mejorar la eficiencia y la organización en los centros de salud, permitiendo un manejo adecuado de la información relacionada con pacientes y profesionales de la salud.

**Componentes Principales:**

* Citas Médicas: Los pacientes pueden reservar citas médicas, las cuales se asignan a médicos disponibles y se gestionan en el sistema.
* Historial Clínico: Cada cita médica se asocia a un historial clínico, donde se registran diagnósticos y procedimientos realizados.
* Gestión de Personal: El sistema permite gestionar los datos de los profesionales de la salud (médicos, enfermeros, etc.).
* Generación de Reportes Clínicos: Los médicos pueden generar reportes basados en los exámenes realizados a los pacientes.
* Establecimientos de Salud: Los centros médicos y hospitales se gestionan en el sistema, permitiendo asociar a cada paciente y profesional con un establecimiento específico.

1. **Base de Datos**

La base de datos utilizada en el sistema está basada en SQL Server y contiene diversas tablas relacionadas entre sí para gestionar la información del sistema. Algunas de las tablas más relevantes son:

* Cita: Almacena información sobre las citas médicas de los pacientes, como la fecha, hora, y estado de la cita, y asocia a los pacientes con los profesionales y establecimientos de salud correspondientes.
* Historial: Registra los diagnósticos y resultados de cada cita médica, generando un historial clínico para cada paciente.
* Establecimiento: Contiene los datos de los hospitales y centros de salud, con información como nombre, red y microred.
* Paciente: Almacena los datos personales de los pacientes, incluyendo su DNI, nombre, y fecha de nacimiento.
* Personal: Gestiona la información de los médicos y otros profesionales de la salud, asociándolos con su profesión y el establecimiento de trabajo.
* Profesion: Define las profesiones de los empleados de salud (por ejemplo, médico, enfermero, etc.).

Las relaciones entre las tablas se gestionan a través de claves foráneas. Por ejemplo, las citas están relacionadas con los pacientes, el personal de salud y el establecimiento de atención.

1. **Software Utilizado (Programa centroSalud-main)**

El software está desarrollado en Java, utilizando una estructura Modelo-Vista-Controlador (MVC). La aplicación está organizada de la siguiente manera:

* Backend (Lógica de Negocio): Utiliza Spring Boot para crear una API RESTful que permite la interacción con la base de datos y la lógica del sistema.
* Interfaz Gráfica (GUI): La interfaz está construida con Java Swing o JavaFX, con formularios diseñados para facilitar la interacción del usuario.

Archivos relevantes:

Atencion.java / Atencion.form: Gestión de la atención médica.

Cita.java / Cita.form: Gestión de citas médicas.

GestionarPersonal.java / GestionarPersonal.form: Gestión de personal médico.

Reportes.java / Reportes.form: Generación de reportes médicos.

1. **APIs Utilizadas y Descripción**

Aunque no se especifican APIs externas directamente en el código, el sistema puede integrarse con varias APIs para mejorar su funcionalidad:

* API de Autenticación: Para gestionar el inicio de sesión de usuarios y mantener sesiones seguras.
* API de Generación de Reportes: Para exportar los reportes clínicos a formatos como PDF o Excel.
* API de Consulta de Identidad: La clase ConsultaReniec.java indica que podría haber integración con sistemas externos (como RENIEC) para validar la identidad de los pacientes.

1. **Extensiones y Funcionalidades Adicionales**

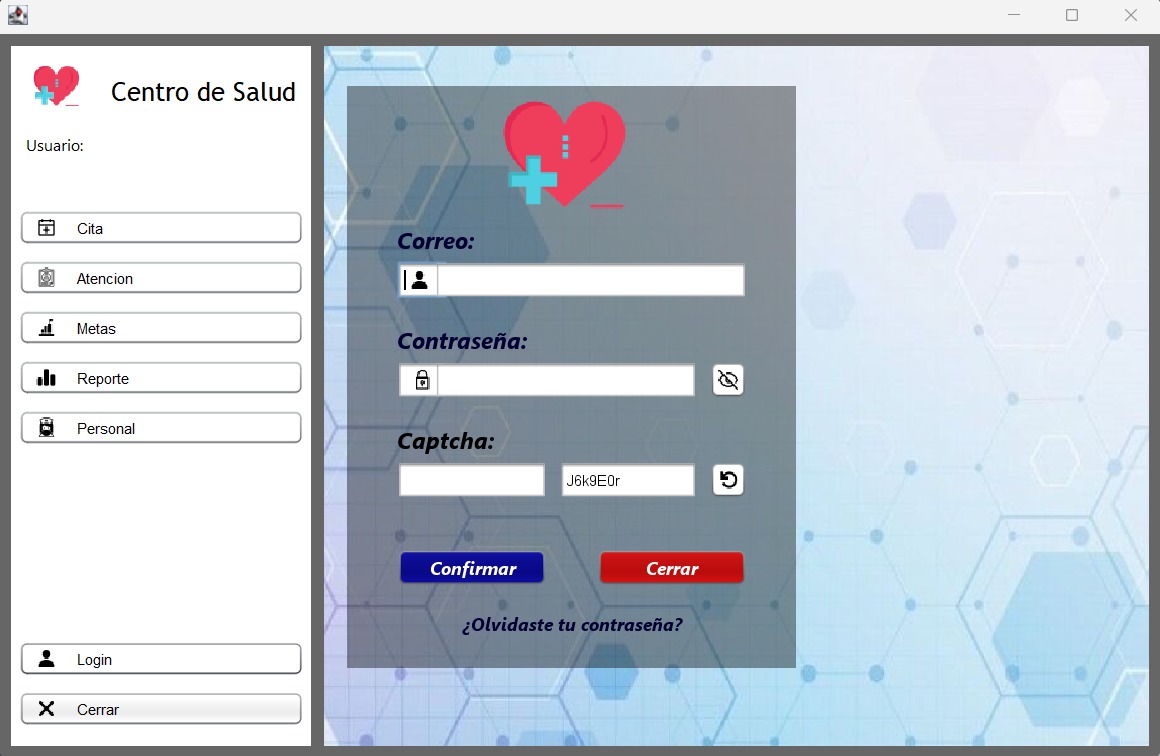
El sistema puede ampliarse con varias extensiones según las necesidades del proyecto. Algunas posibles extensiones incluyen:

* Autenticación avanzada: Implementar autenticación mediante tokens JWT o integración con servicios de autenticación de terceros (OAuth).
* Generación de informes personalizados: Uso de bibliotecas como Apache POI o iText para la generación de reportes en formatos PDF o Excel.
* Integración con servicios externos: Posibilidad de conectar el sistema con bases de datos o servicios de salud externos para la verificación de datos de pacientes o la importación de resultados de exámenes médicos.

1. **Cómo Trabaja el Sistema**

El sistema opera de manera modular, con una clara separación entre la interfaz gráfica (GUI), la lógica de negocio y la base de datos. Los usuarios pueden interactuar con la aplicación a través de formularios que permiten:

* Registro de Citas: Los pacientes pueden agendar citas con los médicos disponibles, y el sistema mantiene un registro detallado de cada cita.
* Atención Médica: Los médicos realizan la atención de los pacientes durante las citas y registran los diagnósticos y resultados de los exámenes en el historial del paciente.
* Generación de Reportes: Los médicos pueden generar reportes sobre los exámenes realizados y sus resultados.
* Gestión de Usuarios: Los administradores pueden gestionar los usuarios del sistema, asignar roles y controlar el acceso a las funcionalidades.



Para poder ingresar al Login, existe 2 modos:

* **Modo administrador:**

**Usuario**: Julianazapata@gmail.com

Contraseña: Juliana10.

UsuarioJhonatanfarro@gmail.com

Contraseña: Jhonatan10.

* **Modo personal**

**Usuario limitado**: gestiona solo citas y atención

Usuario: Dianalitano@gmail.com

Contraseña: Diana10.

**Jefe de área:** gestiona todo menos usuarios

Usuario: Julianazapata@gmail.com

Contraseña: Juliana10.

**Procesos Clave del Sistema**

1. Proceso de Gestión de Citas Médicas

Descripción del Proceso:

El sistema permite a los pacientes gestionar sus citas médicas a través de una interfaz gráfica sencilla e intuitiva. Los pacientes pueden seleccionar la fecha y hora que les convenga, además de elegir al médico disponible según su especialidad.

Flujo del Proceso:

* Creación de la Cita: El paciente ingresa sus datos personales y selecciona el médico, fecha y hora disponible a través del formulario de Cita en la interfaz.
* Confirmación: Una vez creada la cita, el sistema guarda la información en la base de datos bajo la tabla Cita, asociando el paciente con el médico y el establecimiento de salud correspondiente.
* Estado de la Cita: El sistema mantiene un seguimiento del estado de la cita (pendiente, atendida, vencida) para facilitar la administración.
* Notificación: El sistema envía una confirmación al paciente y a la consulta médica correspondiente, registrando cualquier cambio o cancelación de la cita.

Tablas Involucradas:

* Cita: Almacena los detalles de la cita (fecha, hora, estado, ID del paciente y médico).
* Paciente: Relaciona al paciente con la cita.
* Personal: Asocia al médico con la cita.
* Establecimiento: Relaciona la cita con el centro de salud donde se realizará.

2. Proceso de Generación de Reportes Clínicos

Descripción del Proceso:

Este proceso permite a los médicos generar reportes clínicos basados en los exámenes realizados durante las citas médicas, proporcionando un historial detallado y un análisis de los resultados obtenidos.

Flujo del Proceso:

Registro de Exámenes: Durante la cita, el médico realiza el examen correspondiente (por ejemplo, PAP, VIH, VPH, entre otros) y registra los resultados en el Historial Clínico del paciente.

Generación del Reporte:

Una vez que se han registrado los resultados de los exámenes, el médico puede generar un Reporte Clínico desde la interfaz de usuario, seleccionando el tipo de examen y el período de tiempo relevante.

Exportación de Reporte:

El sistema permite al médico exportar el reporte en formato PDF o Excel, proporcionando una versión imprimible del informe médico para el paciente o para otros profesionales de la salud.

Almacenamiento:

El reporte generado se guarda en la base de datos, asociando los detalles del reporte con el paciente, el médico y el establecimiento de salud.

Tablas Involucradas:

* ReporteClinico: Almacena los detalles del reporte generado (nombre del reporte, fecha, cantidad de exámenes realizados).
* Paciente: Asocia el reporte con el paciente que ha recibido la atención médica.
* Atención: Vincula el reporte con el tipo de atención médica recibida.
* Personal: Relaciona al médico que generó el reporte con los exámenes realizados

Informe de Proyecto: Sistema de Gestión de Salud

1. Descripción del Sistema

El Sistema de Gestión de Salud es una aplicación diseñada para facilitar la administración de citas médicas, historial clínico de pacientes, gestión de personal médico y la generación de reportes clínicos. Su objetivo es mejorar la eficiencia y la organización en los centros de salud, permitiendo un manejo adecuado de la información relacionada con pacientes y profesionales de la salud.

Componentes Principales:

* Citas Médicas: Los pacientes pueden reservar citas médicas, las cuales se asignan a médicos disponibles y se gestionan en el sistema.
* Historial Clínico: Cada cita médica se asocia a un historial clínico, donde se registran diagnósticos y procedimientos realizados.
* Gestión de Personal: El sistema permite gestionar los datos de los profesionales de la salud (médicos, enfermeros, etc.).
* Generación de Reportes Clínicos: Los médicos pueden generar reportes basados en los exámenes realizados a los pacientes.
* Establecimientos de Salud: Los centros médicos y hospitales se gestionan en el sistema, permitiendo asociar a cada paciente y profesional con un establecimiento específico.

2. Base de Datos

La base de datos utilizada en el sistema está basada en SQL Server y contiene diversas tablas relacionadas entre sí para gestionar la información del sistema. Algunas de las tablas más relevantes son:

* Cita: Almacena información sobre las citas médicas de los pacientes, como la fecha, hora, y estado de la cita, y asocia a los pacientes con los profesionales y establecimientos de salud correspondientes.
* Historial: Registra los diagnósticos y resultados de cada cita médica, generando un historial clínico para cada paciente.
* Establecimiento: Contiene los datos de los hospitales y centros de salud, con información como nombre, red y microred.
* Paciente: Almacena los datos personales de los pacientes, incluyendo su DNI, nombre, y fecha de nacimiento.
* Personal: Gestiona la información de los médicos y otros profesionales de la salud, asociándolos con su profesión y el establecimiento de trabajo.
* Profesion: Define las profesiones de los empleados de salud (por ejemplo, médico, enfermero, etc.).

Las relaciones entre las tablas se gestionan a través de claves foráneas. Por ejemplo, las citas están relacionadas con los pacientes, el personal de salud y el establecimiento de atención.

3. Software Utilizado (Programa centroSalud-main)

El software está desarrollado en Java, utilizando una estructura Modelo-Vista-Controlador (MVC). La aplicación está organizada de la siguiente manera:

Backend (Lógica de Negocio): Utiliza Spring Boot para crear una API RESTful que permite la interacción con la base de datos y la lógica del sistema.

Interfaz Gráfica (GUI): La interfaz está construida con Java Swing o JavaFX, con formularios diseñados para facilitar la interacción del usuario.

Archivos relevantes:

* Atencion.java / Atencion.form: Gestión de la atención médica.
* Cita.java / Cita.form: Gestión de citas médicas.
* GestionarPersonal.java / GestionarPersonal.form: Gestión de personal médico.
* Reportes.java / Reportes.form: Generación de reportes médicos.

4. APIs Utilizadas y Descripción

Aunque no se especifican APIs externas directamente en el código, el sistema puede integrarse con varias APIs para mejorar su funcionalidad:

API de Autenticación: Para gestionar el inicio de sesión de usuarios y mantener sesiones seguras. Se podría implementar utilizando JWT (JSON Web Tokens).

API de Generación de Reportes: Para exportar los reportes clínicos a formatos como PDF o Excel.

API de Consulta de Identidad: La clase ConsultaReniec.java indica que podría haber integración con sistemas externos (como RENIEC) para validar la identidad de los pacientes.

5. Extensiones y Funcionalidades Adicionales

El sistema puede ampliarse con varias extensiones según las necesidades del proyecto. Algunas posibles extensiones incluyen:

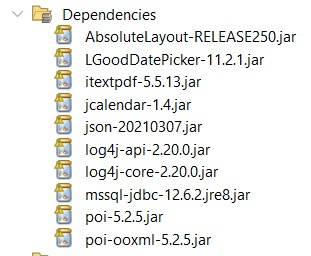
* Autenticación avanzada: Implementar autenticación mediante tokens JWT o integración con servicios de autenticación de terceros (OAuth).
* Generación de informes personalizados: Uso de bibliotecas como Apache POI o iText para la generación de reportes en formatos PDF o Excel.
* Integración con servicios externos: Posibilidad de conectar el sistema con bases de datos o servicios de salud externos para la verificación de datos de pacientes o la importación de resultados de exámenes médicos.

6. Cómo Trabaja el Sistema

El sistema opera de manera modular, con una clara separación entre la interfaz gráfica (GUI), la lógica de negocio y la base de datos. Los usuarios pueden interactuar con la aplicación a través de formularios que permiten:

* Registro de Citas: Los pacientes pueden agendar citas con los médicos disponibles, y el sistema mantiene un registro detallado de cada cita.
* Atención Médica: Los médicos realizan la atención de los pacientes durante las citas y registran los diagnósticos y resultados de los exámenes en el historial del paciente.
* Generación de Reportes: Los médicos pueden generar reportes sobre los exámenes realizados y sus resultados.
* Gestión de Usuarios: Los administradores pueden gestionar los usuarios del sistema, asignar roles y controlar el acceso a las funcionalidades.

**Extensiones:**



* AbsoluteLayout-RELEASE250.jar:

Esta librería proporciona un gestor de diseño absoluto para interfaces gráficas en Java. Permite posicionar los componentes en un contenedor de manera fija y sin restricciones de alineación, facilitando la creación de interfaces personalizadas.

* LGoodDatePicker-11.2.1.jar:

Ofrece un selector de fechas avanzado para aplicaciones gráficas, mejorando la experiencia de usuario al permitir una selección visualmente atractiva y personalizable de fechas.

* itextpdf-5.5.13.jar:

iText PDF es una librería que facilita la creación y manipulación de documentos PDF en Java. Esta librería permite generar documentos, agregar texto, imágenes, tablas, y otras funcionalidades relacionadas con PDFs.

* jcalendar-1.4.jar:

Esta librería proporciona un componente de calendario que se integra en las interfaces gráficas de Java, permitiendo que los usuarios seleccionen fechas de forma interactiva.

* json-20210307.jar:

JSON-java es una librería para trabajar con datos en formato JSON. Facilita la lectura y escritura de JSON en aplicaciones Java, muy útil para intercambiar datos entre aplicaciones.

* log4j-api-2.20.0.jar y log4j-core-2.20.0.jar:

Apache Log4j es una librería para gestionar logs y registros de actividad en aplicaciones Java. Estas dependencias permiten realizar un seguimiento detallado de las operaciones y posibles errores en el sistema, facilitando el diagnóstico y la auditoría.

* mssql-jdbc-12.6.2.jre8.jar:

Esta librería es un controlador JDBC utilizado para conectar aplicaciones Java con bases de datos Microsoft SQL Server. Permite realizar consultas y actualizaciones de datos en SQL Server desde la aplicación.

* poi-5.2.5.jar y poi-ooxml-5.2.5.jar:

Apache POI es una librería que permite leer y escribir archivos Microsoft Office (como Excel, Word, PowerPoint) desde Java. La dependencia poi-ooxml se especializa en trabajar con archivos de Excel en formato OOXML (XLSX)

**Anexos:**

