**Guatemala 14 de 06 de 2025**

**Tema:**

**Sistema Hospitalario**

**Alumno:** Gary Orlando Ochoa Quiroa - 2349076

**Alumno:** Carlos Enrique Orozco Menéndez - 2349077

**Alumno:** Juan José Antonio Pérez García - 2349044

**Alumno:** Bryan Leonardo Santiagos Galicia -

**Catedrático:** Ing. William González

**Curso:** Programación II – Quinto semestre

# **ÍNDICE**

**Navegación general**

[**ÍNDICE** 2](#_Toc196834193)

[**INTRODUCCIÓN** 3](#_Toc196834194)

[**OBJETIVOS** 4](#_Toc196834195)

[**CONTENIDO** 5](#_Toc196834196)

[**CONCLUSION** 6](#_Toc196834197)

[**RECOMENDACIONES** 7](#_Toc196834198)

[**E-GRAFÍAS/REFERENCIAS** 8](#_Toc196834199)

# **INTRODUCCIÓN**

En el ámbito de la tecnología y el desarrollo de software, la integración de múltiples lenguajes de programación y tecnologías es esencial para crear aplicaciones robustas y eficientes. Este proyecto tiene como objetivo desarrollar una aplicación utilizando Java, SQL y C# para la creación y gestión de APIs. A lo largo del documento, se detallan los procesos, componentes y objetivos del proyecto.

El proyecto se desarrollará siguiendo una metodología ágil, permitiendo iteraciones rápidas y adaptaciones según las necesidades del usuario y los requisitos del sistema. Se utilizarán herramientas de control de versiones como Git para gestionar el código fuente y asegurar la colaboración efectiva entre los miembros del equipo con “commits” periódicos para el avance de los objetivos listados.

# **OBJETIVOS**

1. Desarrollo en Java: Utilizar Java como el lenguaje principal para la lógica de negocio y la interfaz de usuario. Java es conocido por su portabilidad y robustez, lo que lo convierte en una excelente opción para aplicaciones empresariales.
2. Base de Datos con SQL: Implementar una base de datos relacional utilizando SQL para el almacenamiento y gestión de datos. SQL es el estándar para la manipulación de bases de datos y permite realizar consultas eficientes y seguras.
3. Creación de APIs con C#: Desarrollar APIs utilizando C# y el framework .NET. Las APIs permitirán la comunicación entre diferentes componentes de la aplicación y facilitarán la integración con otros sistemas.

# **CONTENIDO**

# DEFINICIÓN DE ACRÓNIMOS Y ABREVIATURAS

|  |  |
| --- | --- |
| Término | Concepto |
| API | Una API, o interfaz de programación de aplicaciones, es un conjunto de reglas o protocolos que permiten que las aplicaciones de software se comuniquen entre sí para intercambiar datos, características y funcionalidades. |
| Back-end | Arquitectura interna de un sitio web. Esta área lógica, que no es visible a los ojos del usuario; se encarga de la lógica de negocio, de recibir y devolver datos procesados. |
| (UML)Casos de uso | Un caso de uso es la descripción de una acción o actividad. Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso. |
| CRUD | Significa "Crear, Leer, Actualizar y Eliminar" y son las cuatro operaciones básicas de las bases de datos. |
| Front-end | Es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla. |
| HTTP | El protocolo de transferencia de hipertexto o HTTP (HyperText Transfer Protocol) es el protocolo de red que permite la transferencia de documentos de hipermedia en la red, generalmente entre un navegador y un servidor, para que los humanos puedan leerlos. |
| MySQL | Sistema de gestión de bases de datos relacional (forma de estructurar información en tablas, filas y columnas. |
| PDF | Formato Portátil de Documento, usado para mostrar documentos en la forma electrónica. |
| POO | Programación Orientada a Objetos. Enfoque de desarrollo donde el sistema se estructura en clases que modelan entidades con atributos y métodos. |
| REST | Interfaz para conectar varios sistemas basados en el protocolo HTTP; sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como JSON (formato de texto; utiliza convenios que resultan familiares entre distintos lenguajes de programación). |
| XAMPP | Paquete de software libre que incluye Apache, MySQL, PHP y Perl para montar un servidor local de pruebas. |
| Postman | Herramienta utilizada para probar APIs. Permite enviar peticiones HTTP y visualizar respuestas. |
| Java Swing | Es una biblioteca gráfica (GUI) de Java que permite crear interfaces de usuario de aplicaciones de escritorio. Proporciona componentes como ventanas, botones, tablas, campos de texto, etc. |
| UML | lenguaje de modelado estándar que se utiliza para visualizar un plan arquitectónico para elementos como actividades, procesos de negocio y esquemas de base de datos. |
| (UML) Diagrama de clases | Representación visual de la estructura de un sistema, describe los tipos de objetos del sistema y los diversos tipos de relaciones estáticas que existen entre ellos. |
| (UML) Diagrama de secuencia | Tipo de diagrama de interacción que muestra cómo, y en qué orden, un conjunto de objetos interactúa entre sí en un proceso. |
| Inyección SQL | Es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una base de datos. |
| ORM | Es un modelo de programación que permite mapear las estructuras de una base de datos relacional. |
| Java | Lenguaje de programación multiplataforma orientada a objetos y de propósito general ampliamente utilizado en el desarrollo de software. |
| C# | Lenguaje de programación de alto nivel y propósito general, desarrollado por Microsoft, se utiliza para crear una amplia variedad de aplicaciones. |
| .NET | marco multiplataforma y de código abierto que nos sirve para compilar aplicaciones que puedan ejecutarse en cualquier plataforma**,** admite lenguajes de programación mas no que es uno. Algunos de estos que admite son: Visual Basic, C# y F#. |
| MVC | Siglas de Modelo-Vista-Controlador. Patrón de arquitectura de software que divide la aplicación en tres componentes interconectados. el Modelo (que maneja los datos), la Vista (que muestra la interfaz de usuario) y el Controlador (que interacciona entre la Vista y el Modelo). |
| Apache NetBeans | Entorno de Desarrollo Integrado (IDE) de código abierto y gratuito que facilita el desarrollo de aplicaciones web, corporativas, de escritorio y móviles, basado en el lenguaje Java y ejecutado en Swing. |

# Actores

|  |  |
| --- | --- |
| Rol | Permisos principales |
| Administrador | Acceso total al sistema, incluyendo gestión de usuarios, médicos, pacientes, servicios, reportes, historial y facturas. |
| Médico | Visualiza y actualiza historial médico de pacientes. |
| Recepcionista | Gestión de citas y facturación. |
| Gestor | CRUD completo sobre módulos de pacientes, médicos, servicios, facturas. *(Puedes redefinir este rol como “gestor” o “secretaria administrativa” por claridad)* |

# Diagrama de flujoC:\Users\juana\OneDrive\Desktop\Universidad\Progra II\diagrama de flujo pacientes.png

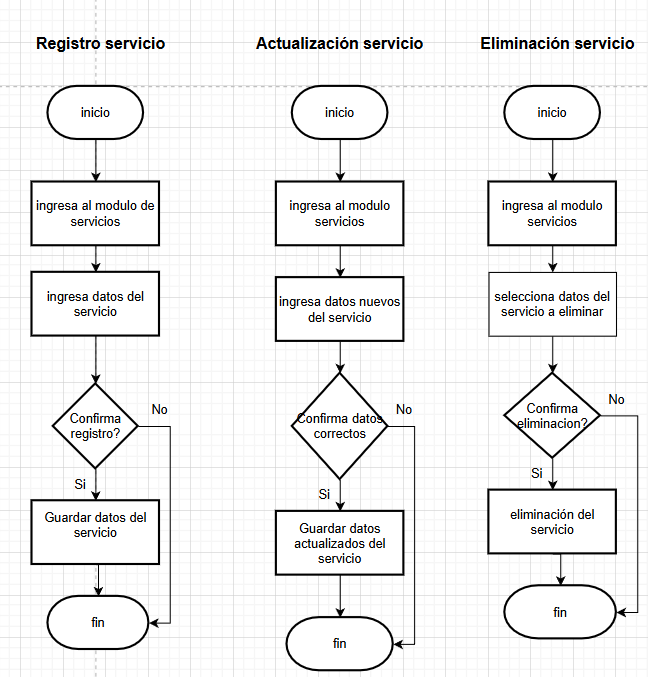
No

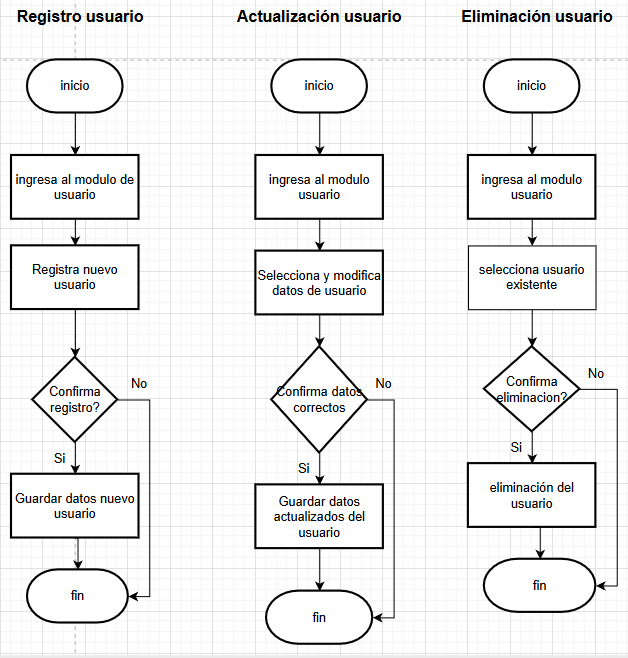
**Eliminar medico**

**Actualizar medico**

**Registrar medico**

# 





# Módulos

## Modulo pacientes:

El módulo de pacientes permite la gestión básica de los registros de pacientes dentro del sistema hospitalario a través de operaciones CRUD. Este módulo es fundamental para mantener organizada y actualizada la información esencial de los pacientes.

## Modulo Medico:

Este módulo gestiona la información de los médicos que prestan servicios en el hospital. Permite mantener un registro actualizado de su perfil profesional y asociarlos con especialidades y horarios de atención.

## Módulo de Servicios:

Gestiona los diferentes servicios médicos y administrativos que ofrece el hospital, como consultas, estudios de laboratorio, cirugías, y demás servicios que se puedan integrar posteriormente.

## Módulo de citas:

Permite gestionar las citas médicas entre pacientes y médicos, registrando fecha, hora, y el medico asignado.

## Módulo de facturación:

Este módulo registra los cargos asociados a los servicios prestados a un paciente, generando el detalle correspondiente a los servicios prestados por el hospital.

# Casos de uso

## Gestión de pacientes

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Registrar paciente |
| Actor | Administrador, Gestor |
| Descripción breve | Permite ingresar a un nuevo paciente al sistema hospitalario. |
| Precondiciones | • El usuario debe haber iniciado sesión.  • Tener permisos en el módulo de pacientes. |
| Flujo básico | 1. El usuario accede al módulo de pacientes. 2. Completa los datos requeridos (nombre, apellido, etc). 3. Confirma el registro. 4. 5. El sistema guarda el paciente en la base de datos. |
| Flujos alternativos | N/A |
| Excepciones | 4a. Error de conexión con la base de datos: el sistema muestra un mensaje de error. |
| Resultado esperado | El paciente queda registrado correctamente en el sistema. |

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Actualizar datos del paciente |
| Actor | Administrador, Gestor |
| Descripción breve | Permite modificar la información existente de un paciente. |
| Precondiciones | • El paciente debe estar registrado.  • El usuario debe tener permisos adecuados. |
| Flujo básico | 1. El usuario selecciona un registro de la tabla.  2. Los datos se cargan en los campos del formulario  3. Modifica los datos necesarios.  4. clic en el botón actualizar.  5. El sistema actualiza los datos en la base. |
| Flujos alternativos | N/A |
| Excepciones | 4a. Error al guardar: el sistema muestra Error al actualizar paciente. |
| Resultado esperado | Los datos del paciente son actualizados correctamente. |

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Eliminar paciente |
| Actor | Administrador, Gestor |
| Descripción breve | Permite eliminar del sistema el registro de un paciente existente. |
| Precondiciones | • El paciente debe estar registrado.  • El usuario debe tener permisos de eliminación. |
| Flujo básico | 1. El usuario accede al módulo de pacientes.  2. Selecciona un registro de la tabla.  3. Selecciona la opción "Eliminar".  4. Confirma la eliminación.  5. El sistema elimina el registro del paciente de la base de datos. |
| Flujos alternativos | 2A al seleccionar el registro, el usuario puede solo inactivar al paciente, el cual se marcará en un color distintivo. |
| Excepciones | 4a. Error en la eliminación por dependencia de datos: el sistema impide el proceso y muestra un mensaje. |
| Resultado esperado | El paciente es eliminado. |

## Gestión de médicos

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Registrar médico |
| Actor | Administrador, Gestor |
| Descripción breve | Permite registrar un nuevo médico en el sistema con sus datos personales y profesionales. |
| Precondiciones | • El usuario debe haber iniciado sesión.  • Tener permisos sobre el módulo de médicos. |
| Flujo básico | 1. El usuario accede al módulo de médicos.  2. Ingresa los datos personales (nombre, apellido, especialidad, correo, etc.).  3. Confirma el registro.  4. El sistema almacena el nuevo médico. |
| Flujos alternativos | N/A |
| Excepciones | 4a. Error al registrar paciente |
| Resultado esperado | El médico queda registrado exitosamente. |

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Actualizar médico |
| Actor | Administrador, Gestor |
| Descripción breve | Permite modificar la información registrada de un médico. |
| Precondiciones | • El médico debe estar registrado. |
| Flujo básico | 1. El usuario selecciona un registro de la tabla.  2. Modifica los campos necesarios.  3. Clic en actualizar.  4. El sistema actualiza los datos. |
| Flujos alternativos | Ninguno. |
| Excepciones | 4a. Error al actualizar médico. |
| Resultado esperado | El médico queda actualizado correctamente. |

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Eliminar médico |
| Actor | Administrador, Gestor |
| Descripción breve | Permite eliminar un médico del sistema si no tiene datos dependientes. |
| Precondiciones | • El médico debe estar registrado.  • El usuario debe tener permisos. |
| Flujo básico | 1. El usuario accede al listado de médicos.  2. Selecciona el registro de la tabla.  3. Hace clic en “Eliminar”.  4. Confirma la eliminación.  5. El sistema borra el registro. |
| Flujos alternativos | N/A |
| Excepciones | 3a. al seleccionar el registro el usuario puede solamente inactivar el registro del médico.  4a. Error al eliminar médico. |
| Resultado esperado | El médico es eliminado o marcado como inactivo según políticas. |

## Modulo gestión de servicios

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Registrar servicio médico |
| Actor | Administrador, Gestor |
| Descripción breve | Permite crear un nuevo servicio médico (consulta, laboratorio, cirugía, etc.). |
| Precondiciones | • El usuario debe haber iniciado sesión.  • Tener permisos en el módulo de servicios. |
| Flujo básico | 1. El usuario accede al módulo de servicios.  2. Ingresa los datos del servicio (nombre, precio, descripción).  3. Hace clic en registrar.  4. El sistema guarda el servicio. |
| Flujos alternativos | N/A |
| Excepciones | 3ª. Error al registrar servicio |
| Resultado esperado | El nuevo servicio médico queda registrado correctamente. |

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Actualizar servicio médico |
| Actor | Administrador, Gestor |
| Descripción breve | Permite modificar los datos de un servicio médico registrado. |
| Precondiciones | • El servicio debe estar registrado. |
| Flujo básico | 1. El usuario selecciona el servicio de la tabla.  2. Cambia los campos necesarios (nombre, precio, etc.).  3. Hace clic en actualizar.  4. El sistema actualiza la información. |
| Flujos alternativos | Ninguno. |
| Excepciones | 4a. Error al actualizar servicio. |
| Resultado esperado | El servicio se actualiza correctamente. |

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Eliminar servicio médico |
| Actor | Administrador, Gestor |
| Descripción breve | Permite eliminar un servicio del catálogo del hospital. |
| Precondiciones | • El servicio debe existir. |
| Flujo básico | 1. El usuario accede al listado de servicios.  2. Selecciona el servicio a eliminar de la tabla.  3. Hace clic en “Eliminar”.  4. El sistema elimina el servicio. |
| Flujos alternativos | N/A |
| Excepciones | 4a. Error al eliminar servicio. |
| Resultado esperado | El servicio queda eliminado del sistema |

## Modulo gestión de usuario

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Registrar usuario nuevo |
| Actor | Administrador, Gestor |
| Descripción breve | Permite crear un nuevo usuario (gestionar módulos). |
| Precondiciones | • El usuario debe haber iniciado sesión.  • Tener permisos en el módulo de usuario. |
| Flujo básico | 1. El usuario accede al módulo de usuario.  2. Ingresa los datos del nuevo usuario (nombre, correo electrónico contraseña, rol).  3. Hace clic en registrar.  4. El sistema guarda el servicio. |
| Flujos alternativos | N/A |
| Excepciones | 3ª. Error al registrar servicio |
| Resultado esperado | El nuevo usuario queda registrado correctamente. |

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Actualizar usuario existente |
| Actor | Administrador, Gestor |
| Descripción breve | Permite modificar los datos de un usuario existente |
| Precondiciones | • El usuario debe estar registrado. |
| Flujo básico | 1. El administrador selecciona el usuario existente  2. Cambia los campos necesarios (contraseña, correo electrónico etc.).  3. Hace clic en actualizar.  4. El sistema actualiza la información. |
| Flujos alternativos | Ninguno. |
| Excepciones | 4a. Error al actualizar servicio. |
| Resultado esperado | El servicio se actualiza correctamente. |

|  |  |
| --- | --- |
| Campo | Detalle |
| Nombre | Eliminar usuario |
| Actor | Administrador, Gestor |
| Descripción breve | Permite eliminar un usuario del sistema. |
| Precondiciones | • El usuario debe existir. |
| Flujo básico | 1. El administrador accede al módulo.  2. Selecciona el usuario a eliminar del módulo.  3. Hace clic en “Eliminar”.  4. El sistema elimina el servicio. |
| Flujos alternativos | N/A |
| Excepciones | 4a. Error al eliminar servicio. |
| Resultado esperado | El usuario queda eliminado del sistema |

# **CONCLUSION**

Este proyecto representa una oportunidad para aplicar conocimientos adquiridos en el ámbito académico y desarrollar habilidades prácticas en el desarrollo de software. La integración de Java, SQL y C# permitirá crear una aplicación completa y funcional, demostrando la capacidad de trabajar con múltiples tecnologías y lenguajes de programación.

# **RECOMENDACIONES**

# **E-GRAFÍAS/REFERENCIAS**