



Ejercicio Agenda de citas.

Juan David Ruge Garzón.

Gestor del conocimiento.

William Alexander Matallana Parra.

Universidad de Cundinamarca, extensión Chía.

Ingeniería de Sistemas y computación.

18 de febrero de 2026.



Contenido.

Ejercicio Agenda de citas.....	1
Formato de Requerimientos del sistema	3
1. Información general del proyecto.....	3
2. Descripción general del sistema	3
3. Requerimientos Funcionales (RF)	4
4. Requerimientos No Funcionales (RNF)	4
5. Relación Requerimiento – POO	5
Análisis.	6
Relaciones.....	6
Diagrama de Clases.	7
Diseño	9
GITHUB.	10



Formato de Requerimientos del sistema.

1. Información general del proyecto

Nombre del proyecto	Sistema de Gestión de Agenda de Citas
Integrantes	Juan David Ruge Garzón
Programa académico	Ingeniería de Sistemas y Computación
Fecha	18/02/2026
Lenguaje de programación	Java
Tipo de aplicación	Consola

2. Descripción general del sistema

- El sistema permite registrar citas médicas ingresando los datos del paciente y la fecha y hora de atención.
- Cada cita puede cambiar de estado según su proceso, aplicando reglas definidas.
- El sistema muestra todas las citas registradas y permite modificarlas o eliminarlas según id.



3. Requerimientos Funcionales (RF)

ID	Nombre	Descripción	Entrada(s)	Proceso	Salida
RF-01	Registrar cita	Permite registrar una nueva cita médica en el sistema	nombrePaciente, fecha, hora	Valida datos y crea una nueva cita con estado	Cita registrada correctamente
RF-02	Listar citas	Muestra todas las citas registradas en el sistema		Recorre la lista de citas	Lista de citas
RF-03	Cambiar estado de cita	Permite modificar el estado de una existente	idCita, nuevoEstado	Busca la cita por ID y actualiza su estado	Estado de cita actualizado
RF-04	Eliminar cita	Permite eliminar una cita del sistema	idCita	Busca la cita por ID y la elimina	Cita eliminada correctamente

4. Requerimientos No Funcionales (RNF)

ID	Tipo	Descripción
RNF-01	Validación	El sistema no debe permitir registrar citas con nombre vacío
RNF-02	Validación	La fecha y hora ingresadas deben ser validadas
RNF-03	Estructura	El sistema debe estar organizado siguiendo una estructura tipo MVC
RNF-04	Calidad	El sistema debe aplicar encapsulamiento, uso de enums y separación de responsabilidades.
RNF-05	Usabilidad	La aplicación debe mostrar mensajes claros en caso de error o éxito



5. Relación Requerimiento – POO

ID Requerimiento	Clase	Método	Tipo
RF-01	CitaService	registrarCita()	Funcional
RF-02	CitaService	listarCitas()	Funcional
RF-03	CitaService	cambiarEstado()	Funcional
RF-04	CitaService	eliminarCita()	Funcional
RNF-01	ValidacionUtil	validarNombre()	No Funcional
RNF-02	ValidacionUtil	validarFecha	No Funcional



Análisis.

Entidades Identificadas:

- Cita (id, nombrePaciente, fechaHora, estado)

Entidad principal del sistema.

- EstadoCita (PROGRAMADA, CANCELADA, ATENDIDA)

Posibles estados de una cita

- CitaService (gestiona la lista de citas)

Define operaciones del sistema: registrar, listar, cambiar estado y eliminar citas.

- CitaServiceImpl

Implement de CitaService. Gestiona lista interna de citas y ejecuta la lógica de negocio.

- CitaView

Muestra mensajes e información de citas.

- Main

Inicia componentes y gestiona el menú principal

Relaciones.

- Una Cita tiene un único EstadoCita.
- El CitaService puede gestionar muchas Citas.
- CitaController depende de CitaService.
- CitaController se comunica con CitaView para mostrar resultados.
- Main crea e inyecta las dependencias del sistema.



Diagrama de Clases.

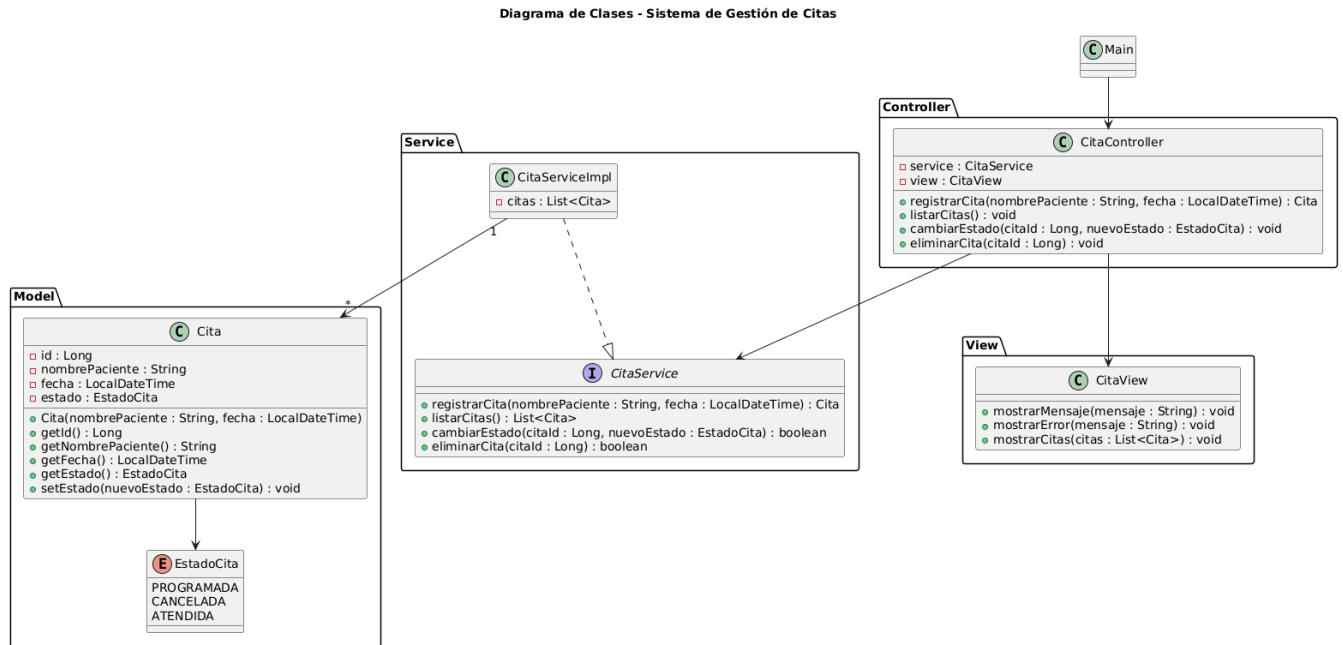


Diagrama de casos de uso.





Diagrama de Flujo: Registrar cita.

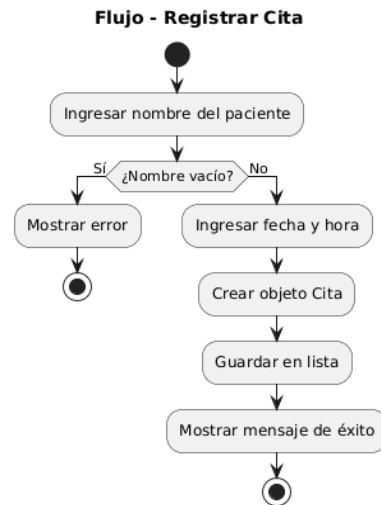


Diagrama de flujo: Cambiar cita.

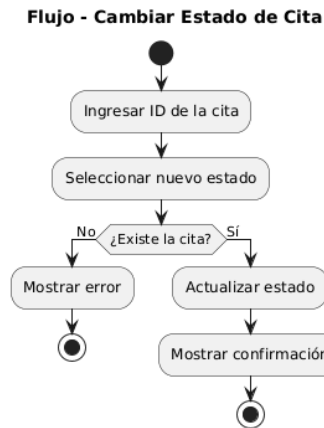


Diagrama de flujo: Eliminar cita





Diseño

Modelo

Cita

EstadoCita

Vista

CitaView

Controlador

CitaController

Servicios

CitaService

CitaServiceImpl

Aplicación

Main



GITHUB.

https://github.com/DavidAikoo/RequerimientosPOO_Programa