



FORMATO DE REQUERIMIENTOS DEL SISTEMA ASISTENCIA DE ESTUDIANTES

Creador de oportunidades

María Paula Ramírez Gómez

Gestor del Conocimiento

William Alexander Matallana

Universidad de Cundinamarca, Extensión Chía

Ingeniería de Sistemas y Computación

Ingeniería de Software I

2026

FORMATO DE REQUERIMIENTOS DEL SISTEMA

1. Información general del proyecto

Nombre del proyecto	Sistema de Gestión de Asistencia de Estudiantes
Integrantes	María Paula Ramírez Gómez
Programa académico	Ingeniería de Sistemas y Computación
Fecha	17/02/2026
Lenguaje de programación	Java
Tipo de aplicación	Consola

1.1 Descripción general del sistema

- El sistema permite gestionar la asistencia de estudiantes mediante el registro de estudiantes y el control de su asistencia en fechas específicas.
- El usuario puede registrar estudiantes, marcar su asistencia indicando si asistieron o no, consultar los registros existentes y eliminar registros cuando sea necesario.
- La aplicación está desarrollada en Java como una aplicación de consola, aplicando principios de Programación Orientada a Objetos y una arquitectura organizada en capas: modelo, servicio, controlador y vista.

1.2 Requerimientos Funcionales (RF)

ID	Nombre	Descripción	Entrada(s)	Proceso	Salida
RF-01	Registrar estudiante	Permite registrar un estudiante en el sistema.	id del estudiante, nombre	Se crea un objeto estudiante con los datos ingresados	Estudiante registrado
RF-02	Marcar asistencia	Permite registrar la asistencia de un estudiante en una fecha determinada.	estudiante, fecha, estado de asistencia	Se crea un registro de asistencia asociado al estudiante	Asistencia registrada
RF-03	Consultar asistencia	Permite visualizar los registros de asistencia de un estudiante.	estudiante	Se buscan y muestran los registros asociados	Lista de asistencias del estudiante
RF-04	Eliminar registro de asistencia	Permite eliminar un registro de asistencia existente.	registro de asistencia	Se elimina el registro del sistema	Confirmación de eliminación

1.3 Requerimientos No Funcionales (RNF)

ID	Tipo	Descripción
RNF-01	Validación de datos	El sistema debe validar que los datos ingresados no estén vacíos.

RNF-02	Estructura del sistema	El sistema debe estar organizado en paquetes: model, service, controller y view.
RNF-03	Arquitectura	El sistema debe implementar separación de responsabilidades entre capas.
RNF-04	Programación Orientada a Objetos	El sistema debe aplicar encapsulamiento, uso de clases, enumeraciones e interfaces.

1.4 Relación Requerimiento – POO

ID Requerimiento	Clase	Método	Tipo
RF-01	Estudiante	Constructor/métodos de acceso	Funcional
RF-02	RegistroAsistencia	constructor()	Funcional
RF-03	AsistenciaService	consultarAsistencia	Funcional
RF-04	AsistenciaService	eliminarRegistro	Funcional
RNF-01	Controller /Service	Validaciones	No Funcional
RNF-02	Paquetes del Sistema	Organización	No Funcional

2. Análisis

2.1 Diagrama UML de Clases

El diagrama UML de clases representa la estructura estática del sistema, mostrando las clases principales, sus atributos, métodos y las relaciones entre ellas.

Este diagrama permite visualizar cómo se organiza el sistema de asistencia académica siguiendo el patrón Modelo-Vista-Controlador (MVC).

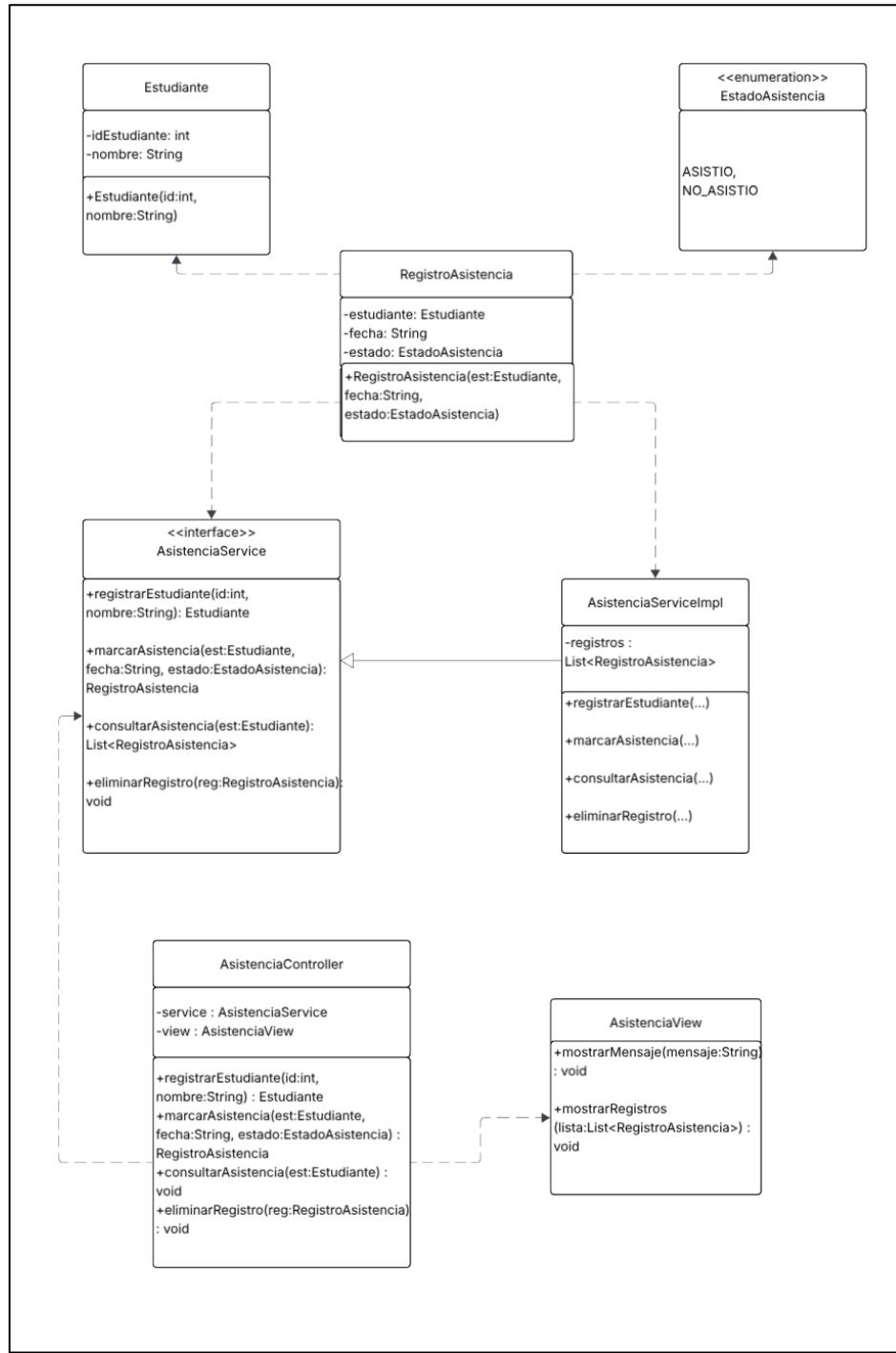


Figura 1. Diagrama UML de clases del sistema de asistencia.

Fuente: Elaboración propia.

2.2 Diagrama de casos de uso

El diagrama de casos de uso representa las funcionalidades del sistema desde la perspectiva del usuario. En el sistema de control de asistencia, el actor principal es el docente, quien puede registrar estudiantes, marcar asistencia, consultar registros y eliminar asistencias. Este diagrama permite visualizar de manera sencilla la interacción entre el usuario y el sistema sin mostrar detalles técnicos internos.

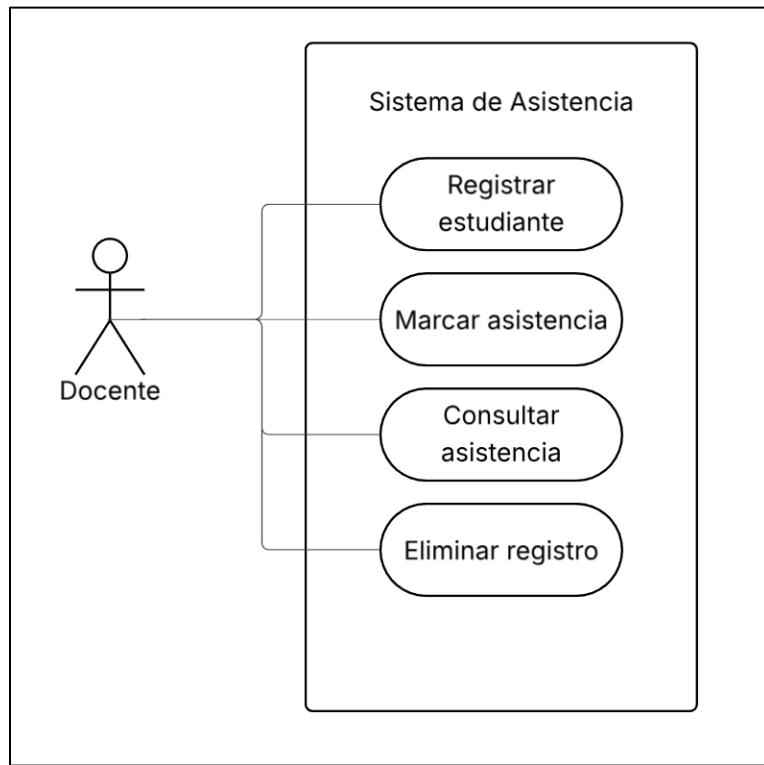


Figura 2. Diagrama UML de casos de uso del sistema de asistencia.

Fuente: Elaboración propia.

2.3 Diagrama de flujo

El diagrama de flujo representa de manera gráfica la secuencia de pasos y decisiones necesarias para el funcionamiento del sistema de control de asistencia. Permite

visualizar el proceso desde el inicio, pasando por las opciones disponibles para el docente —como registrar estudiantes, marcar asistencia, consultar registros o eliminar datos— hasta la finalización del programa. Este diagrama facilita la comprensión del funcionamiento lógico del sistema y del flujo de ejecución de las acciones.

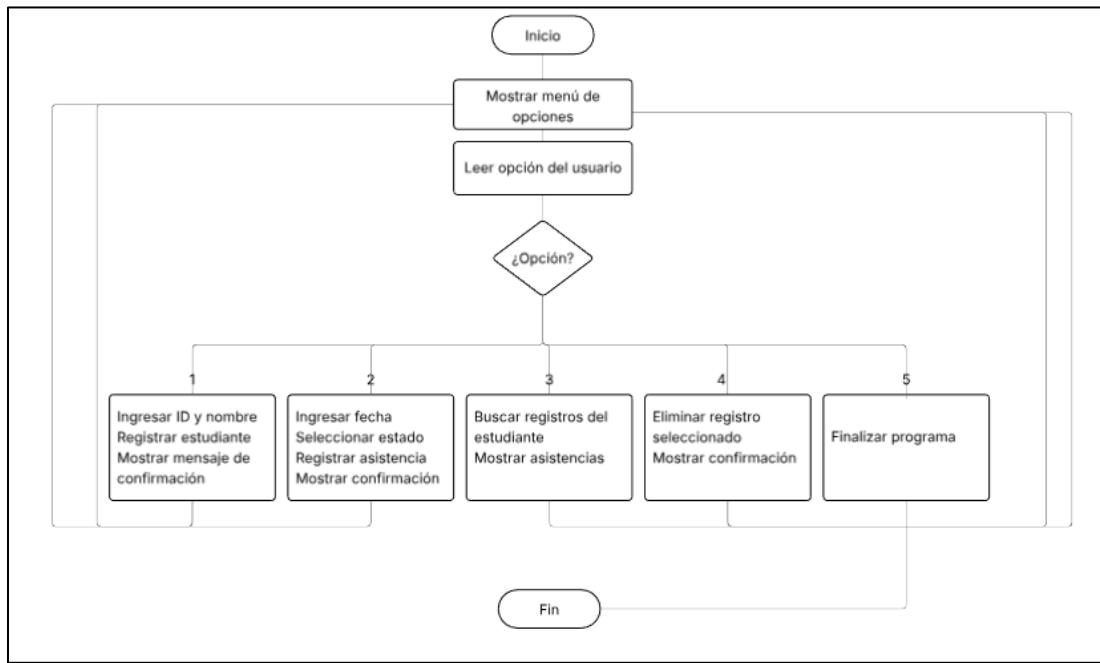


Figura 3. Diagrama de flujo del sistema de asistencia.

Fuente: Elaboración propia.

3. Diseño del Sistema

◊ 3.1 Arquitectura del sistema

El sistema se desarrolla bajo el modelo **MVC (Modelo–Vista–Controlador)**, el cual permite separar la lógica de negocio, la interfaz gráfica y el control de las acciones del usuario.

- **Modelo:** Gestiona los datos y reglas del sistema (Asistencia, EstadoAsistencia, Estudiante).
- **Vista:** Muestra la información al usuario y permite la interacción.
- **Controlador:** Recibe las acciones del usuario y coordina la comunicación entre vista y modelo.

- **Servicio:** Contiene la lógica del negocio y operaciones principales.

◊ **3.2 Diseño de datos**

Define las entidades principales y sus atributos.

Entidades principales:

- **Asistencia**
 - id
 - fecha
 - estado
 - estudiante
- **EstadoAsistencia**
 - idEstado
 - nombreEstado (Presente, Ausente, Tarde)
- **Estudiante**
 - id
 - nombre
 - curso

◊ **3.3 Diseño de la interfaz**

Describe las pantallas y elementos con los que interactúa el usuario.

Pantalla principal de asistencia:

- Lista de estudiantes
- Selección del estado de asistencia
- Campo de fecha
- Botón “Registrar asistencia”
- Mensaje de confirmación

◊ 3.4 Diseño funcional

Describe el comportamiento del sistema ante las acciones del usuario.

Funcionamiento general:

1. El usuario accede al módulo de asistencia.
2. El sistema muestra la lista de estudiantes.
3. El usuario selecciona un estudiante.
4. Elige el estado (Presente, Ausente o Tarde).
5. Presiona el botón de registrar.
6. El sistema guarda la información.
7. Se muestra un mensaje de confirmación.

En conclusión, el desarrollo del sistema de asistencia permite optimizar el registro y control de la participación de los estudiantes, facilitando las labores del docente y reduciendo errores en los procesos manuales. Mediante el uso de herramientas de análisis y diseño UML, se logró estructurar una solución organizada, funcional y adaptable a diferentes contextos educativos. Este proyecto demuestra la importancia de aplicar metodologías de ingeniería de sistemas para crear aplicaciones eficientes que respondan a necesidades reales.