

## **FORMATO DE REQUERIMIENTOS DEL SISTEMA**

Lizeth V. Moreno

Universidad de Cundinamarca, Extensión Chia

Facultad de ingeniería, Programa Sistemas y computación

Ingeniería de Software I

William Matallana

17 de feb. de 26

## 1. Información general del proyecto

Nombre del proyecto	SISTEMA DE CONTROL DE ASISTENCIA
Integrantes	Lizeth Vanessa Moreno Moreno
Programa académico	Ingeniería de Sistemas y Computación
Fecha	18/02/2026
Lenguaje de programación	Java
Tipo de aplicación	Consola

## 2. Descripción general del sistema

- Registrar estudiantes
- Marcar asistencia
- Consultar asistencia por estudiante
- Eliminar registros de asistencia

## 3. Requerimientos Funcionales (RF)

ID	Nombre	Descripción	Entrada(s)	Proceso	Salida
RF-01	Registrar estudiantes	Permite registrar un estudiante	Fecha Nombre Identificación	Guarda datos del estudiante	Estudiante registrado
RF-02	Marcar asistencia	Permite registrar la asistencia del estudiante	Asistió/ no asistió	Guarda información de asistencia	Asistencia registrada
RF-03	Consultar asistencia	Permite buscar la asistencia de un estudiante	estudiante	Busca la información de asistencia del estudiante	Asistencia registrada del estudiante

RF-04	Eliminar registros de asistencia	Permite eliminar la asistencia de un estudiante	Estudiante Fecha	Busca y elimina el registro	Registro Eliminado
-------	----------------------------------	---	------------------	-----------------------------	--------------------

#### 4. Requerimientos No Funcionales (RNF)

ID	Tipo	Descripción
RNF-01	Validación	El sistema no debe permitir registrar asistencia dos veces al día
RNF-02	Validación	El estado de la asistencia debe de ser solo asistió o no asistió
RNF-03	Estructura	El sistema debe usar el patrón modelo vista controlador para organizar el sistema
RNF-04	Calidad	El sistema debe aplicar encapsulamiento para proteger los datos y evitar modificaciones incorrectas

#### 5. Relación Requerimiento – POO

ID Requerimiento	Clase	Método	Tipo
RF-01	EstudianteService	registrarEstudiante()	Funcional
RF-02	AsistenciaService	registrarAsistencia()	Funcional
RF-03	AsistenciaService	consultarAsistencia()	Funcional
RF-04	AsistenciaService	eliminarAsistencia()	Funcional
RNF-01	AsistenciaService	validarAsistenciadoble()	No Funcional

## SDLC (Ciclo de Vida de Desarrollo de Software)

### 1. Levantamiento de requerimientos

#### *Funciones claves del sistema*

El sistema de control de asistencia tiene como propósito colaborar con el registro de asistencia de los estudiantes en sus funciones esta también la validación de no permitir que no se registre asistencia duplicada el mismo día.

#### *Requerimientos funcionales*

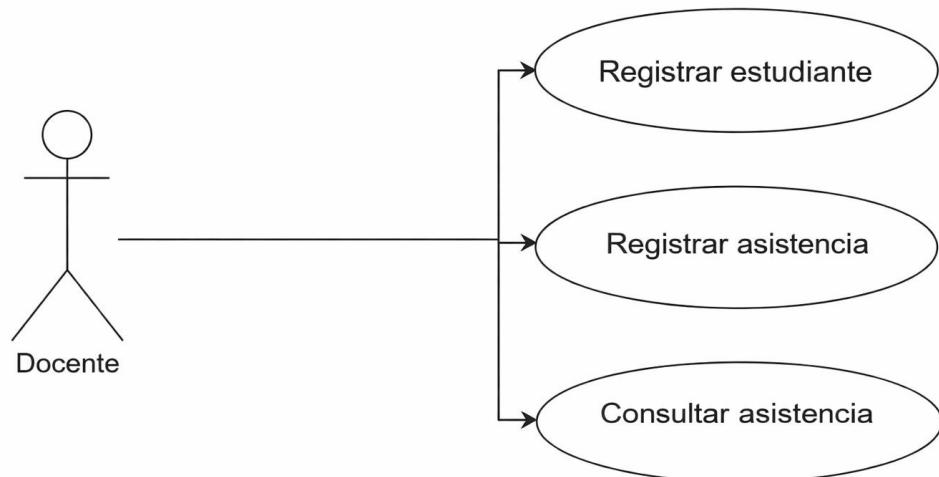
Permite registrar estudiantes con nombre y número de identificación, debe permitir registrar la asistencia de un estudiante (asistió o no asistió) y guardarla. También debe verificar que no se registre la asistencia más de una vez en la misma fecha para el mismo estudiante. Además, permite buscar la asistencia registrada de un estudiante mostrando sus datos fecha y el estado de asistencia.

#### *Requerimientos no funcionales*

El sistema está desarrollado en Java, se usa el patrón Modelo Vista Controlador (MVC) para organizar el código, aplica encapsulamiento para proteger los datos y principalmente debe de únicamente dejar que el estado de asistencia pueda ser solo asistió o no asistió, debe estar estructurado en paquete de forma ordenada

### 2. Análisis del sistema

#### *Diagrama de casos de uso (explicación en texto)*



### *Diagrama de flujo*



### *Viabilidad técnica*

No requiere conexión a internet y Puede ejecutarse en cualquier computador con Java instalado.

### *Análisis del flujo de trabajo*

El usuario inicia el sistema, registra un estudiante, registra la asistencia, guarda y el sistema muestra el resultado.

*Evaluación de tecnologías Java a usar:* Lenguaje Java, POO, encapsulamiento, enumeraciones (enum), patrón MVC, Clases

### 3. Diseño

#### *Estructura usada*

La estructura por paquetes que se ha usado para la codificación es

1. Modelo :guarda información del estudiante y su asistencia
  - Estudiante
  - Asistencia
  - EstadoAsistencia
2. Vista: muestra la información al usuario en la consola
  - AsistenciaView
3. Controlador: comunica la vista con el servicio
  - AsistenciaController
4. Servicio: contiene la lógica del sistema
  - AsistenciaService
  - AsistenciaServiceImpl
5. Main: inicia el sistema y ejecuta el flujo principal

#### *Identificación de entidades del dominio*

Las entidades principales son: Estudiante (nombre, identificación), Asistencia (estudiante, fecha), enum (asistió, no asistió) Estas representan los objetos principales del sistema.

#### *Patrones de diseño utilizados*

MVC, Encapsulamiento (atributos privados, getters y setters en una sola clase), Abstracción

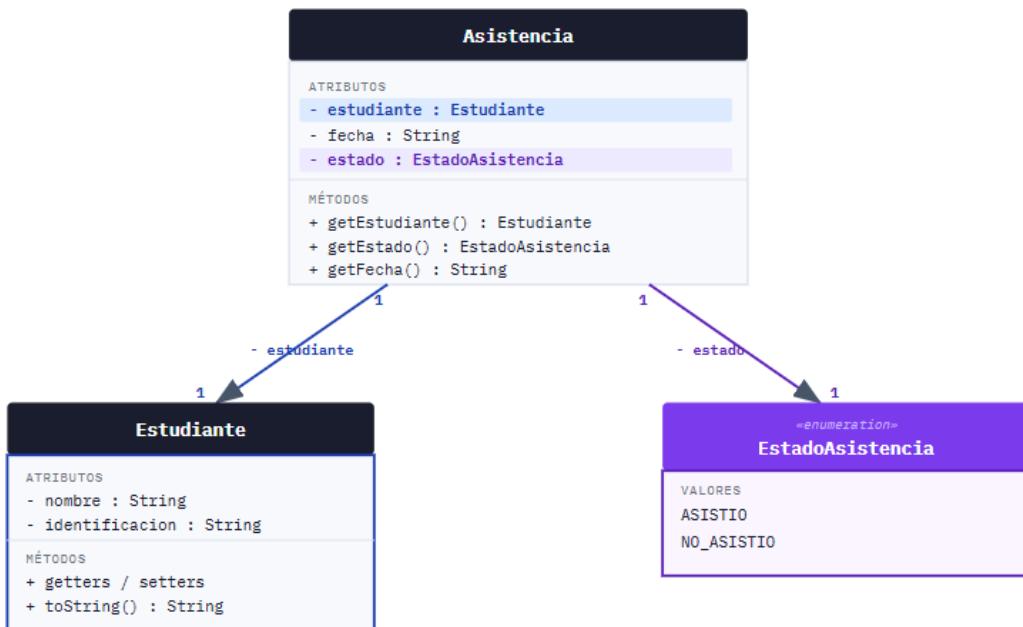
#### *Diagramas casos de clases*

De implementación: una clase implementa una interfaz y los métodos de la clase implementan todos los métodos de la interfaz.



## De asociación

Una clase usa a otra como atributo



Repositorio : <https://github.com/morenolizeth-lang/actividad3.git>