



Departamento de Computación  
Asignatura: Ingeniería de Software I (3385)  
Año 2025

# INGENIERÍA DE SOFTWARE I (3385)

PRÁCTICO 1B: SISTEMA DE  
GESTIÓN ESTUDIANTIL

Grupo 1 (Comisión 3)

- Irusta, Juan Ignacio. DNI: 41.525.387
- Macció, Facundo Damián. DNI: 45.704.335
- Pereno, Juan Cruz. DNI: 45.353.298
- Trabucco, Esteban Daniel. DNI: 43.883.138

## **1. Análisis del enunciado y actores.**

Tras analizar la situación problemática, interpretamos que la necesidad principal de la Universidad es contar con una herramienta que centralice y organice la gestión de la información académica. Actualmente, los procesos están fragmentados en planillas y sistemas aislados, lo que dificulta la comunicación entre Oficina de Alumnos, docentes y estudiantes.

El desafío que el sistema debe abordar es brindar una plataforma integrada que permita registrar y consultar datos de alumnos y profesores, organizar las carreras y planes de estudio, validar correlatividades, gestionar inscripciones y notas, y facilitar el seguimiento académico. De esta manera, se busca modularizar y simplificar la administración, ofreciendo un registro claro y confiable de la situación de cada estudiante y de la asignación de docentes a las materias.

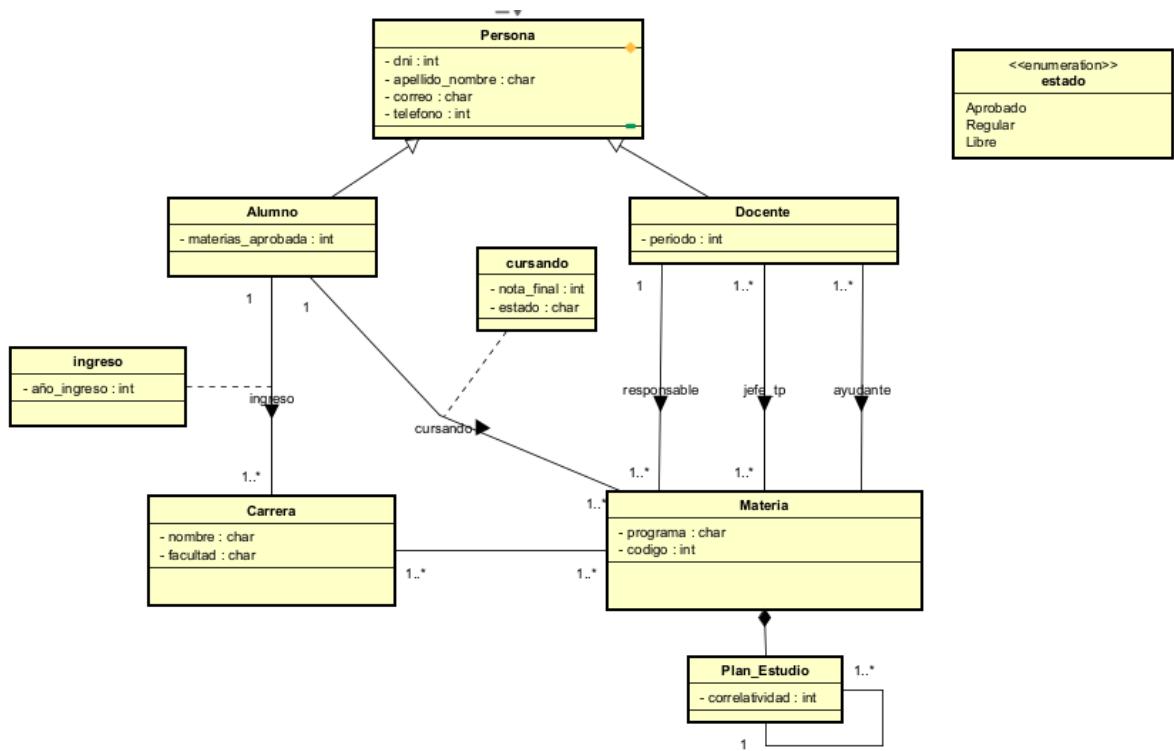
### **Actores principales del sistema:**

Oficina de Alumnos (Administradores): gestionan datos de alumnos, docentes, carreras, materias, correlatividades e inscripciones.

Estudiantes: consultan su información académica, se inscriben a materias y visualizan sus notas.

Profesores: registran calificaciones y acceden a listados de alumnos en sus materias.

## **2. Diagrama de clase de contexto (UML).**



### **3. Requerimientos funcionales.**

- Gestionar Usuario
- Gestionar Materia
- Gestionar Plan de Estudio
- Gestionar Carrera
- Gestionar Docente
- Gestionar Alumno
- Consultar Nota
- Consultar Listado de alumnos
- Gestionar Correlatividad
- Gestionar Situación Académica
- Gestionar Comunicación

### **4. Requerimientos NO funcionales.**

- Disponible 24/7 interfaz amigable interfaz interactiva seguridad de usuario fácil mantenimiento Varios accesos simultáneos Optimización de comunicación
- El sistema debe contar con la posibilidad de poder ingresar en todo momento, los siete días de la semana, las 24 horas. Garantizando muchos accesos simultáneos.
- El sistema debe poseer una interfaz amigable e interactiva para aquellos usuarios que utilicen la misma.
- El sistema debe garantizar seguridad para la información de los usuarios que guardan su acceso bajo una contraseña, y en el caso de perderla, debe haber un patrón a seguir para recuperar la misma de forma verídica.
- El sistema debe proveer información a los estudiantes por medio de alertas/notificaciones, también se debe crear un foro donde se anuncien noticias, y un chat para intercambiar mensajes entre docentes y estudiantes.

## **5. Historia de usuario.**

<b>Historia de HU</b>	HU-001
<b>Título</b>	Gestion de docentes
<b>Declaración</b>	como administrador, quiero gestionar la información de los docentes y sus roles en las materias, para organizar los equipos docentes de manera eficiente
<b>Descripción detallada</b>	El sistema debe permitir al administrador dar de alta, modificar y eliminar registros de docentes. La información a guardar incluye datos personales (nombre, apellido, DNI), datos de contacto (teléfono, correo), y el rol que desempeñan en las materias (responsable de cátedra, jefe de trabajos prácticos o ayudante).
<b>Criterios de validación</b>	<ul style="list-style-type: none"><li>- El sistema debe permitir registrar un docente con todos los campos obligatorios.</li><li>- El administrador debe poder asignar y modificar los roles de un docente en una materia.</li><li>- El sistema debe reflejar correctamente los docentes asignados en cada período académico.</li></ul>
<b>Tareas asociadas a la implementación</b>	<ul style="list-style-type: none"><li>- Implementar formulario de registro y edición de docentes.</li><li>- Implementar asignación de roles y materias a docentes.</li></ul>

<b>ID de HU</b>	HU-002
<b>Título</b>	Gestionar Materia
<b>Declaración</b>	Como Administrador de la Oficina de Alumnos, quiero poder crear, editar y eliminar materias para mantener los planes de estudios actualizados.
<b>Descripción Detallada</b>	Esta función permitirá al administrador dar de alta nuevas materias (creando su código, su nombre), también permitirán la modificación de sus atributos y la eliminación en caso de que el plan de estudio de la carrera sea modificado.
<b>Criterios de Validación</b>	<ul style="list-style-type: none"> <li>- El sistema debe permitir al Administrador crear una materia con código y nombre.</li> <li>- El sistema debe permitir al Administrador modificar los datos de las materias existentes.</li> <li>- Al crear o modificar la materia, el sistema debe validar que el código de la materia sea único.</li> <li>- El sistema debe permitir al Administrador eliminar una materia.</li> <li>- El sistema debe verificar si la materia que se está por eliminar tiene inscripciones activas</li> <li>- El sistema debe mostrar un mensaje de confirmación al eliminar una materia.</li> </ul>
<b>Tareas Asociadas a la Implementación</b>	<ul style="list-style-type: none"> <li>- Crear Materia con atributos: código, nombre.</li> <li>- Desarrollar endpoint REST para CRUD de materias.</li> <li>- Implementar validación de código único.</li> <li>- Implementar validación de inscripciones a la materia al eliminar.</li> </ul>

<b>ID de Hu</b>	HU-ALUM-001
<b>Título</b>	Gestionar Alumnos
<b>Declaración</b>	Como administrador, quiero registrar y gestionar estudiantes para mantener actualizada la base de datos académica.
<b>Descripción Detallada</b>	<ul style="list-style-type: none"> <li>- El sistema debe permitir al administrador ingresar a un módulo “Gestión de Estudiantes” mediante un botón en el menú principal. Visualizar una tabla/listado de estudiantes ya cargados, con columnas como: Nombre completo, DNI, Dirección, Email, Estado (Activo/Inactivo).</li> <li>- Tener disponibles los botones:</li> <li>- “Agregar Estudiante”: abre un formulario emergente con campos obligatorios (Nombre, Apellido, DNI, Dirección, Teléfono, Email).</li> <li>- “Editar”: despliega un formulario con los datos cargados del estudiante seleccionado, permitiendo modificarlos.</li> <li>- “Eliminar”: pide confirmación mediante un cuadro de diálogo (¿Está seguro que desea eliminar al estudiante seleccionado?).</li> </ul>
<b>Criterios de validación</b>	<ul style="list-style-type: none"> <li>- No se debe poder guardar un estudiante sin completar los campos obligatorios.</li> <li>- El sistema debe validar que el DNI sea único.</li> <li>- El administrador debe ver un mensaje de confirmación al finalizar cada acción.</li> <li>- El listado de estudiantes debe actualizarse automáticamente después de cada operación.</li> </ul>
<b>Tareas asociadas a la implementación</b>	<ul style="list-style-type: none"> <li>- Crear la clase Estudiante heredando de Persona.</li> <li>- Implementar la interfaz gráfica con tabla y botones (Agregar, Editar, Eliminar).</li> <li>- Desarrollar validaciones de campos obligatorios y DNI único.</li> <li>- Programar lógica de activación/inactivación del estudiante en la base de datos.</li> <li>- Implementar notificaciones visuales (alertas de éxito o error).</li> </ul>

## **6. Backlog Inicial.**

- 1°. Gestionar Usuario
- 2°. Gestionar Materia
- 3°. Gestionar Plan de Estudio
- 4°. Gestionar Carrera
- 5°. Gestionar Docente
- 6°. Gestionar Alumno
- 7°. Consultar Nota
- 8°. Consultar Listado de alumnos
- 9°. Gestionar Correlatividad
- 10°. Gestionar Situación Académica.
- 11°. Gestionar Comunicación.