

# Sistema de Administración Universitario

## Proyecto de Programación Orientada a Objetos

Isidora Santa Cruz   David Reyes   Matías Silva   Josefina Cabeza   Matías Follert

Programación Orientada a Objetos

12 de diciembre de 2025

# Índice

1 Descripción de la problemática

2 Objetivos del proyecto

3 Justificación

# Descripción de la problemática

El sistema actual enfrenta desafíos que afectan la eficiencia:

## Procesos manuales y errores

- Dependencia de formularios físicos y hojas de cálculo sin conexión.
- Alta probabilidad de error humano.
- Pérdida de documentos y conflictos en la administración.

## Gestión de cupos y espacios

- Falta de control en tiempo real de la capacidad de aulas.
- Demasiados cupos o utilización precaria de recursos.

# Descripción de la problemática

## Descentralización y Comunicación

- Información fragmentada entre estudiantes y profesores.
- Falta en la comunicación: Listas desactualizadas para profesores y falta de información del docente para alumnos.

## Gestión curricular ineficiente

- No existe un sistema integrado para crear y administrar facultades y carreras.
- Las mallas curriculares se gestionan de forma manual y desconectada del sistema.
- Configurar prerrequisitos entre asignaturas es un proceso complejo y propenso a errores.
- No hay un mecanismo claro para habilitar carreras en períodos de matrícula específicos.

# Objetivo general

## Objetivo principal

Desarrollar un sistema integral de gestión académica en Python que unifique y optimice el proceso de matrícula, asignación docente y administración para las operaciones del instituto.

# Objetivos específicos

## 1 Gestión de estudiantes:

- Registro digital, historial académico y actualización de datos.
- Búsqueda y visualización de carga académica.

## 2 Administración centralizada:

- Panel de control con gestión de roles y permisos.
- Generación de reportes, estadísticas y respaldo de seguridad.
- Implementar ventanas para agregar nuevas facultades y carreras.

## 3 Inscripción y asignación:

- Validación automática de pre-requisitos y topes de horario.
- Gestión de cupos.
- Crear mallas curriculares digitales para las carreras.
- Habilitar carreras para procesos de matrícula.
- Añadir y configurar prerrequisitos a las asignaturas.

# Justificación del proyecto

## **Eficiencia**

- Reducción del tiempo de matrícula.
- Automatización de tareas repetitivas.

## **Confiabilidad de datos**

- Eliminación de discrepancias entre departamentos.
- Minimización de errores.

## **Experiencia educativa**

- Procesos gestionables para alumnos.
- Herramientas de gestión efectivas para los docentes.

## **Toma de decisiones**

- Análisis e informes para optimizar los recursos.
- Gestión basada en los datos.

# Diagrama de Clases

