

	UNIVERSIDAD ARGENTINA DE LA EMPRESA Departamento de Tecnología Informática PROGRAMACIÓN I – ALGORITMOS Y ESTRUCTURAS DE DATOS I				
	Proyecto	Sistema de Alumnado			
	Fecha	16/06/2025	N° Equipo	10	Versión 0.8

DESCRIPCIÓN

Es un sistema de administración de una escuela secundaria, donde podrán añadir, modificar y quitar a la base de datos, desde alumnos y profesores hasta materias y cursos con sus características individuales

LISTA DE FUNCIONALIDADES

Título	Descripción
Añadir	Podremos añadir a nuestra base de datos alumnos, profesores, cursos y materias para permitir la mayor flexibilidad de orden de una escuela
Eliminar	Podremos eliminar de manera sencilla de nuestra base de datos alumnos, profesores, cursos y materias para permitir un orden fácil
Modificar	Podremos modificar de manera sencilla de nuestra base de datos alumnos, profesores, cursos y materias para permitir un orden fácil y evitar errores en la carga
Muestreo de datos a usuarios	Nuestros usuarios (Alumnos y profesores) tendrán a disposición toda la base de datos para revisar los datos que deseen basado en los permisos que tengan

CONSIDERACIONES Y EXCLUSIONES

Título	Descripción
Profesores y materias	Un profesor puede estar en varias materias, pero en una materia no puede haber mas de 3 profesores
Horas	Si bien las materias incluyen nombre y curso, no se asignan horarios de cursada
Cursos	Hay solo de 1ero a 6to, el sistema no contempla mayores cursos, pero si contempla mañana y tarde
Menú	Todo el sistema funciona en modo consola
Profesores	No hay limite en cuantas materias puede estar un profesor
Graduación de alumnos	No tenemos la funcionalidad específica para hacer pasar de año o graduar alumnos. Actualmente debe hacerse eliminando y volviendo a cargar manualmente

USUARIOS DEL SISTEMA

Rol	Uso del sistema
Admin	Aquel que puede modificar cada aspecto del sistema, desde las materias y cursos, hasta cada dato de alumnos o profesores
Profesor	Aquel que puede incidir en las notas y ver en que materias se encuentra y todos los datos de sus materias
Alumno	Aquel que puede solo ver sus datos, materias, notas y profesores

MODELO DE DATOS (entidades involucradas)

Profesores #lista	Alumnos #lista	Materias #lista	Registro_notas #matriz	Cursos #lista	Registro_cambios #matriz de tuplas
<pre> profes{ dni nombre apellido nacimiento mail telefono password } </pre>	<pre> alumno{ dni nombre apellido nacimiento curso turno password etapa telefono } </pre>	<pre> materia{ año profesores dias turno } </pre>	<pre> desc_nota[profesor (nombre - dni) curso turno alumno (nombre - dni) nota instancia fecha] </pre>	<pre> curso{ nombre cant_alumnos alumnos materias turno } </pre>	<pre> cambio(cambio fecha donde quien) </pre>
archivo json		archivo tipo json	archivo tipo CSV	archivo JSON	archivo txt
	entidad dentro de curso			posee a todos los alumnos	

ESTRUCTURA DE MENÚ

Opción nivel admin	Opción nivel profesor	Opción nivel alumno
[1] Gestión de profesores	[1] Gestión de notas	[1] Ver mis notas
[2] Gestión de alumnos	[2] Ver mis materias	[2] Ver mis materias
[3] Gestión de materias		
[4] Registro de cambios		
[0] Salir del sistema	[0] Salir del sistema	[0] Salir del sistema

EXPERIENCIA DE DESARROLLO (problemas encontrados, soluciones implementadas)

Al principio no sabíamos bien cómo organizar el proyecto, y se mezclaban funciones de materias, alumnos y profesores. Nos terminamos decidiendo por separar todo en módulos diferentes según la entidad (alumnos.py, materias.py, etc), lo que ayudó a que cada parte tenga su archivo y sea más fácil de encontrar y editar

Dificultad a la hora de realizar funciones las cuales puedan servir para diferentes objetos (profesores y alumnos, por ejemplo), pero se intentó hacer lo más global y reutilizable posible

A la hora de validar datos, al no tener una lista precargada fue tedioso, pero se pudo resolver implementando una carga aleatoria de profesores

Tuvimos algunos problemas con el guardado y carga de archivos (por ejemplo, sobrescritura o datos mal formateados), pero usamos json para los datos estructurados y csv para las notas, y revisamos siempre que el formato coincida al escribir o leer.

BACKLOG

<https://trello.com/b/hALGTEUP/trabajo>

BREVE DETALLE DE LOS SPRINTS

Sprint N°1:

- Diseño inicial de estructuras de datos.
- Creación de `almacen_datos.py` y funciones básicas.

Sprint N°2:

- Implementación de carga, validación y eliminación de datos.
- Manejo de validaciones con expresiones regulares.

Sprint N°3:

- Mejora de manejo de datos internamente.
- Separación de funciones en módulos.

Sprint N°4:

- Implementación de la gestión de materias y cursos.
- Asociación de materias a cursos y profesores.
- Carga y lectura de materias desde JSON.

Sprint N°5:

- Incorporación de sistema de notas: carga, modificación y consulta.
- Lectura y escritura de notas en archivo notas.csv.
- Implementación de visualización personalizada de notas según rol (profesor o alumno).

Sprint N°6:

- Mejoras generales en la experiencia de usuario
- Optimización de lectura/escritura en archivos.
- Pruebas de integración de módulos con distintos escenarios de datos.