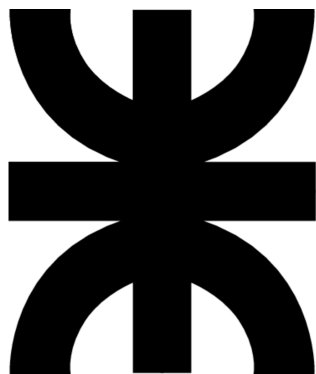


---

**UNIVERSIDAD TECNOLÓGICA NACIONAL**

**Centro Universitario Vicente López**



**Segundo Parcial Programación  
Avanzada I - Laboratorio V**

**Libreta del alumno**

Institución:

- Universidad Tecnológica Nacional

Carrera:

- Tecnicatura superior en sistemas informáticos - TUSI

Tutor Académico:

- Diego Cañizares

Participantes

- Bacchini, Ezequiel
  - Galetto, Nicolás
  - Mendoza, Germán
-

## ÍNDICE

<b>Concepto.....</b>	<b>2</b>
<b>Funciones.....</b>	<b>2</b>
<b>Decisiones que el sistema toma.....</b>	<b>2</b>
<b>Dominio.....</b>	<b>2</b>
❖ Funciones de cálculo de notas.....	3
❖ Funciones de lógica del negocio.....	3
<b>Persistencia.....</b>	<b>3</b>
❖ Donde se almacenan los datos.....	3
❖ Cómo se manejan los datos.....	4
❖ Diagrama Entidad - Relación - D.E.R.....	4
<b>Manejo de Consola.....</b>	<b>5</b>
❖ Acciones.....	5
❖ Funciones de manejo de consola.....	5
<b>Testing.....</b>	<b>5</b>

## Concepto

Una aplicación de escritorio instalada en tu computadora que permite cargar materias y registrar las notas de parciales, con el fin de facilitar la organización y el seguimiento del rendimiento académico.

## Funciones

- Determina automáticamente si el estudiante promociona una materia o debe rendir un examen final, en base a las notas ingresadas.
- Calcula el promedio general de cada materia según los parciales registrados.
- Lleva un control de las oportunidades disponibles para rendir un examen final, de acuerdo con la reglamentación académica establecida.
- Presenta información general del rendimiento académico, como la cantidad total de materias aprobadas.
- Guarda los datos ingresados en un archivo local, permitiendo su recuperación al volver a abrir la aplicación.
- Permite utilizar tanto una interfaz por consola como una API REST para la entrada y salida de datos. Estos son dos ejemplos implementados, pero se pueden añadir más métodos de entrada y salida.

## Decisiones que el sistema toma

- Evaluar si la nota obtenida por el alumno es aprobatoria o desaprobatoria, en base a los criterios establecidos.
- Determinar si el alumno promociona una materia o debe rendir examen final, considerando sus calificaciones parciales (dos o más).
- Verificar, según la aprobación o desaprobación de una materia, si el alumno puede acceder a materias correlativas correspondientes.
- Calcular y mostrar el promedio general de cada materia.
- Establecer cuántas oportunidades tiene el alumno para rendir un examen final.
- Permitir configurar los valores mínimos de nota necesarios para aprobar y promocionar una materia. Estos valores podrán definirse al momento de crear cada materia.

## Dominio

El dominio está compuesto por las funciones y clases que encapsulan la lógica central del sistema. Esto incluye tanto las operaciones necesarias para calcular notas y determinar condiciones académicas (como aprobación, promoción).

Además de la lógica de negocio, el dominio define la estructura de los elementos principales del sistema, como Materias, Finales y Parciales, especificando sus

atributos y relaciones. Estas entidades conforman la base sobre la cual se modela el comportamiento académico del usuario dentro de la aplicación.

#### ❖ Funciones de cálculo de notas

Este componente está conformado por un conjunto de clases y funciones dedicadas a procesar y analizar los datos asociados a una materia. Su propósito es brindar herramientas para determinar el estado académico del alumno en relación con cada asignatura.

Entre las principales operaciones que realiza se incluyen:

- ❖ Verificar si una materia está aprobada o desaprobada, en base a las notas obtenidas.
- ❖ Calcular el promedio general de calificaciones, cuando los datos disponibles lo permitan.
- ❖ Determinar si una materia fue promocionada según los criterios establecidos.
- ❖ Calcular la cantidad de intentos de examen final que le quedan disponibles al alumno.

#### ❖ Funciones de lógica del negocio

Este componente incluye el builder, la clase principal y las funciones asociadas que permiten definir y aplicar los criterios necesarios para determinar el estado actual de una materia según los datos proporcionados por el usuario.

Su objetivo es establecer, de forma estructurada y flexible, si una materia está aprobada, desaprobada o promocionada, en función de las calificaciones, parámetros configurables y condiciones académicas definidas. Gracias al uso del *builder*, es posible construir objetos personalizados que reflejan las reglas específicas de cada materia.

Este módulo centraliza la lógica de evaluación, asegurando coherencia en los resultados y facilitando su extensión o modificación futura.

## Persistencia

#### ❖ Donde se almacenan los datos

Los datos del sistema se almacenan en un archivo con extensión `.db`, correspondiente a una base de datos SQLite3.

Se ha elegido SQLite3 como motor de base de datos debido a su eficiencia para realizar consultas rápidas y su conveniencia en entornos de escritorio, al

tratarse de una solución liviana, sin necesidad de configuración adicional ni servidores externos.

SQLite3 permite un manejo eficaz de las tablas que representan entidades clave del sistema (como materias, parciales, finales y notas), garantizando integridad en los datos y accesos consistentes desde la lógica de aplicación.

### ❖ Cómo se manejan los datos

El acceso y manejo de los datos se realiza a través de una Facade de persistencia, la cual encapsula el uso directo de la biblioteca `sqlite3`. Esta abstracción permite desacoplar la lógica del sistema del motor de base de datos, facilitando así la mantenibilidad, escalabilidad y futuras modificaciones.

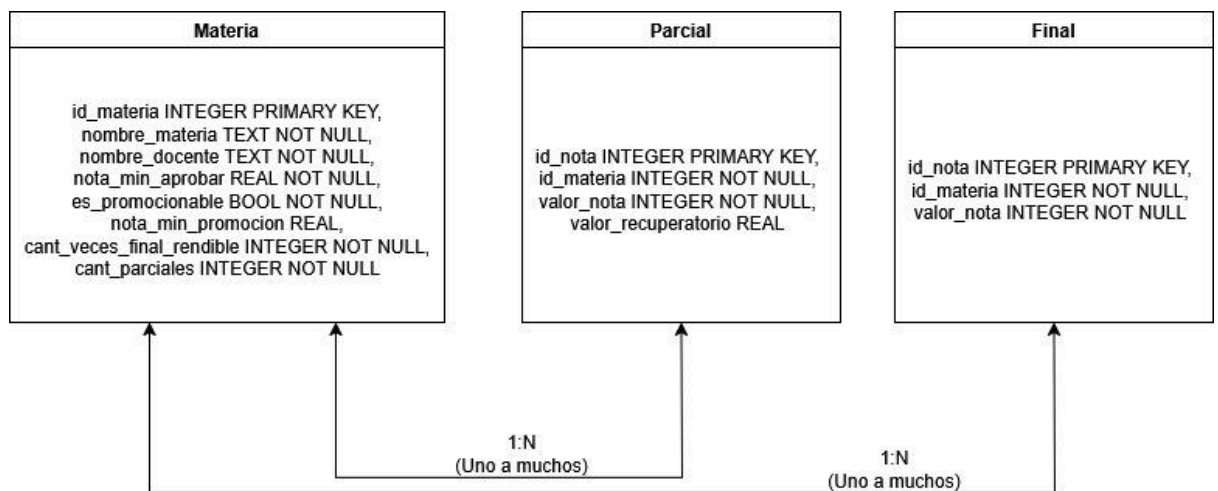
Dentro de esta fachada se implementan las operaciones fundamentales sobre la base de datos SQLite3, tales como:

- ❖ Inserción de registros en las tablas.
- ❖ Consultas y recuperación de datos.
- ❖ Actualización y eliminación de registros.
- ❖ Validaciones y transformaciones asociadas a las entidades persistidas.

El uso de este patrón permite centralizar la lógica de acceso a datos, simplificando el uso de SQLite3 por parte de otros componentes del sistema y mejorando la organización del código.

### ❖ Diagrama Entidad - Relación - D.E.R

A continuación, adjuntamos un diagrama Entidad - Relación (D.E.R.) de nuestra base de datos:



## Manejo de Consola

### ❖ Acciones

Representan las distintas operaciones que el sistema puede ejecutar en función de las solicitudes del usuario. Cada vez que el usuario ingresa un comando, el sistema evalúa su validez y, si es posible, ejecuta la acción correspondiente.

Una vez finalizada la operación solicitada, el sistema muestra nuevamente al usuario el listado de acciones disponibles, permitiendo una navegación fluida y continua a través de las funcionalidades ofrecidas por la aplicación.

### ❖ Funciones de manejo de consola

Este componente contiene la clase responsable del manejo de la Interfaz de Línea de Comandos (CLI), cuya función principal es facilitar la interacción entre el usuario y el sistema.

La CLI permite:

- ❖ Mostrar en pantalla los datos obtenidos y procesados desde el módulo de persistencia.
- ❖ Recibir comandos y entradas del usuario para ejecutar las distintas funcionalidades del sistema.

Este módulo actúa como puente entre el usuario y la lógica interna de la aplicación, ofreciendo una interfaz textual clara y funcional para navegar, consultar y operar sobre los datos académicos registrados.

## Testing

Tanto el módulo de Dominio como el de Persistencia cuentan con pruebas unitarias exhaustivas que cubren todas las funciones y constructores implementados en el sistema.

Estas pruebas evalúan múltiples casos y escenarios, con el objetivo de garantizar el correcto funcionamiento de cada componente y verificar que los resultados obtenidos coinciden con los esperados según la lógica definida.

Este enfoque asegura la calidad, robustez y confiabilidad del sistema durante su desarrollo y mantenimiento.