

<b><u>UTN – FR Mar del Plata – TSP</u></b> <b>Laboratorio III - Primer Parcial Lab 3</b> Abril 2023	<b>Nombre y Apellido</b>	<b>Nota</b>
---	--------------------------	-------------

### IMPORTANTE:

- Crear un proyecto con su Nombre y Apellido (**solamente**).
- Añadir comentarios a su código, modularice y sea prolijo en la codificación, no sumará puntos pero podría restar.
- Al finalizar, comprímalo en un archivo **solamente** con su apellido para luego subirlo al campus
- Desarrolle un main() e invoque únicamente a las funciones que compilen correctamente. La resolución del mismo es obligatoria y sumará puntos extras en caso de necesitarlos. **No se pide ingreso por teclado de ningún tipo.**

En este examen haremos un mini sistema de gestión de un solo estudiante. Nuestra aplicación deberá administrar las notas de las materias. Del estudiante sabemos sus datos básicos: el nombre, el apellido, fecha de nacimiento, su legajo y el mail de contacto. Las materias que el alumno cursa deberán ser almacenadas en una colección. Como el estudiante es aplicado, cursa muchas materias y no tiene tiempo (porque justamente es estudioso), a cada materia le asignó un código único de tres letras (programación sería prog, matemática sería mat y así). Cada materia tiene su nombre descriptivo largo, el docente titular, el ayudante y los días de cursada (es un texto descriptivo, nada extravagante). Además cada materia tiene un listado de notas. Cada nota tiene un número, una instancia y una fecha. Estos tres atributos la hacen única y no puede ni debe haber una repetida en la colección de notas. Suficientes pistas. Ambas colecciones (las materias y las notas) tendrán los mismos métodos (agregar, contar, listar y eliminar). Acá matamos dos temas de un solo tiro. Basta de pistas. Dije basta. Pasemos a las reglas del negocio. Si un alumno quiere agregar una materia que ya existe, será castigado, digo, lanzará una excepción (creada por ustedes donde agregaran el código de la materia repetida). Si el estudiante quiere agregar una nota para un examen que ya rindió, también lanzará una excepción con los datos de esa nota así luego el profe revisa esa información. Dicho esto, pensemos que cada materia tiene también un límite de instancias por lo cual si quiero agregar más notas, también podemos castigar al usuario con otra excepción (aquí no guardaremos nada en especial). Y JSON? Y Candela? ¿Y la moto? (quizás sean muy jóvenes para entender la referencia pero yo me debo a los oldies +30). Vamos a guardar en un archivo .json toda la información del estudiante, sus materias y notas. Hay muchas formas de hacerlo y la que no resta puntos es la que usa funciones en cada clase, es prolija y extensible. Cualquiera otra vuelve con menos puntos y observaciones en la corrección. Tenemos las suficientes herramientas para hacer un buen trabajo. ¿Archivos binarios? No. Cada pregunta que me hagan y esté en el enunciado suma un poroto. A los dos porotos se habilita un enunciado extra. Hijos del rigor? Hijos del rigor.

[https://github.com/benoffi7/Programacion-Laboratorio-3-UTN-MDP/blob/master/2023/json\\_archivos/src/JsonUtiles.java](https://github.com/benoffi7/Programacion-Laboratorio-3-UTN-MDP/blob/master/2023/json_archivos/src/JsonUtiles.java)

[https://github.com/benoffi7/Programacion-Laboratorio-3-UTN-MDP/blob/master/2023/json\\_archivos/java-json.jar](https://github.com/benoffi7/Programacion-Laboratorio-3-UTN-MDP/blob/master/2023/json_archivos/java-json.jar)

Diseño de clases -> 15pts

Interfaces -> 15pts

JSON -> 20pts

Genericidad -> 20pts

Excepciones -> 15pts

Colecciones -> 15pts