

UTN – UA Mar del Plata – TSP Laboratorio de Computación 3 Segundo Parcial Junio 2024	Nombre y Apellido:	Nota:
---	--------------------	-------

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.
- 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.**

George Lucas y Disney nos contratan para hacer una mini app para el magnífico y maravilloso mundo de Star Wars.

- 1) Copiar el contenido de este archivo <https://pastebin.com/30sAQYka> y pegarlo en un archivo .json dentro del proyecto.
- 2) Usar este archivo https://github.com/benoffi7/Programacion-Laboratorio-3-UTN-MDP/blob/master/2023/json_archivos/src/JsonUtiles.java para recuperar el contenido de JSON. Bajar la librería .jar https://github.com/benoffi7/Programacion-Laboratorio-3-UTN-MDP/blob/master/2023/json_archivos/java-json.jar
- 3) Modelar las clases necesarias para pasar el contenido de JSON a JAVA. (20 pts)
- 4) Escribir una clase genérica llamada StarWarsRegistro<T> que actúe como un registro de personajes del universo de Star Wars. Esta clase debe utilizar una colección genérica para almacenar objetos de tipo T, donde T representa un personaje de Star Wars. Además, implementa los siguientes métodos: (30 pts)

agregarPersonaje(T personaje): Agrega un personaje al registro.

obtenerPersonajePorNombre(String nombre): Devuelve el primer personaje encontrado en el registro que coincida con el nombre especificado.

eliminarPersonaje(T personaje): Elimina la primera aparición del personaje especificado del registro.

listarPersonajes(): Devuelve una lista con todos los personajes del registro.

limpiar(): Elimina todos los personajes del registro.

Los elementos no se deben repetir dentro del registro.

- 5) La clase StarWarsRegistro<T> debe manejar y lanzar una excepción personalizada llamada PersonajeNoEncontradoException cuando el método obtenerPersonajePorNombre() no encuentra ningún personaje con el nombre especificado. Almacenar en dicha excepción el nombre ingresado. (20 pts)
- 6) Guardar en un archivo binario todas las naves de los personajes. Mostrar dicho archivo. (20 pts)

7) Escribir un main para probar toda la funcionalidad (10 pts)