



UNLP. Facultad de Informática.  
**Seminario de Lenguajes opción Rust**  
**Cursada 2023**

## **Práctica 5**

### **Archivos, Testing, Manejo de errores**

1- En base al ejercicio 7 del tp#3 implemente lo siguiente:

a- Al agregar un auto si supera el límite de la concesionaria debe arrojar un error propio con un mensaje de contexto.

b- Haga todos los tests correspondientes para probar en profundidad los métodos que agregan un auto y eliminan un auto de la concesionaria , obteniendo el mayor porcentaje de coverage sobre el código que realiza las operaciones.

c- Una vez hecho el punto anterior debe hacer que los autos de la concesionaria se almacenen en un archivo en formato JSON. Agregue y modifique lo que considere necesario para que:

- Al agregar un nuevo auto se abra el archivo de autos guardados y lo agregue a dicho archivo, para ello debería abrirlo en modo r+w mire lo siguiente para ver cómo realizarlo: <https://doc.rust-lang.org/std/fs/struct.OpenOptions.html>
- Eliminar un auto: al eliminar un auto se debe eliminar este del archivo.

No debe modificar los tests hechos en el punto b. Si puede agregar más en caso de que haga nueva funcionalidad..

2- En base al ejercicio 8 del tp#3 implemente lo siguiente:

a- Realice todos los tests de la funcionalidad implementada obteniendo un coverage de por lo menos 90%

b- Una vez obtenido dicho coverage, las canciones de la playlist deben ser guardadas en un archivo en formato JSON, por lo tanto las operaciones que agreguen, quiten o modifiquen la playlist deben estar respaldadas sobre dicho archivo.

No debe modificar los tests hechos en el punto a. Si puede agregar más en caso de que haga métodos nuevos. Recuerde también que se debe seguir manteniendo un coverage de al menos 90%,



**UNLP. Facultad de Informática.**  
**Seminario de Lenguajes opción Rust**  
**Cursada 2023**

3- En base al ejercicio 9 del tp#3 implemente lo siguiente:

a- Realice todos los tests de la funcionalidad implementada obteniendo un coverage de por lo menos 90%

b - Ahora el registro de atenciones debe persistir en un archivo en formato JSON, es decir todas la operaciones que lectura, agregar y modificación de atenciones se realizan sobre un archivo.No debe modificar los tests hechos en el punto a. Si puede agregar más en caso de que haga métodos nuevos para cumplir con este punto. Recuerde también que se debe seguir manteniendo un coverage de al menos 90%,

4- En base al ejercicio 10 del tp#3 implemente lo siguiente:

a- Realice todos los tests de la funcionalidad implementada obteniendo un coverage de por lo menos 90%

b- Tanto los libros con sus copias como la administración de préstamos se realizan sobre archivos en formato JSON. Realice las modificaciones pertinentes para poder hacerlo así. No debe modificar los tests hechos en el punto a. Si puede agregar más en caso de que haga métodos nuevos para cumplir con este punto . Recuerde también que se debe seguir manteniendo un coverage de al menos 90%.

5- En base al ejercicio 3 del tp#4 implemente lo siguiente:

a- Realice todos los tests de la funcionalidad implementada obteniendo un coverage de por lo menos 90%

b- Todas las suscripciones deben almacenarse en un archivo en formato JSON, implemente lo necesario para que toda la funcionalidad de las suscripciones se realice guardando, leyendo o modificando archivos.No debe modificar los tests hechos en el punto a. Si puede agregar más en caso de que haga métodos nuevos para cumplir con este punto. Recuerde también que se debe seguir manteniendo un coverage de al menos 90%.

6- En base al ejercicio 5 del tp#4 implemente lo siguiente:

a- Realice todos los tests de la funcionalidad implementada obteniendo un coverage de por lo menos 90%

b- Todos los balances de los usuarios así como las transacciones deben persistir en archivos en formato JSON. No debe modificar los tests hechos en el punto a. Si puede agregar más en caso de que haga métodos nuevos para cumplir con este punto . Recuerde también que se debe seguir manteniendo un coverage de al menos 90%.