



UNLP. Facultad de Informática.  
**Seminario de Lenguajes opción Rust**  
**Cursada 2024**

## **Práctica 3**

### **Structs, Enums, Option, Collections**

Nota: Para todos los ejercicios realizar los tests de unidad correspondientes.

1- Escribir un programa que defina una estructura Persona que tenga campos para el nombre, la edad y la dirección(que puede ser nulo al momento de la creación de una persona). Para dicha estructura implemente los siguientes métodos:

- new: que pasando los parámetros correspondientes, crea una Persona y la retorna.
- to\_string: que retorna un string con los datos de la persona concatenados sobre el mensaje ejecutado por ej:  
person.to\_string() , donde person es una variable del tipo Persona.
- obtener\_edad: retorna la edad de la persona.
- actualizar\_direccion(nueva\_direccion)

2- Escribir un programa que defina la estructura Rectángulo que tenga campos para la longitud y el ancho. Para dicha estructura implemente los siguientes métodos:

- new: que pasando los parámetros correspondientes, crea un Rectángulo y lo retorna.
- calcular\_area: calcular el área y la retorna.
- calcular\_perimetro: calcula el perímetro y lo retorna.
- es\_cuadrado: retorna true si es cuadrado, false caso contrario

3- Escribir un programa que defina una estructura Fecha que tenga campos para el día, el mes y el año. Para dicha estructura implemente los siguientes métodos:

- new: que pasando los parámetros correspondientes, crea una Fecha y la retorna.
- es\_fecha\_valida: retorna true si es una fecha válida, false caso contrario.//tenga en cuenta los años bisiestos también.
- es\_bisiesto: retorna true si el año de la fecha pertenece a un año bisiesto.
- sumar\_dias(dias): suma la cantidad de días a la fecha, modificándose
- restar\_dias(dias): resta la cantidad de días a la fecha, modificándose
- es\_mayor(una\_fecha): que retorna true si la fecha que recibe el mensaje es mayor a la fecha pasada por parámetro.



UNLP. Facultad de Informática.  
**Seminario de Lenguajes opción Rust**  
**Cursada 2024**

4- Escribir un programa que defina la estructura Triángulo que tenga campos para las longitudes de sus tres lados. Para dicha estructura implemente los siguientes métodos:

- new: que pasando los parámetros correspondientes, crea un Triángulo y lo retorna.
- determinar\_tipo: retorna el tipo del triángulo, los tipos pueden ser equilátero, isósceles o escaleno.
- calcular\_area: calcular el área y la retorna.
- calcular\_perimetro: calcula el perímetro y lo retorna.

5- Escribir un programa que defina una estructura Producto que tenga campos para el nombre, el precio bruto y un número identificador. Para dicha estructura implemente los siguientes métodos:

- new: que pasando los parámetros correspondientes, crea un Producto y lo retorna.
- calcular\_impuestos(porcentaje\_de\_impuestos): retorna el valor de impuestos sobre el precio bruto
- aplicar\_descuento(porcentaje\_de\_descuento): retorna el valor del porcentaje de descuento sobre el precio bruto
- calcular\_precio\_total(porcentaje\_de\_impuestos, porcentaje\_descuento): retorna el precio total a pagar aplicando impuesto y descuento. Tenga en cuenta que los parámetros son opcionales.

6- Escribir un programa que defina una estructura Estudiante que tenga campos para el nombre, el número de identificación y las calificaciones de exámenes. De cada Examen se conoce el nombre de la materia y la nota. Para dichas estructuras implemente los siguientes métodos:

- ❖ Examen:
  - new: que pasando los parámetros correspondientes, crea un Examen y lo retorna.
- ❖ Estudiante:
  - new: que pasando los parámetros correspondientes, crea un Estudiante y lo retorna.
  - obtener\_promedio: retorna el promedio de las notas.
  - obtener\_calificacion\_mas\_alta: retorna la nota más alta.
  - obtener\_calificacion\_mas\_baja: retorna la nota más baja.

Nota: Tenga en cuenta que el Estudiante puede tener entre 0 y n notas de examen.



UNLP. Facultad de Informática.  
**Seminario de Lenguajes opción Rust**  
**Cursada 2024**

7- Defina una estructura llamada ConcesionarioAuto donde se conoce el nombre, la dirección y tiene una capacidad máxima para albergar X cantidad de autos. De los autos se conocen los campos de la marca, modelo, año, precio bruto y color que pueden ser: rojo, verde, azul, amarillo, blanco o negro.

Para dichas estructuras implemente los siguientes métodos:

- ❖ ConcesionarioAuto:
  - new: que pasando los parámetros correspondientes, crea un ConcesionarioAuto y lo retorna.
  - agregar\_auto(auto): agrega un auto a la lista de autos que tiene sin superar la máxima cantidad para albergarlos y retorna true, en caso de que lo supere no lo agrega y retorna false.
  - eliminar\_auto(auto): elimina un auto de la lista de autos.
  - buscar\_auto(auto): busca un auto y si lo encuentra lo retorna.
- ❖ Auto:
  - new: que pasando los parámetros correspondientes, crea un Auto y lo retorna.
  - calcular\_precio: retorna el precio del auto aplicando los siguientes criterios:
    - si es de color primario le aplica un recargo del 25%, sino le aplica un descuento del 10%.
    - si la marca es BMW le aplica un recargo del 15%-
    - si el año es menor a 2000 le aplica un descuento del 5%.

8- Defina la estructura Cancion con campos para el título, el artista y el género. El género puede ser rock, pop, rap, jazz, otros. Luego modele una playlist. La playlist está compuesta por una lista de canciones y un nombre, y se permiten hacer las siguientes acciones sobre ella:

- agregar canción.
- eliminar canción.
- mover canción // mueve la canción a una determinada posición de la playlist.
- buscar canción por nombre.
- obtener las canciones de un determinado género.
- obtener las canciones de un determinado artista.
- modificar título de la playlist.
- eliminar todas las canciones.x1



UNLP. Facultad de Informática.  
**Seminario de Lenguajes opción Rust**  
**Cursada 2024**

9.-Dada una cadena de veterinarias se desea implementar un sistema de atención de pacientes para cada veterinaria, de la veterinaria se conoce el nombre, la dirección y un id. Para la atención de mascotas se requiere administrar una cola de atención. De la mascota se conoce el nombre, la edad, el tipo de animal(perro, gato, caballo, otros) y su dueño. Del dueño se conoce el nombre, la dirección y un teléfono de contacto. Luego de la atención se desea tener un registro de las atenciones realizadas guardando los datos de la mascota, el diagnóstico final, tratamiento y fecha de la próxima visita si es que se requiere.

Dado todo lo mencionado anteriormente implemente los métodos para realizar las siguientes acciones:

- crear una veterinaria.
- agregar una nueva mascota a la cola de atención de la veterinaria.
- agregar una nueva mascota a la cola de atención pero que sea la siguiente en atender porque tiene la máxima prioridad.
- atender la próxima mascota de la cola.
- eliminar una mascota específica de la cola de atención dado que se retira.
- registrar una atención.
- buscar una atención dado el nombre de la mascota, el nombre del dueño y el teléfono.
- modificar el diagnóstico de una determinada atención.
- modificar la fecha de la próxima visita de una determinada atención.
- eliminar una determinada atención.

**Nota:** para la fecha utilice lo implementado en el punto 3.

10-Para una biblioteca se desea implementar un sistema de préstamos de libros. De la biblioteca se conoce el nombre y la dirección, las copias de los libros a disposición para prestar y los préstamos efectuados. Los libros a disposición es un registro donde se indica la cantidad de ejemplares que tiene a disposición para prestar de determinado libro. De cada libro se conoce el isbn, el título, autor, número de páginas, género(novela, infantil, técnico, otros). Para registrar un préstamo se requiere el libro, el cliente, la fecha de vencimiento del préstamo, la fecha de devolución y el estado que puede ser devuelto o en préstamo. Del cliente se conoce el nombre, teléfono y dirección de correo electrónico. Implemente los métodos necesarios para realizar las siguientes acciones:



UNLP. Facultad de Informática.  
**Seminario de Lenguajes opción Rust**  
**Cursada 2024**

- obtener cantidad de copias: dado un determinado libro retorna la cantidad de copias a disposición que hay para prestar de dicho libro.
  - decrementar cantidad de copias a disposición; dado un libro decrementa en 1 la cantidad de copias de libros a disposición para prestar.
  - incrementar cantidad de copias a disposición: dado un libro incrementa en 1 la cantidad de copias del libro a disposición para ser prestado.
  - contar préstamos de un cliente: devuelve la cantidad de préstamos en estado “en préstamo” de un determinado cliente.
  - realizar un préstamo de un libro para un cliente: crea un préstamo de un libro para un determinado cliente cumpliendo con lo siguiente
    - ◆ el cliente no tenga más de 5 préstamos en el estado “en préstamo”
    - ◆ haya al menos una copia disponible en el registro de copias a disposición.
- De ser así descuenta 1 en el registro de “copias a disposición” y retorna true, si no cumple con alguna de las condiciones retorna false.
- ver préstamos a vencer el los próximos días: retorna una lista de préstamos a vencer el los próximos días, el valor de días es pasado por parámetro.
  - ver los préstamos vencidos: retorna una lista de préstamos en el estado “en préstamos” donde la fecha de vencimiento es menor a la fecha actual.
  - buscar préstamo: dado un libro y un cliente busca un préstamo y lo retorna si existe.
  - devolver libro: dado un libro y un cliente se busca el préstamo y se cambia al estado “devuelto”, se registra la fecha de devolución y se incrementa la cantidad de libros en 1 del libro devuelto en el registro de copias a disposición.

**Nota:** para la fecha utilice lo implementado en el punto 3.