**Junior**

Topics

- Collections, Annotations, JavaIO

- List, Queue, Set, Map

- Inheritance, Interfaces

- Relational DB, SQL Basics

- Web Basics

1. Crea una lista de números enteros utilizando ArrayList y agrega algunos números. Luego, utiliza un bucle para imprimir los números en la lista.
2. Crea un HashSet de nombres de frutas y agrega algunas frutas. Luego, utiliza un bucle para imprimir las frutas en el conjunto.
3. Crea una clase abstracta llamada "Vehiculo" con un método abstracto acelerar(). Luego, crea clases derivadas como "Automovil" y "Motocicleta" que implementen el método acelerar() según el tipo de vehículo.
4. Crea una tabla llamada "Producto" en una base de datos utilizando SQL. La tabla debe tener columnas como nombre, precio y cantidad. Luego, escribe una consulta SQL para seleccionar todos los productos de la tabla.
5. Crea una aplicación web utilizando el framework Spring Boot. Crea un controlador que maneje una solicitud GET y devuelva un mensaje de bienvenida en formato JSON.
6. Crea una clase llamada "Estudiante" con atributos como nombre, edad y lista de asignaturas. Utiliza la anotación @Override para sobrescribir el método toString() y mostrar los detalles del estudiante.
7. Crea un LinkedList de nombres de ciudades y agrega algunas ciudades. Luego, utiliza un bucle para imprimir las ciudades en la lista.
8. Crea una interfaz llamada "Animal" con un método hacerSonido(). Luego, crea clases como "Perro" y "Gato" que implementen la interfaz y proporcionen una implementación del método hacerSonido() para cada animal.
9. Crea una tabla llamada "Empleado" en una base de datos utilizando SQL. La tabla debe tener columnas como nombre, salario y cargo. Luego, escribe una consulta SQL para seleccionar todos los empleados cuyo salario sea mayor a cierto valor.
10. Crea una aplicación web utilizando el framework Spring Boot. Crea un controlador que maneje una solicitud POST para registrar un nuevo usuario y muestre un mensaje de éxito en una página.

Consideraciones:

Para lograr los objetivos el desarrollador puede:

Utilizar la IDE que le resulte más cómodo.

Herramientas como Docker, Git, UML.

Organizar el proyecto como mejor le parezca.

Los ejercicios tienen que ser ejecutables.