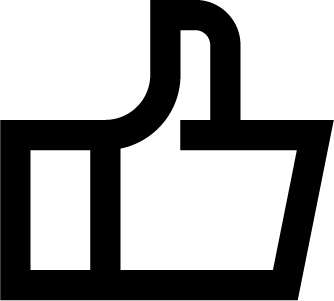
Programación Java

 **Generics + Wrapper + Lambda + Streams**

// Práctica integradora

Objetivo

El objetivo de esta guía práctica es que podamos profundizar el uso de los Generics, haciendo uso en de las diferentes colecciones de datos y de la Api Stream de java para el manejo de la programación funcional junto con las expresiones lambda. Para esto vamos a plantear una serie de ejercicios simples e incrementales (ya que vamos a ir trabajando y agregando lógica a las clases que tenemos que construir), lo que nos permitirá repasar los temas que estudiamos.

**¿Are you ready? **

# Ejercicio 1

Inicia creando una clase Vehículo con los atributos modelo, marca y costo. Luego crea una clase garaje con los atributos id o identificador único y una lista de vehículos. Crea además los constructores de las clases y los métodos Setter y Getter.



Ejercicio 2

Haz una clase Main con el método main para representar un escenario donde se crea una instancia de la clase garaje con una lista de vehículos según la tabla.

| **Marca** | **Modelo** | **Precio** |
| --- | --- | --- |
| Ford | Fiesta | 1000 |
| Ford | Focus | 1200 |
| Ford | Explorer | 2500 |
| Fiat | Uno | 500 |
| Fiat | Cronos | 1000 |
| Fiat | Torino | 1250 |
| Chevrolet | Aveo | 1250 |
| Chevrolet | Spin | 2500 |
| Toyota | Corola | 1200 |
| Toyota | Fortuner | 3000 |
| Renault | Logan | 950 |

****Ejercicio 3

Haciendo uso del método *sort* en la lista de Vehículos con expresiones lambda, obtén una lista de vehículos ordenados por precio de menor a mayor, imprime por pantalla el resultado.

****Ejercicio 4

De la misma forma que el ejercicio anterior, imprime una lista ordenada por marca y a su vez por precio.

****Ejercicio 5

Se desea extraer una lista de vehículos con precio no mayor a 1000, luego otra con precios mayor o igual 1000 y por último, obtén el promedio total de precios de toda la lista de vehículos.