INTERFACES FUNCIONALES

INTERFACES

- *Contrato* que compromete a una clase a implementar una serie de métodos.
- Se pueden utilizar como referencias a la hora de crear objetos.

```
List<String> lista = new ArrayList<>();
```

- Desde Java SE 8, pueden incluir implementación (static y default).

INTERFACES

```
public interface Interfaz {

    public void metodo();

    default public void metodoPorDefecto() {
            System.out.println("Este es uno de los nuevos
                                        métodos por defecto");
    }

    public static void metodoEstatico() {
            System.out.println("Método estático en
                                            un interfaz");
    }

}
```

Abstracto

Por defecto

Estático

INTERFAZ FUNCIONAL

- **Interfaz que solo tiene un método abstracto.**
- Puede tener uno o varios métodos por defecto y/o estáticos
- Puede tener varios métodos abstractos, siempre que *todos menos uno* sobrescriban un método público de la clase *Object*.
- ***Usualmente, los implementamos con una clase anónima.***
- Muchos interfaces conocidos son *funcionales*.

INTERFAZ FUNCIONAL

- Java SE 8 también incorpora la anotación @FunctionalInterface que comprueba en tiempo de compilación si se cumplen con las condiciones anteriores.

INTERFACES FUNCIONALES Y EXPRESIONES LAMBDA

- Altamente relacionados
- De alguna forma, allí donde se espera una instancia de una clase que implemente una interfaz funcional, podremos usar una expresión lambda.

```
Collections.sort(lista, (str1, str2)-> str1.length()-str2.length());
```