Clase 21: Java 8 (Parte II)

Programación & Laboratorio III

Stream

- Un Stream no es ni más ni menos que un conjunto de funciones que se ejecutan de forma anidada.
- Las operaciones para el procesamiento de sus datos:
  - De forma declarativa usando expresiones lambda.
  - Permitiendo el posible encadenamiento de varias operaciones, con lo que se logra tener un código fácil de leer y con un objetivo claro.
  - De forma secuencial o paralela
- La API de Stream nos permite realizar operaciones sobre colecciones de datos usando el modelo **filtro/mapeo/reducción**, en el cual se seleccionan los datos que se van a procesar (filtro), se convierten a otro tipo de dato (mapeo) y al final se obtiene el resultado deseado (reducción).

Stream - Operaciones sobre colecciones de datos

- Casi todas las aplicaciones de Java crean y procesan colecciones. Son esenciales para muchas tareas de programación: permiten agrupar y procesar datos.

Operaciones sobre colecciones de datos (2)

- Ejemplo:

```
public class Transaccion {
   private LocalDateTime dateTime;
   private int precio;
   private Proveedor proveedor;

   //getters & setters
}

public class Proveedor {
   private String nombre;
   private String ciudad;

   //getters & setters
}
```

Operaciones sobre colecciones de datos (3)

- Sumar los precios de las transacciones correspondientes a la ciudad de Mar del Plata:

```
public static void main(String[] args) {
    List<Transaccion> transacciones = new ArrayList<>();
    …
    int suma = 0;
    for (Transaccion t : transacciones) {
        if (t.getProveedor().getCiudad().equals(“Mar del Plata”)) {
            suma += t.getPrecio();
        }
    }
    System.out.println(suma);
}
```

Operaciones sobre colecciones de datos (4)

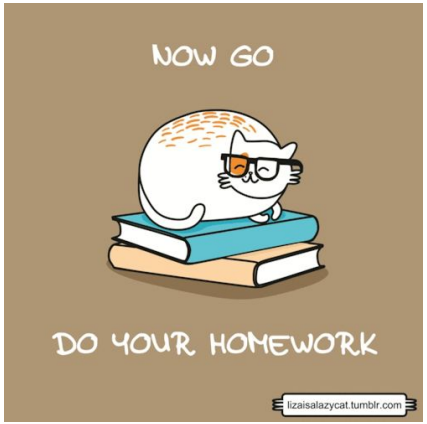
```
public static void main(String[] args) {
	List<Transaccion> transacciones = new ArrayList<>();
	…
	int suma = transacciones.stream()
			.filter(t -> t.getProveedor().getCiudad().equals(“Mar del Plata”))
			.mapToInt(Transaccion::getPrecio)
			.sum();
}
```

1. Fuente de información → La lista de transacciones
2. Operaciones intermedias → operación filter y mapToInt.
3. Operación terminal que produce un resultado → operación sum.

Operaciones sobre colecciones de datos (4)

- Las operaciones intermedias retornan otro Stream. Esto permite que podamos hacer un encadenamiento de operaciones .
- Funcionamiento → Las operaciones intermedias son encoladas hasta que una operación terminal sea invocada. Ninguna de las operaciones intermedias es ejecutada hasta que se invoca una operación terminal.

Investigar Date API



Bibliografía oficial

- http://www.oracle.com/technetwork/articles/java/ma14-java-se-8-streams-2177646.html
- https://docs.oracle.com/javase/8/docs/api/java/util/stream/package-summary.html
- http://www.oracle.com/technetwork/articles/java/jf14-date-time-2125367.html
- http://www.baeldung.com/java-8-date-time-intro