**Lambdas:**

* Estructura básica: (parámetros) -> {cuerpo-lambda}

**Se utilizan únicamente en caso de:**

* Código con un tiempo de vida corto.
* Encapsular código específico.

A partir de Java 8 se incluyeron muchas funciones lambda como él **.foreach**

*ArrayList*<Film> films = **new** ArrayList();

films.forEach(f *->* System.out.println(f.toString()));

**Stream**

Es un método que se añadió a partir de Java 8 y puede ser utilizado en las colecciones, y esto para que puedan manejar las expresiones lambda.

objects.stream()

**Filter**

Es otro de los métodos para el manejo de lambdas y este sirve para utilizarlo como si fuese un If en programación estructurada

objects.stream().filter()

**forEach**

Este es otro metodo que no necesita ser trabajado específicamente con stream() amenos que se quira trabajar en conjunto con otro como el filter()

objects.stream().forEach()

**Ejemplo**

movies.stream()

.filter(m *->* m.vista == true)

.forEach(m *->* System.out.println("Nombre "+m.Nombre));

**Predicate**

Es utilizado como forma de buenas prácticas para pasar como parámetros en la lambda cuando tiene muchas condicionales es decir muchas líneas de código, estas deben ser de formato condicional

*Predicate*<Book> bookReaded = b *->* b.getIsReaded();

books.stream().filter(bookReaded);

**Comsumers**

Son todas las acciones que se realizaran en una iteración

*Consumer*<Book> bookConsumer = b *->* b.UpperCase();

books.stream().forEach(bookConsumer);