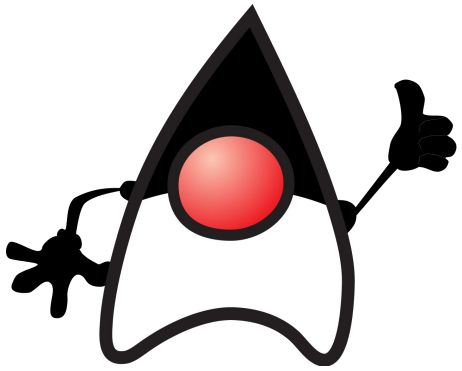
PROGRAMACIÓN REACTIVA EN JAVA 101

ISC ROSARIO ELENA GORDILLO PADILLA



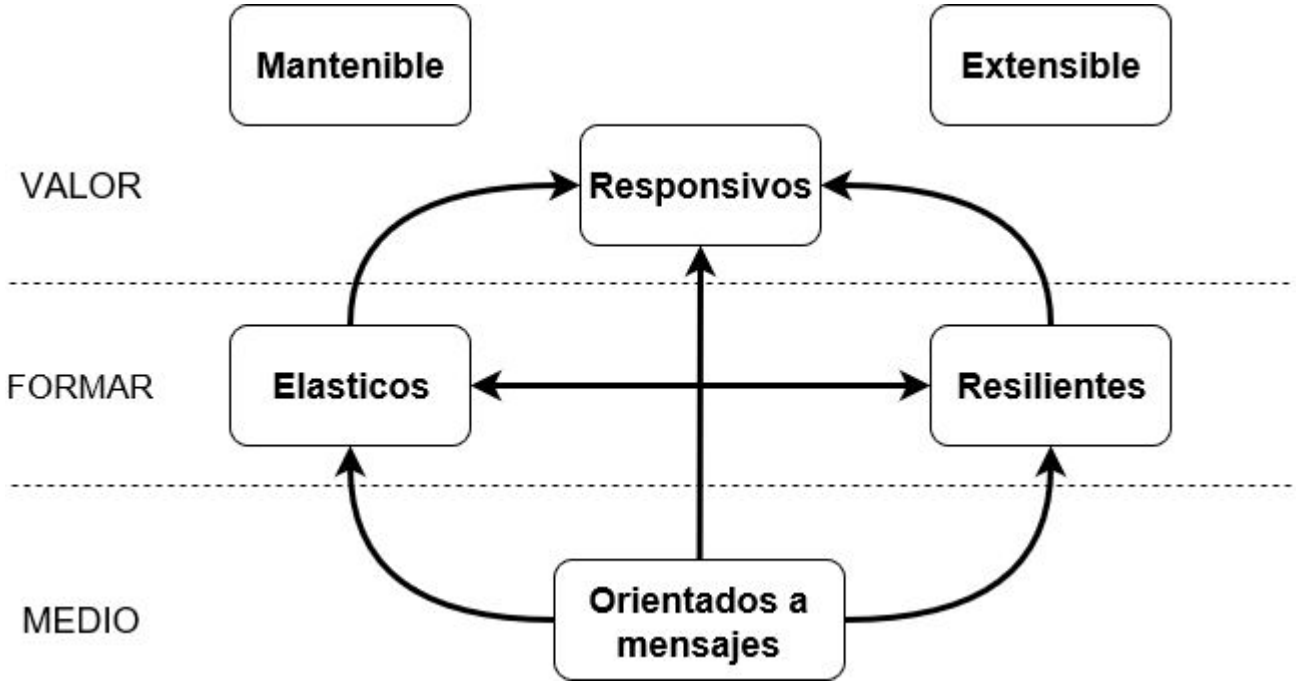
USO DE LA PROGRAMACIÓN REACTIVA

- **Aplicaciones con interfaz gráfica en la que el usuario interactúe con la aplicación:**
  - Web
  - Móviles
- **Trabajo con información en tiempo real**
  - Estadísticas para reportes en tiempo real
- **Flujos interminables de datos**
  - Redes Sociales
- **Envío de información desde Backend**
- **Eventos de I/O**
  - Escritura lectura de archivos
  - Senso a traves de hardware

¿PORQUÉ EL BOOM DE LA PROGRAMACIÓN REACTIVA?

Los usuarios esperan tiempos de respuesta de milisegundos y un tiempo de actividad de 100%, con una gran cantidad de datos.

MANIFIESTO REACTIVO



CONCEPTOS

- **Reactividad:** Propiedad de reaccionar ante una situación
- **Asincronía:** Objetos o eventos que no estar coordinados con el tiempo
- **Flujo de datos:** Movimiento que tienen los datos dentro de un sistema (datos de entrada y datos de salida)

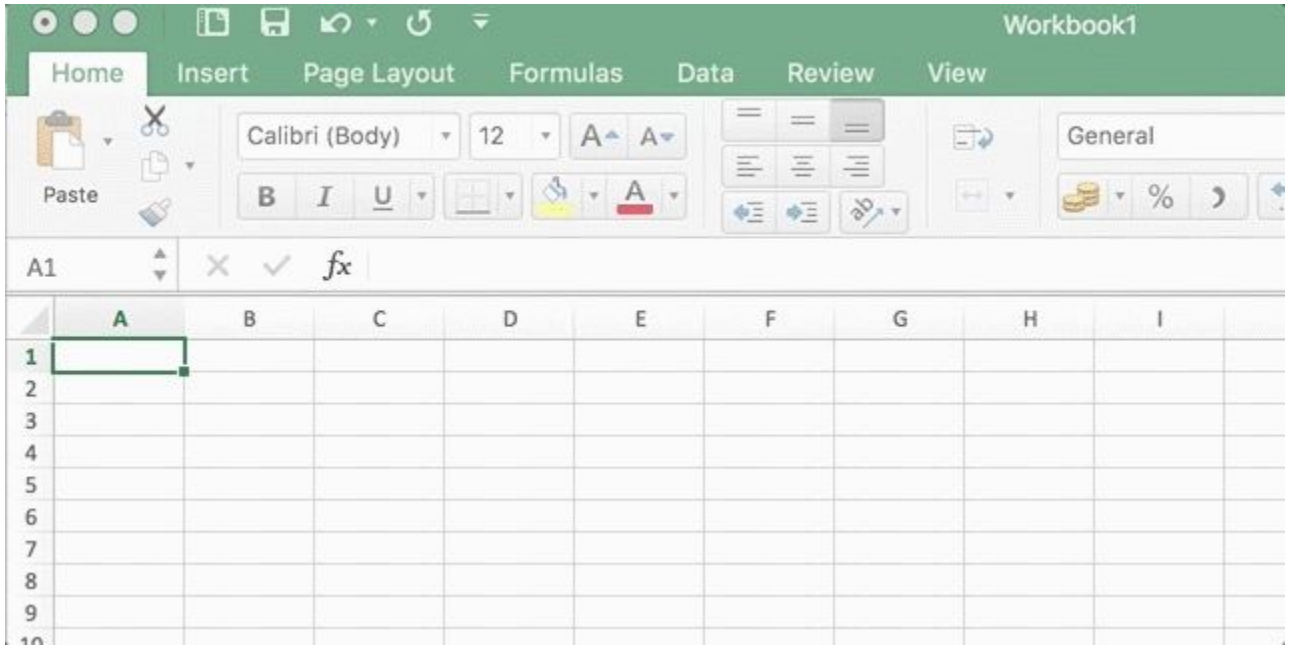
¿QUÉ ES LA PROGRAMACIÓN REACTIVA?

“Paradigma enfocado en el trabajo con flujo de datos finitos o infinitos de manera asíncrona ante los cuales podemos reaccionar y actuar en consecuencia”

- **Rx (Reactive Extension)**

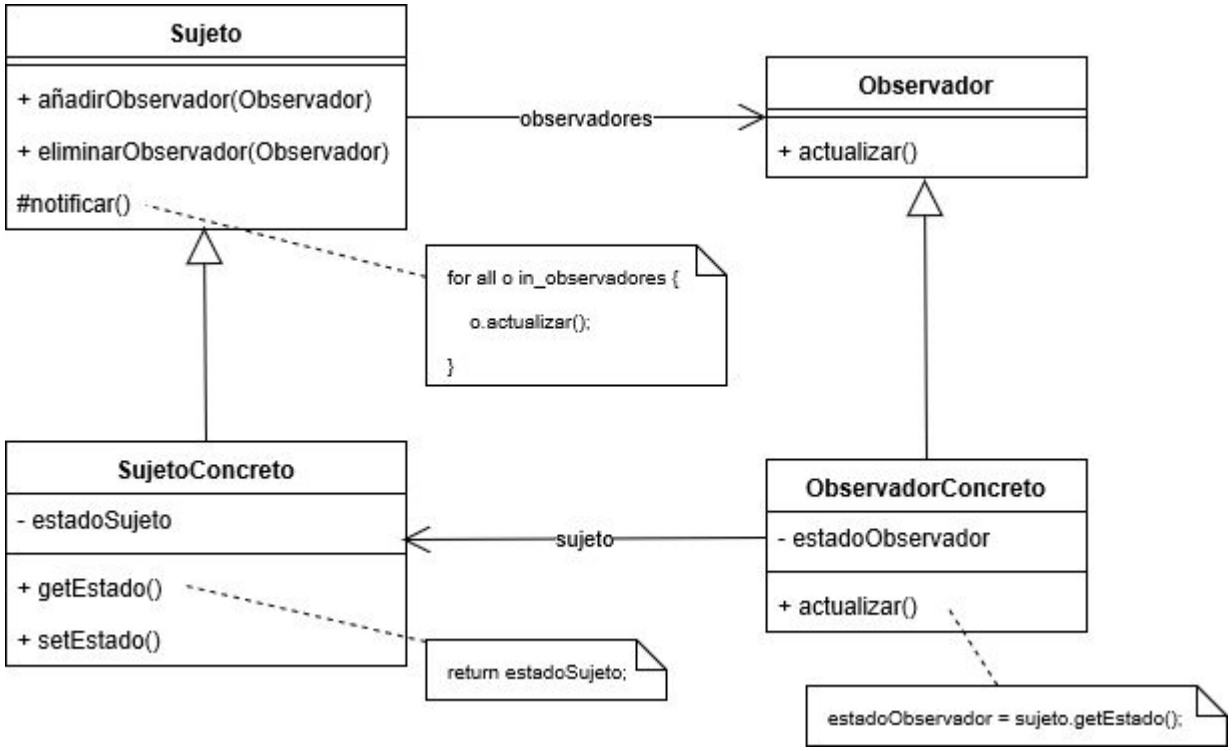
- **FRP (Functional Reactive Programming)**
- **Observer Pattern (Gang of four)**
- **Iterator Pattern**

EJEMPLO


Workbook1
Home
Insert
Page Layout
Formulas
Data
Review
View
Paste
Calibri (Body)
12
General
A1
fx
A
B
C
D
E
F
G
H
I
1
2
3
4
5
6
7
8
9

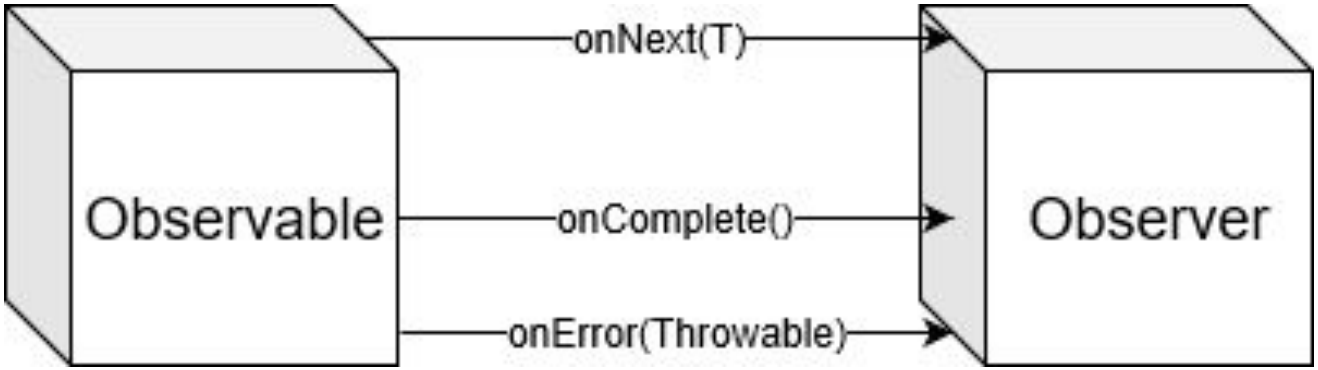
GENERACIONES

GENERACIÓN 0



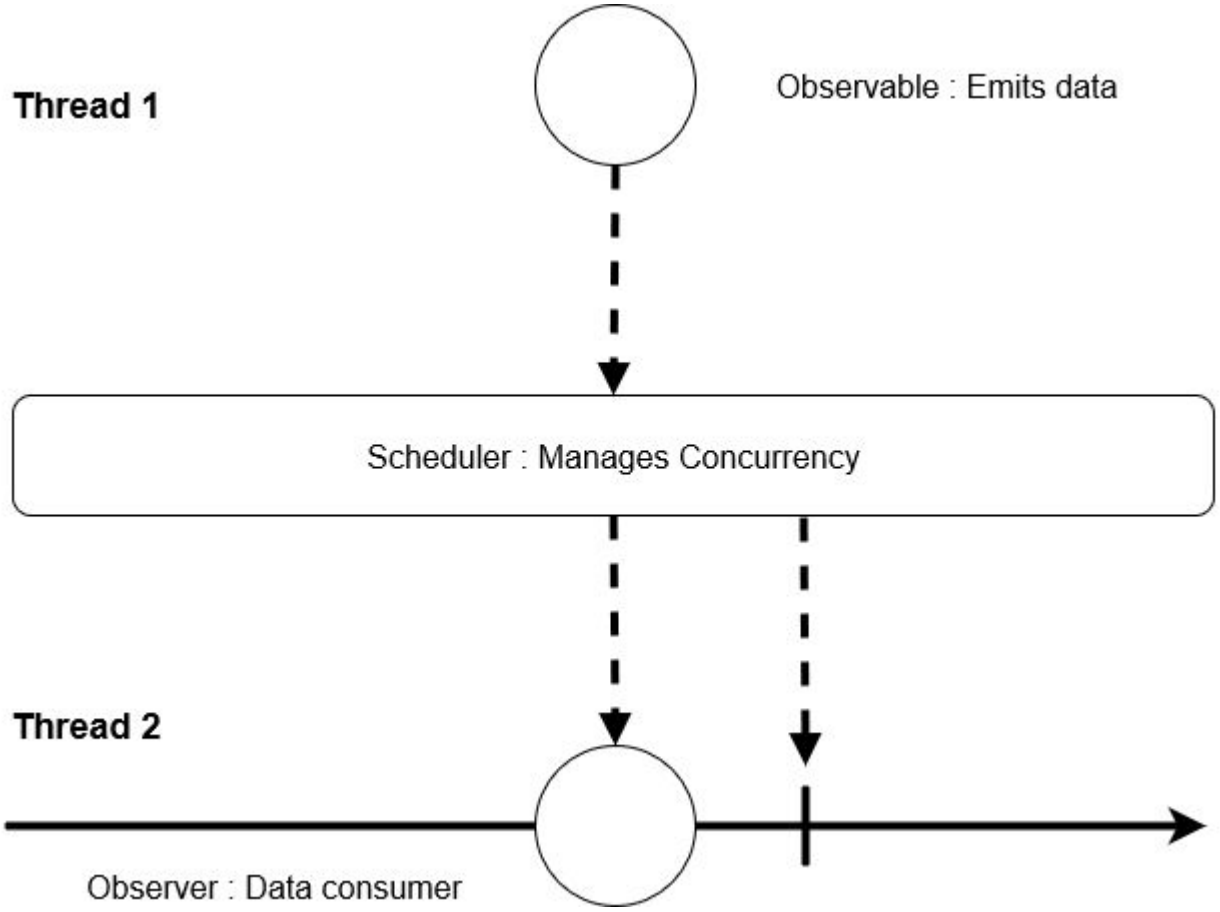
- java.util.Observable/Observer

GENERACIÓN 1

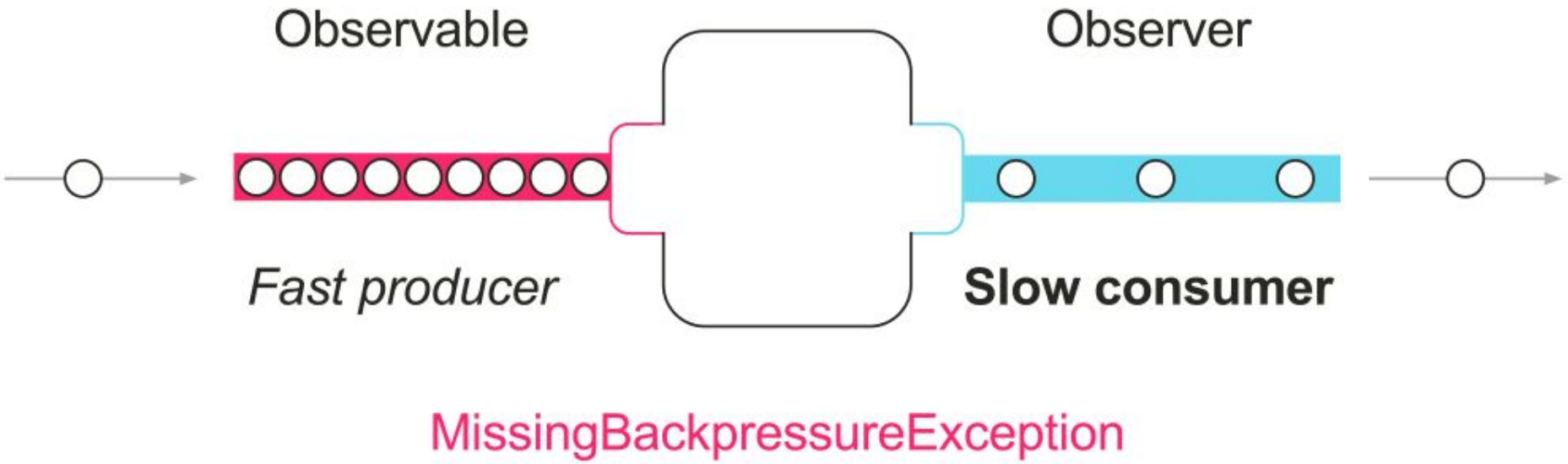


- Año 2013 nace RX.NET
- Se porta a Java con la librería RxJava
- io.reactivex.Observable/Observer

GENERACIÓN 1

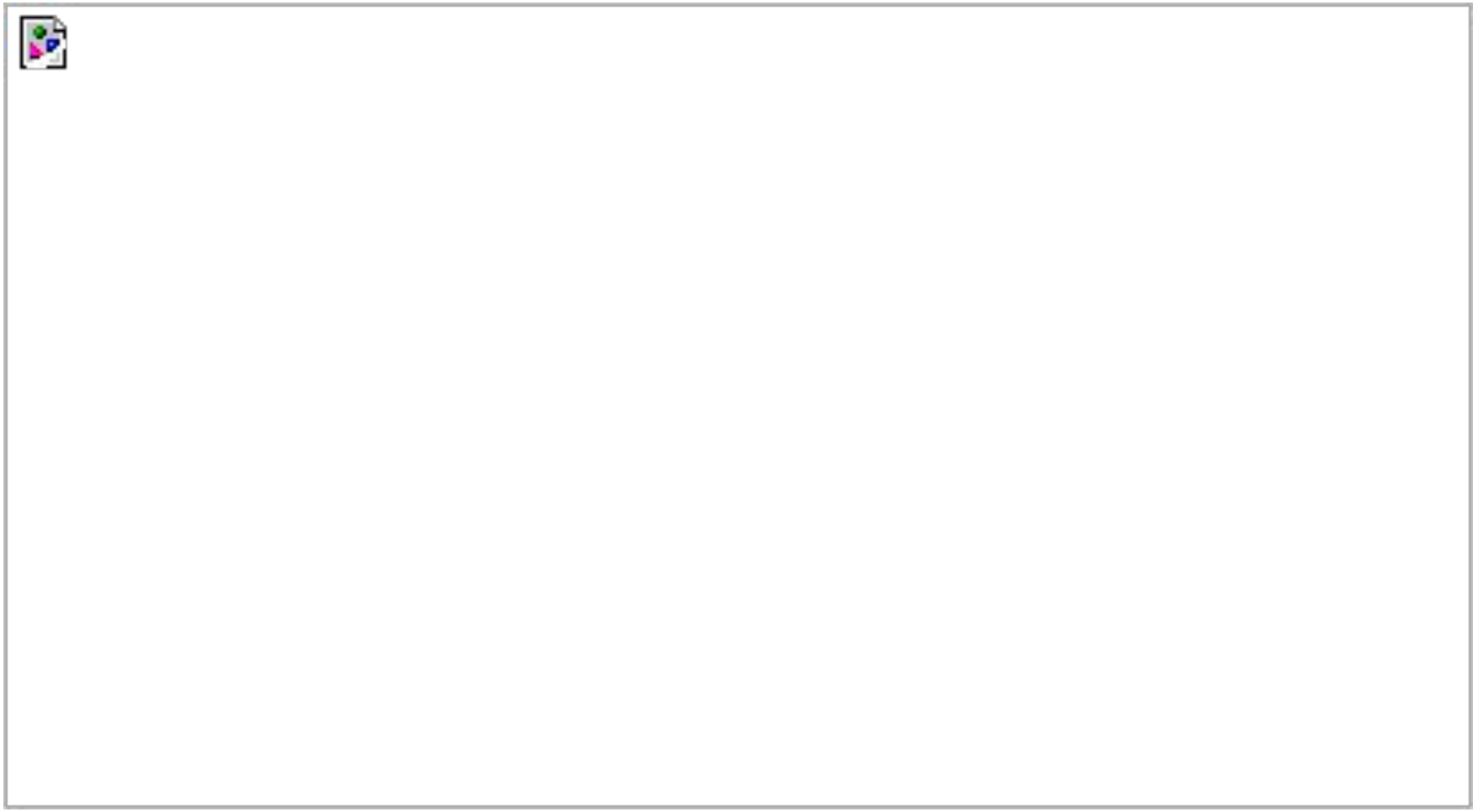


GENERACIÓN 2



- Se solucionan problemas de **backpressure**
- io.reactivex.Subscriber/Producer

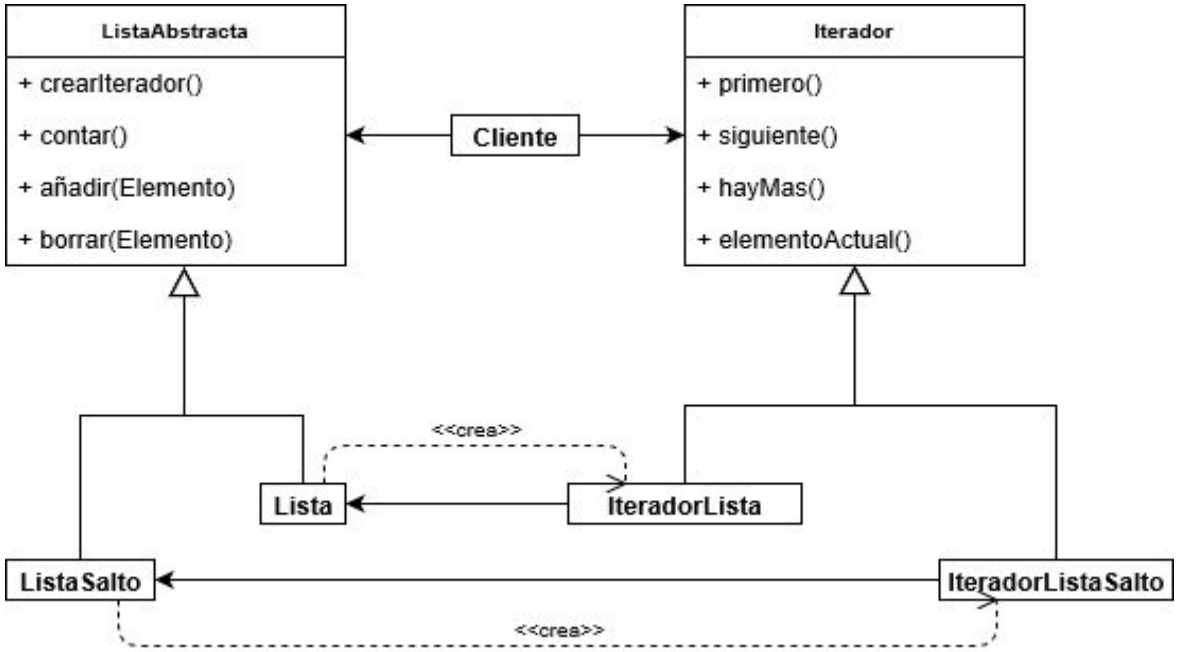
GENERACIÓN 2



GENERACIÓN 3 Y 4

- **Se elimino la incompatibilidad entre multiples librerías con la especificación Reactive Stream (JDK 9)**
- **Se fijaron clases principales Publisher y Subscriber (implementadas en RxJava 2.x, ProjectReactor y AkkaStreams)**
- **Soluciona el backpressure así:**
  - Subscribers indican el número de datos que quieren recibir o procesar
  - Publisher aplica operaciones para evitar saturar a los subscribers: buffer, descarte de datos, etc.

REACTIVE STREAM



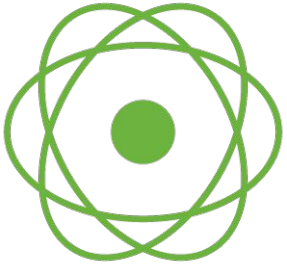
EVENTO	ITERABLE (PUSH)	REACTIVE (PULL)
Obtener dato	next();	onNext(Object data);
Error	Throws Exception	onError(Exception);
fin	!hasNext()	onComplete()

CARACTERISTICAS DE UNA LIBRERÍA REACTIVA

- Control de la ejecución
- De fácil administración de subprocesos
- Combinable
- Reducción al mínimo de efectos secundarios
- Cumplan con las interfaces estándar (Observable, Observer, Subscriber, Publisher, Producer)

LIBRERÍAS Y FRAMEWORKS

- **RxJava 2.x**
  - SpringMVC
  - SpringCloud
  - Netflix OSS
- **ProjectReactor**
  - Forma parte de Pivotal
  - Spring 5 WebFlux



- **RxScala**
- **RxJS**

CONCLUSIÓN

CONTACTOS

Rosanele Gopa

@RosaneleGopa

rosanele7@gmail.com