S4N

SEATTLE / SÃO PAULO / PANAMÁ
BOGOTÁ / **MEDELLÍN**

S4N

Colombia - 09 de Diciembre de 2020

Acerca de mí

S4N

Roger Sepúlveda

BACKEND DEV EN S4N

4+ años en s4n

- Programación funcional
- Seguros
- Retail
- Publicidad

Objetivo

Implementar una pequeña aplicación en Scala con un estilo funcional.

Agenda

01

¿Statectomy?

02

Conceptos básicos

03

Implementación
de una aplicación ETL

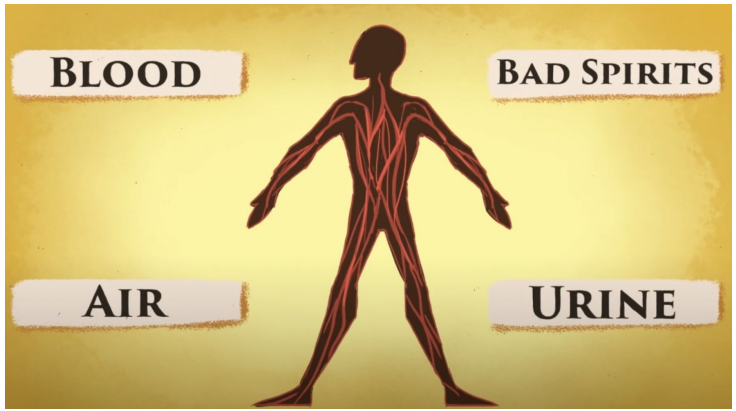
04

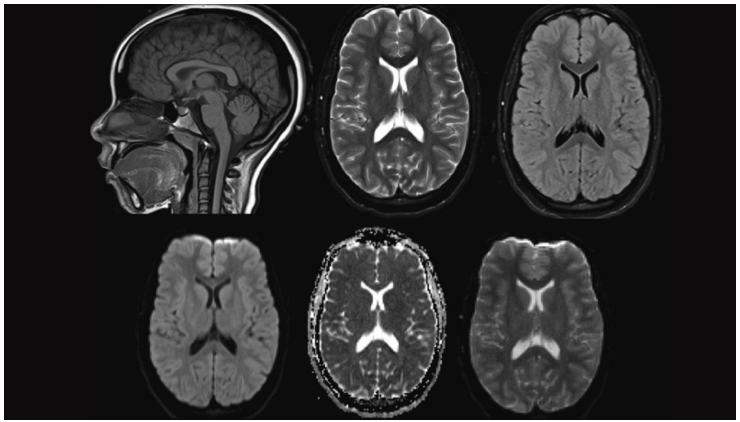
Demo

01

¿Statectomy?

¿Statectomy?





¿Statectomy?

S4N

¿Statectomy?

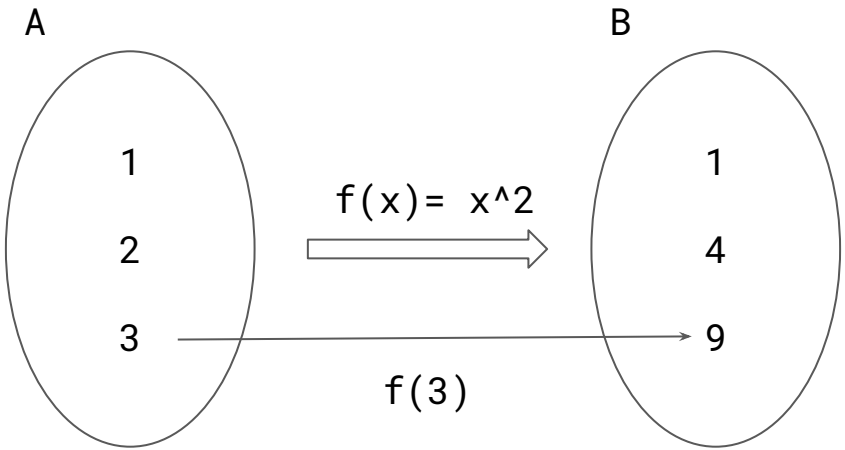
Pregunta:

¿Qué herramientas ofrece Scala para un manejo de estado más limpio?

02

Conceptos básicos

Funciones




```
f: A => B
```

Funciones

```
val f: Int => Int = x => x * x

var b = 1
val g: Int => Int = x => x * b

// def f(x: Int): Int = x * x
// def g(x: Int): Int = x * b
```

Inmutabilidad

```
val a = java.time.ZonedDateTime.now()
var b = java.util.Date()
```

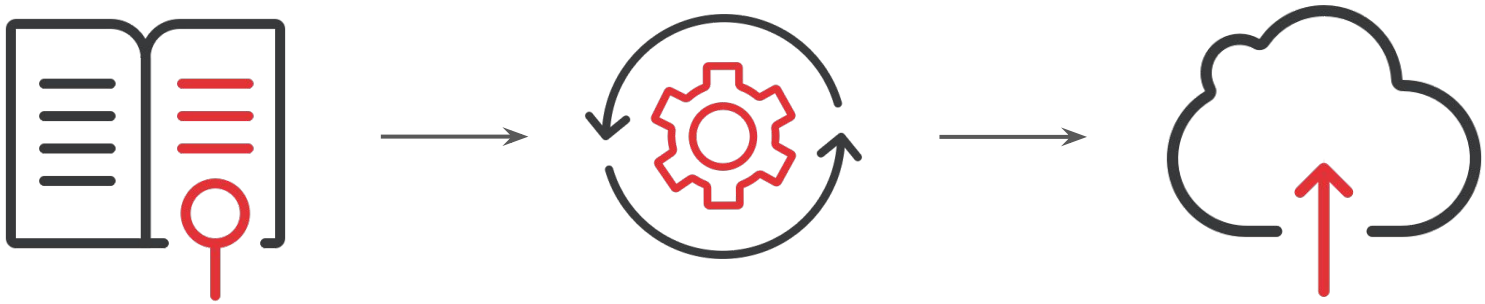
Inmutabilidad

```
class MyObject[T](val x: T) {
 def extractInfo(): String = x.toString
}

val a = java.time.ZonedDateTime.now()
var b = java.util.Date()

val c = new MyObject(a)
val d = new MyObject(b)

interval(1 second)
 .subscribe{ _ => Future {
   println(c.extractInfo())
   println(d.extractInfo())
   ...
```

03

Implementación de una aplicación ETL

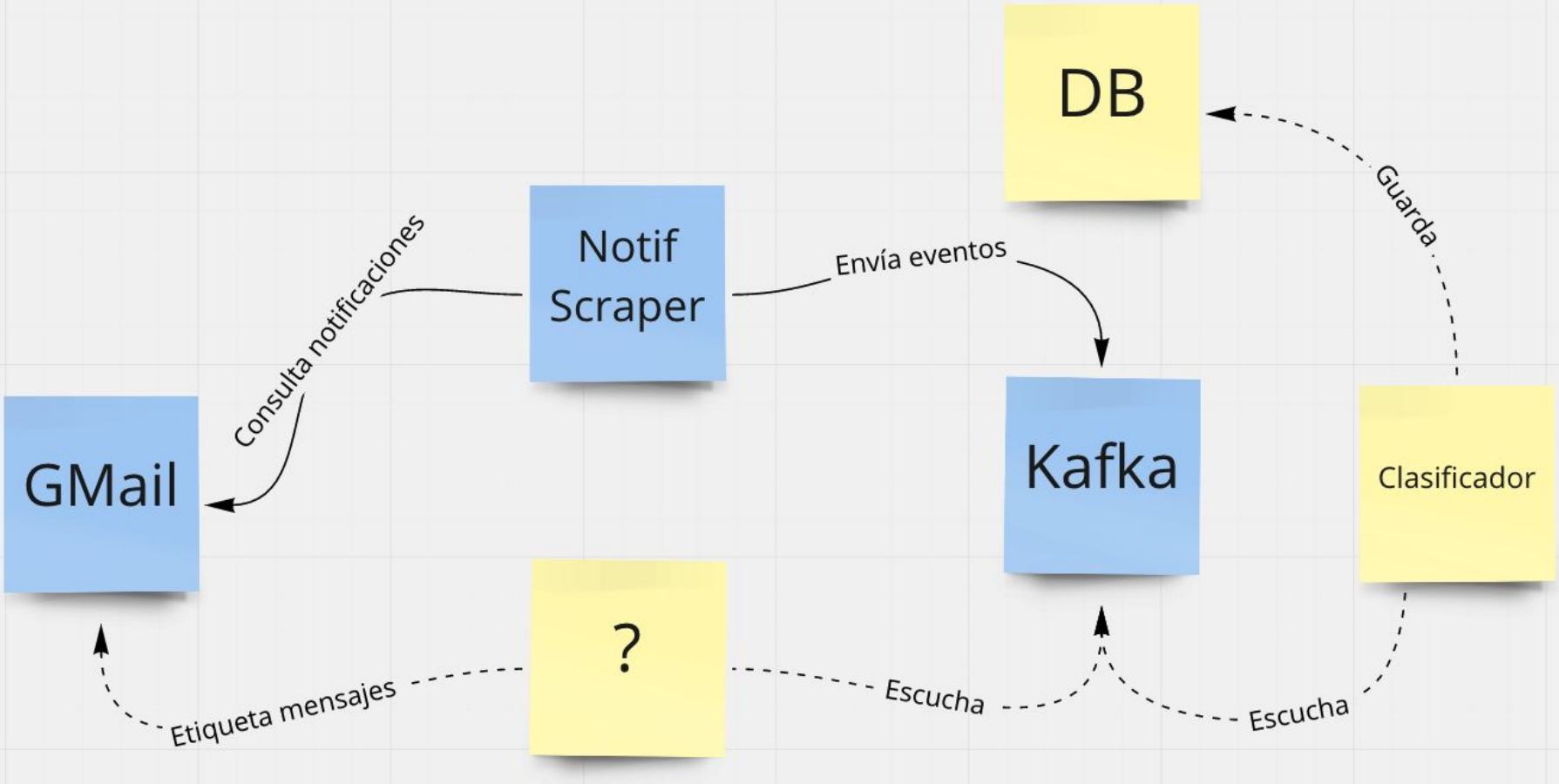
ETL



Extract **Transform** **Load**

ETL


Notificación
Transaccional
Banco le informa Transferencia por $60,000
desde cta *1111 a cta 11111111111. 04/12/2020
08:49. Inquietudes al 0345109095/018000931987.
Esta es una notificación automática, por favor no responda este mensaje


DB
Guarda
Notif Scraper
Consulta notificaciones
Envía eventos
GMail
Kafka
Clasificador
?
Etiqueta mensajes
Escucha
Escucha

Notification scraper

```
new GmailClient(GmailClient.gmailService)   : GmailClient
  .transacciones()                          : Observable[modelos.Notificacion]
  .mapEval(Parser.transform)                : Observable[modelos.Mensaje]
  .mapEval(KafkaClient.load)                : Observable[Option[RecordMetadata]]
```



S4N

Dominio

```scala
package object modelos {

  sealed trait TipoTransaccion
  case object Credito extends TipoTransaccion
  case object Debito extends TipoTransaccion

  case class Transaccion(
    valor: BigDecimal,
    tipo: TipoTransaccion,
    producto: Option[String],
    descripcion: String,
    fecha: Option[ZonedDateTime]
  )

  sealed trait Mensaje { val id: String }
  case class MensajeConTransaccion(id: String, transaccion: Transaccion) extends Mensaje
  case class OtroMensaje(id: String) extends Mensaje

  case class Notificacion(id: String, contenido: String)

}
```

03.1

Extract

Monix

https://monix.io/

Monix is a high-performance Scala / Scala.js library for composing asynchronous, event-based programs.

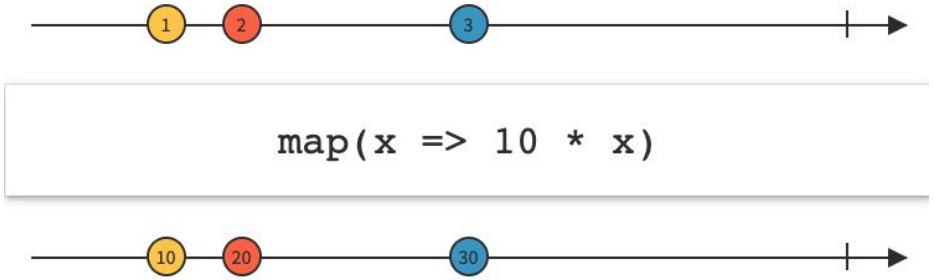
S4N

Observable

https://monix.io/docs/current/reactive/observable.html

The `Observable` is a data type for modeling and processing asynchronous and reactive streaming of events with non-blocking back-pressure.

Observable



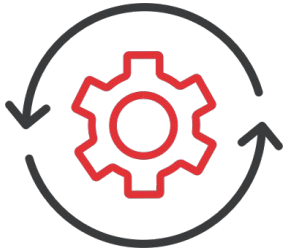

```
import monix.reactive.Observable

Observable.range( from = 1,  until = 3)
  .map(x ⇒ x * 10)
```

Notification scraper

```
def transacciones(): Observable[Notificacion] = {
  val messagePages: Observable[ListMessagesResponse] =
    Observable.fromTask(fetchMessages()).flatMap { response ⇒
      Observable.create(OverflowStrategy.Unbounded) { sub ⇒
        producerLoop(sub, response).runToFuture(sub.scheduler)
      }
    }

  messagePages
    .flatMap(response ⇒ Observable.fromIterable(response.getMessages.asScala))
    .flatMap(m ⇒ Observable.fromTask(fetchFullMessage(m)))
    .mapEval(bodyToNotif)
}
```

03.2

Transform



Notification scraper

```
new GmailClient(GmailClient.gmailService)
  .transacciones()
  .mapEval(Parser.transform)
  .mapEval(KafkaClient.load)
```

Transform


Notificación
Transaccional
Banco le informa Transferencia por $60,000
desde cta *1111 a cta 11111111111. 04/12/2020
08:49. Inquietudes al 0345109095/018000931987.
Esta es una notificación automática, por favor no responda este mensaje

Transform

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
...
<tr>
                                    <td align="center" style="font-family: Arial,
Helvetica, sans-serif; font-size: 14px; line-height: 17px; color:
#000000;font-weight: bold;"><strong>Banco le informa Compra por
$19.700,00 en PEPITO PEREZ 15:43. 30/11/2020 T.Cred *1111.
Inquietudes al 0345109095/018000931987.</strong></td>
                                </tr>
```

S4N

Fast parse

https://www.lihaoyi.com/fastparse/

FastParse is a Scala library for parsing strings and bytes into structured data. This lets you easily write a parser for any arbitrary textual data formats

```
import fastparse._, NoWhitespace._

def number[_: P]: P[Int] = P(CharIn("0-9").rep( min = 1).!.map(_.toInt))
```

Transform

```
// Banco le informa Retiro por $50.000,00 en PARQ. Hora 20:42 21/01/2020 T.Deb *1111
def retiro[_ : P]: P[Transaccion] = P(
    TiposTransaccion.retiro ~ valorTransaccion ~ "en" ~ palabrasAlfanumericas ~ horaFecha ~ producto
  )
  .map(x ⇒ Transaccion(
    tipo = Credito,
    descripcion = x._2.replaceAll( regex = "\\. Hora$",  replacement = ""),
    valor = x._1,
    fecha = Option(x._3),
    producto = Option(x._4)
  ) )
```

03.3

Load

Load

Kafka monix

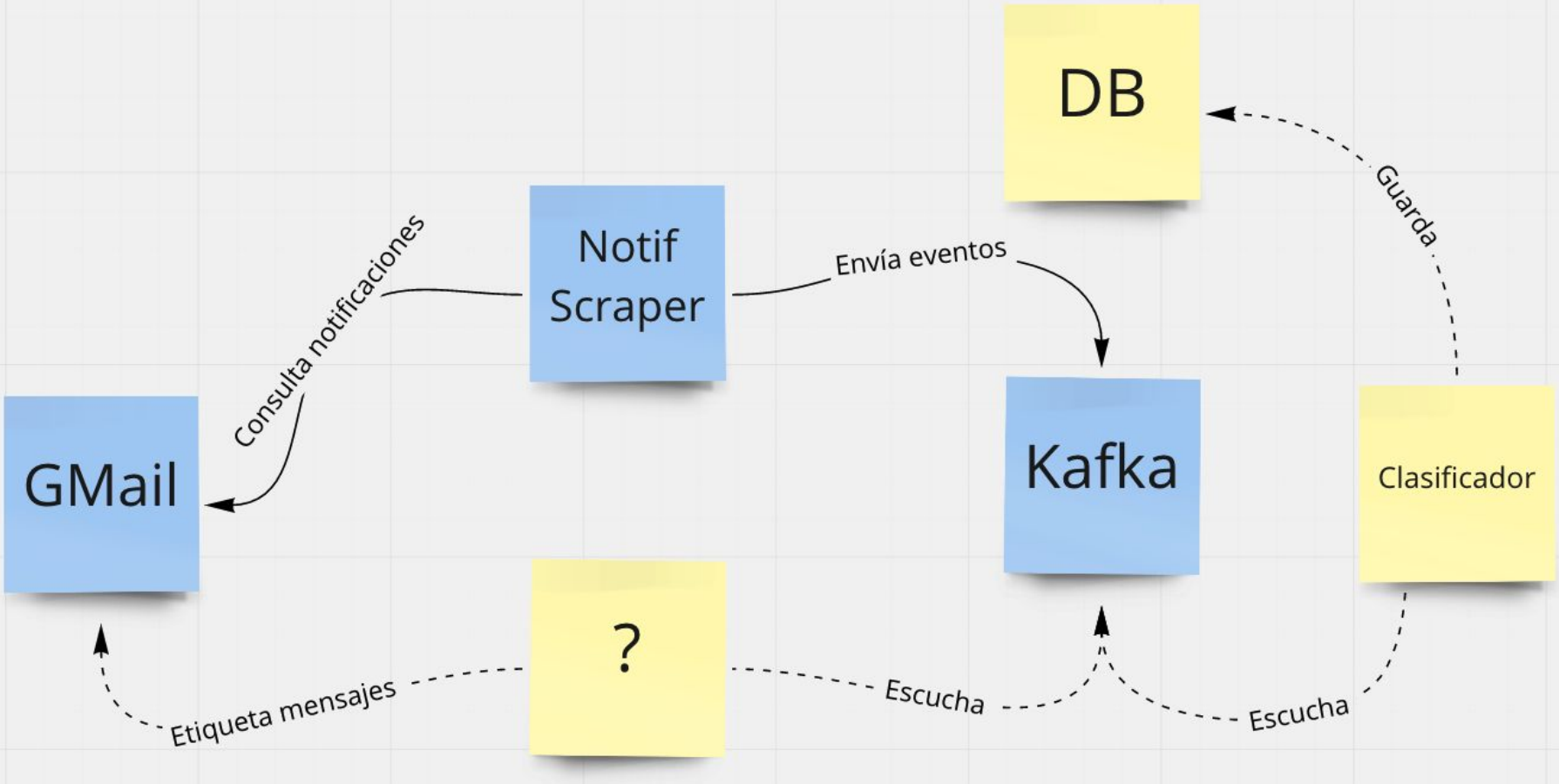
https://github.com/monix/monix-kafka

```
def load(mensaje: Mensaje): Task[Option[RecordMetadata]] =
  producer.send(
    topic = "notificaciones",
    mensaje.id,
    mensaje match {
      case MensajeConTransaccion(id, transaccion) ⇒
        s"${id}," +
          s"${transaccion.tipo}," +
          s"${transaccion.valor.toString().limited()}," +
          s"${transaccion.descripcion.redacted}," +
          s"${transaccion.producto.getOrElse("").redacted}," +
          s"${transaccion.fecha.getOrElse("")}"
      case OtroMensaje(id) ⇒ s"${id},otro tipo de mensaje,,,,"
    }
  )
```

04

Demo

Resumen


DB
Guarda
Notif Scraper
Consulta notificaciones
Envía eventos
GMail
Kafka
Clasificador
?
Etiqueta mensajes
Escucha
Escucha

Preguntas

¿Qué aprendimos en esta charla?

¡Gracias!