

+ Introducción a Elixir.

Santiago Posada

SofkaU

#ElDesafíoEsContigo



Imperativo:
Como quiero que se
haga.

Declarativo:
Que quiero que se
haga.





Imperativo

```
List<Integer> numeros = new ArrayList<>();
numeros.add(1);
numeros.add(2);
numeros.add(3);
numeros.add(4);
numeros.add(5);

// Implementación Imperativa
int sumaParesImperativo = 0;

// Recorremos la lista de números
for (int num : numeros) {
    // Verificamos si el número es par
    if (num % 2 == 0) {
        // Sumamos el número par a la suma total
        sumaPares += num;
    }
}

System.out.println("La suma de los números pares es: " + sumaParesImperativo);
```


Declarativo

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

```
SELECT * FROM Customers  
WHERE Country='Mexico';
```

Paradigma Funcional

Es llevar a cabo toda la transformación y manipulación de la información de nuestros programas a través de funciones pero ...





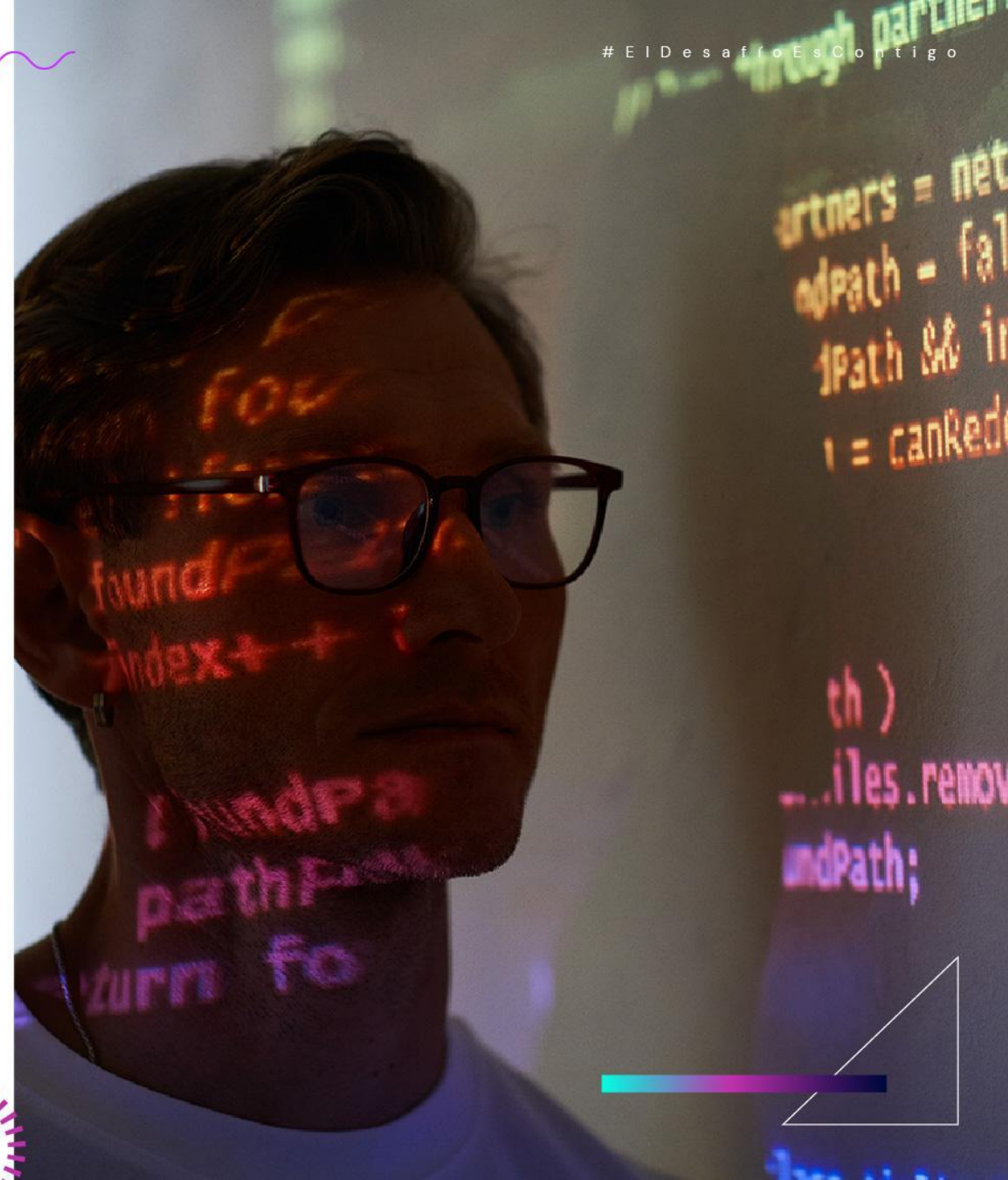
Inmutabilidad:

Un objeto inmutable es aquel que no puede ser modificado una vez haya sido creado.



Funciones puras:

Las funciones puras, no son más que funciones, las cuales, dando el mismo input, siempre retornan el mismo output, además de no tener efectos secundarios.





Introducción a Elixir:

Elixir es un lenguaje de programación funcional y concurrente que se ejecuta sobre la máquina virtual de Erlang (BEAM). Fue creado por José Valim, un conocido desarrollador de Ruby on Rails, y lanzado por primera vez en 2011.

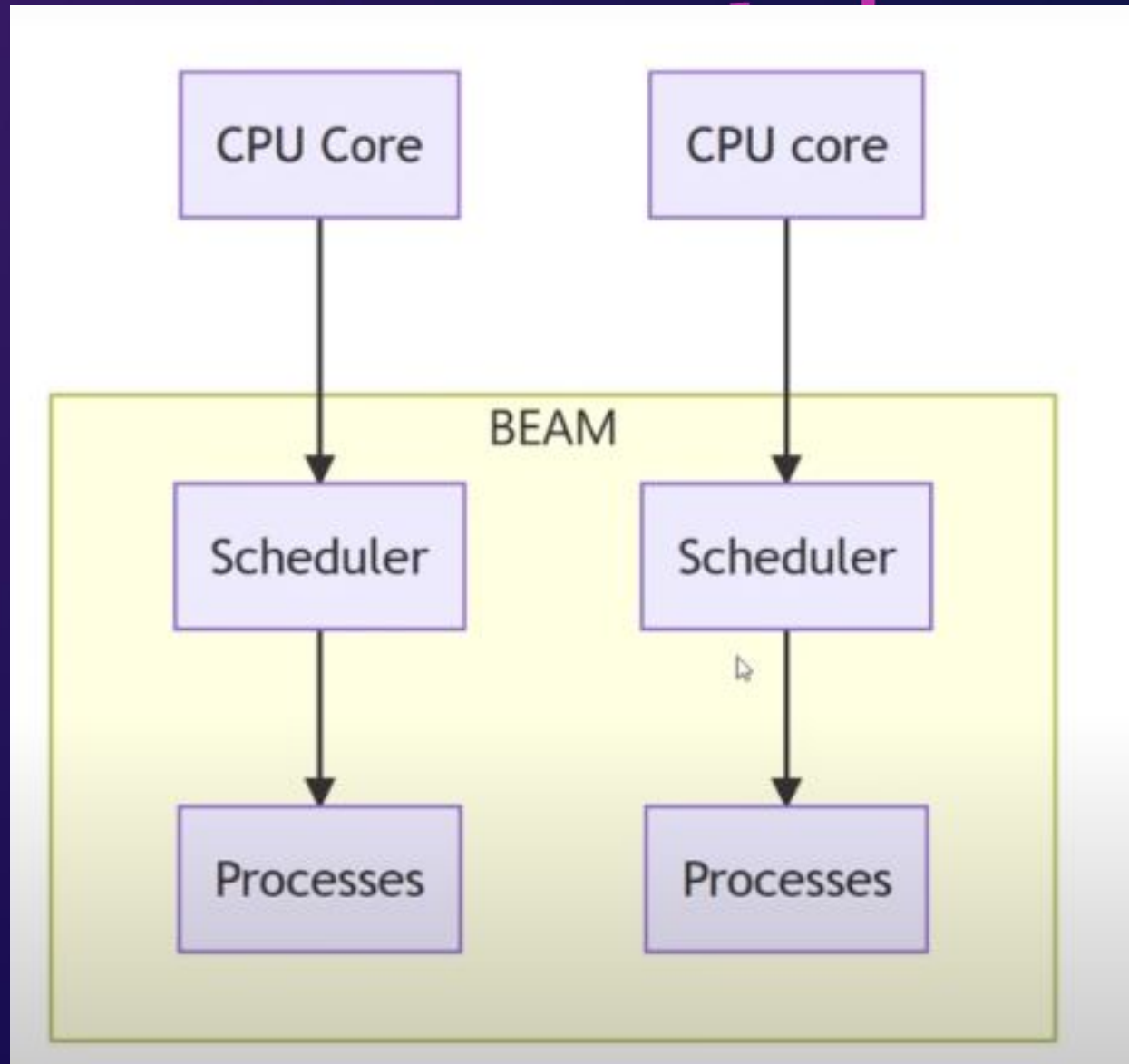
ERLANG: Conociendo al predecesor.

- Creado por Ericsson en 1980
- Su propósito era soportar el flujo de las telecomunicaciones
 - Alta disponibilidad y habilidad para el manejo de múltiples conexiones concurrentes.
- Concurrencia.
- Distribución
- Tolerancia a fallos.
- Alta disponibilidad.
- Hot code swapping.
- Robustez: Modelo de actores y supervisores.

Limitaciones de ERLANG

- Sintaxis y legibilidad.
- + ● Ecosistema y librerías.
- Curva de aprendizaje.





Bogdan and Bjorn's ERLANG abstract machine

¿Por qué se creó Elixir?

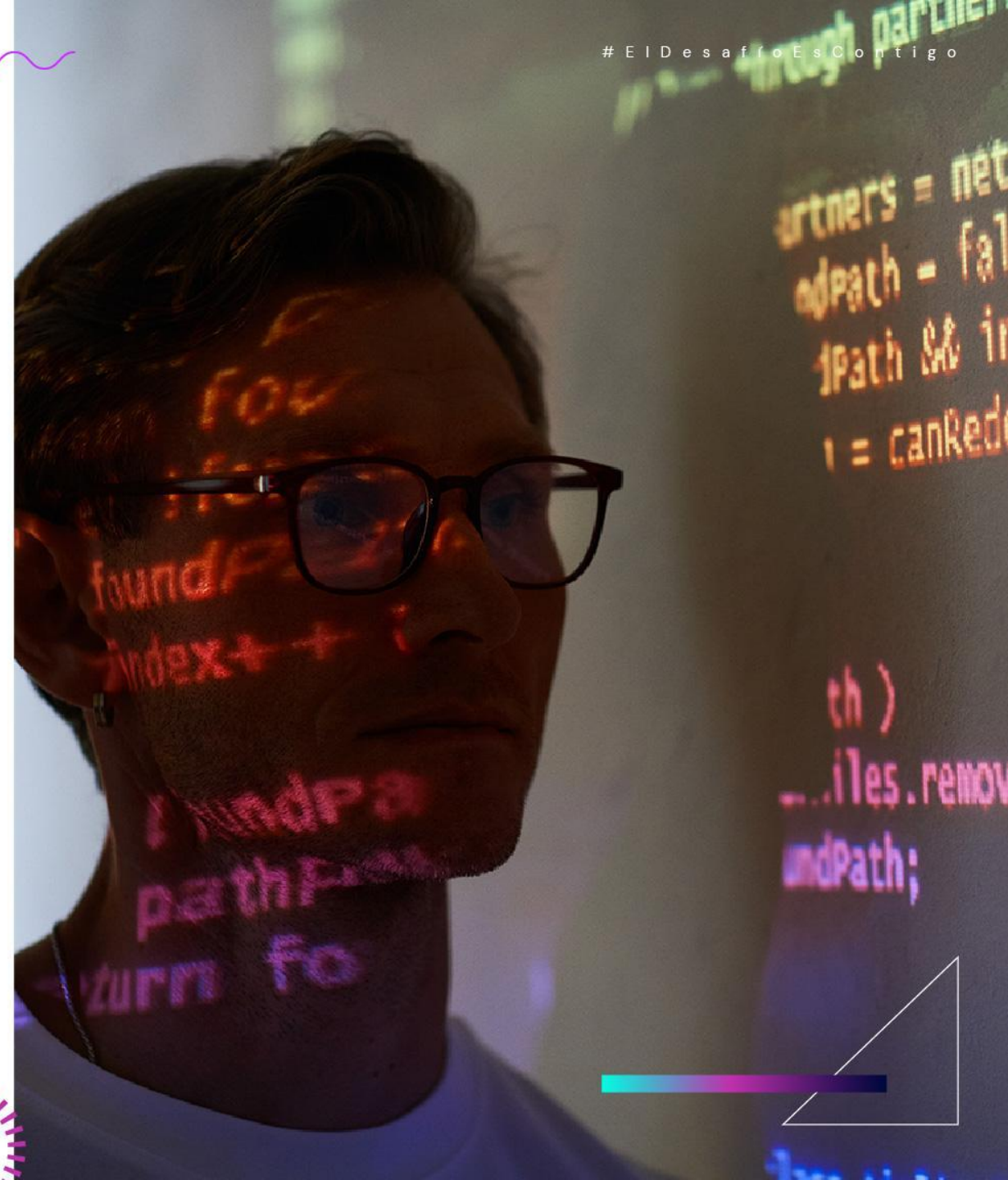
+ José Valim creó Elixir para abordar algunas de las limitaciones que había experimentado con otros lenguajes al desarrollar aplicaciones concurrentes y distribuidas. Quería combinar la simplicidad y elegancia de Ruby con la robustez y escalabilidad de Erlang.

Propósito de Elixir:

- Aplicaciones Web
- Sistemas distribuidos:

Ejemplos:

- Sistema de Mensajería Instantánea (WhatsApp).
- Sistemas de Archivos Distribuidos (Google File System - GFS).
- Blockchain y Criptomonedas (Bitcoin):
- Plataformas de Comercio Electrónico (Amazon).
- Plataformas de Streaming (Netflix).





Características clave de Elixir:

- Tipado dinámico.
- Todo es una expresión.
- Asignación de memoria dinámica.
- Sintaxis Simple y Legible.
- Concurrency (Concurrencia).
- Fault Tolerance (Tolerancia a Fallos).

- ¿Qué es el modelo de Actores?
- ¿Que les pareció interesante de Elixir o Erlang?
- Desde su rol ¿Como ven que elixir puede aportar?

Instalación de Elixir:

- <https://elixir-lang.org/install.html>

Recurso de aprendizaje:

- <https://elixirschool.com/en>

< / >



Calle 12 # 30-80 Medellín

Calle 85 # 11 – 53 Int 6 Of. 301 Bogotá



+57 604 266 4547



info@sofka.com.co



www.sofka.com.co



Síguenos



Sofka Technologies



Sofka_Technologies

