

PROGRAMANDO EN HASKELL



Capítulo 1 - Introducción

¿Qué es un Lenguaje Funcional?

Las opiniones difieren, y es difícil dar una definición precisa, pero de manera general:

- La programación funcional es un estilo de programación en el cual el método básico de computación es la aplicación de funciones a argumentos;
- Un lenguaje funcional es aquel que respalda y motiva el estilo funcional.

Ejemplo

Sumar los enteros del 1 al 10 en Java:

```
int total = 0;
for (int i = 1; i ≤ 10; i++)
    total = total + i;
```

El método de computación es la asignación de variables.

Ejemplo

Sumar los enteros del 1 al 10 en Haskell:

```
sum [1..10]
```

El método de computación es la aplicación de funciones.

Trasfondo Histórico

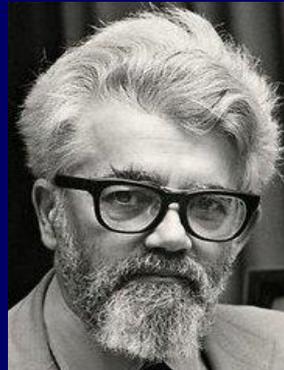
1930s:



Alonzo Church desarrolla el cálculo lambda, una teoría simple de funciones pero poderosa.

Trasfondo Histórico

1950s:



John McCarthy desarrolla Lisp, el primer lenguaje funcional, con algunas influencias del cálculo lambda, pero reteniendo las asignaciones de variable.

Trasfondo Histórico

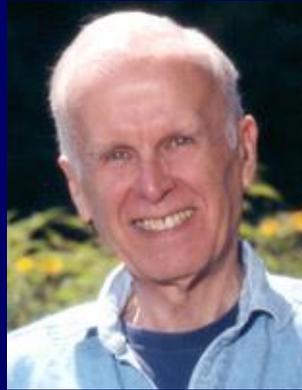
1960s:



Peter Landin desarrolla ISWIM, el primer language funcional *puro*, apegado al cálculo lambda, sin asignaciones.

Trasfondo Histórico

1970s:



John Backus desarrolla FP, un lenguaje funcional que enfatiza las *funciones de alto-orden* y el *razonamiento sobre los programas*.

Trasfondo Histórico

1970s:



Robin Milner y otros desarrollan ML, el primer lenguaje funcional moderno, el cual introdujo la *inferencia de tipos* y los *tipos polimórficos*.

Trasfondo Histórico

1970s - 1980s:



David Turner desarrolla varios lenguajes funcionales de evaluación *lazy*, culminando en el sistema Miranda.

Trasfondo Histórico

1987:



Un comité internacional comienza el desarrollo de Haskell, un lenguaje funcional lazy estándar.

Trasfondo Histórico

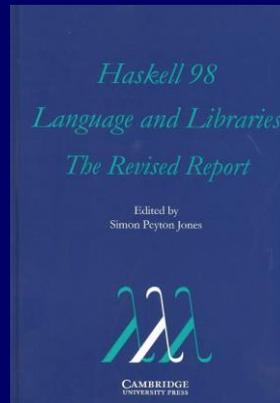
1990s:



Phil Wadler y otros desarrollan las *clases de tipo* y las *mónadas*, dos de las principales innovaciones de Haskell.

Trasfondo Histórico

2003:



El comité publica el Reporte de Haskell, definiendo una versión estable del lenguaje; una versión actualizada se publicó en el 2010.

Trasfondo Histórico

2010-a la fecha:



Distribución estándar, soporte de librerías, nuevas características de lenguaje, herramientas de desarrollo, uso en la industria, influencia sobre otros lenguajes, etc.

Trasfondo Histórico

```
f [] = []
```

```
f (x:xs) = f ys ++ [x] ++ f zs
```

where

```
ys = [a | a ← xs, a ≤ x]
```

```
zs = [b | b ← xs, b > x]
```

?

Fuente.

Profesor Graham Hutton.

<http://www.cs.nott.ac.uk/~pszgmh/>

Traducción al español.

Caleb Josue Ruiz Torres.

<https://www.calebjosue.com>