

# 6 Dimensiones

## Representando un arreglo de 6 dimensiones con clases en Object Oriented Haskell aka Patito++

Un programa tan sencillo como

```
1 class Cuenta{
2     [+] Integer [10][10] numeros;
3 }
4
5 class Humano{
6     [+] Cuenta [10][10] cuentas;
7 }
8
9 class Empresa{
10     [+] Humano [10][10] humanos;
11 }
12
13 class Google : Empresa{
14
15 }
16
17 main{
18     Humano humano;
19     Cuenta cuenta;
20     Integer [10][10] numeros = [[1,2,3,4,5,6,7,8,9,10],
21     [+][+][+][+][+][+][+][+] [3,4,5,6,7,8,9,100]];
22     cuenta.numeros = numeros;
23     Cuenta [10][10] matrizCuentas = [[cuenta,cuenta,cuenta]];
24
25     humano.cuentas = matrizCuentas;
26     Humano [10][10] humanos = [[humano,humano,humano,humano],
27     [+][+][+][+][+][+][+][+] [humano,humano]];
28     Empresa e,e2;
29     e.humanos = humanos;
30     Cuenta [10][10] cuentas;
31     cuentas = humano.cuentas;
32     Google g,g2;
33     g.humanos = humanos;
34     g2 = g;
35     display(g2.humanos);
36 }
```

Requiere una vasta cantidad de memoria.

## ¿Por qué?

Analizando detalladamente el código, podemos concluir lo siguiente:

1 empresa tiene 10 x 10 **humanos**

1 humano tiene 10 x 10 **cuentas**

1 cuenta tiene 10 x 10 **numeros**

El componente encargado de realizar las asignaciones en memoria virtual arroja que para este programa, se reservaron 5,030,219 millones de enteros.

Aproximadamente esto equivale a la siguiente ecuación:

$$1 \text{ Ints: } = 10^6 * 5 + (10^4 * 3) + (10^2) \approx 5,030,219$$

$10^6$  por 5 porque hay 4 **empresas** cada una con matrices de **humanos** y una matriz de **humanos**

$10^4$  por 4 porque hay 2 matrices de **cuentas** y un **humano** con 1 matriz de **cuentas**

$10^2$  por 2 porque hay 2 matrices de **enteros**, matrizDeCuentas y la matriz **cuentas** que contiene las **cuentas** de **humano**

## Ejecución

Utilizando una máquina virtual hecha en Haskell, la ejecución toma **45 segundos**. Esto se debe a que en asignaciones de matriz a matriz, se copian los elementos de una en la otra.

Se debe de imprimir 18 veces el 100 y 18 veces el 10.

El resultado es el siguiente:

```
[VM] Execution in process...
```

```
1
2
3
4
5
6
7
8
9
10
3
4
5
6
7
8
9
100
1
2
3
```

4	
5	
6	
7	
8	
9	
10	
3	
4	
5	
6	
7	
8	
9	
100	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
3	
4	
5	
6	
7	
8	
9	
100	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
3	
4	
5	
6	
7	
8	
9	
100	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

3  
4  
5  
6  
7  
8  
9  
100  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
3  
4  
5  
6  
7  
8  
9  
100  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
3  
4  
5  
6  
7  
8  
9  
100  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
3  
4  
5  
6  
7  
8  
9

100

1

2

3

4

5

6

7

8

9

10

3

4

5

6

7

8

9

100

1

2

3

4

5

6

7

8

9

10

3

4

5

6

7

8

9

100

1

2

3

4

5

6

7

8

9

10

3

4

5

6

7

8

9

100

1

2

3

4

5

6

7  
8  
9  
10  
3  
4  
5  
6  
7  
8  
9  
100  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
3  
4  
5  
6  
7  
8  
9  
100  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
3  
4  
5  
6  
7  
8  
9  
100  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
3  
4  
5

6  
7  
8  
9  
100  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
3  
4  
5  
6  
7  
8  
9  
100  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
3  
4  
5  
6  
7  
8  
9  
100  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
3  
4  
5  
6  
7  
8  
9  
100  
1  
2  
3  
4  
5  
6  
7  
8  
9  
100  
[VM] Finished in 45.488617 sec

