



CURRIFICACIÓN



BASES DEL LAMBDA-CÁLCULO - II

Currificación

- Es una transformación de **funciones de más de un parámetro**.

- Supongamos la función del ejemplo anterior $(x,y) \rightarrow x^y$

$(x,y) \rightarrow x^y$ “currificada” será ... $x \rightarrow y \rightarrow x^y$

Donde ahora **x** ya no se evalúa a un escalar como en el caso $x \rightarrow x+2$ que se evaluaba a **x+2**, sino que **se evalúa a otra función anónima** $y \rightarrow x^y$

Además... $(x \rightarrow y \rightarrow x^y)(2)(10) = (y \rightarrow 2^y)(10) = 2^{10} = 1024$

- Decimos de las funciones de n parámetros en ese “formato” de función con un solo parámetro que devuelve una función con n -1 parámetros en ese mismo “formato” está en forma “currificada” (currified), por Haskell Curry (de quien toma nombre el lenguaje Haskell).
- Currifica $(x, y, z) \rightarrow x^2 + y^2 + z^2$



BASES DEL LAMBDA-CÁLCULO – III

Currificación Con Expresiones Lambda En C#

¿Cómo Llevamos $(x \rightarrow Y \rightarrow X^y)(2)(10) = (y \rightarrow 2^y)(10) = 2^{10} = 1024$ A C#?

```
Func<double, Func<double, double>> potencia =  
x => y => Math.Pow(x, y);  
Console.WriteLine(potencia(2)(10)); // Mostrará 1024
```

Entonces...¿ $(X, Y, Z) \rightarrow X^2 + Y^2 + Z^2$ “currificada” En C#?

```
Func<double, Func<double, Func<double, double>>> f =  
x => y => z => x * x + y * y + z * z;  
Console.WriteLine(f(2)(3)(4)); // Mostrará 29
```



CURRIFICACIÓN EN C# - EJEMPLO1

- Supongamos nuestra expresión lambda $(x,y) \rightarrow x^y$
- Si ahora la “currificamos” como $y \rightarrow x \rightarrow x^y$ podremos hacer...

```
static readonly Func<double, Func<double, double>> Potencia =  
    y => x => Math.Pow(x, y);  
  
static readonly Func<double, double> Cuadrado = Potencia(2);  
static readonly Func<double, double> Cubo = Potencia(3);  
  
static void Main()  
{  
    Console.WriteLine(Cuadrado(10));           // calcularía 102  
    Console.WriteLine(Cubo(10));               // calcularía 103  
    Console.WriteLine(Potencia(4)(10));        // calcularía 104  
}
```

- Vale, le has puesto un nombre más específico a `Potencia(2)(10)` llamando le `Cuadrado(10)` pero.... yo ya podía hacer `Math.Pow(2, 10)`

¿Algún Ejemplo Más?



CURRIFICACIÓN EN C# - EJEMPLO2 - I

- Supongamos la siguiente definición sin currificar...

```
Action<int, int, ConsoleColor, string>  
escribeEnPosConColor = (fila, columna, color, texto) => {  
    ConsoleColor backupColor = Console.ForegroundColor;  
    Console.ForegroundColor = color;  
    Console.SetCursorPosition(columna, fila);  
    Console.Write(texto);  
    Console.ForegroundColor = backupColor;  
};
```

- Supongamos la definición currificada...

```
Func<int, Func<int, Func<ConsoleColor, Action<string>>>>  
escribeEnPosConColorCurrificada = fila => columna => color => texto => {  
    ConsoleColor backupColor = Console.ForegroundColor;  
    Console.ForegroundColor = color;  
    Console.SetCursorPosition(columna, fila);  
    Console.Write(texto);  
    Console.ForegroundColor = backupColor;  
};
```

- Hemos ido aislando de más abtracto como escribir un texto a escribirlo con un color y por último a escribirlo con un color y en una posición.



CURRIFICACIÓN EN C# - EJEMPLO2 – II

- De esta forma podremos definir delegados más específicos tales como...

```
Action<string> escribeCentradoRojo = texto =>  
    escribeEnPosConColorCurrificada(12)(40)(ConsoleColor.Red)(texto);  
escribeCentradoRojo("Hola");
```

- Al final la llamada `escribeCentradoRojo("Hola")` escribirá será la máxima especificación de `escribeEnPosConColor`.