




PROGRAMACIÓN FUNCIONAL

Conceptos Preliminares



A vertical bar on the left side of the page, composed of two parallel lines: a thin dark gray line on the left and a slightly thicker light gray line on the right.

“Cuando sepas reconocer la cuatrifolia en todos sus estados, raíz, hoja y flor, por la vista y el olfato, y la semilla, podrás aprender el verdadero nombre de la planta, ya que entonces conocerás su esencia, que es más que su utilidad.”

Un mago de Terramar
Úrsula K. Le Guin

Three light gray triangles pointing downwards, arranged vertically on the right side of the page. Each triangle has a thin dark gray line along its left edge.

Conceptos Preliminares

- ◆ Conceptos de programación y programa
- ◆ Propiedades de los programas
- ◆ Motivación para Programación Funcional

Definiciones

◆ Programación:

- ◆ es una tarea que comprende los siguientes puntos
 - ◆ analizar problemas
 - ◆ diseñar soluciones para ellos que puedan ejecutarse
 - ◆ codificar dichas soluciones
 - ◆ verificar propiedades deseadas de las mismas

◆ Programa:

- ◆ **descripción** de una solución a un problema, que puede **ejecutarse** de alguna manera para obtener una instancia particular de dicha solución.

Programación

- ◆ ¿Cuáles son los dos aspectos fundamentales?
 - ◆ transformación de información
 - ◆ interacción con el medio
- ◆ Ejemplos:
 - ◆ calcular el promedio de notas de examen
 - ◆ cargar datos de un paciente en su historia clínica
- ◆ Este curso se concentrará en el primero de estos aspectos.

Preguntas

◆ ¿Cuáles propiedades de un programa son importantes?

- ◆ eficiencia
- ◆ corrección
- ◆ claridad
- ◆ modificabilidad
- ◆ terminación
- ◆ equivalencia
- ◆ generalidad
- ◆ simplicidad

◆ ¿En cuáles debería focalizarse un programador?
¿Por qué?

Propiedades

- ◆ Si podemos probar fácilmente equivalencia de programas, podemos
 - ◆ reemplazar un programa por otro más eficiente
 - ◆ usar un programa correcto para ver que otro lo es
 - ◆ ver que no alteramos el significado al modificarlo
- ◆ ¿Qué necesitamos para poder probar equivalencia de programas con sencillez?

Preguntas

- ◆ ¿Cómo saber cuándo dos programas son iguales?
- ◆ Ejemplo:
 - ◆ ¿Son equivalentes ' $f(3)+f(3)$ ' y ' $2*f(3)$ '?
 - ◆ ¿Siempre?
 - ◆ ¿Sería deseable que siempre lo fueran?
¿Por qué?

Ejemplo

- ◆ ¿Qué imprime este programa en Javascript?

// Test.js (gentileza de Martín Goffan, 2018)

let x = 0;

**function f(y) { x = x + 1;
 return x + y; }**

console.log(f(3) + f(3));

- ◆ ¿Y con '2*f(3)' en lugar de 'f(3)+f(3)'?

Ejemplo

- ◆ ¿Qué imprime este programa en Java?

```
public class Example {  
    static int x = 0; // (by Juan Li Puma 2018)  
    static int f(int y) { return ++x + y; }  
    public static void main(String[] args)  
    { System.out.printf(f(3) + f(3)); }  
}
```

- ◆ ¿Y con '2*f(3)' en lugar de 'f(3)+f(3)'?

Ejemplo (original)

- ◆ ¿Qué imprime este programa en Pascal?

```
Program test;
```

```
var x : integer;
```

```
function f(y:integer):integer;
```

```
begin x := x+1; f :=x+y; end;
```

```
begin x := 0; writeln(f(3)+f(3)); end;
```

- ◆ ¿Y con '**2*f(3)**' en lugar de '**f(3)+f(3)**'?

Preguntas

- ◆ ¿Qué haría falta para que este fenómeno no ocurriese?
 - ◆ En matemáticas no sucede.
 - ◆ Precisamos un modelo de computación con características similares a las de matemáticas...

Conclusiones

- ◆ Revisamos las nociones de programación y programa
- ◆ Revalorizamos la importancia de demostrar propiedades de programas
- ◆ Encontramos problemas en la forma tradicional de los lenguajes de programación
- ◆ Vimos que se hace necesaria una solución a dichos problemas