

# Programación como disciplina básica

Sylvia da Rosa

Instituto de Computación  
Universidad de la República  
Montevideo - Uruguay

e-mail: [darosa@fing.edu.uy](mailto:darosa@fing.edu.uy)

# Estructura de la charla

---

- La informática en la enseñanza media
  - Los dos perfiles
  - Práctica docente
  - Investigación
- Programas de didáctica de la informática
  - Anterior - Bibliografía
  - Programa actual Basado en What (Else) Should CS educators Know? de Judith Gal-Ezer and David Harel (Communications of the ACM, 1998)

- Tema de interés para Butía: Programación
  - "Qué informática enseñar en el liceo?", Gilles Dowek (Revista EPI, 2005)
  - "El pensamiento computacional", Jeannette Wing (Communications of the ACM, 2006)
  - "Research Agenda for CSE", Christian Holmboe et al (PPIG, 2001)
- Algunos ejemplos

# La informática en la enseñanza media

---

- Los dos perfiles
- Qué es y qué NO es informática (ciencia de la computación)?
- Qué es y qué NO es CSE (didáctica de la informática)?

*Judith Gal and David Harel: What (Else) Should CS Educators Know? (1998)*

It goes without saying that CS educators must be educators, and not only experts in teaching computing. For example, a high school CS teacher must have studied the various educational material required for any high school teacher.

# Los dos perfiles

---

- Por qué existen dos perfiles?
- Cuáles son?

Surge un problema directamente relacionado con la educación, que tiene sus raíces en la necesidad imperiosa de estos tiempos, de **alfabetizar en informática** a la población, de ahí los grandes esfuerzos por integrar los computadores en la educación a todos los niveles y mediante una variedad de maneras. Esto a su vez, ocasiona una **gran confusión entre CC y alfabetización en informática**, por ejemplo, algoritmia con uso de planillas de cálculo, programación con procesadores de texto, etc.

Ejemplos: opiniones muestran que los profesores provienen de distintas asignaturas y buscan regularizar su situación a través del profesorado.

# La práctica docente

---

- Programar en scratch
- Cursos de matemática
- Usando las salas de informática

No existe aún la asignatura informática con el enfoque adecuado.

Resuelve el profesorado los problemas? (opinión 5)

Es necesario diferenciar desde el inicio en la formación de los estudiantes de profesorado, las tres direcciones que muchas veces se confunden y donde sólo una de ellas tiene que ver con CS: **alfabetización en informática, uso de computadores en la educación de otras asignaturas y educar en CS.**

# Investigación

---

Opinones 6, 7, 8

- ANII
- PEDECIBA (papi)

Algunas respuestas

# Programas de Didáctica de la Informática

---

- Programa anterior
  - Objetivos
  - Bibliografía
- Programa actual



# Programa actual

---

*C.Holmboe, L.Mclver, C.George: Research Agenda for Computer Science Education (2001)*

The academic discipline CSE consists in focusing research on the application of principles from educational-related disciplines -*pedagogy, epistemology, curriculum studies and psychology*- to the teaching and learning of the scientific discipline computer science as a school subject.

# Programa actual

---

*Judith Gal and David Harel What (Else) Should CS Educators Know? (1998)*

Beyond the mastery of core CS material, good CS educators should also be familiar with a significant body of material that will expand their perspectives on the field, and consequently, enhance the quality of their teaching.

# Tema de interés para Butiá

---

- Algoritmia
  - La CC es el estudio de lo que es computable y de cómo computarlo. Sus fundamentos esenciales derivan de la matemática y la lógica y se extienden más allá de ellas, hacia la ingeniería por el hecho de trabajar con sistemas que interactúan con el mundo real.
  - El pensamiento computacional es humano no de máquina.
  - Conceptualizar, no programar. Pensar computacionalmente es más que ser capaz de escribir un programa, se requieren diferentes niveles de abstracción.

# Tema de interés para Butiá

---

- Ideas, no artefactos: no son el hardware y el software lo que debemos incorporar a nuestra vida sino los **conceptos computacionales que usamos para abordar y resolver problemas**.
- Es una gran visión para guiar a los educadores en CC, a los investigadores y a los profesionales, de modo de **cambiar la imagen que tiene la sociedad acerca de este campo de estudio**.

# Tema de interés para Butiá

---

- Programación
  - Son las limitaciones de los dispositivos computacionales subyacentes las que obligan al científico a pensar computacionalmente.
  - Concebir y ejecutar un proyecto: escribir un programa es un proyecto.
  - Poner en práctica los conocimientos adquiridos: interacción entre utilizar nociones y comprenderlas.

# Temas de interés para Butiá

---

- Un puente entre el lenguaje y la acción: Un programa perteneciente a un lenguaje, es decir es un texto, y ese texto es ejecutable es decir, es el agente de una acción. El aprendizaje de programación enseña que decir  $x = 1$  o  $x = 2$ , no es de ninguna manera lo mismo.
- Una escuela del rigor: el rigor no es algo impuesto del exterior, sino que es una condición de buena comunicación entre el estudiante y la máquina.

## Algunos ejemplos

---

```
function calculate(a, b, c): return (a+b)*c
```

```
example1 = calculate (1, 2, 3)
```

```
example2 = calculate ([1, 2, 3], [4, 5, 6], 2)
```

```
example3 = calculate ('apples ', 'and oranges, ', 3)
```

## Qué hacemos en didáctica?

---

*Las reglas pueden ser aplicadas siguiendo distintas estrategias, pero por cualquiera de ellas se llega a una misma forma canónica (si la misma existe). Por ejemplo, la expresión  $(2 + 1) * 5$  puede reducirse a la expresión  $3 * 5$  y ésta a 15, o a la expresión  $10 + 5$  y luego a 15.*

Dado el problema algorítmico: sean 3 números naturales  $a$ ,  $b$  y  $c$ , hallar el valor de la expresión  $(a + b) * c$ . Escribir los dos algoritmos que solucionan el problema, correspondientes a cada una de las dos estrategias descritas.



# El pensamiento computacional

---

- Cómo sabemos que un algoritmo es apropiado para resolver un determinado problema algorítmico?
- Qué significa que sea apropiado?
- Entre los algoritmos que resuelven un mismo problema, hay algunos más apropiados que otros?

# Ejemplos

---

## Interacción entre formalización e implementación

### Algoritmo 1

Escriba en un papel la letra a, la letra b y la letra c

A la letra a, sumarle la letra b

Al resultado de sumar la letra a y la letra b, multiplicar

Escriba el resultado del producto en un papel

### Algoritmo 2

Escriba en un papel la letra a, la letra b y la letra c

Multiplique la letra a por la letra c

Ahora multiplique la letra b por la letra c

Al resultado de multiplicar la letra a por la letra c, suma

Escriba el resultado de la suma en un papel

# Ejemplos

---

Python permite pasar del problema algorítmico particular para números naturales al problema general para elementos para los que esté definida la operación.

Ejercicio: plantear el problema para elementos  $a$ ,  $b$ ,  $c$  y dos operaciones  $ope1$  y  $ope2$  definidas para los conjuntos a los que pertenecen los elementos. Usar los ejemplos de listas y strings vistos anteriormente.

# Ejemplos

---

- Qué es una variable?
- El problema de nombrar
- El signo de =
- la instrucción `if ... then ... else`

# Ejemplos

---

Tarea: Supongamos ahora que modificamos el algoritmo principal, de modo que si el valor de  $c$  ingresado es 0, el algoritmo obtendrá el resultado 0 sin hacer cuentas:

```
distrib
```

```
ingrese 3 numeros naturales a, b , c
```

```
si  $c = 0$  entonces resultado = 0
```

```
si no
```

```
  elija una estrategia e1 o e2
```

```
  si e1 es elegida entonces resultado = distrib1 (a, b, c)
```

```
  si no resultado = distrib2 (a, b, c)
```

# La programación como disciplina básica

---

Discuta el significado que Ud. atribuye a cada uno de los símbolos de igualdad que aparecen en esta última versión del algoritmo distrib. Formalice cada uno de los significados de  $=$ , escribiendo cuales son las acciones involucradas en la ejecución de una sentencia de la forma  $A = B$ , donde A y B son variables o expresiones cualesquiera.

# La programación como disciplina básica

---

Se ha dicho que una persona no comprende algo realmente hasta que lo enseña a otro. En realidad, una persona no comprende algo profundamente hasta que lo puede enseñar a un computador, es decir, expresarlo como un algoritmo.

*Knuth 1974 en American Mathematical Monthly.*