

# Paradigmas de la Programación

FaMAF 2015

Laura Alonso Alemany

Ezequiel Orbe

# qué veremos en esta materia

- aprender conceptos fundamentales (fundacionales) de los lenguajes de programación
- aprender a identificar estos conceptos en lenguajes de programación concretos
- evaluar el impacto de las decisiones de diseño en lenguajes de programación, para poder elegir el lenguaje más adecuado a una situación

# objetivos de la materia

al terminar ustedes podrán...

- pensar cómo traducir programas de un lenguaje a otro
- entender qué caracteriza a un lenguaje
- conocer paquetes de características útiles
- decidir qué lenguaje es más adecuado usar para un problema y contexto determinado
- leer, preguntar y contestar en stackoverflow

# traducir programas

- C, C++, Java:

```
int fact (int n) { return (n == 0) ? 1 : n * fact (n-1); }
```

- Scheme:

```
(define fact  
  (lambda (n) (if (= n 0) 1 (* n (fact (- n 1))))))
```

- ML:

```
fun fact n = if n=0 then 1 else n*fact(n-1);
```

- Haskell:

```
– fact :: Integer->Integer  
– fact 0 = 1  
– fact n = n*fact(n-1)
```

deberían poder leer opiniones como...

- <http://s3.crashworks.org.s3-website-us-east-1.amazonaws.com/if-programming-languages-were-vehicles/>
- <http://www.whoishostingthis.com/blog/2014/09/04/learn-to-code/>

practicalidades

# practicalidades

- teóricos martes y jueves de 9 a 11, aula 17
- laboratorio y práctico viernes de 9 a 13, lab 30
- las semanas con viernes feriado, práctico  
martes y jueves de 11 a 13, aula 17

# practicalidades

- página de la materia  
<http://cs.famaf.unc.edu.ar/materias/paradigmas/>
- grupo de facebook  
[Paradigmas de la Programación - FaMAF](#)
- docentes:
  - Ezequiel Orbe (laboratorio)
  - Laura Alonso Alemany (teórico-práctico)
  - ?
  - Emiliano Cuenca (ayudante)
  - Armando Soto (ayudante)
  - Ariel Wolfmann (ayudante)



# practicalidades: materiales

- libro: [Concepts in Programming Languages](#)  
sólo algunos capítulos
- y materiales complementarios

# practicalidades: evaluación

- nota = 50% teórico + 50% laboratorio
- regularización:
  - aprobar por lo menos 5 laboratorios
  - mínimo 50% en el teórico
- promoción:
  - aprobar todos los laboratorios
  - más de 80% en el teórico

# practicalidades: evaluación

- teórico: 2 parciales y un recuperatorio:
  - 23 de abril, el día de [Sant Jordi](#) y [del libro](#)!
  - 11 de junio
  - 18 de junio recuperatorio (un solo parcial)
- labs: 6 labs y un recuperatorio (un solo lab)
  - los labs se publican el lunes y se entregan el viernes después de dos semanas y media
  - ya hay un lab publicado, para entregar el 26/3
  - vean el [calendario de la materia](#) (también en la [página de la materia](#))

objetivos de la materia

# para qué hacemos lo que hacemos?

- por ejemplo, traducir programas de un lenguaje a otro
- encontrar bugs, incluso en programas de otro
- saber cómo optimizar
- saber elegir el mejor lenguaje en cada contexto
- saber cuándo hay que diseñar un lenguaje nuevo y cómo hacerlo

# conceptos fundamentales

- qué se puede y no se puede expresar en un lenguaje de programación
- los programas representados como cambios en una máquina
- diferencia entre **lo que** se expresa (*sémantica*) y **cómo** se expresa (*sintaxis*)
- potencial de los lenguajes de programación
- cómo diferentes lenguajes usan el potencial

# diferentes formas de abordar los conceptos

<https://www.destroyallsoftware.com/talks/wat>

pero también:

<http://theory.stanford.edu/people/jcm/papers/aplas08-camera-ready.pdf>

no pedimos que puedan leer el paper, pero sí  
entender qué pretende, por qué y cómo

# mapa

ver el [calendario](#)

- historia y paradigmas
- expresividad, sintaxis y semántica
- paradigma imperativo
- paradigma funcional
- paradigma orientado a objetos
- otros paradigmas