

Paradigmas de la Programación

FaMAF 2023

Laura Alonso Alemany

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

Cutting corners to meet arbitrary management deadlines



Essential

Copying and Pasting from Stack Overflow

O'REILLY®

The Practical Developer
@ThePracticalDev

The internet will make those bad words go away



Essential

Googling the Error Message

O RLY?

The Practical Developer
@ThePracticalDev

Software can be chaotic, but we make it work



Expert

Trying Stuff Until it Works

O RLY?

The Practical Developer
@ThePracticalDev



Me podrías implementar una función en Python que retorne el promedio de una lista de números pasados por parámetros y luego lo imprima?



Claro, aquí tienes una función en Python que retorna el promedio de una lista de números pasados por parámetros y luego lo imprime:



Copy code

```
def promedio(numeros):  
    # Calculamos la suma de todos los números en la lista  
    suma = 0  
    for numero in numeros:  
        suma += numero  
  
    # Calculamos el promedio dividiendo la suma por el número de elementos en la lista  
    promedio = suma / len(numeros)  
  
    # Imprimimos el promedio  
    print(promedio)  
  
    # Retornamos el promedio  
    return promedio
```


User this code is not working like i expect — how do i fix it?

```
resultWorkerErr := make(chan error)
defer close(resultWorkerErr)
go func() {
    defer cancel()
    resultWorkerErr <- b.resultWorker(ctx)
}()

err := b.worker(ctx)
cancel()
if err == nil {
    return <-resultWorkerErr
}
return multierror.Append(err, <-resultWorkerErr)
```

ChatGPT It's difficult to say what's wrong with the code without more context. Can you provide more information about what the code is supposed to do and what isn't working as expected? Also, is this the entire code or just a part of it?

<https://chat.openai.com/>

<https://github.com/features/copilot>

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 tener 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

- lab 17
- clases teóricas martes y jueves de 9 a 11
- laboratorio viernes de 9 a 13
- híbridas, por meet, y videos en la página

practicalidades

- página de la materia
<https://sites.google.com/unc.edu.ar/paradigmas2023>
- docentes:
 - Laura Alonso Alemany (teórico-práctico y laboratorio)
 - Beta Ziliani (laboratorio)
 - Facundo Bustos (laboratorio)
 - Tadeo Cocucci (laboratorio)
 - Francesco Silvetti (ayudante)

practicalidades: materiales

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

practicalidades: materiales

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



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

hoja de ruta

ver el [calendario](#)

- historia y paradigmas
- expresividad, sintaxis y semántica
- conceptos fundamentales
- programación funcional
- tipos
- orientación a objetos
- concurrencia y programación reactiva
- frameworks, scripting, programación lógica
- seguridad en lenguajes de programación