Python, Java, C++, C/C++.¿Qué lenguaje usar?

Miguel Astor Romero

Ecoanova Consulting C.A. - 10 de julio de 2020

Agenda

Sobre el presentador

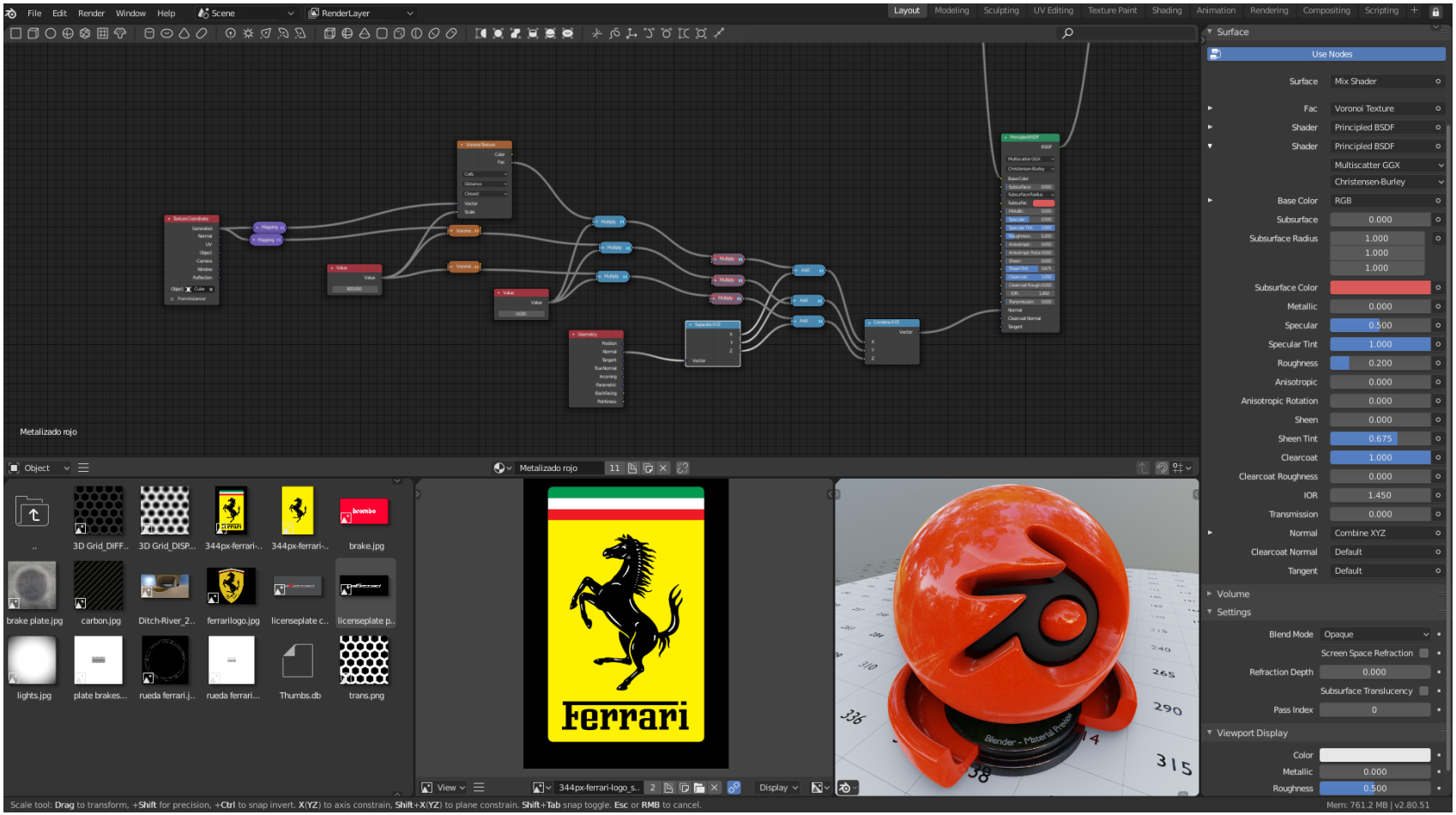
- Licenciado en Computación, UCV.
- Mención Computación Gráfica.
- Profesor Instructor en la Esc. de Computación, UCV.
- Profesor asociado a Ecoanova Consulting C.A.

Quiero Programar, ¿que lenguaje uso?

- Tal vez estoy empezando a aprender.
- Tal vez estoy comenzando un nuevo proyecto.
- Hay infinidad de lenguajes a escoger:
  - ¿Cual lenguaje es mejor para empezar a aprender?
  - ¿Cual se adapta mejor a mis necesidades?

¿Que es un Lenguaje de Programación?

Lenguaje de Programación es una Definición Flexible



¿Qué NO es un Lenguaje de Programación?

Como era al comienzo

- Las computadoras se programan directamente en ensamblador.
- Los ensambladores no tienen estructura.

Hello, AMD64!

```
    .section .text
    .globl main
    .byte 72, 101, 108, 108, 111, 44, 32, 87
    .byte 111, 114, 108, 100, 33, 10, 0
main:
    .quad 0x01c0c748e5894855
    .quad 0xffee358d48000000
    .quad 0xc7480fee8348ffff
    .quad 0xc9050f0000000ec2
    .byte 0xc3
```

La Falta de Estructura es Flexible pero Peligrosa

Paradigmas de los Lenguajes de Programación

Estilo de escritura de programas con un lenguaje específico.

Paradigma Imperativo

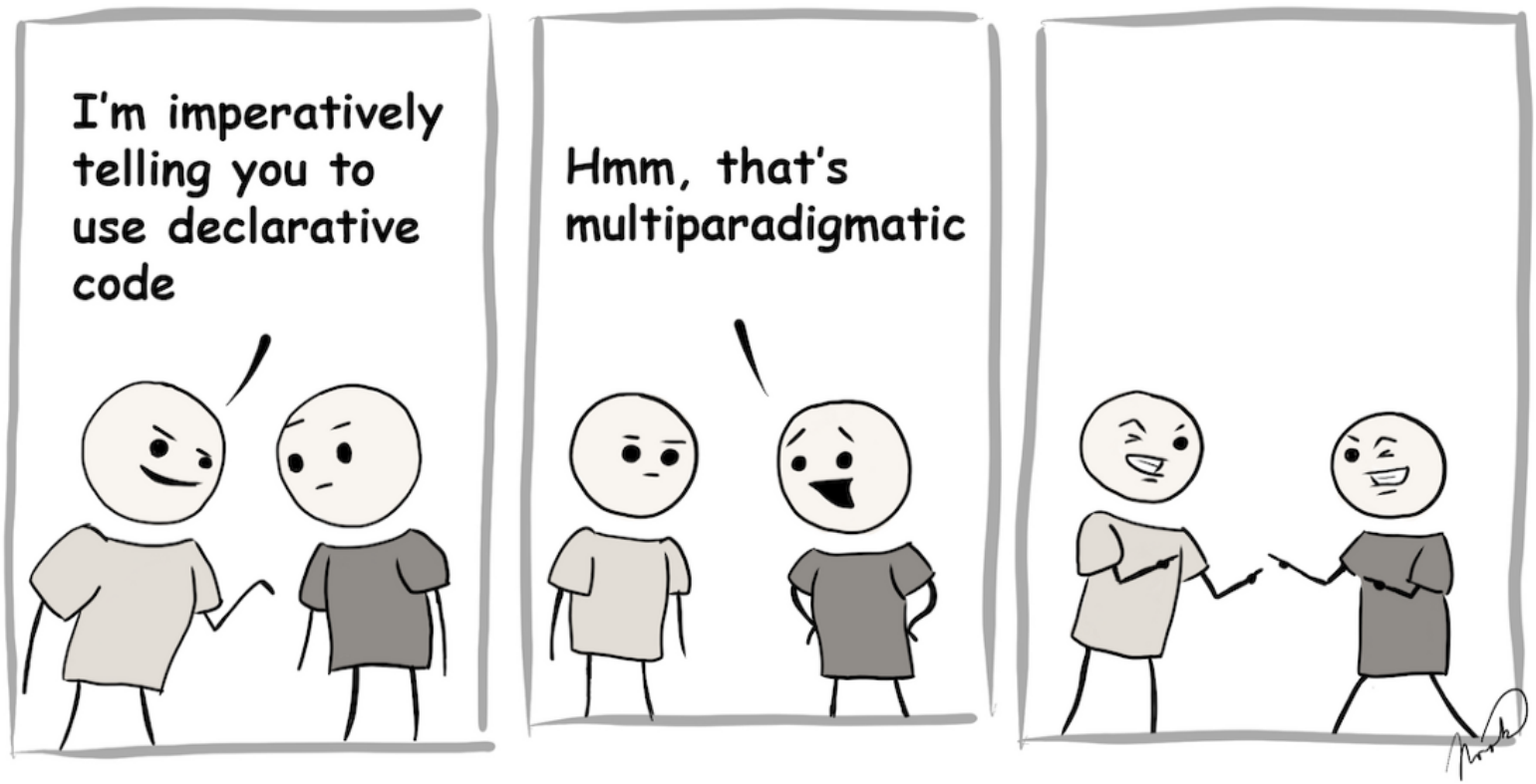
Los programas se componen de instrucciones secuenciales a ejecutar.

Paradigma Declarativo

Los programas se componen de declaraciones de resultados a obtener.

- Estos paradigmas se dividen en múltiples sub-paradigmas.

Los Lenguajes pueden (y suelen) ser Multiparadigma



Lenguajes Declarativos

- Los programas se definen como especificaciones de resultados a obtener.
- Es responsabilidad del entorno de ejecución determinar como obtener el resultado y calcularlo.

```
mother_child(trude, sally).
father_child(tom, sally).
father_child(tom, erica).
father_child(mike, tom).
sibling(X, Y)       :- parent_child(Z, X),
                       parent_child(Z, Y).
parent_child(X, Y) :- father_child(X, Y).
parent_child(X, Y) :- mother_child(X, Y).
```

Otros Lenguajes Declarativos

Flex

```
space       [1-7]
piece       (c|s|t|C|S|T)([1-9][0-9]{0,2})"."
pyramid     (p|P){piece}+"."
boardpiece  ({pyramid}|{piece})
row         (({space}|{boardpiece}){0,7}{boardpiece})?
board       ({row}"/"){15}{row}
```

SQL

```
SELECT * FROM personas WHERE nombre = 'Miguel' ORDER BY ASC;
```

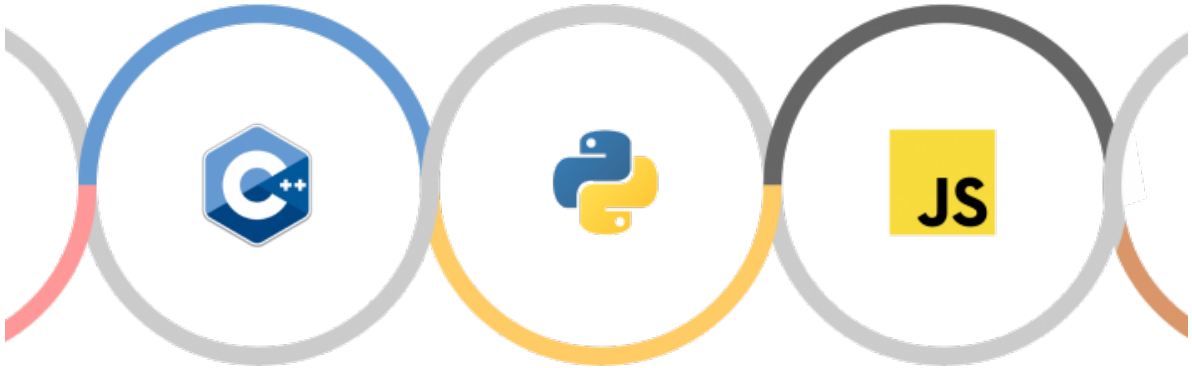
Lenguajes Imperativos

- Los programas se escriben como listas de instrucciones secuenciales.
- Existen tres grandes sub-paradigmas:

  Procedimental: Los programas se componen de subrutinas

  Orientado a objetos: Los programas se componen de clases y objetos

  Funcional: Los programas se componen de funciones



Lenguajes Procedimentales

C

```
#include <stdio.h>
int main(int argc, char ** argv) {
  if (argc > 1) {
    printf("Hello, %s!\n", argv[1]);
  } else {
    printf("Hello, World!\n");
  }
  return 0;
}
```

Lenguajes Orientados a Objetos

Greeter.java

```
class Greeter {
    String msg;
    Greeter(string _msg) {
        if (msg == null)
            msg = "Hello, World!";
        else
            msg = _msg;
    }
    public void greet() {
        system.out.println(msg);
    }
}
```

Main.java

```
class Main {
    public static void
    main(String args) {
        Greeter g =
          new Greeter(null);
        g.greet();
    }
}
```

Lenguajes Funcionales

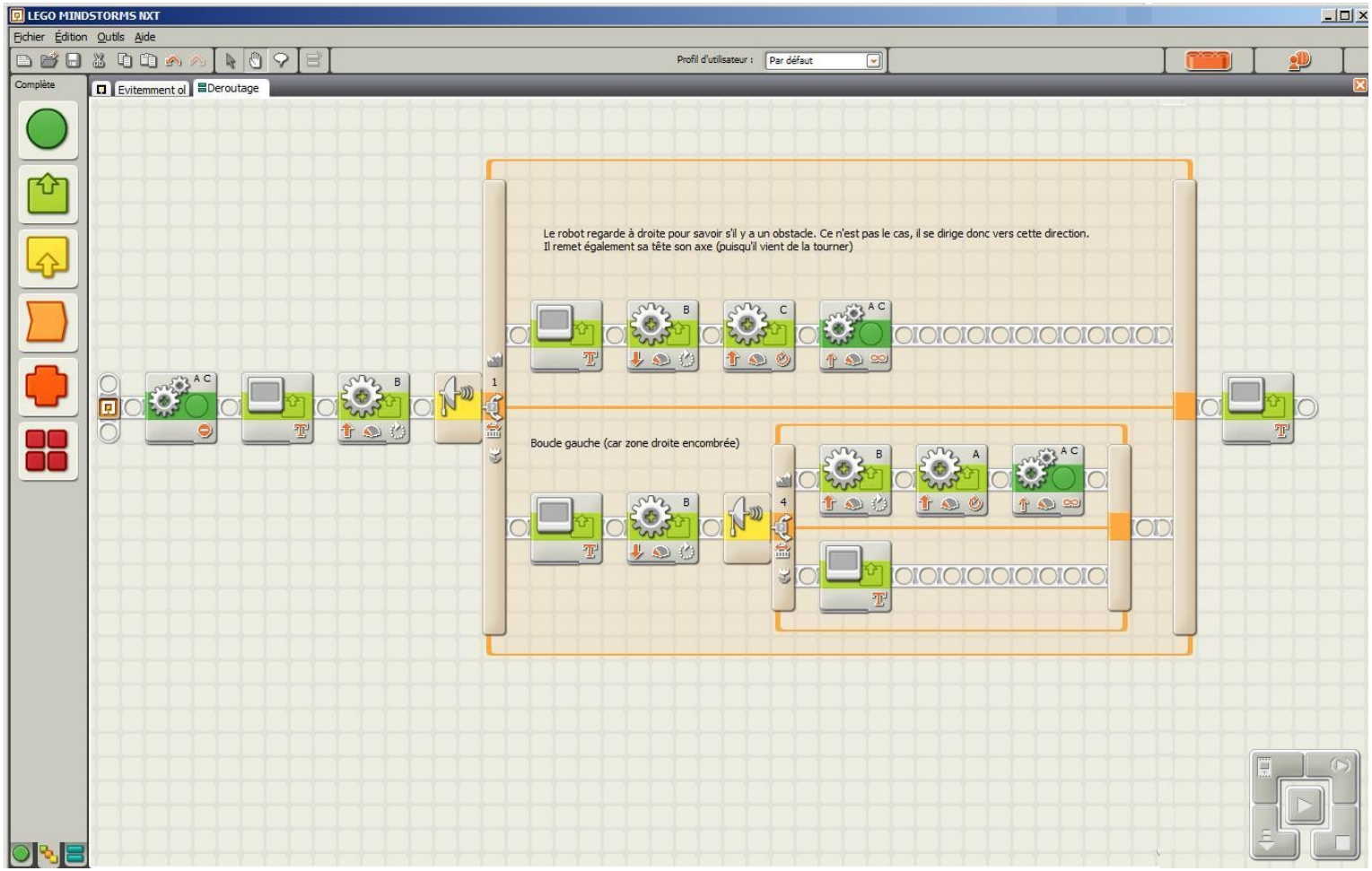
Haskell

```
quickSort []     = []
quickSort (x:xs) = quickSort [a | a <- xs, a < x]
                   ++ [x] ++
                   quickSort [a | a <- xs, a >= x]
```

Clojure

```
(defn -main ; name
  [& args] ; (variable) parameters
  (println "Hello, World!"))
```

Programación Visual



Lenguajes Esotéricos

- Usan paradigmas cada vez más insólitos.
- Poca utilidad práctica.

Befunge

```
2>:3g" "-!v\  g30          <
 |!`"0":+1_:.:03p>03g+:"0"`|
 @               ^  p3\" ":<
2 234567890123456789012345678 ...
```

Brainfuck

```
++++++++[>++++[>++>+
++>+++>+<<<<-]>+>+>-
>>+[<]<-]>>.>---.+++
++++..+++.>>.<-.<.++
+.------.--------.>>
+.>++.
```

Lenguajes Compilados e Interpretados

También se pueden clasificar los lenguajes según su forma de uso.

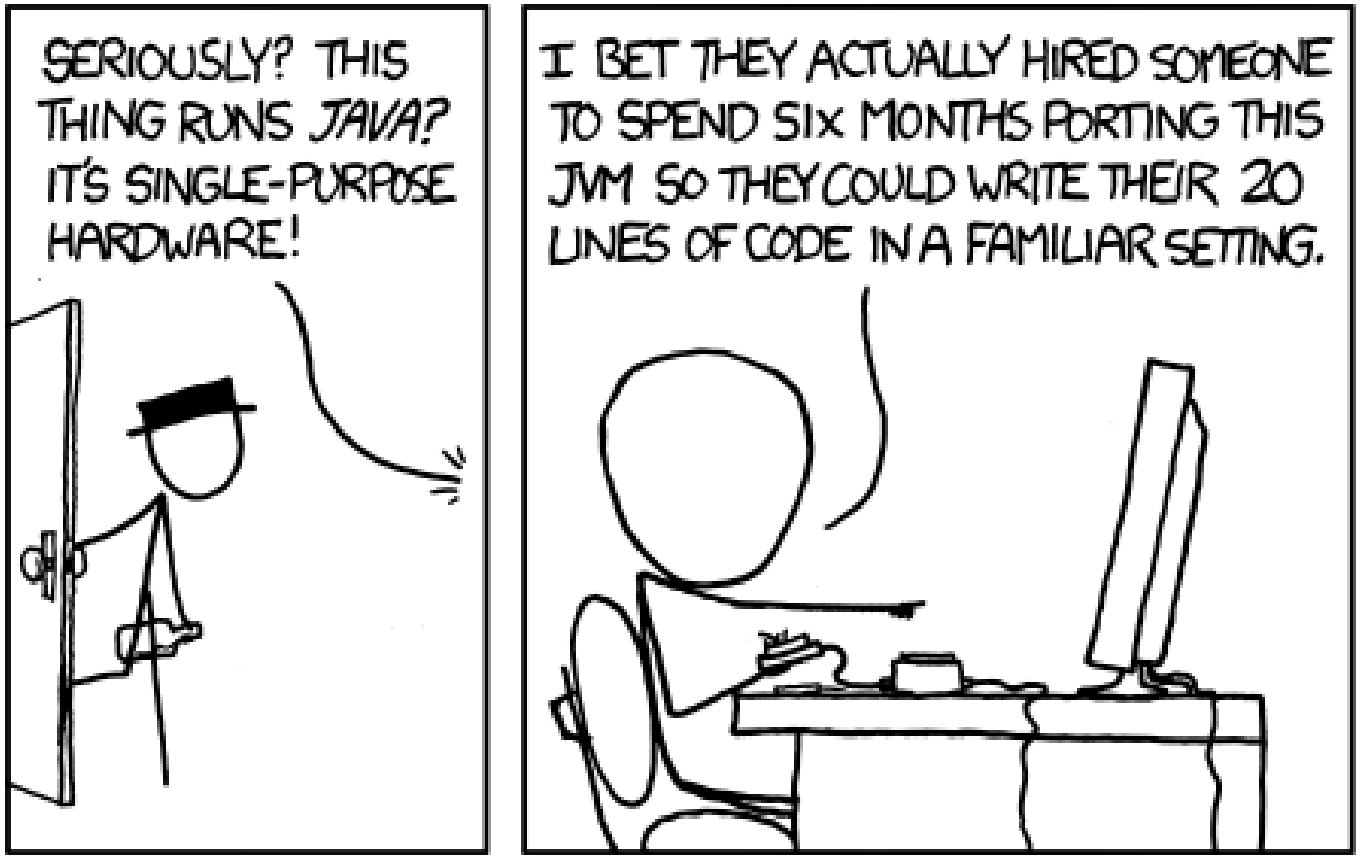
Lenguajes Compilados

Tienen que convertirse a código máquina específico de la computadora mediante un compilador.

Lenguajes Interpretados

Son ejecutados en tiempo real mediante un intérprete o máquina virtual independiente de la computadora real.

Ayer Todo el Mundo Quería Usar Java



Hoy Todo el Mundo Quiere Usar Swift

¿Swift?

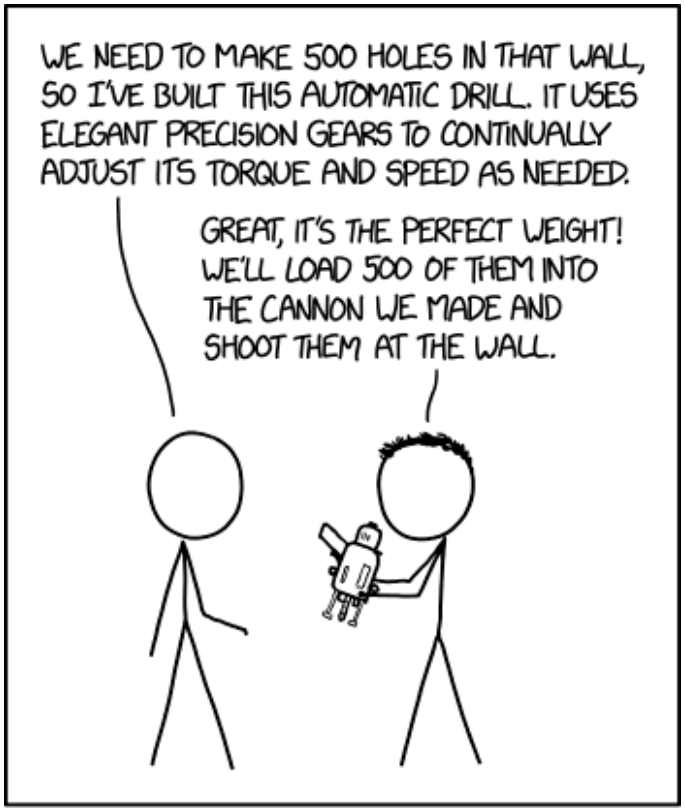
¿Por que no usar solo Java/C/Swift/etc.?

La Décima Ley de Greenspun

Any sufficiently complicated C or Fortran program contains an ad hoc, informally-specified, bug-ridden, slow implementation of half of Common Lisp.

Philip Greenspun

Es Cuestión de Usar la Herramienta Apropiada para el Trabajo



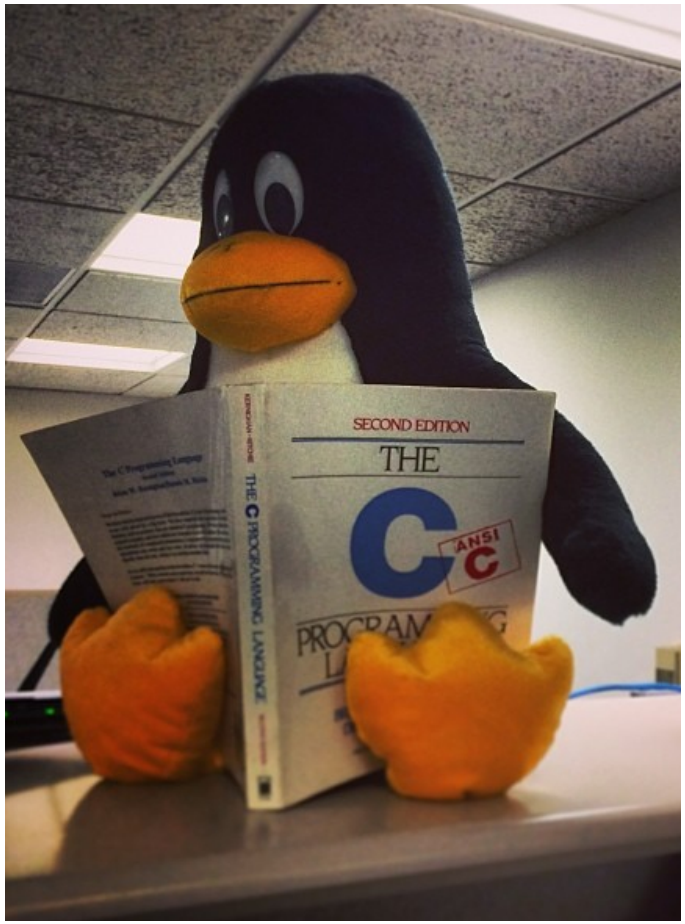
Dominios de Aplicación

Cada dominio de aplicación tiene sus propias características.

- Programación de sistemas.
- Programación de videojuegos.
- Cómputo numérico.
- Desarrollo Web (*frontend* y *backend*).
- Programación paralela.
- Programación matemática.
- Prototipado rápido.

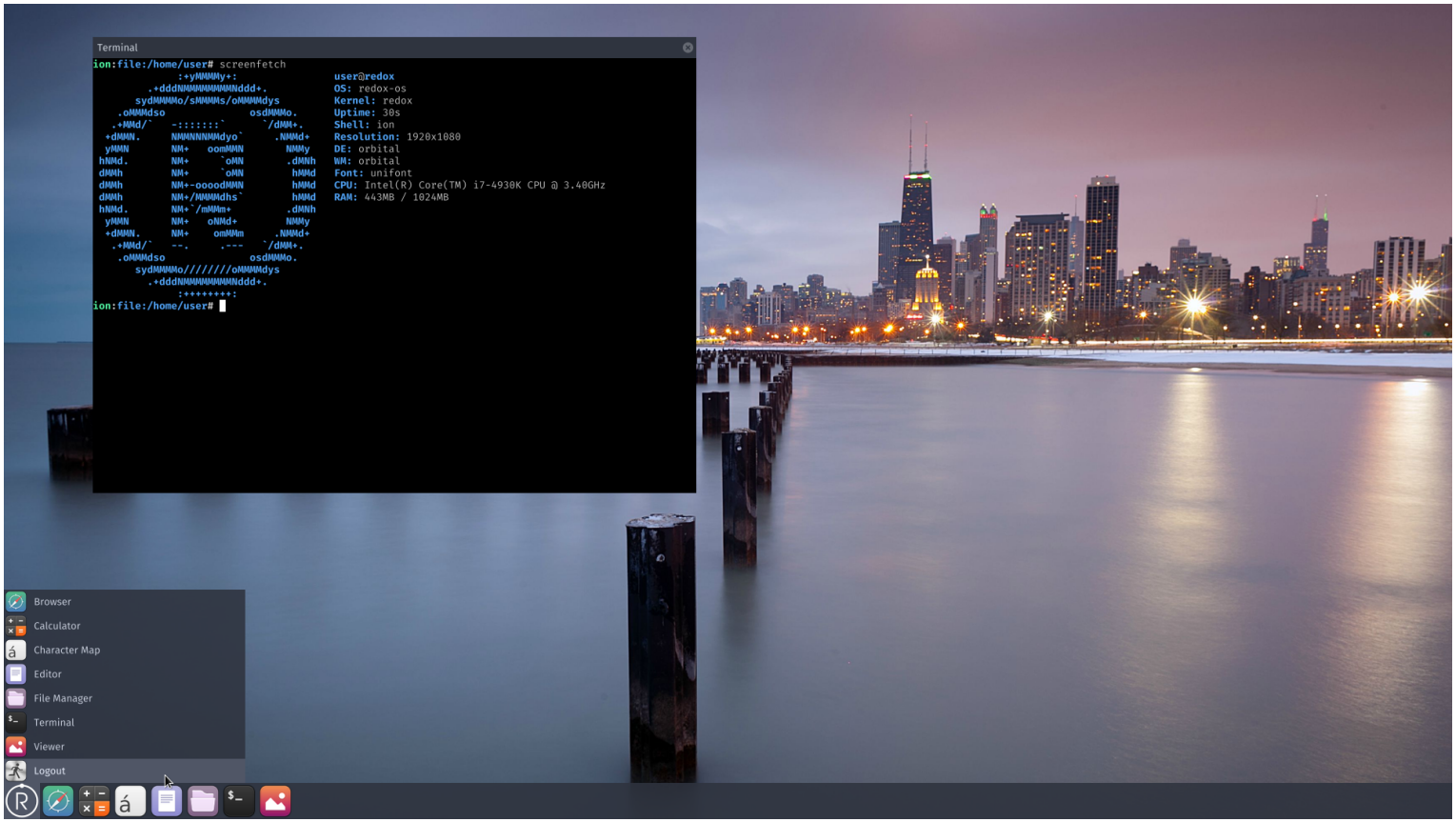
Programación de Sistemas

- Programación directa sobre el hierro.
- No hay entorno de ejecución disponible.
- Es posible usar lenguajes de alto nivel, pero hay que implementar el entorno de ejecución desde cero.
- Acceso directo a la memoria.
- C es el lenguaje de programación de sistemas por excelencia.



Programación de Sistemas de Alto Nivel

Redox es un sistema operativo tipo UNIX programado en Rust.



Programación de Videojuegos y Computación Gráfica

- La computación gráfica se adapta perfectamente a la programación orientada a objetos.
- Entidades con atributos y comportamientos.
- Lenguajes comunes:
  - C++
  - Java
  - C# (.NET y Mono)

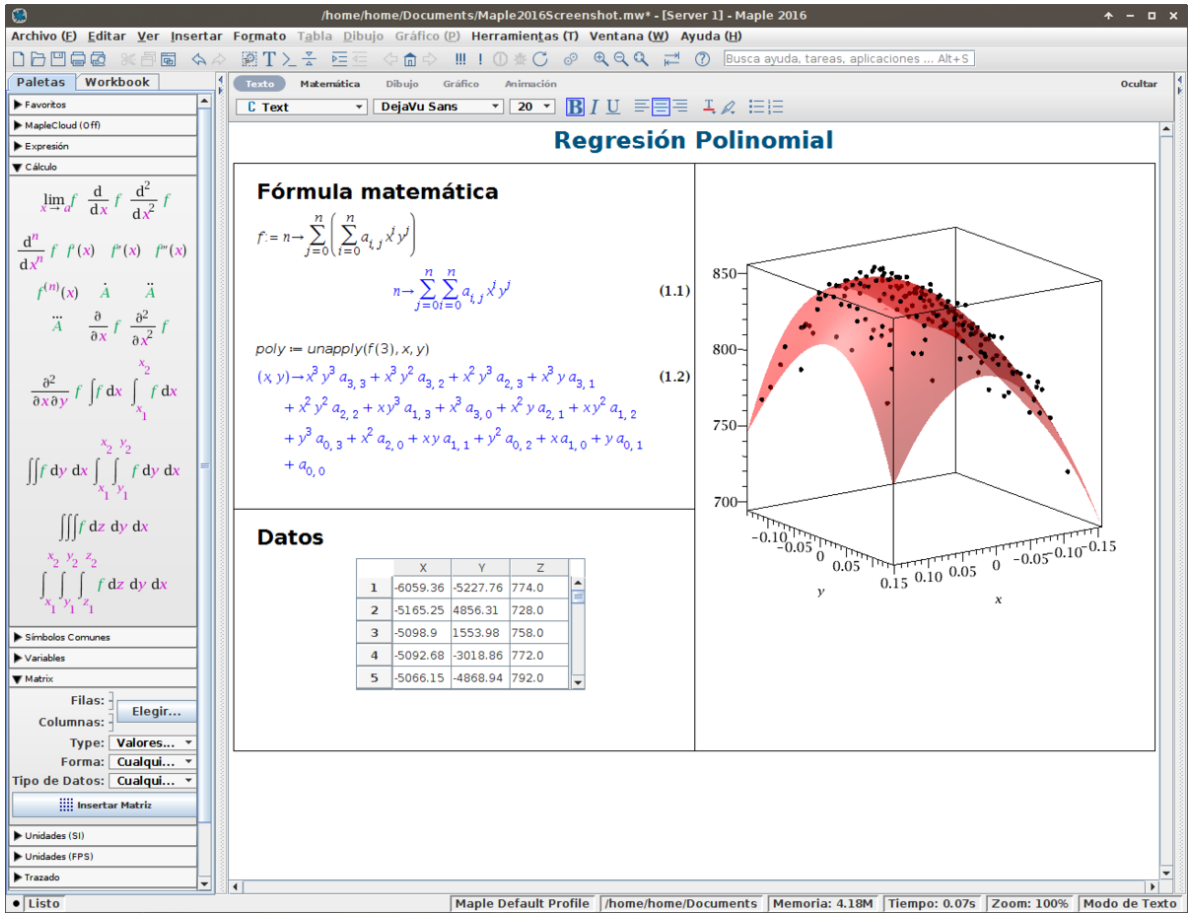
Cómputo Numérico

- Se refiere a cualquier problema que requiera de técnicas de análisis numérico:
  - Simulación
  - Resolución de ecuaciones diferenciales
  - Autovalores y autovectores
  - Integración numérica
  - Elementos finitos
- La matriz es el tipo de dato fundamental

Lenguajes

- MATLAB
- Octave
- SciLab
- Mathematica
- Maple
- R

Cómputo Simbólico



Desarrollo Web

Frontend

- Los navegadores solo ejecutan JavaScript.
- Otros lenguajes se traducen a JavaScript:
  - CoffeeScript
  - TypeScript
  - Dart

Backend

- Se puede usar cualquier lenguaje que pueda interactuar con el servidor.
- Orientado a objetos.
- Puede verse como un problema de programación funcional (!).

El desarrollo de Navegadores es Otra Historia

- Relacionado con la programación de sistemas.

Programación Paralela

- Los programas paralelos son difíciles.
- Los lenguajes procedimentales/orientados a objetos típicos no se adaptan bien a este problema.
  - Variables globales
  - Condiciones de carrera
  - Interbloqueos
- Los lenguajes funcionales como Lisp o Haskell son particularmente útiles.
- Haskell en particular no permite estados globales.
  - Ergo, no más condiciones de carrera

```
https://engineering.fb.com/security/
fighting-spam-with-haskell/
```

Programación Matemática

Lenguajes funcionales como Haskell se adaptan directamente a este tipo de problemas.

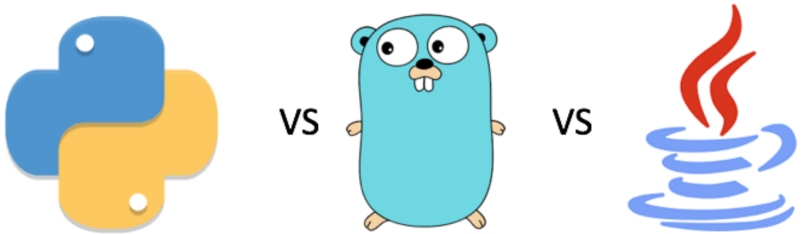
Ecuación Cuadrática en Haskell

```
solveq :: (Double,Double,Double) ->[Double]
solveq (a,b,c)
 | (d < 0) = []
 | (d > 0) = [(- b - sqrt d)/(2*a), (- b + sqrt d)/(2*a)]
 | otherwise = [-b/(2*a)]
 where
 d = b*b - 4*a*c
```

Prototipado Rápido

Los Lenguajes Interpretados Permiten un Ciclo de Desarrollo más Rápido

- Los lenguajes interpretados modernos suelen ser multiparadigma.
- No hay que pasar por una fase de compilación (usualmente).
- Ejemplos:
  - Python
  - Perl
  - Ruby
  - CoffeeScript



Comparando C y Python

```
void quicksort(int number[25],int first,int last){
  int i, j, pivot, temp;
  if(first<last){
    pivot=first;
    i=first;
    j=last;
    while(i<j){
      while(number[i]<=number[pivot]&&i<last)
        i++;
      while(number[j]>number[pivot])
        j--;
      if(i<j){
        temp=number[i];
        number[i]=number[j];
        number[j]=temp;
      }
    }
```

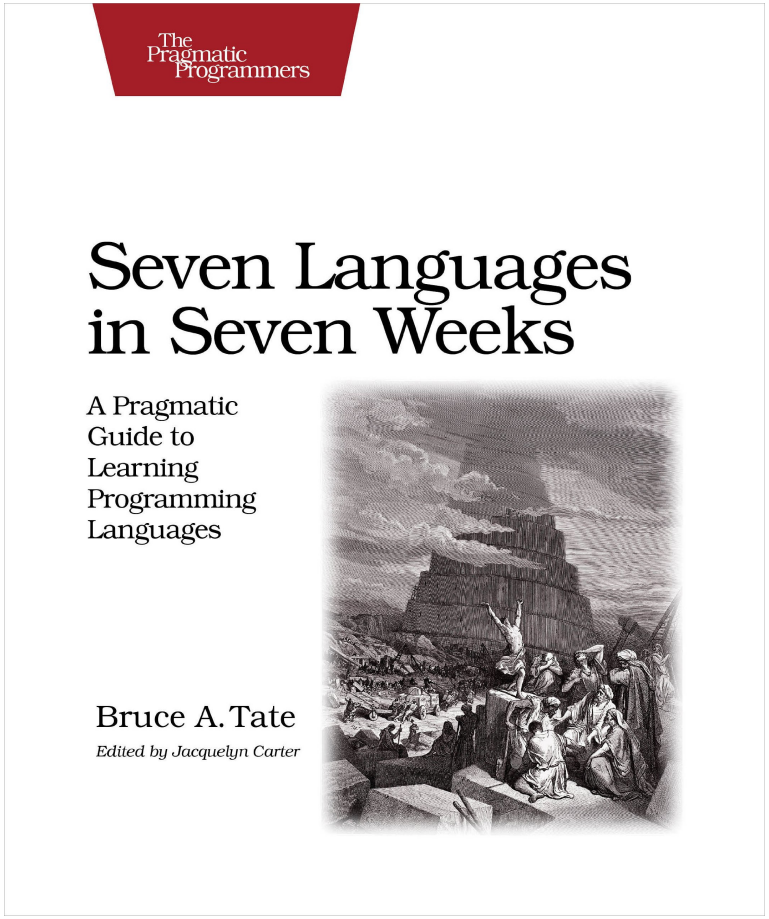
Comparando C y Python

```
def qs(l):
    if len(l) == 0 or len(l) == 1:
        return l
    else:
        p = l.pop()
        less = qs([x for x in l if x < p])
        more = qs([x for x in l if x >= p])
        return less + [p] + more
```

Conclusiones

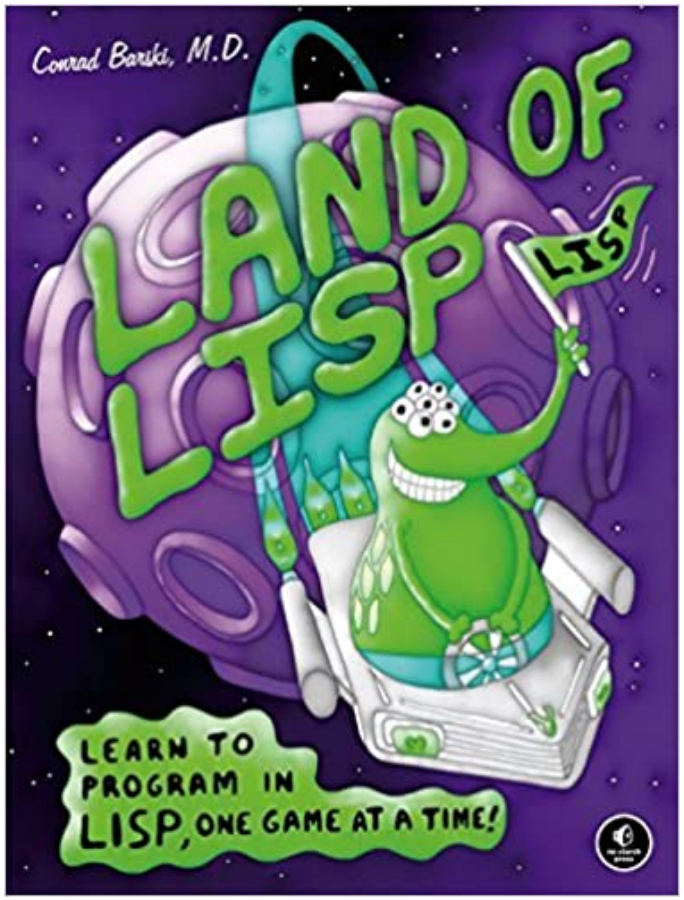
- Para escoger un lenguaje de programación hay que sopesar las propiedades del lenguaje con respecto a las características del problema a resolver:
  - Los lenguajes orientados a objetos ayudan cuando el problema se puede representar como relaciones entre entidades.
  - Los lenguajes funcionales sirven para programación paralela y problemas definidos matemáticamente.
  - Los lenguajes procedimentales y de bajo nivel sirven para programación de sistemas y embebidos.
  - Los lenguajes interpretados sirven para prototipado rápido.

Libros Recomendados (1)



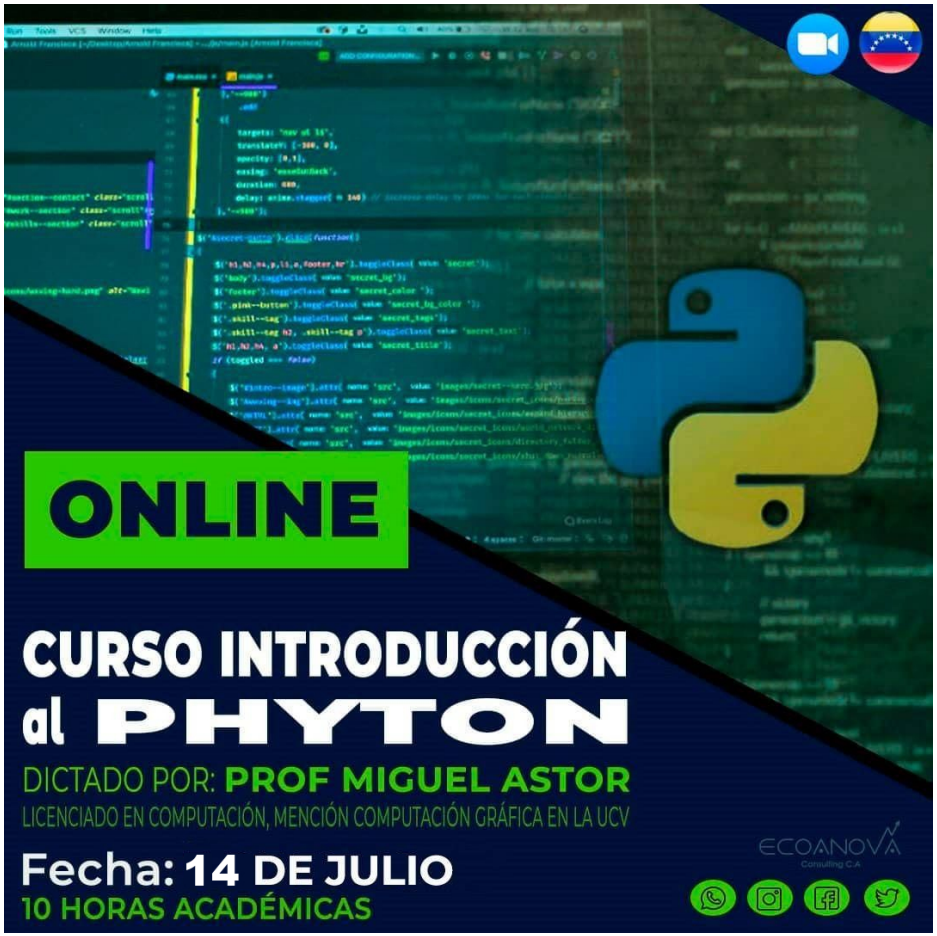
- **Seven Languages in Seven Weeks**
- De Bruce Tate.
- Introducción a siete paradigmas de programación.

Libros Recomendados (2)

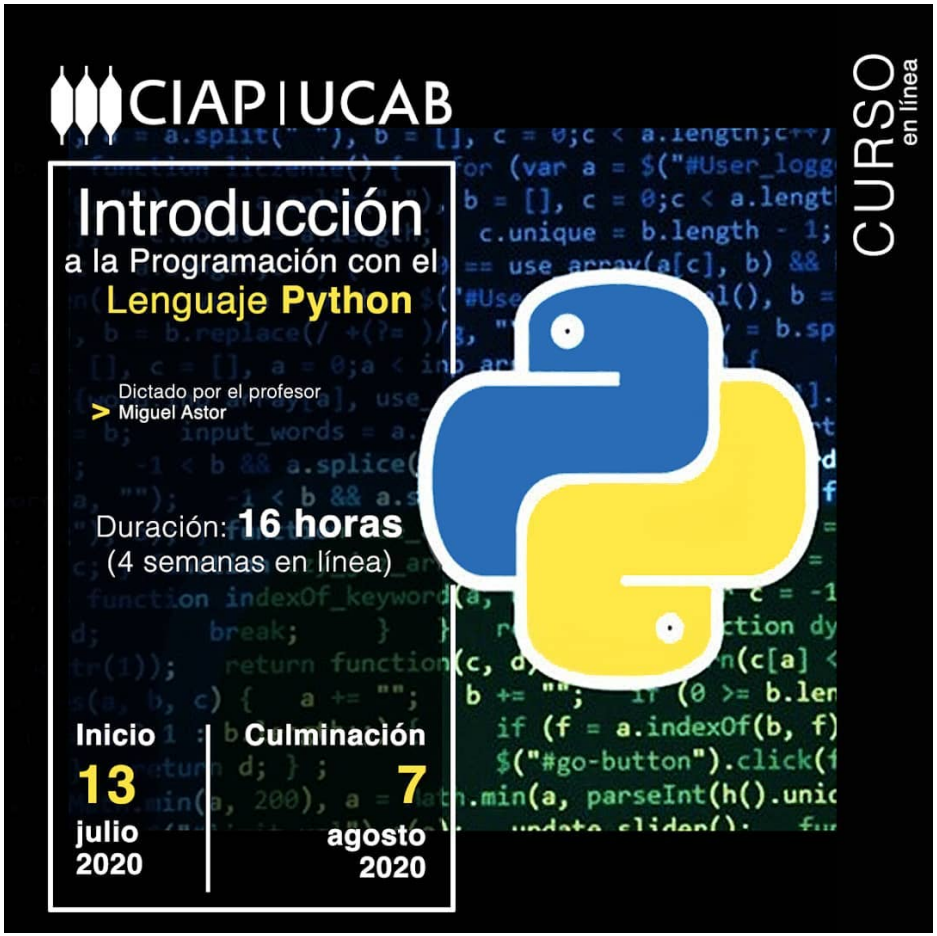


- **Land of Lisp**
- De Conrad Barski.
- Introducción al lenguaje Common Lisp mediante programación de videojuegos.

Invitación


ONLINE
CURSO INTRODUCCIÓN
al PHYTON
DICTADO POR: PROF MIGUEL ASTOR
LICENCIADO EN COMPUTACIÓN, MENCIÓN COMPUTACIÓN GRÁFICA EN LA UCV
Fecha: 14 DE JULIO
10 HORAS ACADÉMICAS
ECOANOVA

Invitación


CIAP | UCAB
CURSO
en línea
Introducción
a la Programación con el
Lenguaje Python
Dictado por el profesor
> Miguel Astor
Duración: 16 horas
(4 semanas en línea)
Inicio
13
julio
2020
Culminación
7
agosto
2020

Contactos

Prof. Miguel Astor

- mastor89@protonmail.com

Ecoanova Consulting C.A.

- @ecoanova en Instagram
- info@ecoanova.com
- WhatsApp +58 412-7334799

Gracias por su atención