





Introducción al pensamiento computacional

Objetivos

-  Explicar cómo aplicar un problema sistemático o la solución de problemas.
-  Discutir cómo crear una definición del problema.
-  Introducir estrategias y consideraciones para encontrar soluciones.
-  Explicar la descomposición como una estrategia en la solución de problemas.

Todos hemos tenido al menos un problema en nuestra vida. Comenzamos analizando la situación para que una computadora nos ayude a resolver nuestro problema. Pero cuando estamos analizando el problema nos aparece la pregunta más difícil.

¿Por donde comenzar?

Los problemas de la vida real tienden a ser complejos y aun más cuando uno de estos es un problema no trivial, ya que revela detalles complejos y varias facetas a considerar.

Cuando nos enfrentamos a un task complejo, esto puede ayudarnos a seguir ciertos tipos de guías o procesos.

Estas guías y procesos suelen considerarse como instrucciones pieza de auto ensamble. Un proceso paso a paso para la solución de problemas parecería un benéfico obvio. Desafortunadamente la solución de problemas es un proceso parcialmente creativo. Cómo pintar un panorama o escribir una novela, esto no puede ser totalmente sistematizado, por estrategias, métodos aleatorios y de buenas prácticas existen para ayudarnos durante procesos creativos.

Creatividad + Estrategia + Aleatorio + Buenas Practicas = Solución de problemas

Otro punto importante a considerar es el conocimiento previo o antecedentes para resolver el problema, ejemplos claros de esto puede ser resolver tipos de problemas inspirados por las mejores tradiciones de matemáticas y ciencias naturales.

Cada área de los métodos científicos siguen los siguientes pasos.

	Formar hipótesis			Entender el problema
	Planear experimentos			Generar un plan
	Ejecutar experimentos			Ejecutar el plan
	Evaluar resultados			Revisar y extender

Que no cunda el pánico

Comenzar es difícil, eso todo lo sabemos. El tamaño, la complejidad y el número de incógnitas pueden alterarte nuestros inicios para empezar, pero pocos intentos racionales pueden ayudarnos.

Tú siempre encontraras que un problema muy grande en realidad es todo un grupo de problemas pequeños relacionados entre si.

Importante

Resístete a la urgencia de brincar directamente en resolver el problema.


Si tu lo haces lo más probable es que solo resuelvas solo una parte muy especifica del problema.

Recuerda que lo ideal es resolverlo completamente.


La solución de problemas involucra transformar puntos falsos o engañosos de comienzo a un punto final no deseable.

Para poder examinar el punto de comienzo, tu debes de tener claramente que es lo no deseado y porque lo es, esto te ayudara a revelar más acerca de cual meta debería de ser.

Tu podrías encontrar una cantidad de estados indeseados por este motivo listaremos algunas ideas o pasos que nos podrían apoyar a comprender el problema.

 Escribir el problema en nuestras propias palabras.

 Tratar de representar el problema, mediante diagramas o imágenes.

 Aclarar todas nuestras dudas antes de comenzar.

Una vez entendido y defino el problema completo, es posible considerar una estrategia para poder resolver el problema asegurando:

 Calidad

 Colaboración

 Interacción

 Descomposición

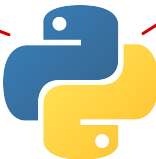
 Recursión



Primera aparición fue en 1991

Enfatiza la lectura de código mediante indentation.

```
print "Starting loop"
for x in [1,2,3,4,5]
    if x%2 == 0
        x += 3
        print x
    print "One execution of the loop!"
print "end of loop"
```



Lenguaje interpretado

lenguaje de programación para el que la mayoría de sus implementaciones ejecuta las instrucciones directamente, sin una previa compilación del programa.

Multi paradigma

- Orientado a objetos
- Orientado a funciones
- Procesos
- Reflective (modificar su propio estado)

Ejercicio 1 - Desplegar un string en tu terminal con la ayuda de Python.

1



+



2

Python tiene cuatro tipos primitivos: enteros, flotantes, booleanos y Strings.

1,2,3,4,5 ...

0 (False) ó 1 (True)

.1, .002, .03 ...

"Hello World"
'Hello World'

3

7. Entrada y salida

print()

Hay diferentes métodos de presentar la salida de un programa; los datos pueden ser impresos de una forma legible por humanos, o escritos a un archivo para uso futuro. Este capítulo discutirá algunas de las posibilidades.

```
>>> print(1)
1
>>> print(.01)
0.01
>>> print(True)
True
>>> print(False)
False
>>> print("Hola")
Hola
>>> print('Hola')
Hola_
```

Ejercicio 2 - Realiza la siguiente operación matemática en Python $100 + 25 - 10 * 3$

1



+



Ctrl + L → Borrar todo

```
[>>> 100 + 25 - 10 * 3  
95  
■
```

Tarea - Realiza un programa que calcule el seno de 25.