

# **Fundamentos de la computación**

## **CI-0202 Principios de informática**

Sivana Hamer - [sivana.hamer@ucr.ac.cr](mailto:sivana.hamer@ucr.ac.cr)

**Escuela de Ciencias de la Computación e Informática**  
**Universidad de Costa Rica**  
Licencia: CC BY-NC-SA 4.0



**UNIVERSIDAD DE COSTA RICA**

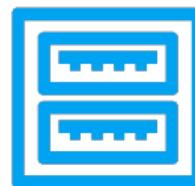
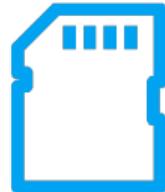
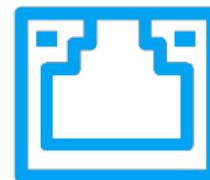
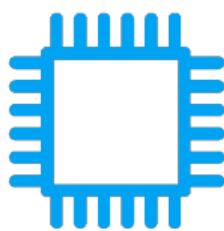
# ¿Qué es una computadora?

**Una computadora es un dispositivo electrónico que permite almacenar y manipular datos (e.g. laptops, teléfonos inteligentes, dispositivos IoT).**



¿Qué partes tiene una  
computadora?

**El hardware es la parte física de la computadora.**

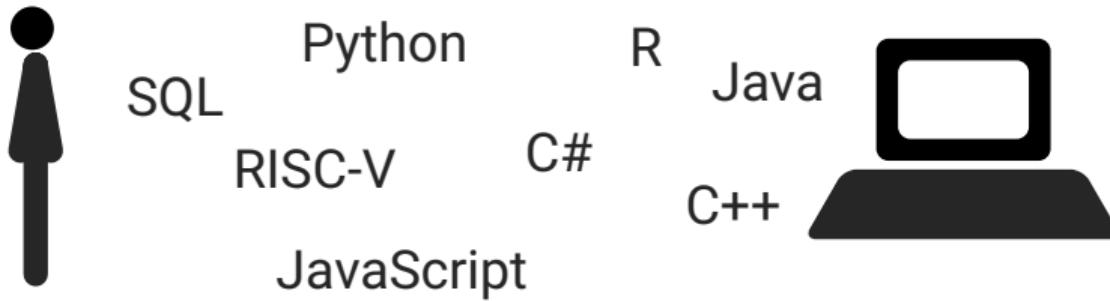


**El software es la parte lógica de la computadora.**

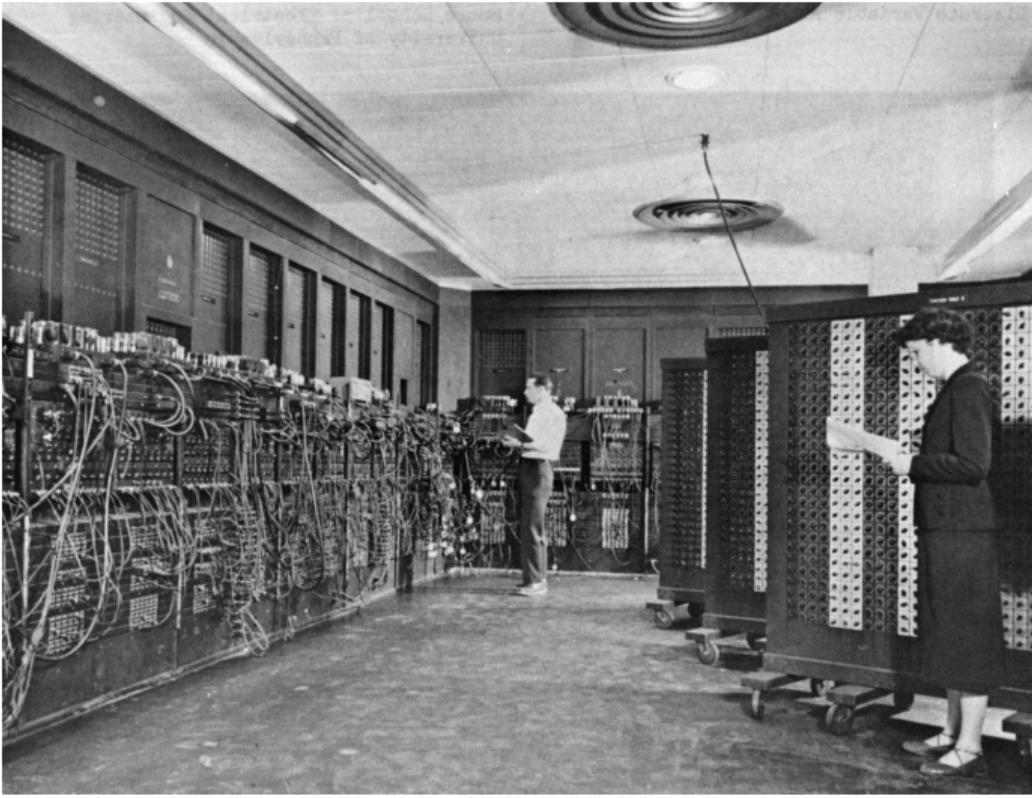


# ¿Qué es un programa?

**Los programas son conjuntos de instrucciones escritos en un lenguaje formal que indica a una computadora cómo realizar una tarea**

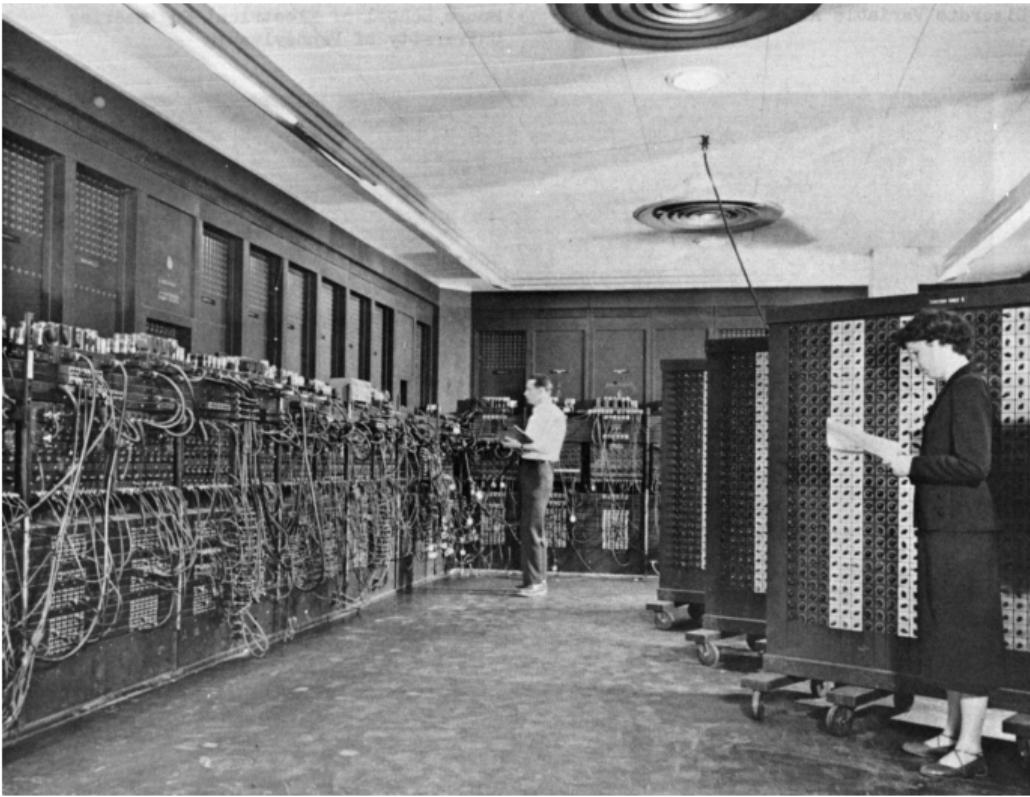


Dado que los programas se escriben en lenguajes, vamos a ver sobre los lenguajes de programación

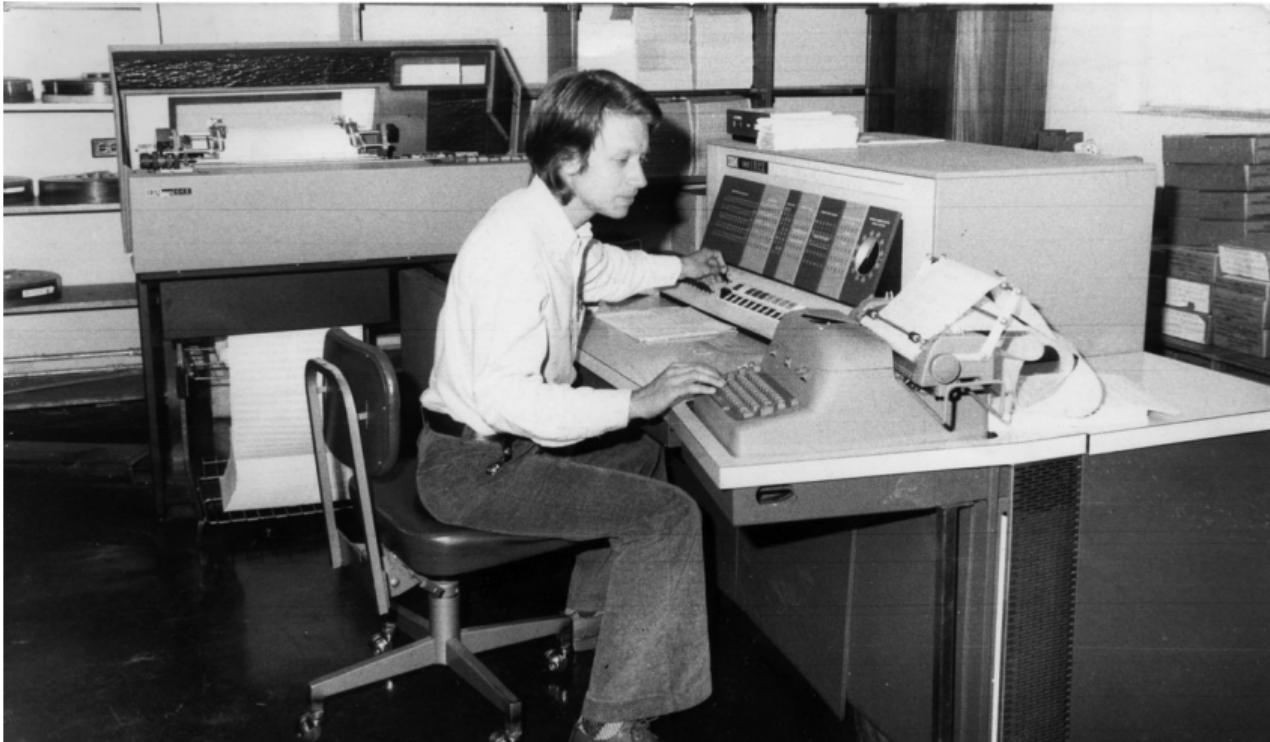


¿Qué era eso?

## ENIAC, una de las primeras computadoras de los 40s



# Matilde, la primera computadora en Costa Rica



¿Cómo se escribía código en esas máquinas?

¿Cómo se escribía código en esas máquinas?  
Lenguaje máquina

# El lenguaje máquina son secuencias de bits para controlar al procesador

## “Hello world” en lenguaje máquina [1]

```
00000000 7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00 00 00 |.ELF.....|
00000010 02 00 03 00 01 00 00 00 54 80 04 08 34 00 00 00 00 |.....T...4...|
00000020 00 00 00 00 00 00 00 00 34 00 20 00 01 00 00 00 |.....4. ....|
00000030 00 00 00 00 01 00 00 00 00 00 00 00 00 80 04 08 |.....|
00000040 00 80 04 08 74 00 00 00 74 00 00 00 05 00 00 00 |....t....t....|
00000050 00 10 00 00 b0 04 31 db 43 b9 69 80 04 08 31 d2 |.....1.C.i...1.|
00000060 b2 0b cd 80 31 c0 40 cd 80 48 65 6c 6c 6f 20 77 |....1.@..Hello w|
00000070 6f 72 6c 64                                |orld|
```

# ¿Cúales errores tiene el lenguaje máquina?

¿Cúales errores tiene el lenguaje  
máquina?

Difícil y propenso a errores en  
programas grandes

Dados las dificultades, decidieron  
crear el lenguaje ensamblador

# El lenguaje ensamblador expresa las operaciones de bajo nivel en abreviaturas nemotécnicas

## “Hello world” en ensamblador RISC-V para Linux [2]

```
.global _start      # Provide program starting address to linker

# Setup the parameters to print hello world and then call Linux to do it.

_start: addi a0, x0, 1      # 1 = StdOut
        la   a1, helloworld # load address of helloworld
        addi a2, x0, 13     # length of our string
        addi a7, x0, 64     # linux write system call
        ecall               # Call linux to output the string

# Setup the parameters to exit the program and then call Linux to do it.

        addi  a0, x0, 0    # Use 0 return code
        addi  a7, x0, 93   # Service command code 93 terminates
        ecall               # Call linux to terminate the program

.data
helloworld: .ascii "Hello World\n"
```

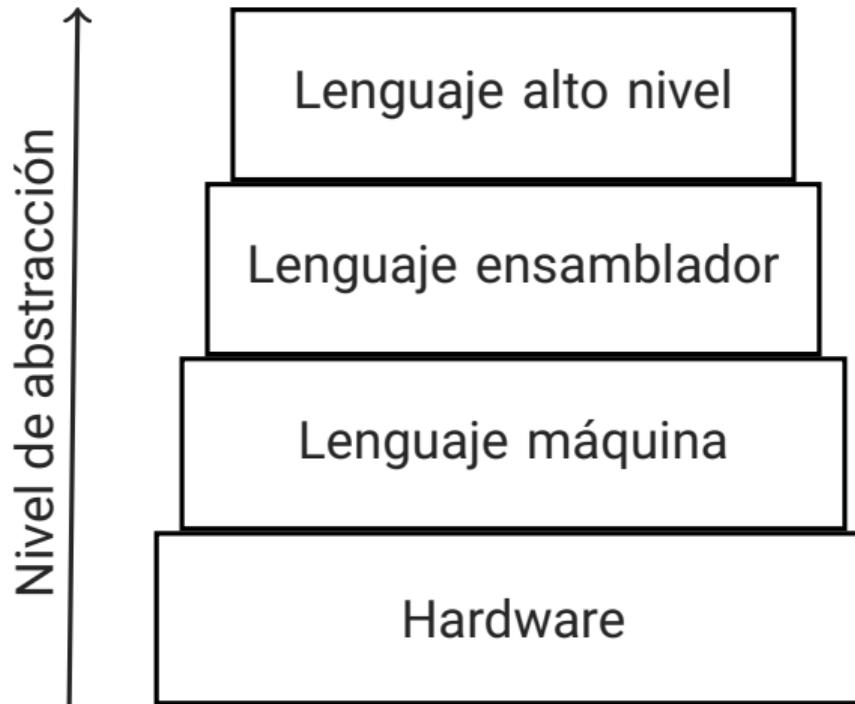
Sin embargo, es complejo escribir en ensamblador y depende de la máquina. Por lo tanto, se crearon lenguajes de alto nivel.

# Los lenguajes de alto nivel son más legibles para humanos y portátiles entre computadoras

## “Hello world” en Java

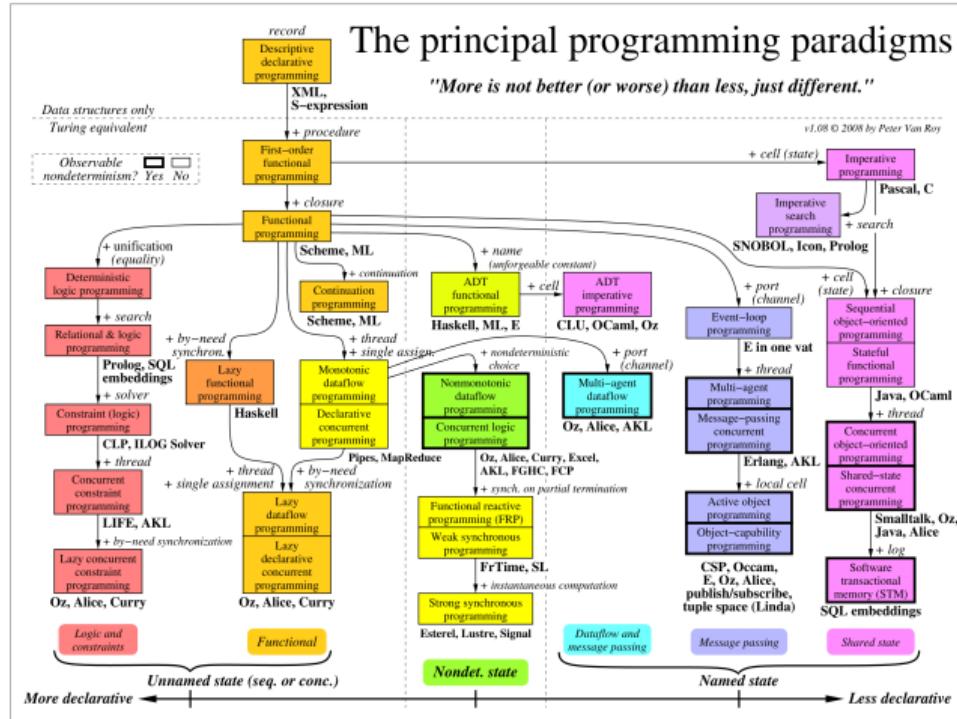
```
print("Hello world") #Prints "Hello world"
```

# Siempre se traduce los lenguajes de alto nivel a los lenguajes de bajo nivel



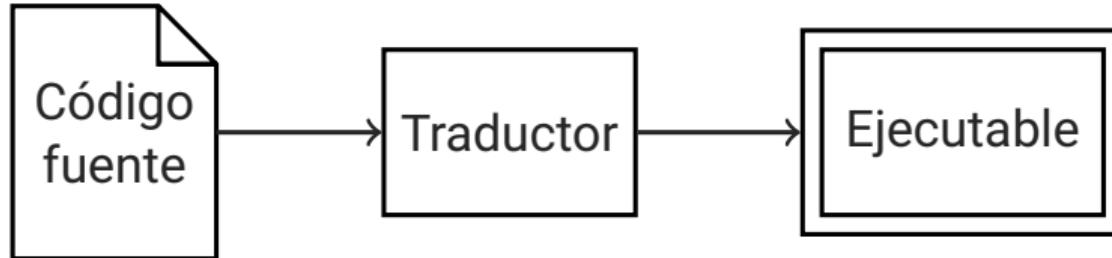
Todos los lenguajes de alto nivel  
tienen un paradigma.

# Un paradigma de programación es una manera de programar en un lenguaje de alto nivel basado en reglas matemáticas o un conjunto de principios



Obtenido de [3]

# Para correr los lenguajes de alto nivel deben ser traducidos a lenguaje máquina



## Nota

Traducir programas tiene un costo en tiempo, aunque tiene ventajas para la facilidad de lectura y portabilidad.

## Se puede traducir el código de dos maneras...

Compilador	Interpretador
<p>Traduce el código completo antes de la ejecución del programa</p> <p>La traducción genera un ejecutable que puede correrse varias veces en la misma computadora.</p> <p>Es como un traductor que traduce un libro de inglés a español, que produce un nuevo libro en español.</p>	<p>Traduce el código por partes durante la ejecución del programa</p> <p>La traducción no genera nada, por lo que ocupa interpretarse cada vez en la misma computadora.</p> <p>Es como un intérprete de LESCO que traduce al mismo tiempo que se está hablando.</p>

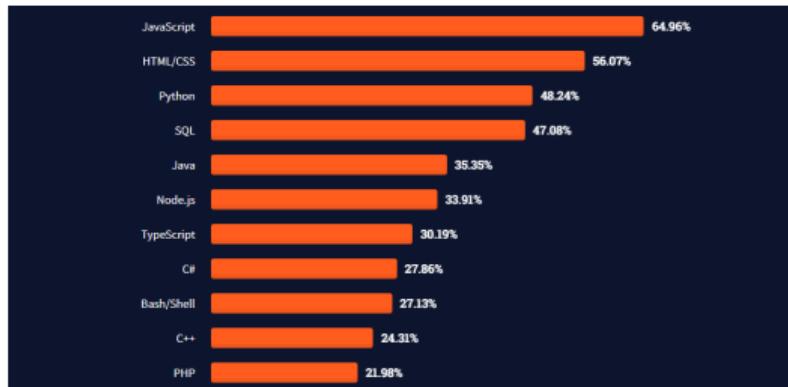
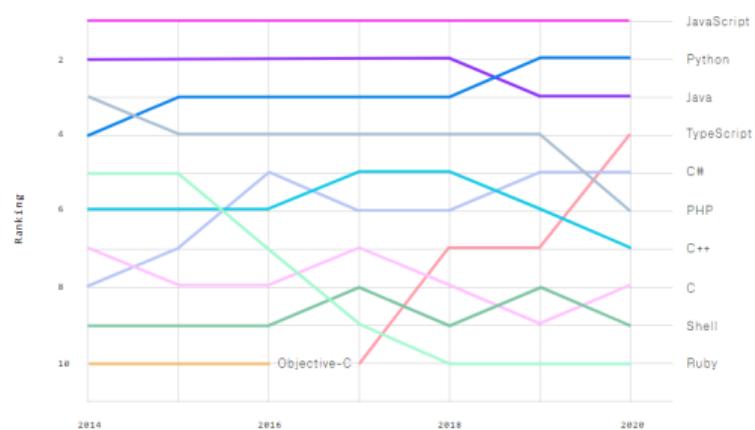
**Python es un lenguaje de alto nivel minimalista que usa el paradigma de programación orientada a objetos**



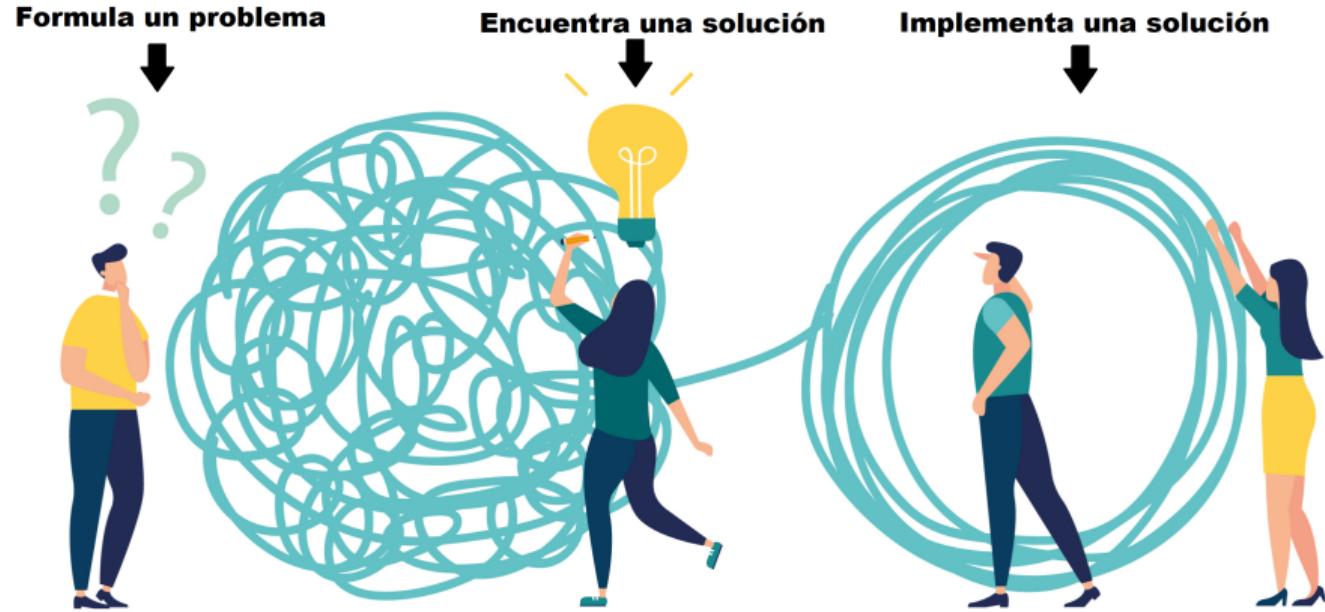
### Fun fact!

El nombre Python viene de la serie de televisión “Monty Python’s Flying Circus”

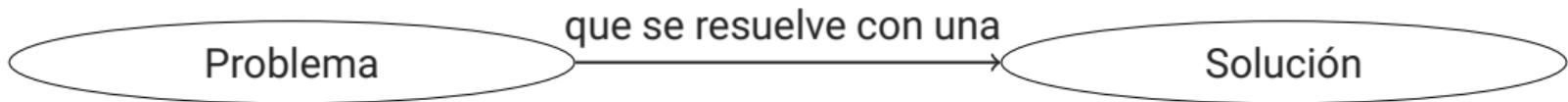
# Python es uno de los lenguajes más populares [4, 5]



# La resolución de problemas es un proceso donde se...



## La resolución de problemas tiene...



**Los problemas son lo que buscamos resolver**

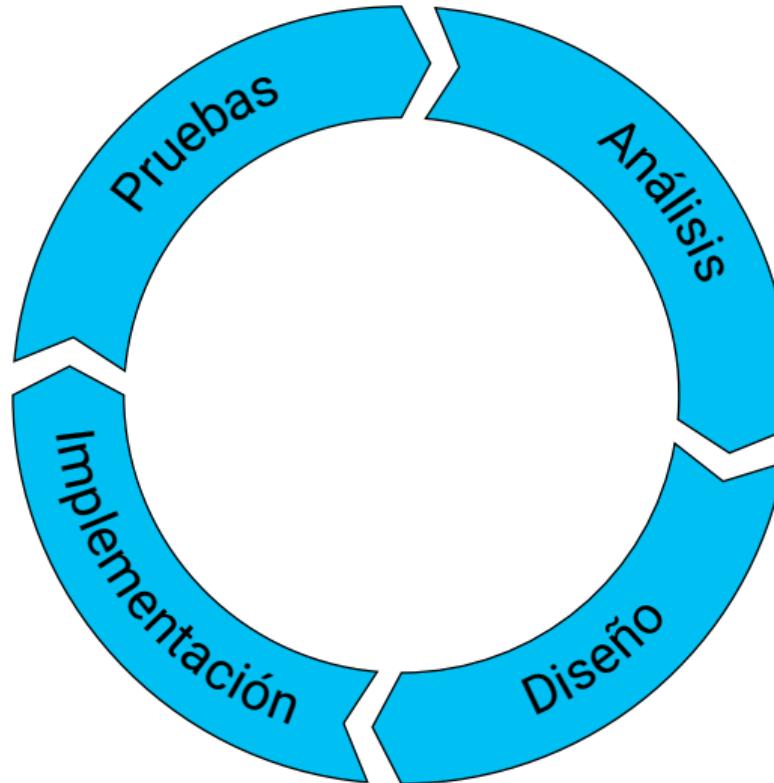


¿Cuáles son ejemplos de problemas que se pueden resolver en computación?

**Problema:** Determine si es un número es primo.

Resuelvan el problema en grupos :)

La resolución de problemas se puede ver como un ciclo



# Análisis

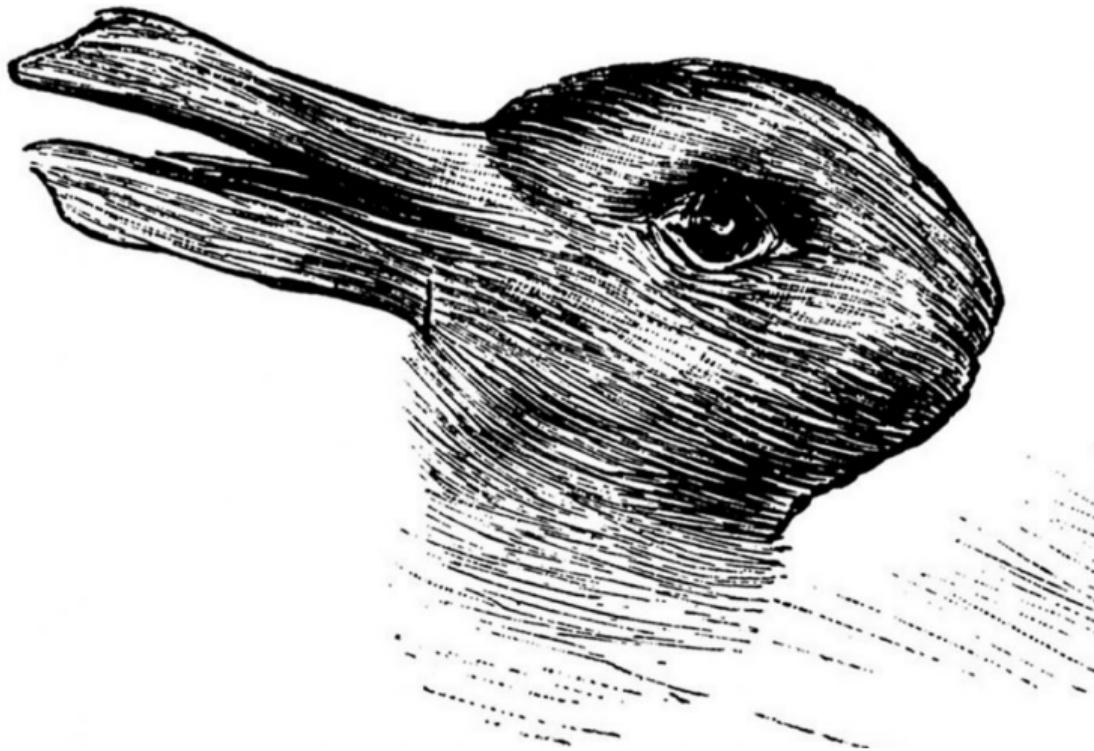
# Analizar es entender el problema



## Vocabulario

Puede ser conocido como requerimientos o ilicitación.

**Los problemas son ambiguos**



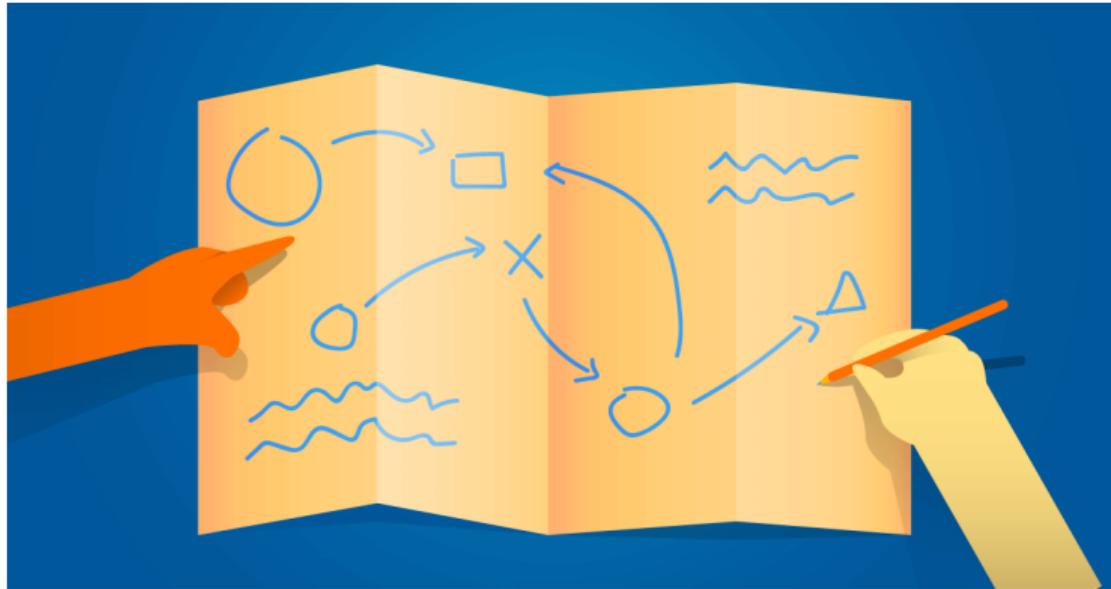
## Dado la ambigüedad, se debe...

- Obtener buena información inicial.
- Entender bien el contexto y la situación.
- Preguntar todo lo que no entienda.
- Establecer los límites del problema.



# Diseño

# Diseñar es dar una solución abstracta



## Nota

Es cuando un arquitecto crea un plano.

Existen algunas estrategias para resolucionar problemas...

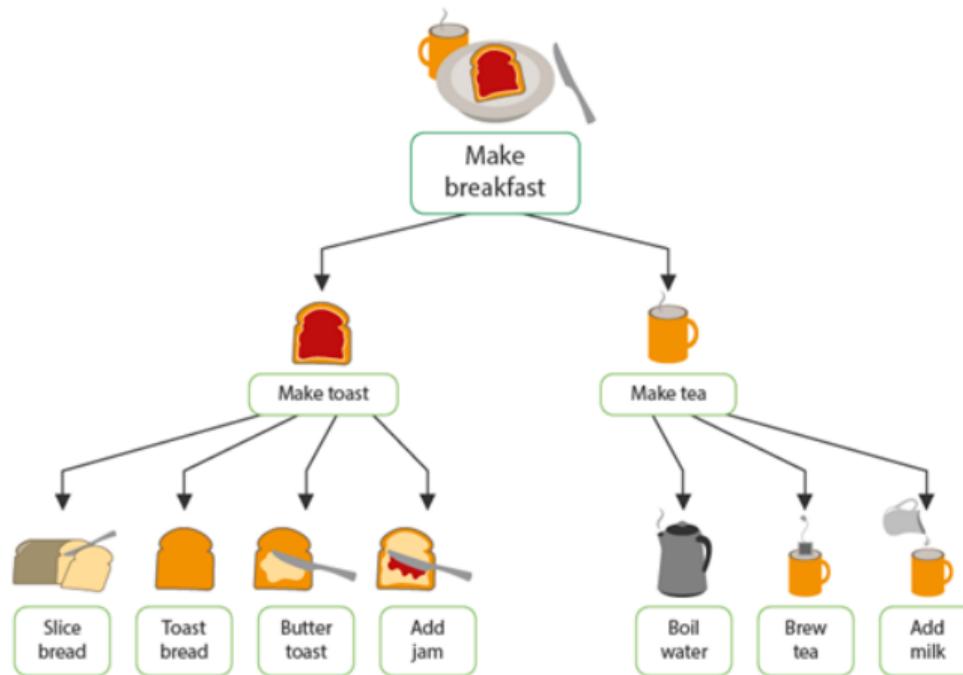
Se puede usar fuerza bruta que es intentar todas las posibles soluciones



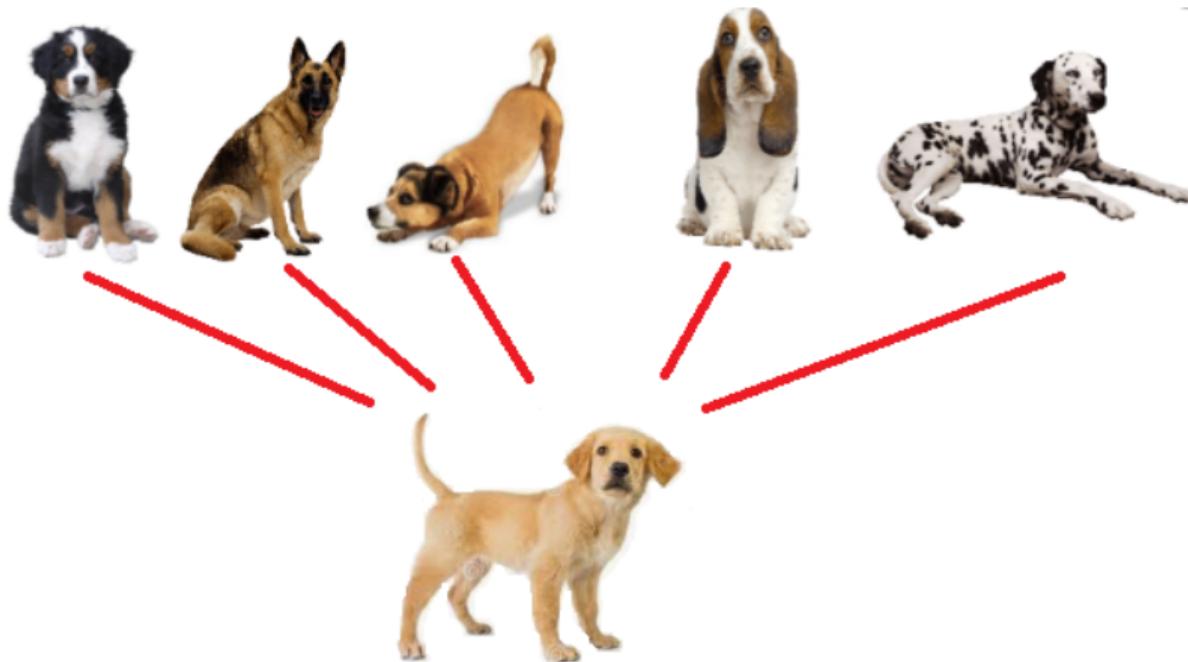
### Nota

Use la fuerza bruta para iniciar el diseño pero no como solución final.

# Se puede descomponer el problema en subproblemas



Se puede utilizar inducir utilizando varios ejemplos para determinar un patrón



Existen distintas maneras en que se puede diseñar una solución...

**Con algoritmos se puede describir una serie de pasos finitos, ordenados y claros para resolver un problema**

## Cocinar palomitas

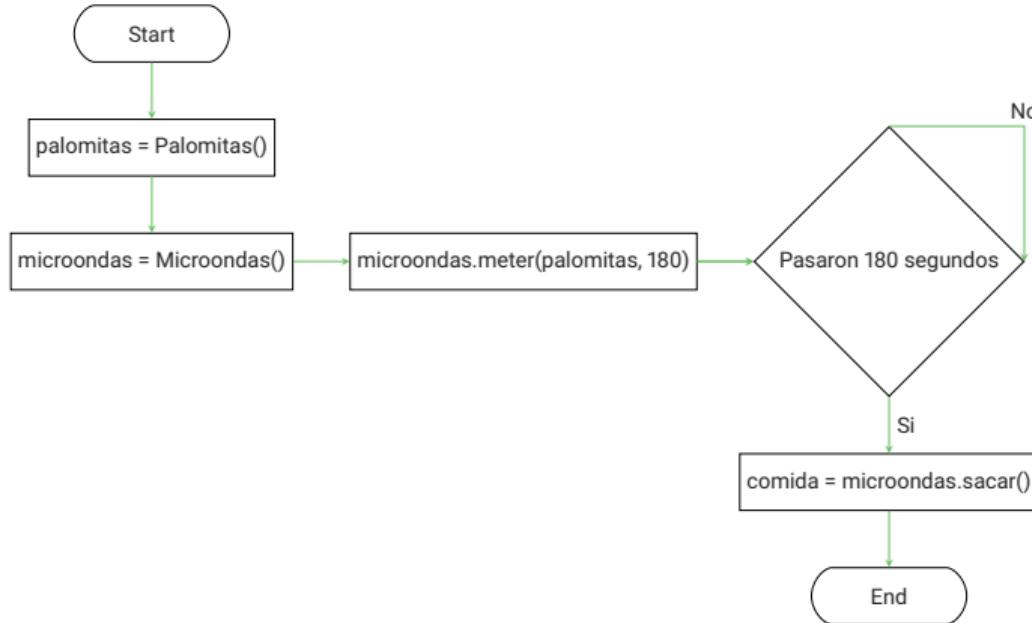
1. Quitar el plástico de la bolsas.
2. Abrir el microondas.
3. Meter la bolsa en el microondas.
4. Cerrar el microondas.
5. Poner de tiempo 3 minutos en el microondas.
6. Iniciar el microondas.
7. Esperar 3 minutos.
8. Abrir el microondas.
9. Sacar las palomitas del microondas.

**Con pseudocódigo se puede escribir algoritmos similares al lenguaje natural**

## Cocinar palomitas

```
// Los tiempos son en segundos
palomitas = Palomitas()
microondas = Microondas()
microondas.meter(palomitas, 180) // Abre, mete y cierra
tiempoEspera = 0
si tiempoEspera < 180:
    tiempoEspera += 1
    esperar(1)
comida = microondas.sacar() // Abre, mete y cierra
```

# Con los modelos se puede visualizar el diseño de un sistema



## Nota

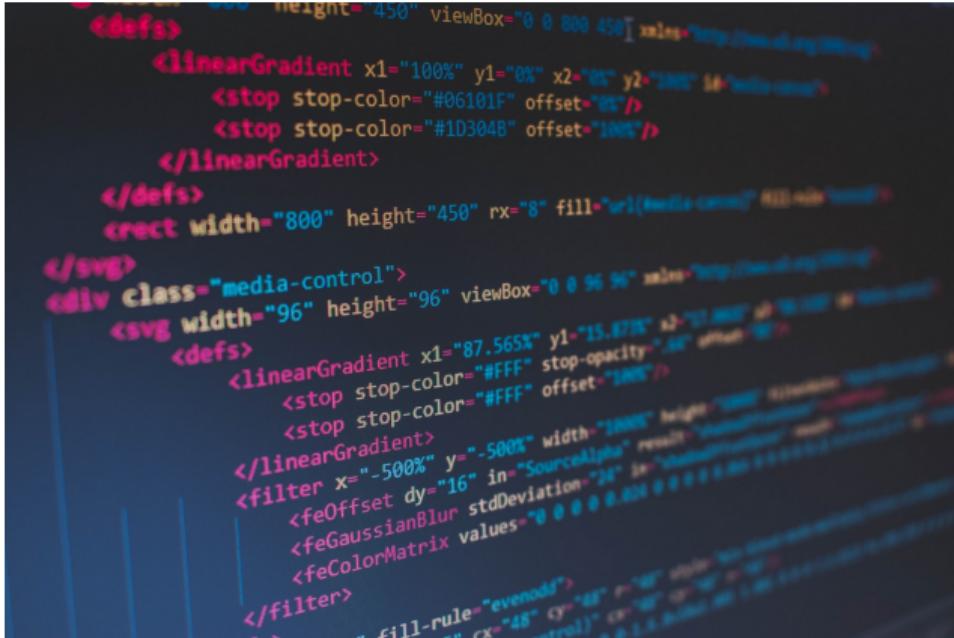
En computación los modelos UML son muy usados.

## Se deben considerar varios aspectos cuando damos una solución

- **Simpleza:** ¿Qué tan sencillo es la implementación del algoritmo?
- **Claridad:** ¿Qué tan claro es la solución?
- **Eficiencia:** ¿Cuánto trabajo debe realizar el algoritmo?
- **Generabilizabilidad:** ¿Qué tan ajustable a nuevas entradas es el algoritmo?
- Otros más avanzados como usabilidad, privacidad, mantenibilidad...

# Implementación

# Implementar es dar una solución programada con código



## Nota

Vamos a estudiar en profundidad como programar en todo el semestre

# Pruebas

# Probar involucrar verificar que funciona



## Nota

Las pruebas son parte de la solución. Si no prueba, no se puede saber si la solución funciona.

## Se prueba tomando en cuenta casos de prueba, con sus entradas y salidas

Test Case Type	Description	Test Step	Expected Result	Status
Functionality	Area should accommodate up to 20 characters	Input up to 20 characters	All 20 characters in the request should be appropriate	Pass or Fail
Security	Verify password rules are working	Create a new password in accordance with rules	The user's password will be accepted if it adheres to the rules	Pass or Fail
Usability	Ensure all links are working properly	Have users click on various links on the page	Links will take users to another web page according to the on-page URL	Pass or Fail

Ahora, ya saben como resolver  
problemas...

Ahora, ya saben como resolver  
problemas...

Resuelvan el problema anterior (de  
nuevo) en grupos :)

# FAQ (Frequently asked questions)

Q: ¿Cuántas maneras existen para resolucionar un problema?

Q: ¿Cuántas maneras existen para resolucionar un problema?  
A: ¡Infinitas!

Q: ¿Cuál es la mejor resolución de un problema?

Q: ¿Cuál es la mejor resolución de un problema?

A: Depende del contexto y la circunstancia.

Q: ¿Cómo puedo mejorar solucionando problemas?

Q: ¿Cómo puedo mejorar solucionando problemas?

A: Práctica, práctica y más práctica. También más práctica.

Q: ¿Se resuelve el problema secuencialmente?

Q: ¿Se resuelve el problema secuencialmente?

A: No. Es un proceso iterativo no secuencial.

## Referencias I

- (2012) How could i write “hello world” in binary? [Online]. Available:  
<https://stackoverflow.com/questions/10821875/how-could-i-write-hello-world-in-binary>
- (2019) Risc-v assembly language hello world. [Online]. Available:  
<https://smist08.wordpress.com/2019/09/07/risc-v-assembly-language-hello-world/>
- P. V. Roy, “Programming Paradigms for Dummies: What Every Programmer Should Know,” *New computational paradigms for computer music*, p. 39, 2009.
- (2020) The state of the 2020 octo-verse. [Online]. Available:  
<https://octoverse.github.com/#overview>
- (2021) Developer survey. [Online]. Available:  
<https://insights.stackoverflow.com/survey/2021>
- L. Villalobos, “Fundamentos de la computación,” Material del curso CI-0202, Universidad de Costa Rica, 2019.
- S. Hamer, “Clases,” Material del curso CI-0112, Universidad de Costa Rica, 2021.

## Referencias II

- C. Swaroop, *A Byte of Python*. Independent, 2020.
- J. Elkner, A. B. Downey, and C. Meyers, "How to think like a computer scientist: Learning with python," 2012.
- R. Wachenchauzer, M. Manterola, M. Curia, M. Medrano, and N. Paez, "Algoritmos y programación con lenguaje python," 2011.
- E. Themes. (2016) Circle icons computer. [Imagen]. [Online]. Available: <https://commons.wikimedia.org/wiki/File:Circle-icons-computer.svg>
- jellyfishwater. Computer hardware icon set. [Imagen]. [Online]. Available: <https://www.vecteezy.com/vector-art/159161-computer-hardware-icon-set>
- pngtree. Animation computer editor monitor software flat icon green. [Imagen]. [Online]. Available: [https://pngtree.com/freepng/animation-computer-editor-monitor-software-flat-icon--green\\_3731423.html](https://pngtree.com/freepng/animation-computer-editor-monitor-software-flat-icon--green_3731423.html)

## Referencias III

[Base image, was modified]. [Online]. Available:

<https://i.kym-cdn.com/photos/images/original/001/292/056/4e7.jpg>

U. Army, "Eniac," [Image]. [Online]. Available:

<https://commons.wikimedia.org/wiki/File:Eniac.jpg>

K. O. Coto, "Matilde: una máquina con historia propia," *Universidad de Costa Rica*, [Image]. [Online]. Available: <https://www.ucr.ac.cr/noticias/2019/01/15/matilde-una-maquina-con-historia-propria.html>

"Can't automate what you don't understand," [Image]. [Online]. Available:

<https://cdn.sysaid.com/wp-content/uploads/2021/06/cant-automate.png>

"Duck and rabbit," [Image]. [Online]. Available:

<https://img.ti-media.net/wp/uploads/sites/46/2016/02/imagemain-920x613.jpg>

"Clarity grow your businessg," [Image]. [Online]. Available: <https://executiveleader.com/wp-content/uploads/2020/08/Clarity-growth-small-copy-862x610.jpg>

## Referencias IV

[Image]. [Online]. Available:

<https://images.unsplash.com/photo-1542831371-29b0f74f9713?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=1950&q=80>

[Image]. [Online]. Available:

<https://static-cms.hotjar.com/images/website-usability-testing.width-1500.png>

[Image]. [Online]. Available: <https://rockcontent.com/es/wp-content/uploads/sites/3/2021/02/plan-de-marketing-1024x538.png.webp>

[Image]. [Online]. Available:

<https://soundcloud.com/sspacelordmusic/game-over-try-again>

[Image]. [Online]. Available:

<https://www.barefootcomputing.org/concepts-and-approaches/decomposition>

[Image]. [Online]. Available:

<https://www.parasoft.com/blog/how-to-write-test-cases-for-software-examples-tutorial/>

## Referencias V

[Image]. [Online]. Available:

[https://miro.medium.com/max/1296/1\\*FDy196B8EFhUnyFbGcIEXA.png1](https://miro.medium.com/max/1296/1*FDy196B8EFhUnyFbGcIEXA.png1)