



IPC



En un rato arrancamos



Encuentro 1




A series of five horizontal bars of varying lengths and shades of gray and red, stacked vertically on the left side of the slide.

Introducción al pensamiento computacional

Presentación de la Materia





Programar es ejercitar la
paciencia

Plan para Hoy



Organización de la Asignatura



El ecosistema de trabajo



¿Qué es un lenguaje de programación?



¿Qué debe cumplir un algoritmo?



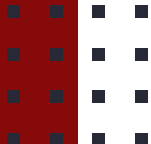
Implementación de un algoritmo



Cierre de Debate



**Base de
programaciòn**



Criterios de aprobación



Desafíos
semanales



Trabajos
Prácticos



Examen



Presentismo

Trabajo práctico Integrador 1

Desarrollar un sistema simple

Tomando los conocimientos adquiridos desarrollamos un programa

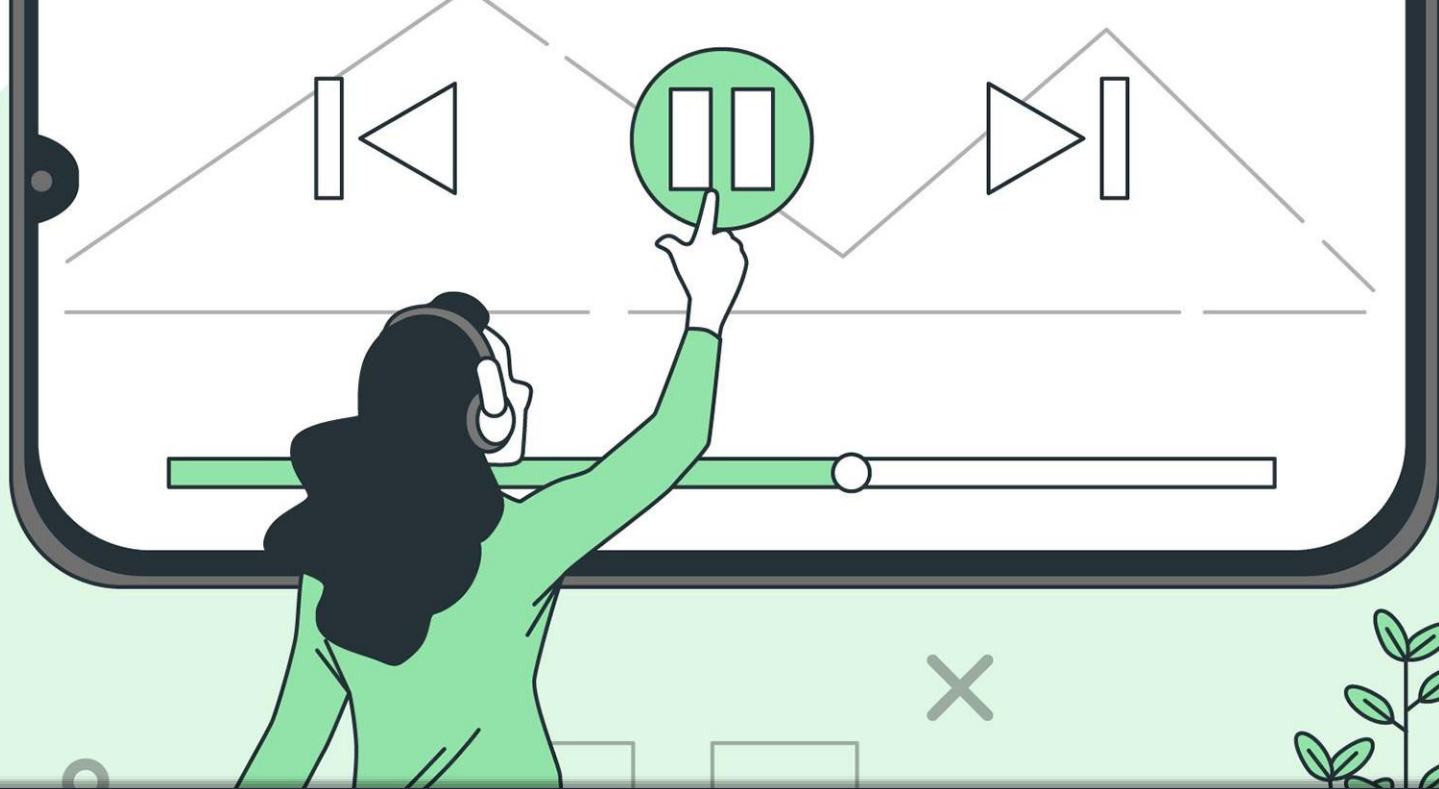


Trabajo práctico integrador 2

Análisis de un dataset

Importamos un dataset o un Excel para hacer un análisis de sus contenidos





Receso 30'



¿Qué es el lenguaje?








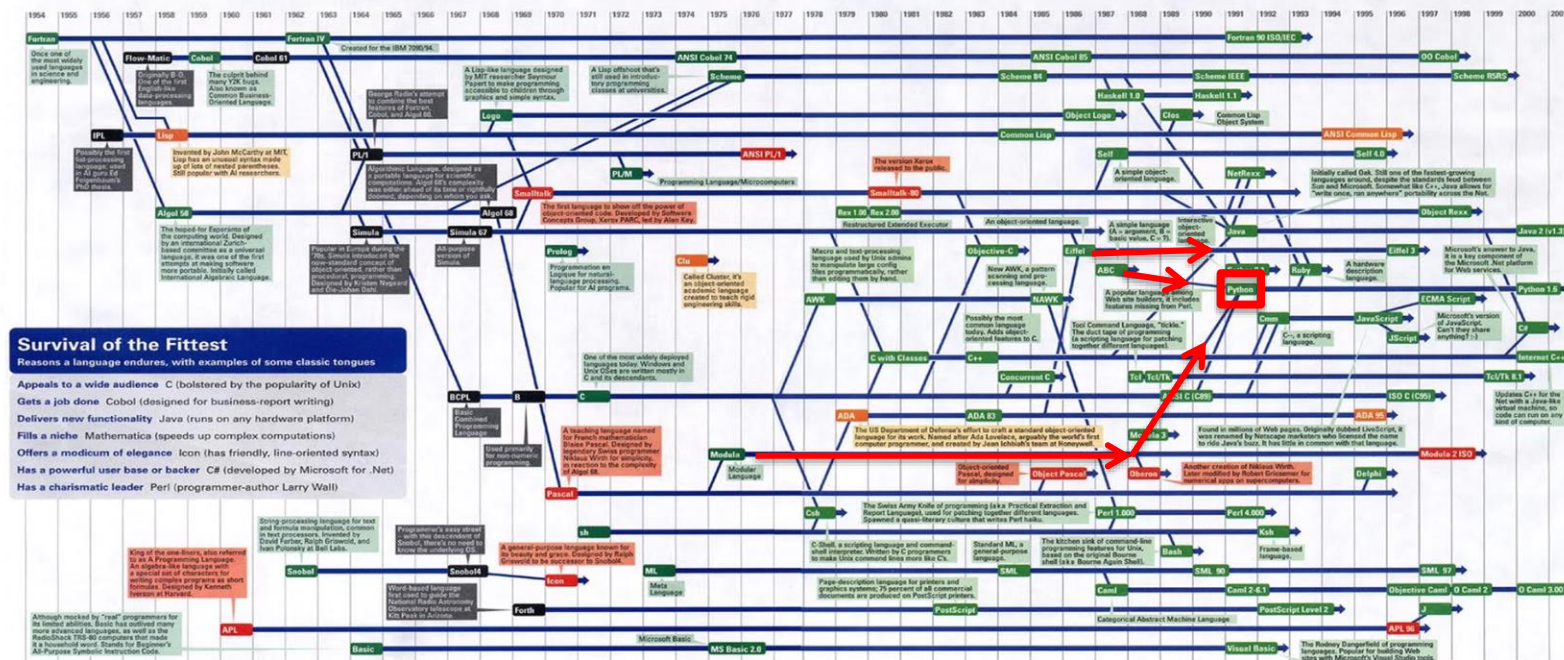
Tracing the roots of computer languages through the ages

An ad hoc collection of engineers – electronic lexicographers, if you will – aim to save, or at least document, the lingo of classic software. They're combing the globe's 9 million developers in search of coders still fluent in these nearly forgotten lingua francas. Among the most endangered are Ada, APL, B (the predecessor of C), Lisp, Oberon, Smalltalk, and Simula.

Code-racer Grady Booch, Rational Software's chief scientist, is working with the Computer History Museum in Silicon Valley to record and, in some cases, maintain languages by writing new compilers so our ever-changing hardware can grok the code. Why bother? They tell us about the state of software practice, the minds of their inventors, and the technical, social and economic forces that shaped history in the code. Booch explains: "They'll provide the raw material for software archaeologists, historians, and developers to use to work out the evolution of what we do today. It's a little like the way we preserve the strongest branches of programming's family tree. For a nicely exhaustive rundown, check out the Language List at www.informatik.uni-freiburg.de/Java/misc/lang_list.html. — Michael Menduno

Key

- 1954 **Year Introduced**
-  **Active:** thousands of users
-  **Protected:** taught at universities; compilers available
-  **Endangered:** usage dropping off
-  **Extinct:** no known active users or up-to-date compilers
-  **Lineage continues**



Sources: Paul Boutin; Brent Hailpern, associate director of computer science at IBM Research; The Retrocomputing Museum

```
for object to mirror_mod.mirror_object:  
    operation == "MIRROR_X":  
        mirror_mod.use_x = True  
        mirror_mod.use_y = False  
        mirror_mod.use_z = False  
    operation == "MIRROR_Y":  
        mirror_mod.use_x = False  
        mirror_mod.use_y = True  
        mirror_mod.use_z = False  
    operation == "MIRROR_Z":  
        mirror_mod.use_x = False  
        mirror_mod.use_y = False  
        mirror_mod.use_z = True
```

```
#selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.name))  
mirror_ob.select = 0  
bpy.context.selected_objects  
data.objects[one.name].select  
print("please select exactly one mirror")
```

--- OPERATOR CLASSES ---

```
types.Operator):  
    X mirror to the selected  
    object.mirror_mirror_x"  
    mirror X"
```

Clasificar un lenguaje de programación

Tipos de lenguajes de programación

- Nivel de complejidad
- Palabras más amigables
- Utilidad

Clasificación





Lenguajes cercanos al programador

Lenguajes de nivel intermedio

```
Void main(){  
    Printf("Hola Mundo")  
}
```



Bajo nivel Lenguaje assembler

MOV AX, @data

MOV DS, AX

MOV DX,offset cadena1

MOV AH,9

INT 21h

```
> pd
0x00000000 90 nop
0x00000001 90 nop
0x00000002 6800009c00 push 0x9c0000 ;
0x00000007 e8c7ace37b call 0x7be3acd3
0x7be3acd3(unk)
0x0000000c bb04009c00 mov ebx, 0x9c0000
0x00000011 8903 mov [ebx], eax
0x00000013 e81903f47b call 0x7bf40331
0x7bf40331()
0x00000018 bb08009c00 mov ebx, 0x9c0000
0x0000001d 8903 mov [ebx], eax
0x0000001f bb00009c00 mov ebx, 0x9c0000
0x00000024 c60300 mov byte [ebx],
0x00000027 68e8030000 push 0x3e8 ; 0x0
0x0000002c e81124e37b call 0x7be32442
0x7be32442(unk)
0x00000031 ebf4 jmp 0x100000027
0x00000033 90 nop
0x00000034 ff invalid
0x00000035 ff invalid
0x00000036 ff invalid
0x00000037 ff invalid
```

Lenguaje máquina

0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0,
0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1,
0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0,
1, 0, 0, 0, 1, 1, 1, 1

4E7C5A 1C4E 4E4F 7C1C 9E 1C4E4D
5A6D5E9E4C 4F9B7C 9E1C 9B4C5E
9B 1C3A4E9B 3A 7F3A 6D4C 7F
5C6C7C4C7C7A 7A 9C 9B 4D 5B
4D4E7A 4C9B9B3B 9B 6C4C9E5A7A5C5E5A
5B5A5A5A 5E 3A4C4C7A 5B7A9B 5E
5E 7A3B 7C5E 1C5B 6D 5A1C
1C5E4C 5E7C9B 9B5B4C 4E 5B
4D5B1C5B 4C7C 5E4C1C1C5A4C4E 5E6C
1C4C4E7C5E4C 9B 4D5B9B5E 1C
9B 4E7A 1C7A7C9E 5A5B9B4E7C 7A
3B 5E 4D 4D6D 1C9B 9B
7A 5A4C3A 9E1C4E3B4E 9E7A7F
7F 7C 9E4E1C9E9B 7F 4C9B4F9B 3B
5C5B6D5A9E 6D7F 9E 3B 7C 4C4C4E4E
7F3A 7F9B4C 4C5E4C 9B 7F 6D3B7F
3A 5A7F 5A5E7A4D 7A 7C7F
3B 6D 4E 1C7A1C 3B 6C 4E 5C5B4F
3B 9B4C 1C 7A9B5A4E9E 7C1C6D9B
1C9B3B 4F 9B9B6D9B 3B6D4C3A
6D5A5E 5E7C 3B3B5B 7A6C 5E
1C5E4C5B 3A4D4F5A4F5E 4E 3B
4C7A5E9B9B4D4E 7F6C 1C 7F 3B 5B
9C4C7C 4C7C9B 7F 4C9E
6D4C 7F3B4C 9B5B6D9B3A 9E4F 4E
4C 7C 3B5B5B4D3B 9E9C7A4C 4C4C7C
3B 5A 4F6C5E4E9B9E4F9C7F 3B7A4F 5A
7A4C 5E4C 5A9E 9E 6D3B9B 4E 6C
1C7F 3A 6D 9E3A 4E 5C9B3B7A
7F1C9E 9C 5B6C4C4C1C4C4E5E3B 4C
4C4C 3B4E7A3B 3A1C5E 7A3B
9E 4E 7F1C5B9B 7C 6C 5B
3B 5E 5A1C4F4F 1C6D9E 9B
7C3A7F4C5E6C 5A 5A 9B 4F 4C4C5B
3B 4D4F 5B6C 4C7F6D 6C 5E 3A

Costo de convertir



Interpretados



Compilados

Paradigmas

- Estructurado
- Orientado a Objetos
- Funcional

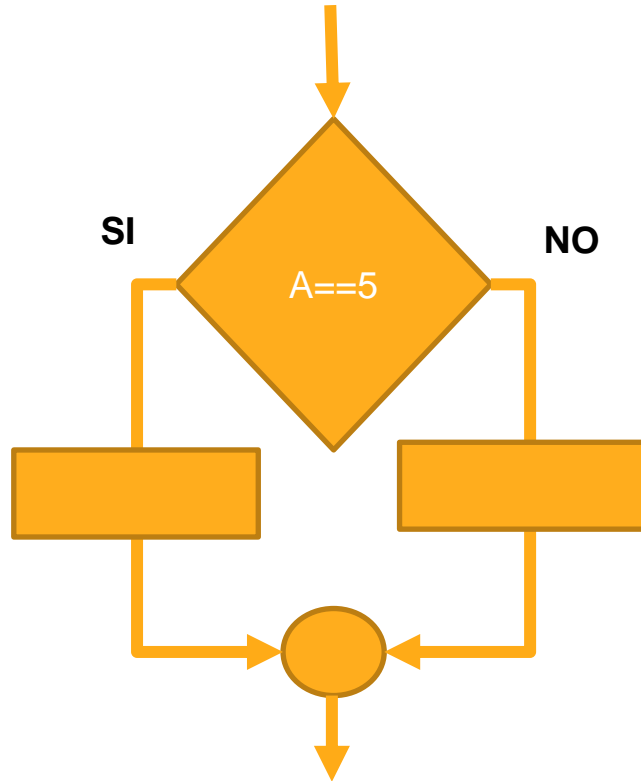


¿Qué tiene un lenguaje de programación?



1. Tipos de instrucciones
 1. Secuenciales
 2. Condicionales
 3. De Repetición

¿Qué es un ALGORITMO ?



Qué debe respetar un ALGORITMO

1. **PRECISO** (Orden)
2. **DEFINIDO** (Resultado)
3. **FINITO**





ALGORITMO “Hacer del 2”

1. Caminar hacia el baño
2. Abrir la puerta
3. Sacarse los pantalones o cualquier vestimenta que tenga puesta
4. Sentarse en el inodoro
5. Hacer fuerza
6. Usar el papel Higiénico
7. Vestirse
8. Salir del baño



ALGORITMO “Hacer del 2” FALLIDO

1. Caminar hacia el baño
2. Abrir la puerta
3. Hacer fuerza
4. Scarse los pantalones o cualquier vestimenta que tenga puesta
5. Sentarse en el inodoro
6. Vestirse
7. Salir del baño
8. Usar el papel Higiénico

Que necesita un algoritmo

- Se debe describir en tres partes
 1. ENTRADA
 2. PROCESO
 3. SALIDA

Técnicas algorítmicas

- Enumeración exhaustiva
- Algoritmo de aproximación



Enumeración exhaustiva

Enumeración exhaustiva – Fuerza Bruta



Inicio

1- conozco mi posición

2- Mi nueva posición es menor al destino

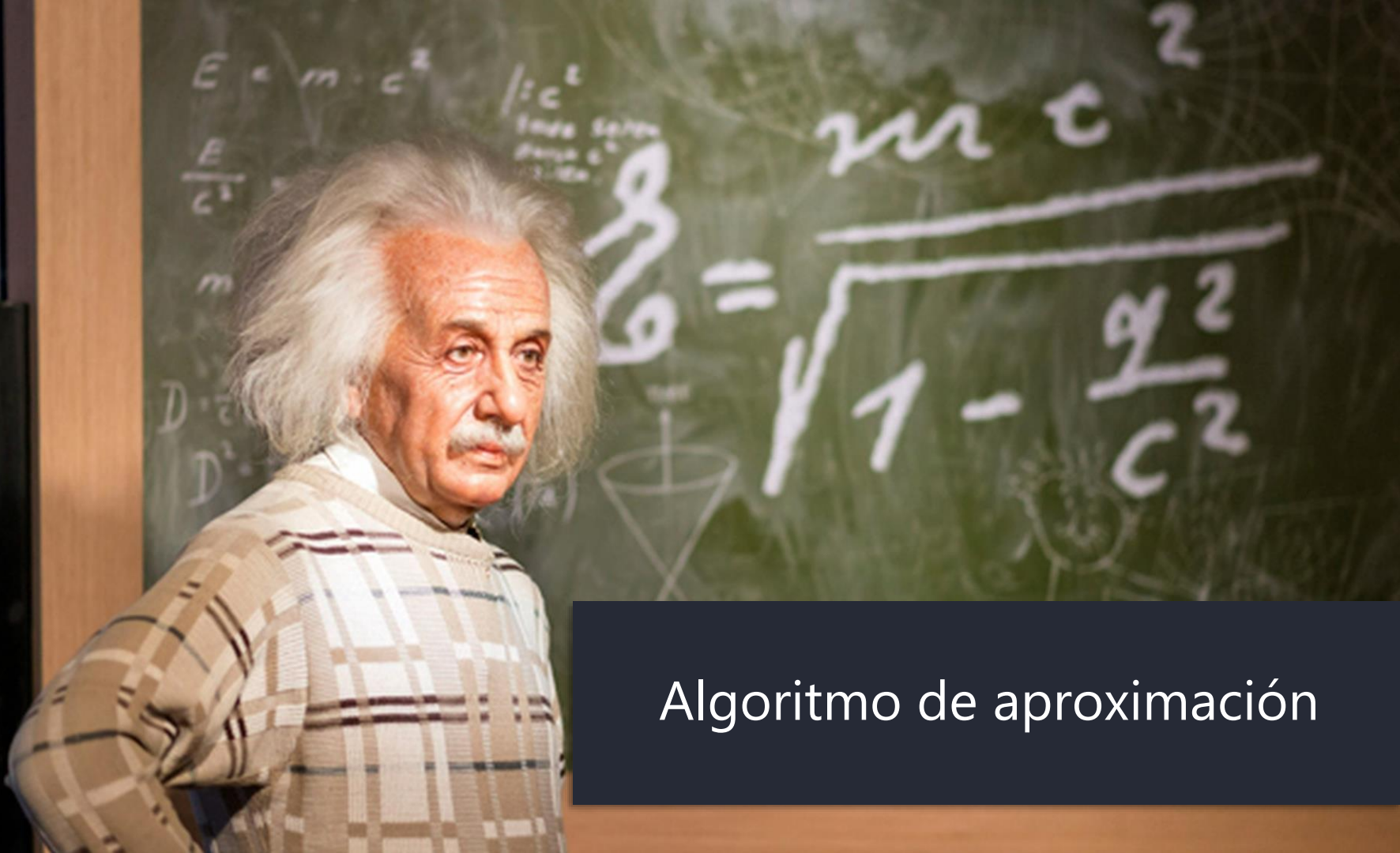
3- sí, sumo otro pie

4- Vuelvo al paso 2

5- mi nueva posición es igual a destino

si- La distancia en pies es exacta

no-la cantidad de pies no es exacto



Algoritmo de aproximación

Algoritmo de aproximación - Heurística

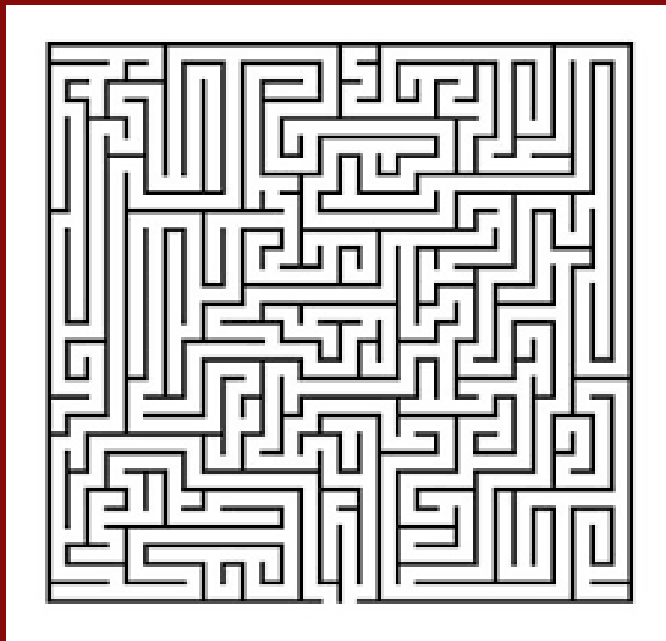


Aproximamos con un
margen de error



Ejercicio para pensar

Resolver un algoritmo que permita a un ser humano de inteligencia media completar un laberinto con éxito





DEBATE

¿Preguntas?

