

# Pensamiento Algorítmico

The background is a grayscale abstract composition. It features a large, light-colored circular shape on the right side, which partially overlaps a piano keyboard. The keyboard's keys are visible as a series of parallel lines. The overall aesthetic is modern and artistic, with a focus on geometric forms and light and shadow.

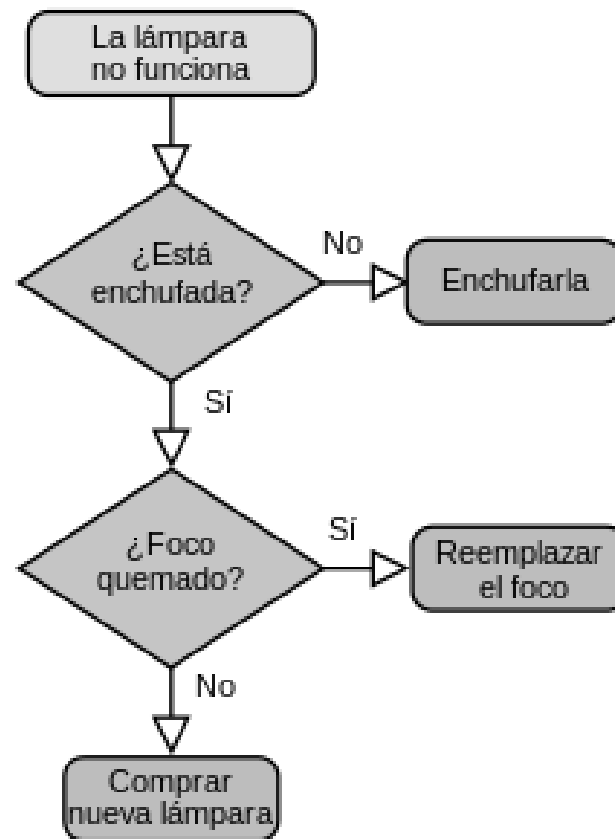
Karly Sandoval

# ¿Qué es el pensamiento algorítmico?

- Es una forma sistemática de resolver problemas
- Se necesita llegar de un punto A a un punto B sin tomar atajos ni asumir detalles.
- Todo debe estar escrito de forma explícita

# ¿Qué es un algoritmo?

- Secuencia de instrucciones definidas e implementables en pasos



# Pasos en Abstracto

- Relacionar la solución al problema a una variable, cantidad o resultado cuantitativo
- Descomponer el problema en partes más pequeñas
- Identificar variables necesarias en cada parte
- Obtener herramientas para conseguir las variables necesarias
- Implementar herramientas y solucionar cada parte
- Combinar todas las anteriores

# Ejemplo: Cocinar

- Resultado: Omelette
- Descripción Tangible: 3 Huevos batidos, cocinados uniformemente a temperatura baja con sal y pimienta.

## Pasos:

- Comprar huevos, sal y pimienta
- Batir huevos
- Calentar sartén
- Agregar huevos a la sartén
- Esperar a que cocine un lado
- Doblar tortilla dos veces
- Esperar a que cocine
- Agregar sal y pimienta

## Herramientas necesarias:

- Dinero
- Bowl, batidora, huevos
- Sartén, hornilla
- Huevos batidos, sartén caliente
- Tiempo
- Espátula
- Tiempo
- Sal, pimienta, omelette

# Ejercicio: Esqueleto del juego piedra, papel o tijera

- Importar librerías
- Asignar un número a cada objeto del juego
- Crear una variable a ser llenada por el usuario
- Crear una variable que genere un número aleatorio del 0 al 2
- Establecer condiciones para ganar, empate y perder

# Ejercicio: Esqueleto del juego piedra, papel o tijera

- Piedra=0
- Papel=1
- Tijeras=2
- Usuario=1
- Comp=Randint (0,3)

Crear condiciones “if” para empate, ganar y perder

Ejercicio: Esqueleto del juego piedra,  
papel o tijera



# Ejercicio: Esqueleto del juego piedra, papel o tijera

## Condiciones elegidas:

- Empate:  $\text{Comp} == \text{Usuario}$
- Perder:  $\text{Comp} - \text{Usuario} == 1$
- Ganar: Todo lo demás

# Ejercicio: Esqueleto del juego piedra, papel o tijera

```
In [1]: import numpy as np
...:
...:
...: """ Piedra=0
...:      Papel=1
...:      Tijeras=2"""
...: |
...: #Para jugar, elige una mano y escribe el numero correspondiente
...: Usuario=1
...:
...: print("Usuario=%i"%(Usuario))
...: Comp=np.random.randint(0,3)
...: print("Comp=%i"%(Comp))
...:
...: if Usuario==Comp:
...:     print("Resultado = Empate")
...: else:
...:     if (Comp-Usuario)==1:
...:         print("Resultado = Perdido")
...:     else:
...:         print ("Resultado = Gano")
Usuario=1
Comp=0
Resultado = Gano
```

# Ejemplo: Escribir la matriz $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

## Pasos:

- Instalar librerías
- Crear espacio a llenar
- Crear un contador para recorrer cada casilla de la matriz
- Crear condición tal que  $i=j$  implique que el valor sea igual a 1
- Activar la condición en cada punto de la matriz usando el contador

## Herramientas

- “import”, “numpy” -> “np”
- “np”, “zeros” -> Variable M
- “numpy”, “range” -> Variables “x”, “y”
- “if”, “for”, “M”, “i”, “j”, “x”, “y” -> Nueva matriz “M”

# Código para activar condiciones en cada punto de la matriz

- $M = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$

- $x = (0, 1)$

- $y = (0, 1)$

- Usar contadores  
“i” y “j”



# Errores más comunes

- Mezclar minúsculas y mayúsculas
- Tabulaciones
- Signos de puntuación
- No especificar ciertas condiciones
- Errores de lógica

# Ejemplos

```
17 if Usuario==Comp:
18     print("Resultado = Empate")
19 else:
20     if (Comp-Usuario)==1:
21         print("Resultado = P")
22     else:
23         print("Resultado = ")
```

Code analysis

invalid syntax

```
16
17 if usuario==Comp:
18     print("Resultado = Empate")
19 else:
20     if (Comp-usuario)==1:
21         print("Resultado = Pe")
22     else:
23         print("Resultado = ")
```

Code analysis

undefined name 'usuario'

```
17 if Usuario==Comp
18     print("Resultado = Empate")
19 else:
20     if (Comp-Usuario)==1:
21         print("Resultado = Perdio")
22     else:
23         print("Resultado = ")
```

Code analysis

invalid syntax

```
In [7]: x=1.9999234
...: y=1.9999586
...:
...:
...: if x-y==0:
...:     print("time=")
...: else:
...:     print("error de logica")
error de logica
```

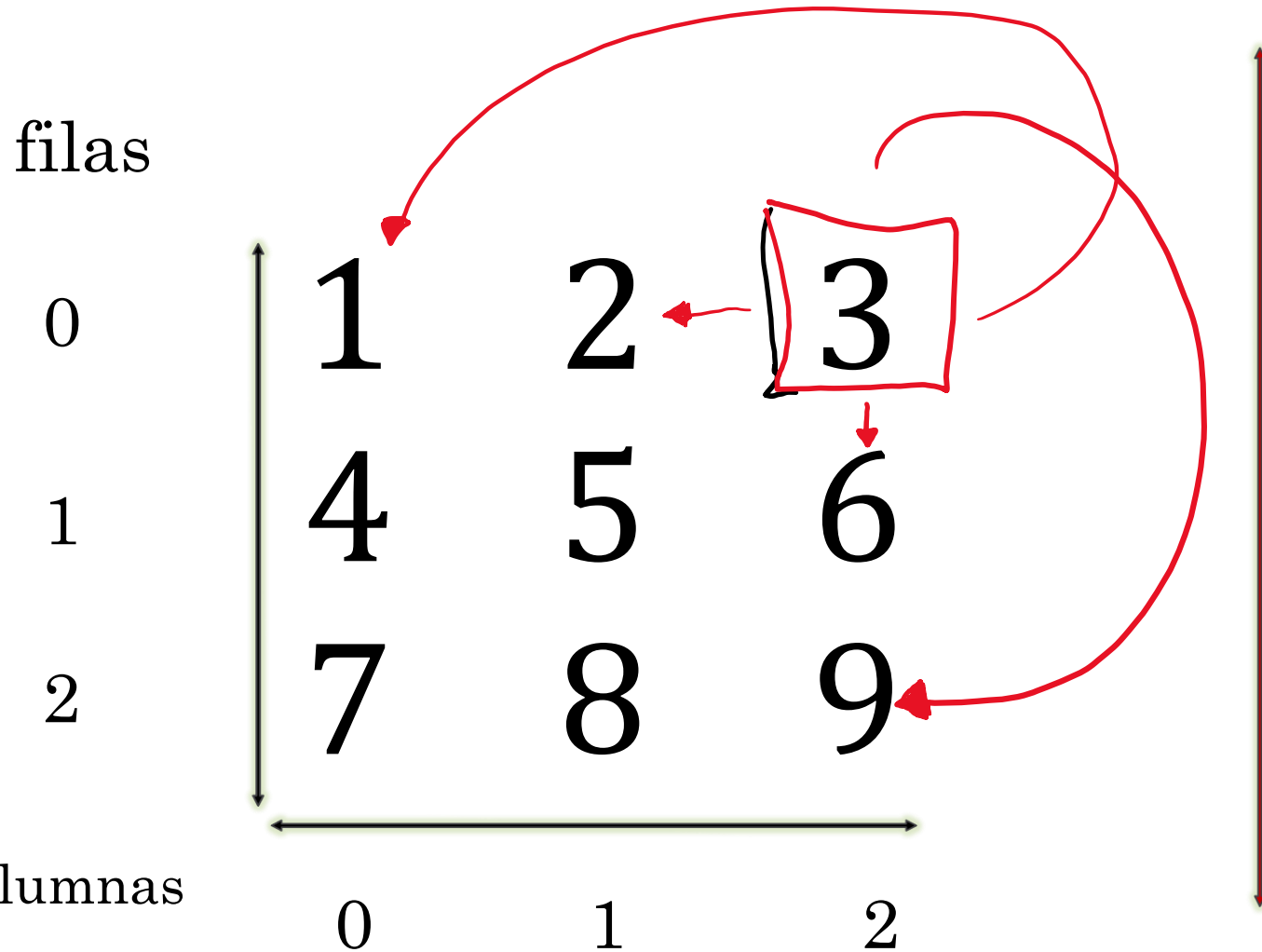
```
In [8]: x-y
Out[8]: -3.520000000012402e-05
```

```
Out[8]: -3.520000000012402e-05
```

# Ejercicio

- Crear un código que me muestre el número más pequeño que Python tiene

# Coordenadas de borde periódicas



## Vecinos

- Arriba=  $-1 \bmod 3 = 2$
- Abajo=  $1 \bmod 3 = 1$
- Izquierda=  $1 \bmod 3 = 1$
- Derecha=  $3 \bmod 3 = 0$



# Ejemplo Final

Propagar un incendio con las siguientes condiciones:

- El bosque comienza sano
- Un árbol se incendia
- Por cada árbol incendiado, se incendian sus vecinos
- Condiciones de borde periódicas

# Ejemplo Final