

# Cache 13



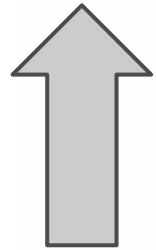
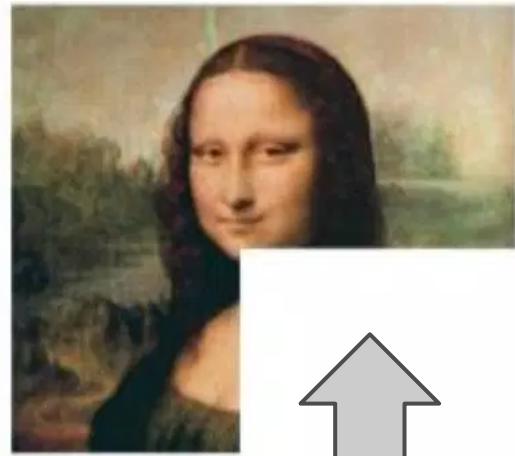
<http://faq.utn.so/enunciado-ppt>

# ¿Qué es Cache 13?

Sistema que emula algunas características de un Sistema Operativo.

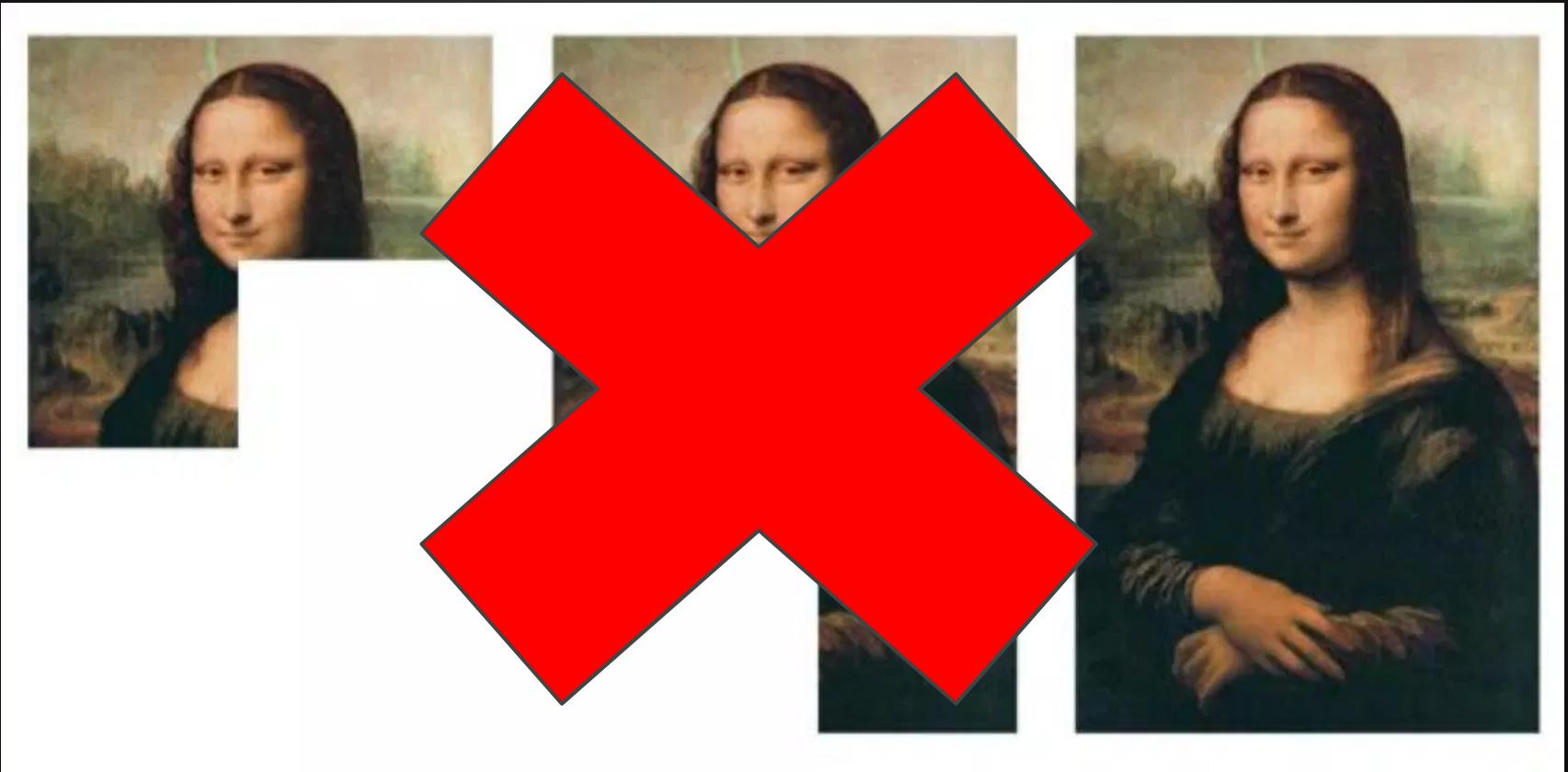
¿Cómo vamos a  
trabajar en esto?

# ¿Desarrollo incremental?

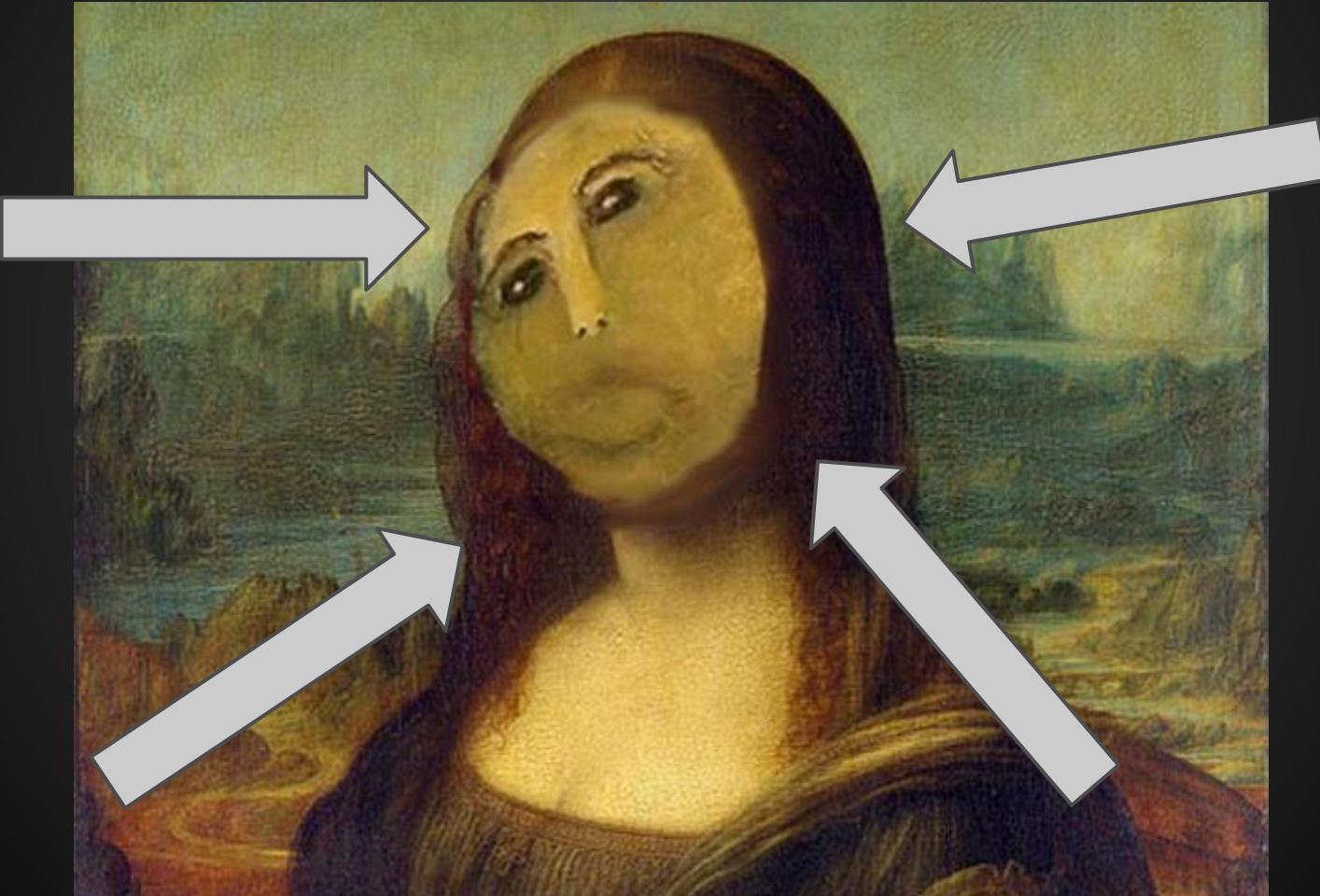


Yo hago mi parte

# ¿Desarrollo incremental?



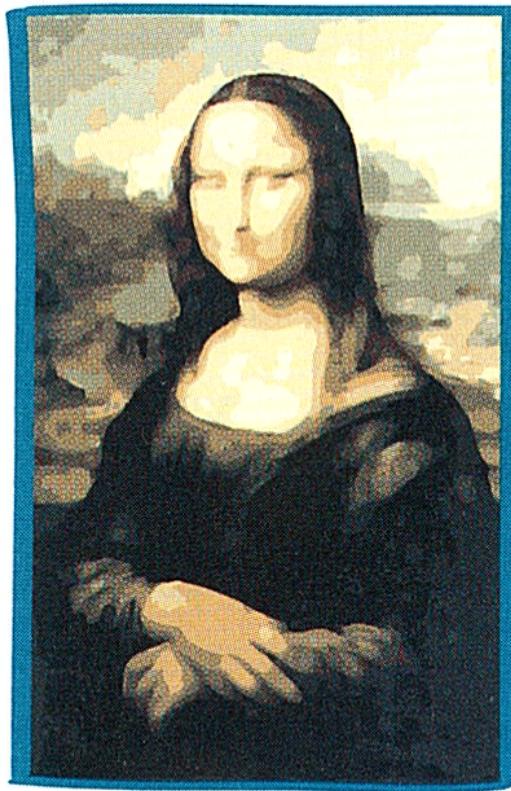
# “Yo hice mi parte”



# ¡Desarrollo iterativo incremental!



# ¡Desarrollo iterativo incremental!



A Los ayudantes de Sistemas Operativos y 350 personas más les gusta esto.



Escribe un comentario...



# Arquitectura de Cache 13

```
graph LR; Planificador[Planificador] --> CPU[CPU]; AdmSwap[Adm. Swap] --> AdmMemoria[Adm. Memoria]; CPU --> AdmMemoria
```

**Planificador**

**Adm. Swap**

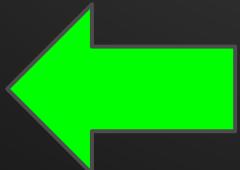
**CPU**

**Adm.  
Memoria**

# ¿Cómo corro mis programas?



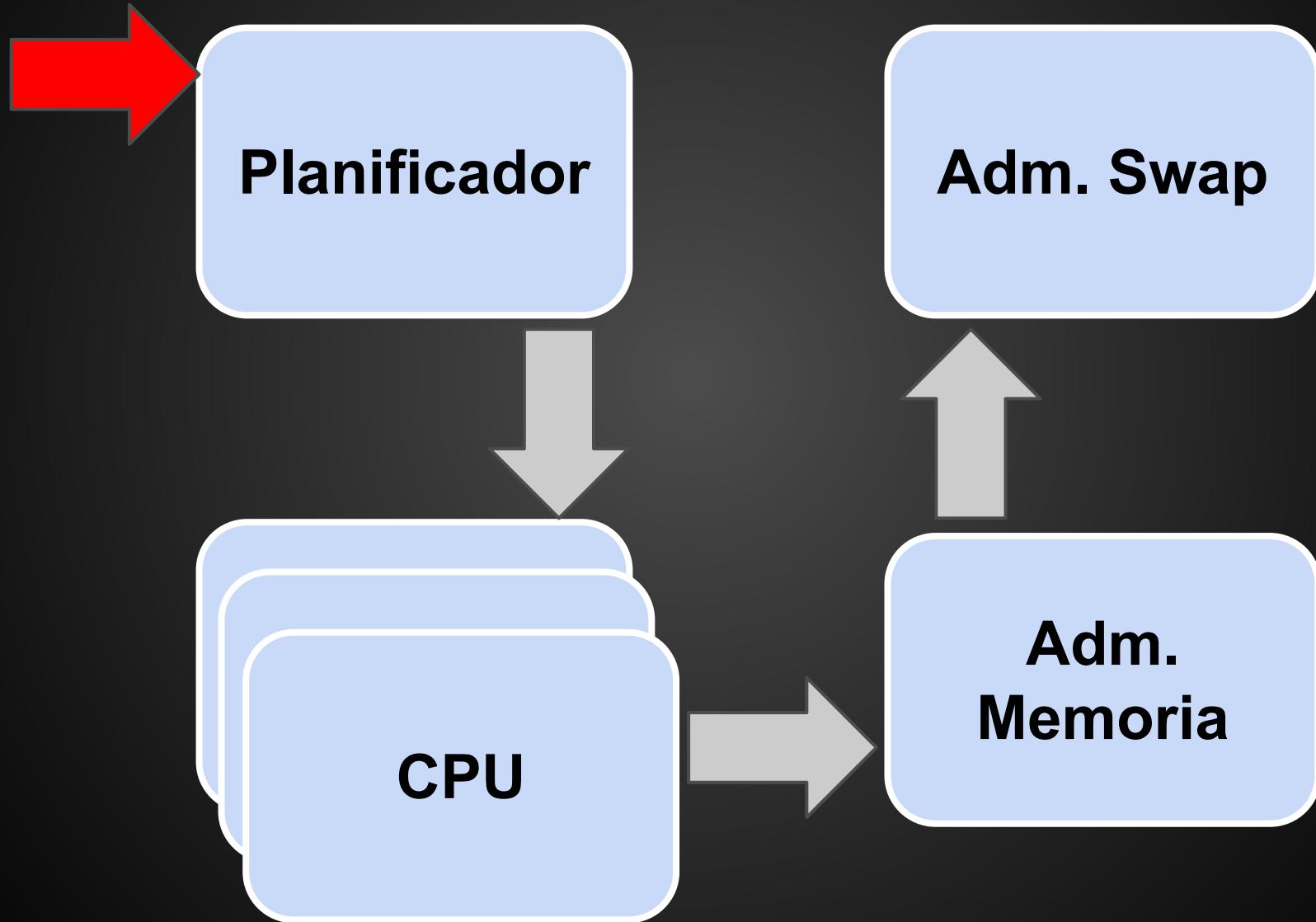
Mi pedido



La respuesta

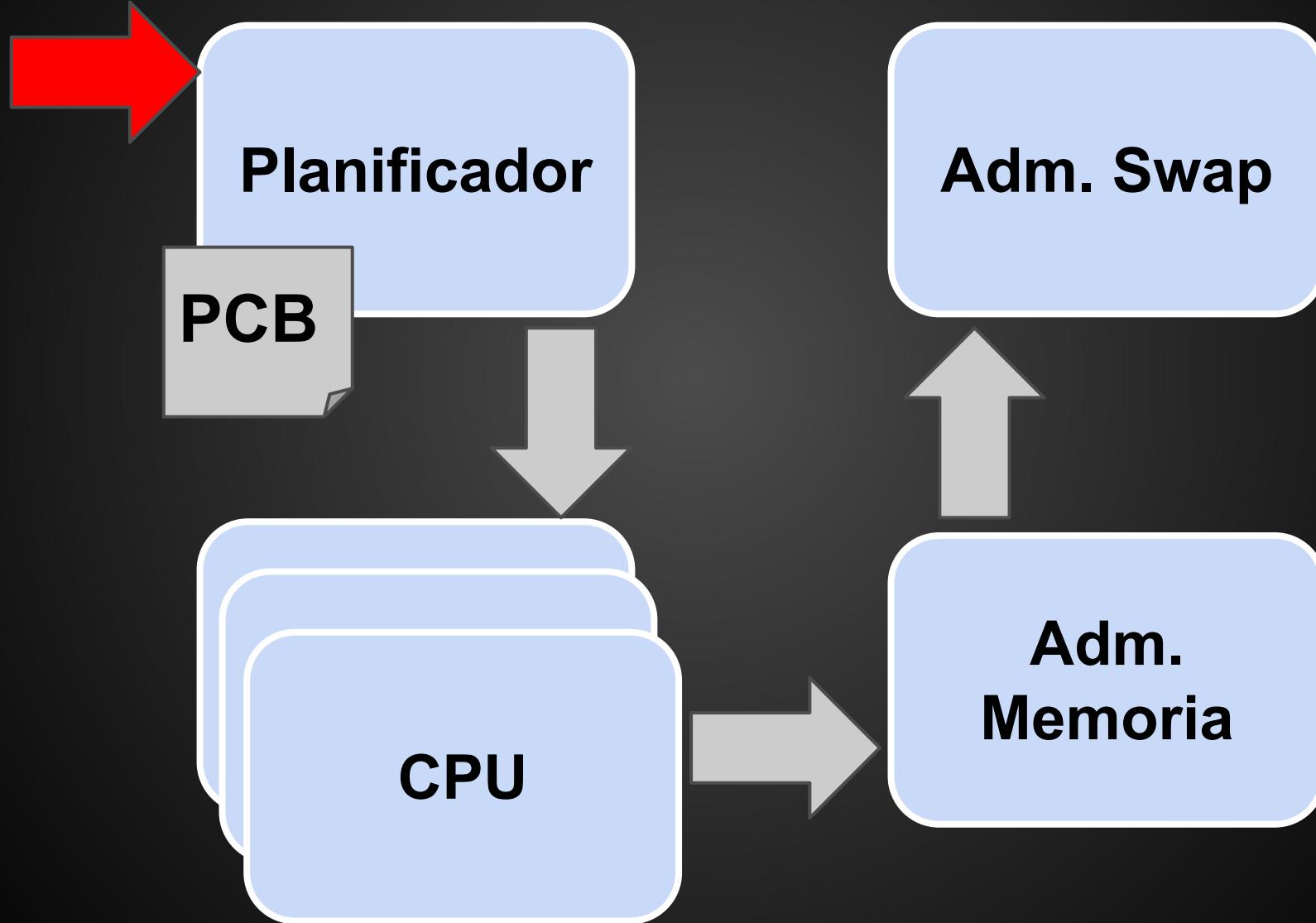
# De mCod a mProc

correr programa.cod



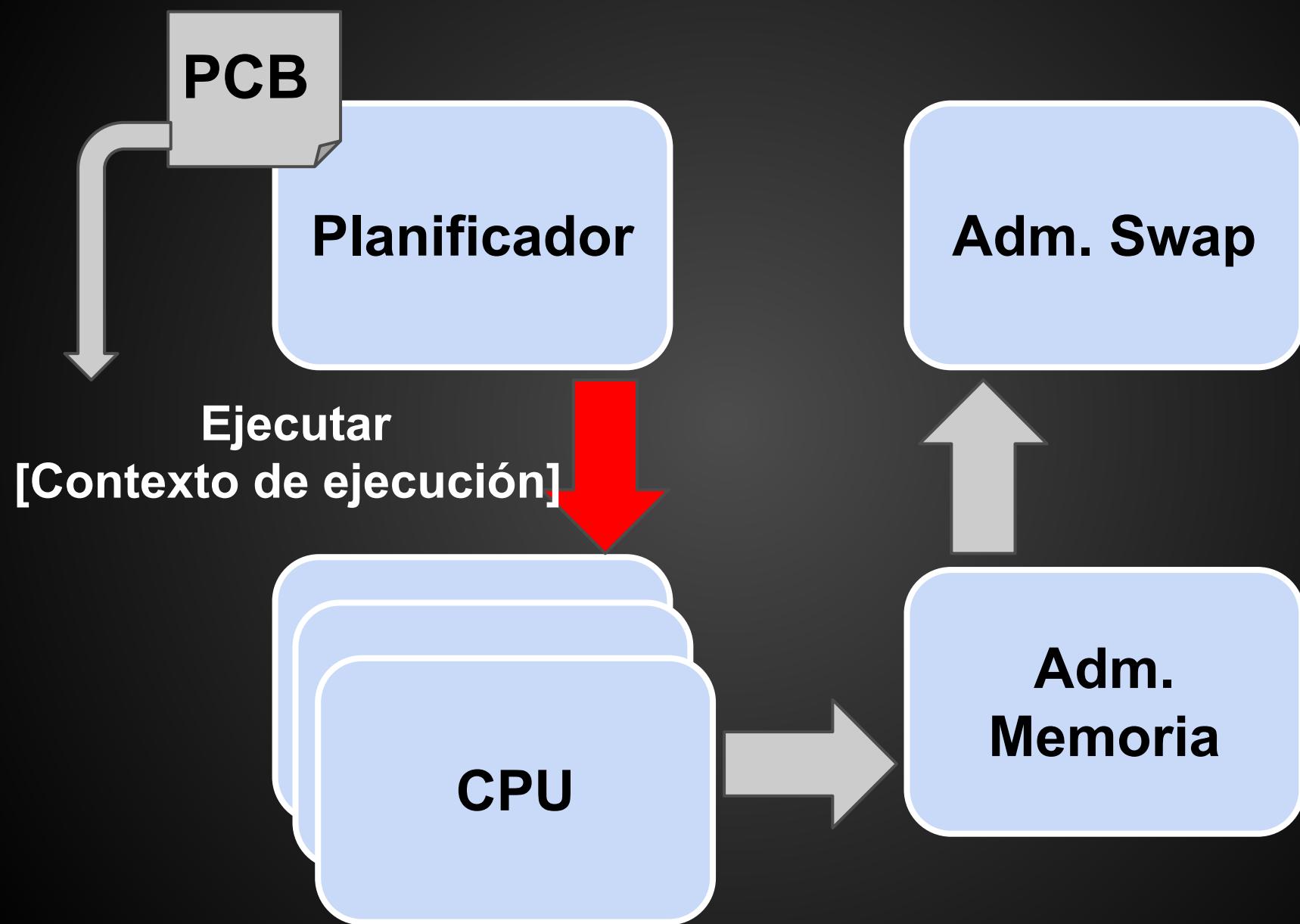
# De mCod a mProc

correr programa.cod

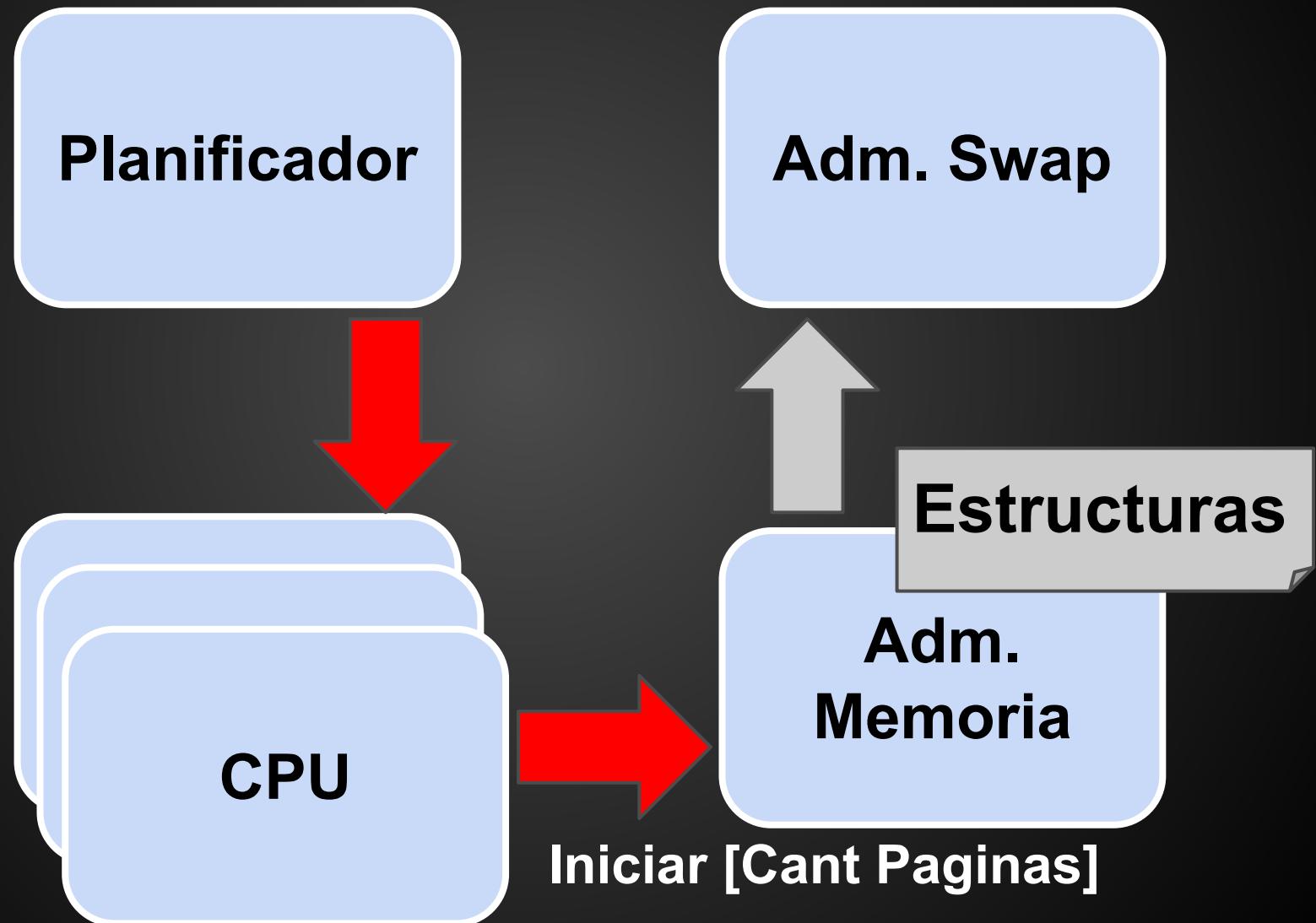


¿Qué sentencias  
puede tener mi  
programa?

# Sentencias: iniciar



# Sentencias: iniciar



# Sentencias: iniciar

Estructuras

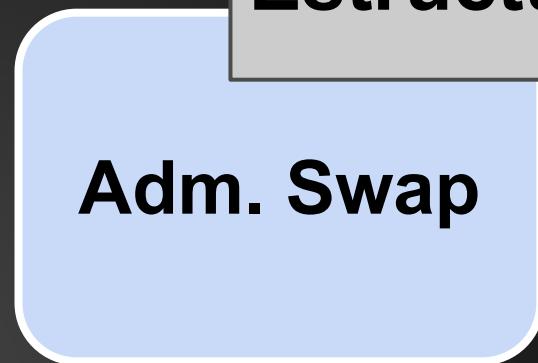
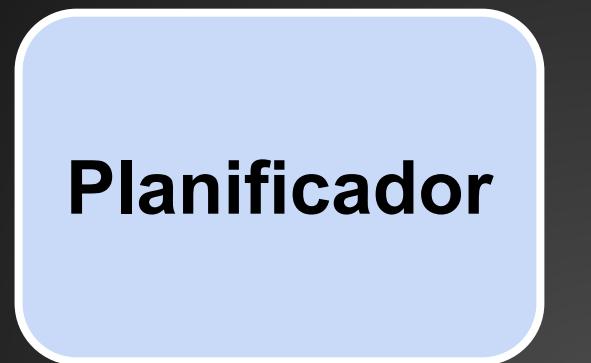
Planificador

Adm. Swap

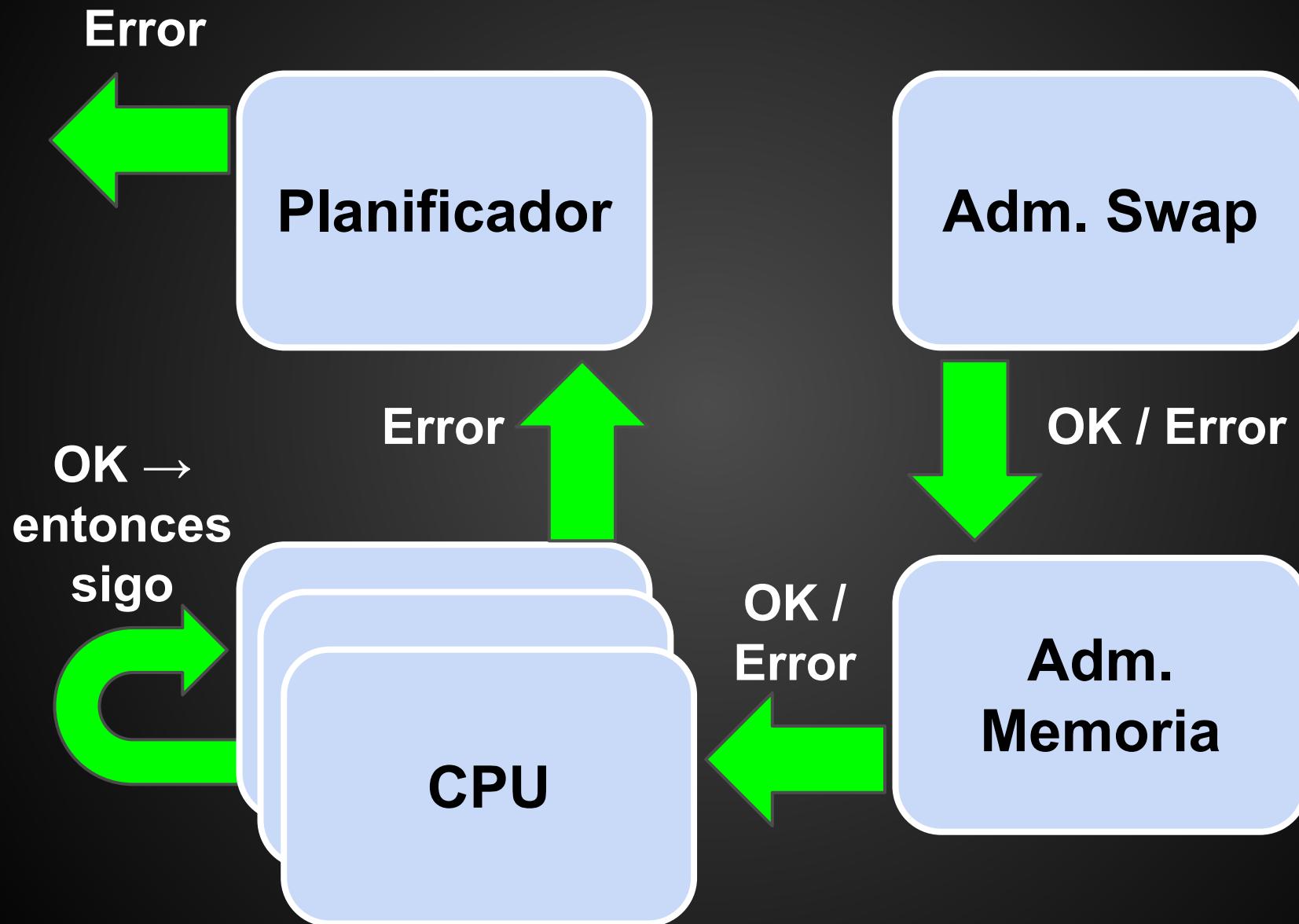
Iniciar [Cant  
Paginas]

Adm.  
Memoria

CPU

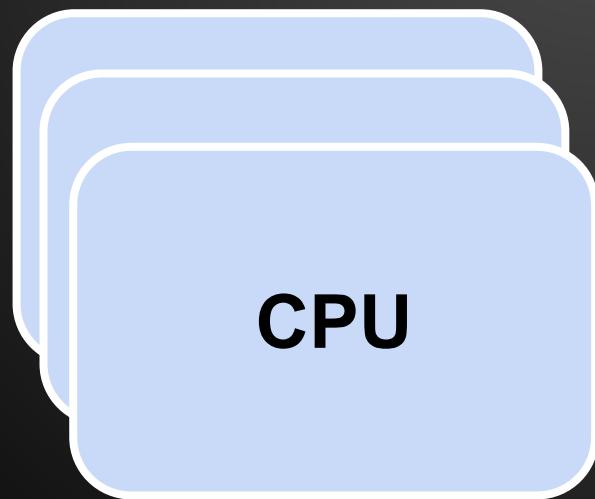


# Sentencias: iniciar

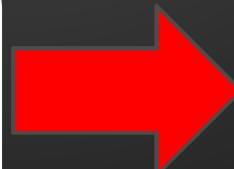


# Sentencias: escribir

El planificador  
no interviene



escribir  
“Hola”



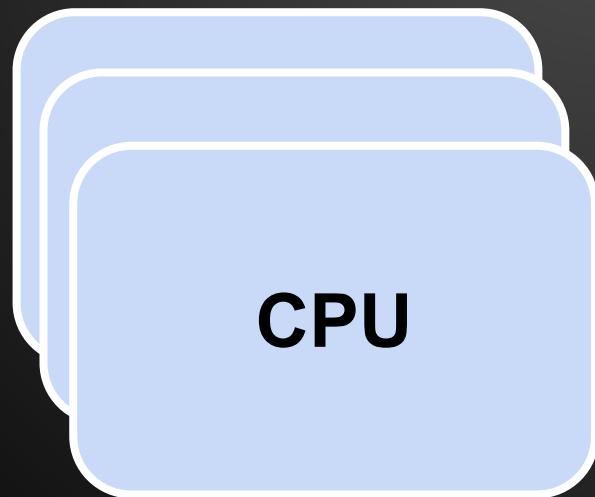
Adm. Swap



Adm.  
Memoria

# Sentencias: escribir

El planificador  
no interviene



Adm. Swap

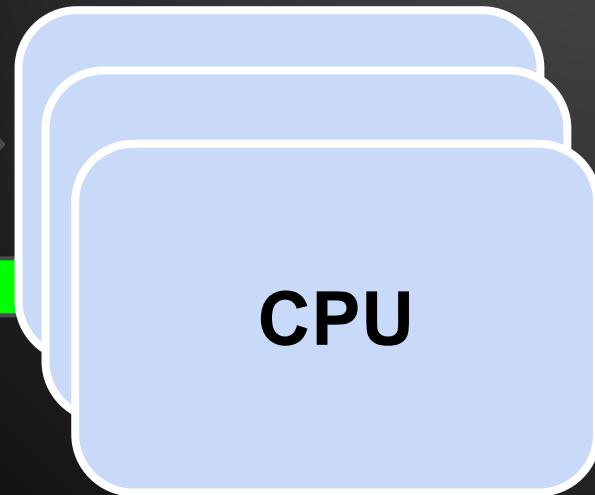
escribir “Hola”  
[ si es necesario ]

Adm.  
Memoria

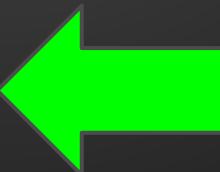
# Sentencias: escribir

El planificador  
no interviene

OK →  
entonces  
sigo



OK



Adm. Swap

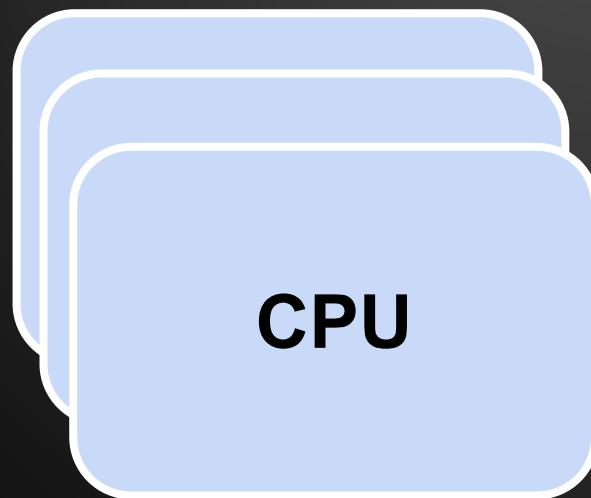


OK → [si fue  
necesario]

Adm.  
Memoria

# Sentencias: leer

El planificador  
no interviene

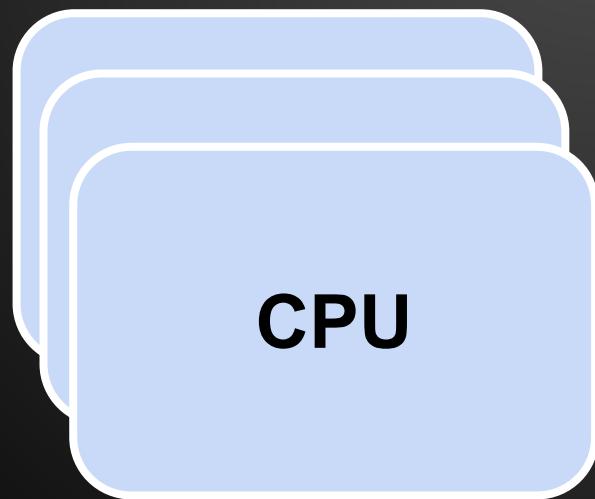


Adm. Swap

Adm.  
Memoria

# Sentencias: leer

El planificador  
no interviene



Adm. Swap

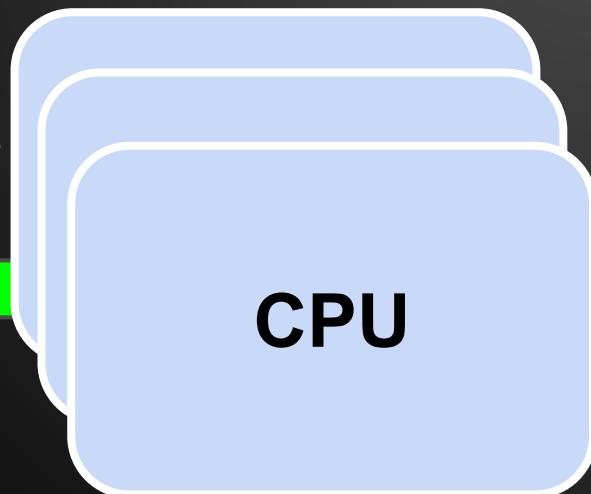
leer 13 → [ si es  
necesario ]

Adm.  
Memoria

# Sentencias: leer

El planificador  
no interviene

OK →  
entonces  
sigo



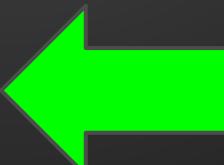
“Hola”

Adm. Swap

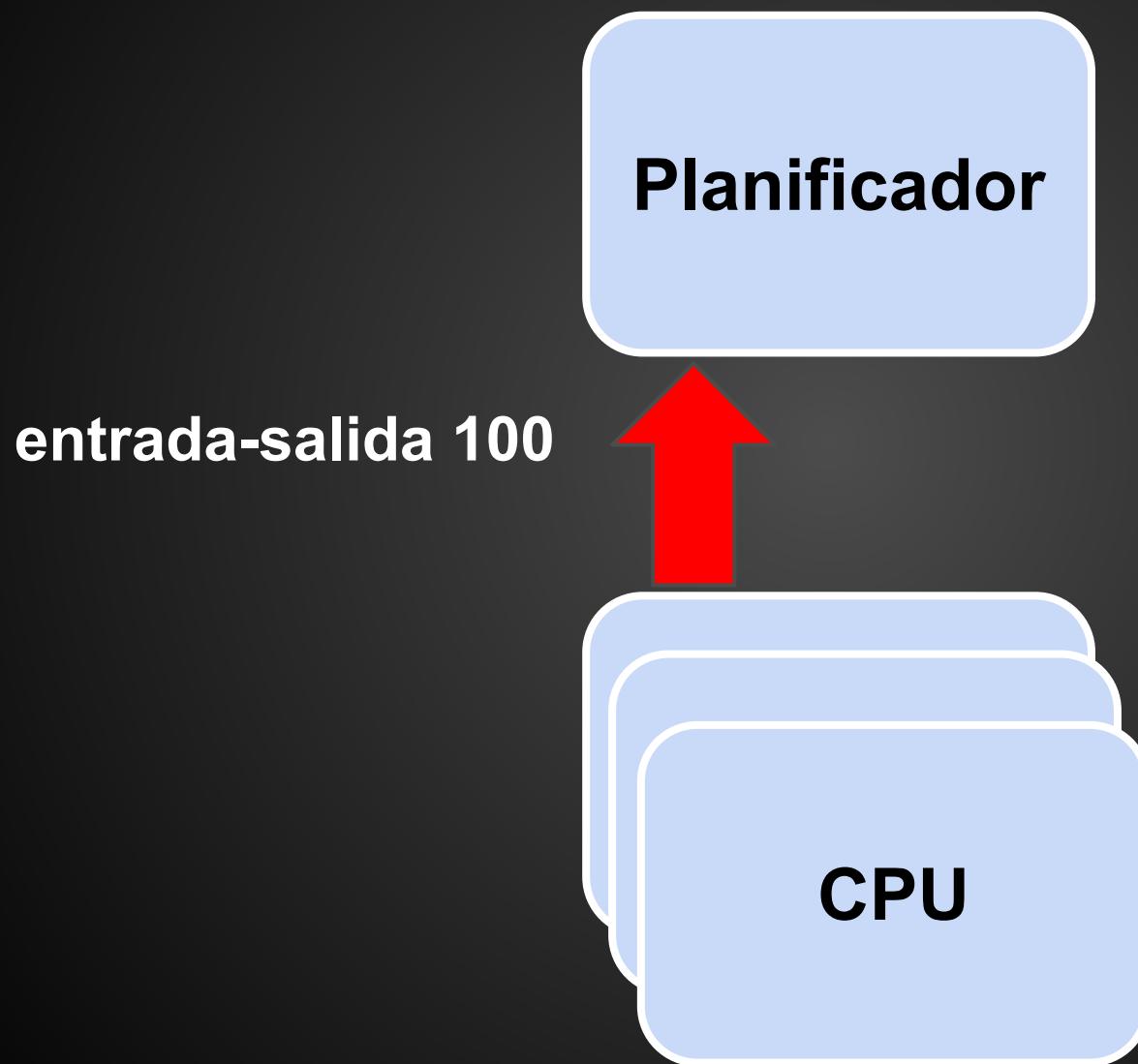


“Hola” → [Si  
fue necesario]

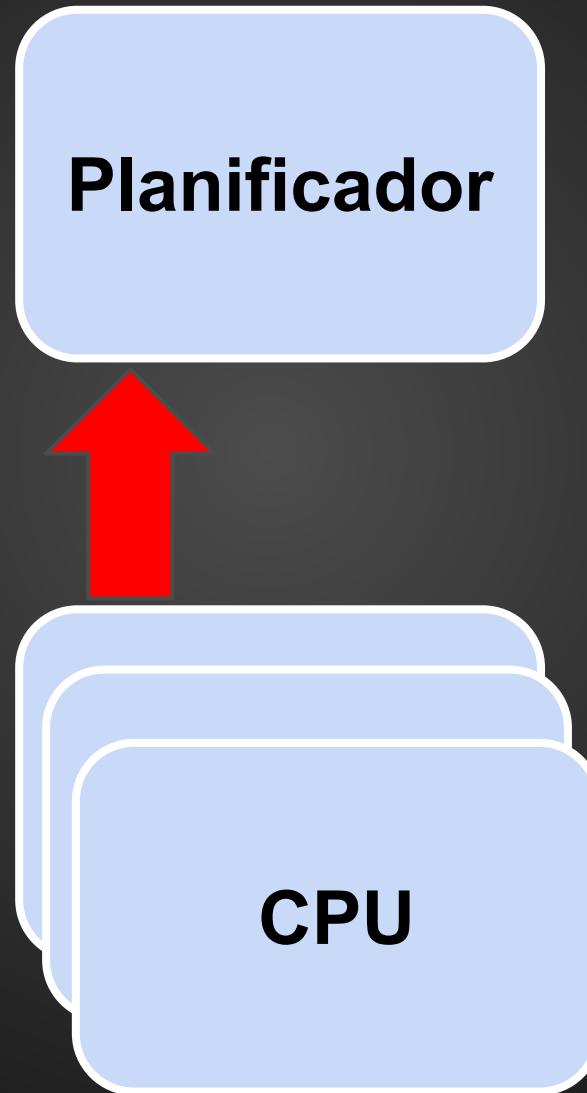
Adm.  
Memoria



# Sentencias: entrada-salida

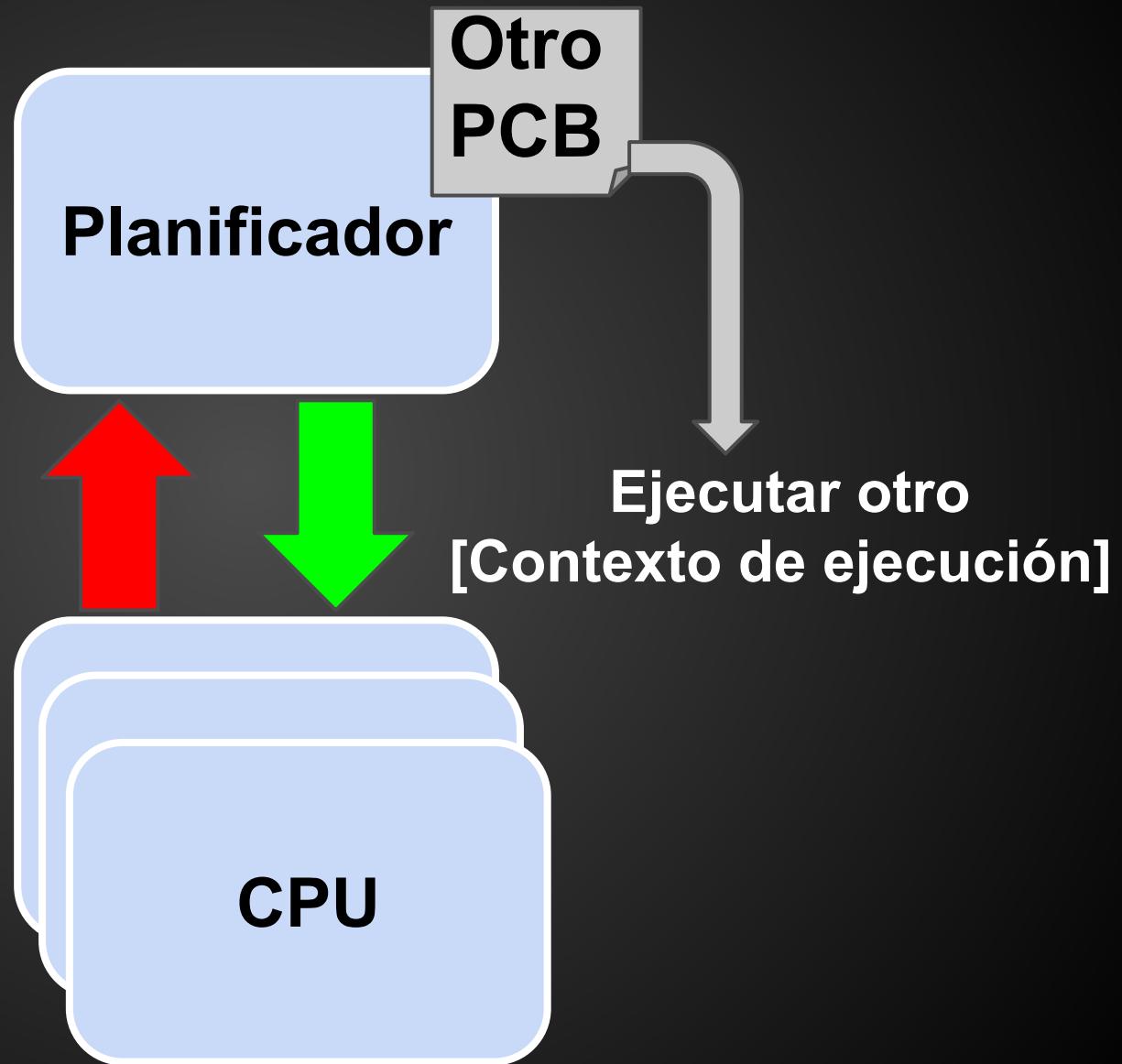


# Sentencias: entrada-salida



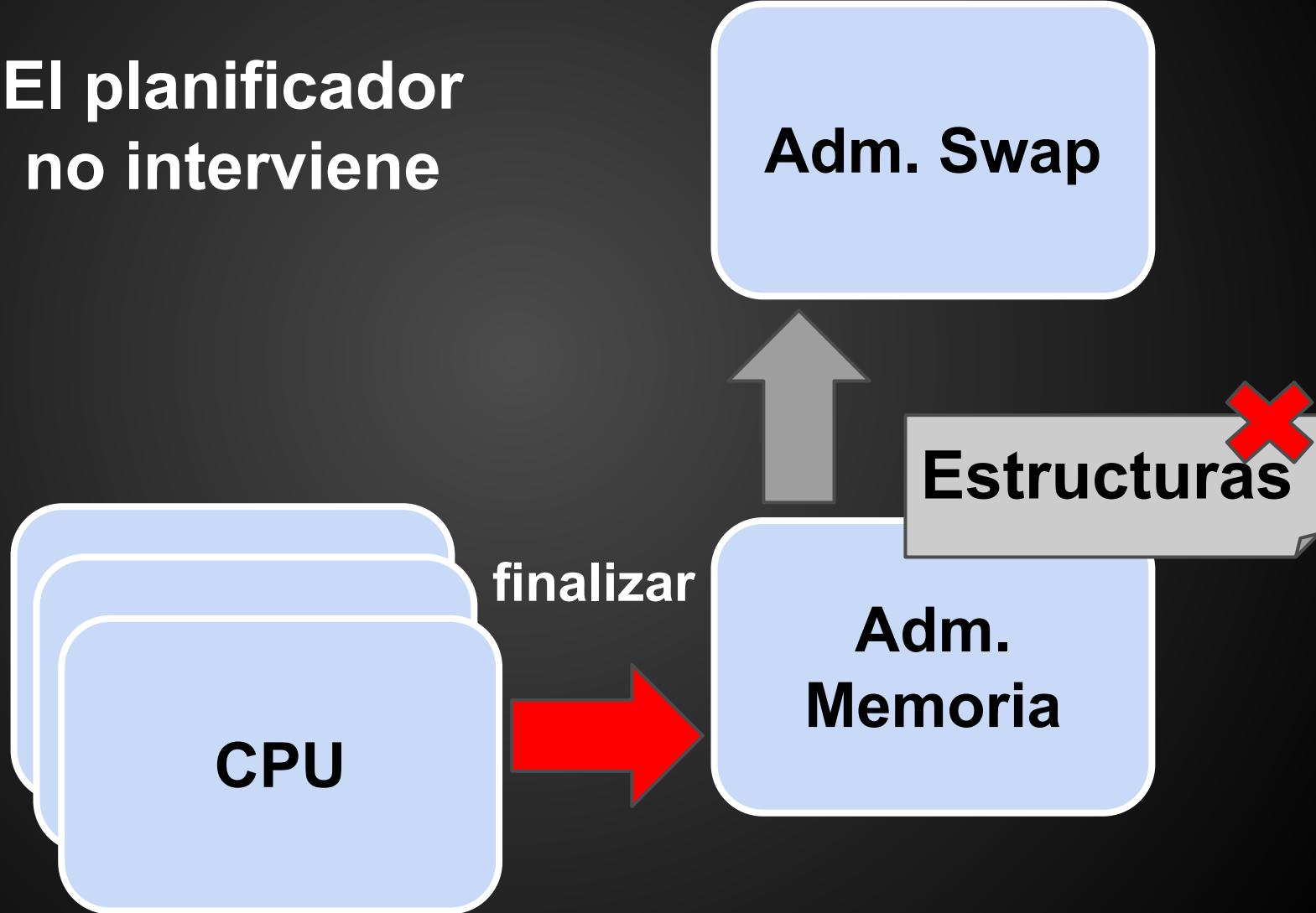
Bloquea al mProc  
Elige otro mProc

# Sentencias: entrada-salida



# Sentencias: finalizar

El planificador  
no interviene



# Sentencias: finalizar

El planificador  
no interviene

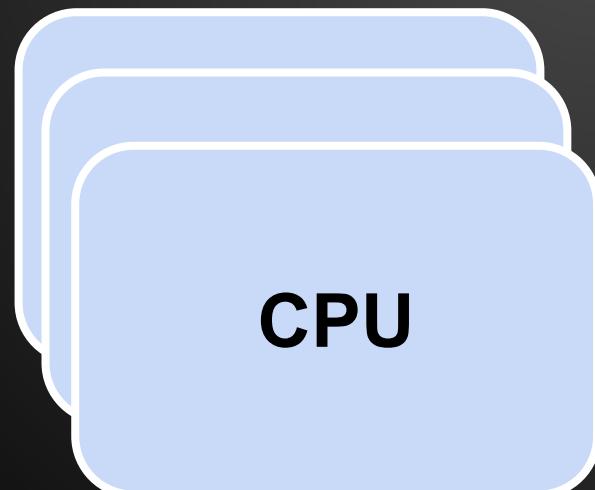
Estructuras 

Adm. Swap

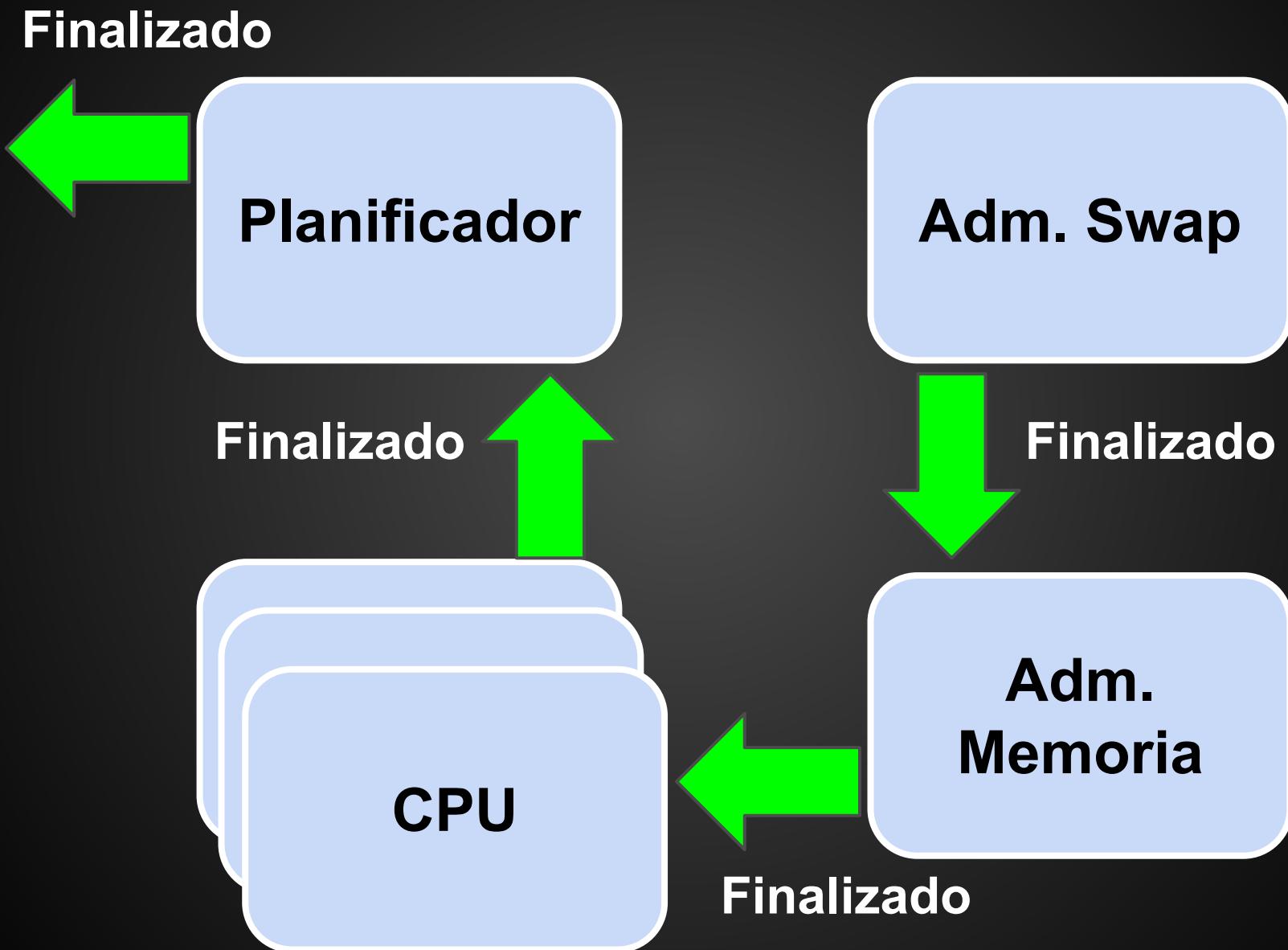
finalizar

Adm.  
Memoria

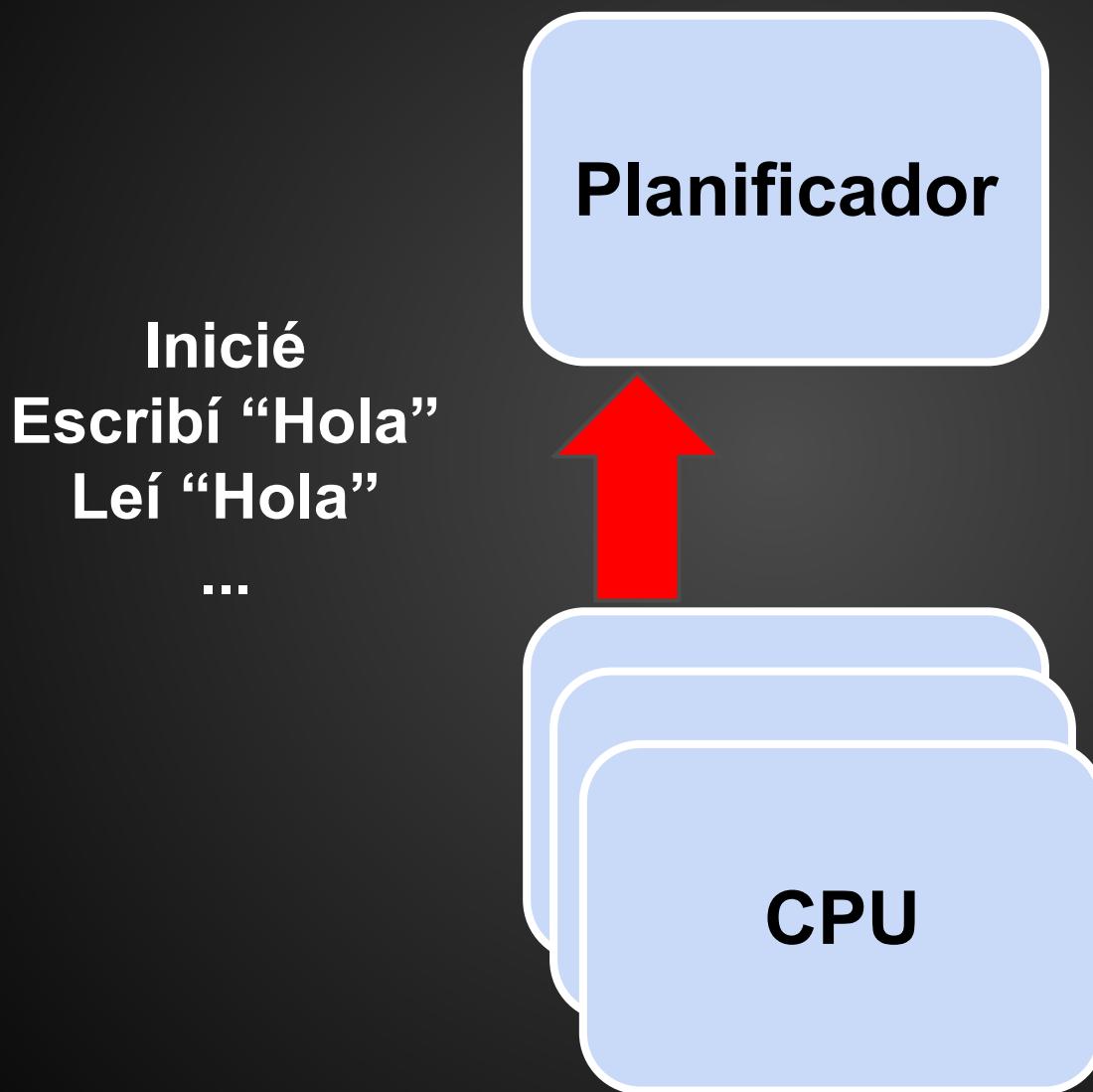
CPU



# Sentencias: finalizar



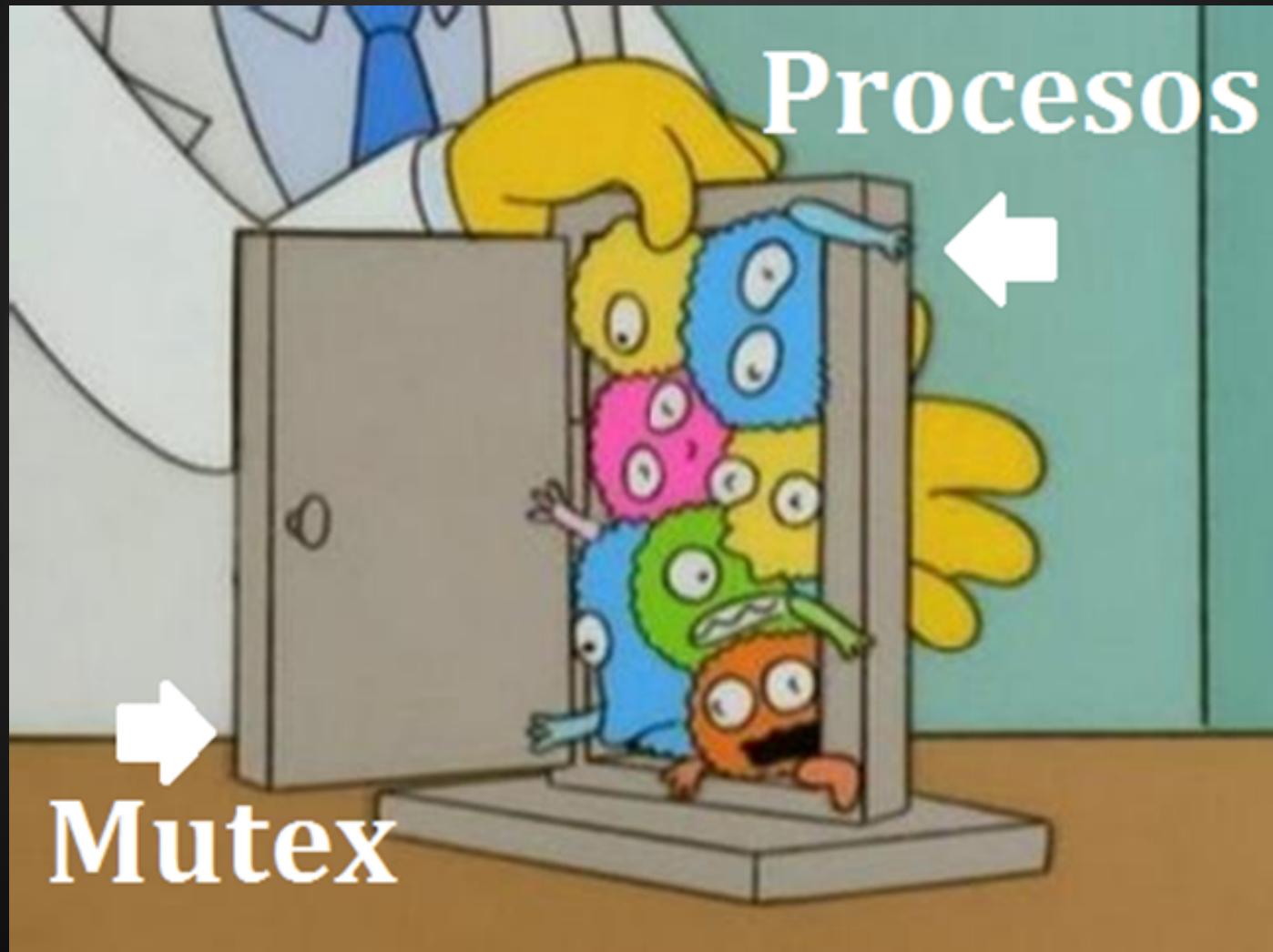
# Termina una ráfaga o fin de Quantum



Muchos mensajes

Mucha concurrencia

# ¡Es MUY importante la sincronización!



# Comandos: correr PATH

correr programa.cod



Pone programa.cod en funcionamiento

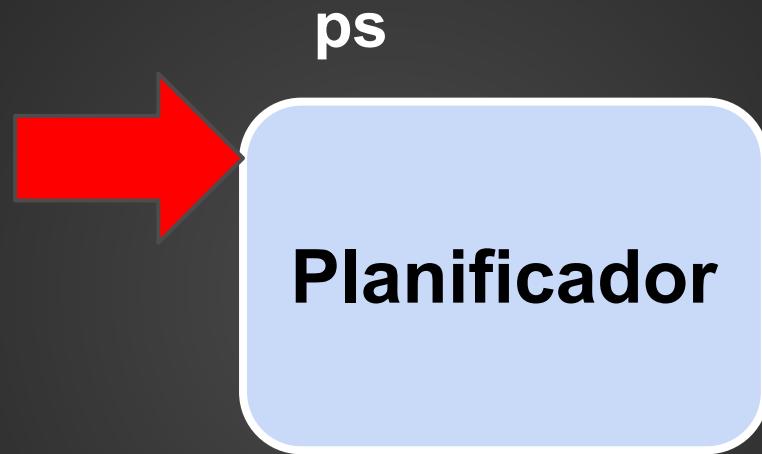
# Comandos: finalizar PID



Pone el puntero de instrucciones de PID en la última.

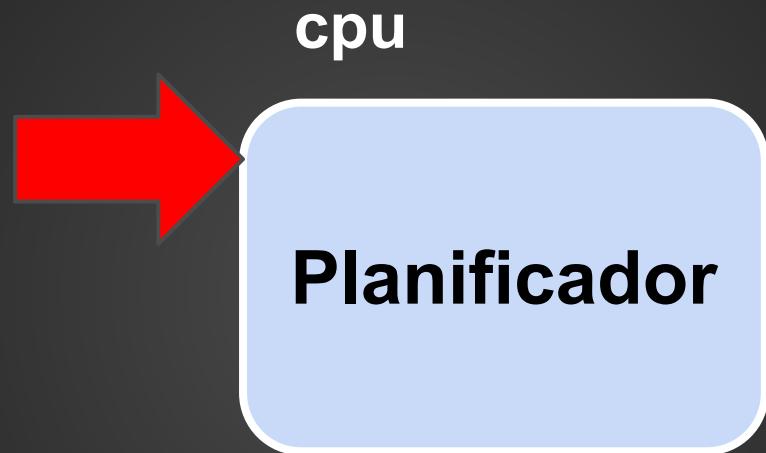
**No lo mata.** Lo fuerza a terminar su ejecución

# Comandos: ps



Devuelve el estado de todos los procesos del sistema

# Comandos: cpu



**Devuelve el % de utilización de cada CPU en el último minuto registrado (no en los últimos 60 segundos)**

¿Qué es Asignación  
Contigua?

# Asignación contigua

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	...								
								...	49

Asignado: -

Libre: 0 a 49

# Asignación contigua

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	...								
								...	49

Asignado: P1 → 0 a 4

Libre: 5 a 49

# Asignación contigua

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	...								
								...	49

Asignado: P1 → 0 a 4  
P2 → 5 a 16

Libre: 17 a 49

# Asignación contigua

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	...								
								...	49

Asignado: P1 → 0 a 4

P2 → 5 a 16

P3 → 17 a 19

Libre: 20 a 49

# Asignación contigua

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	...								
								...	49

Asignado: P1 → 0 a 4  
P3 → 17 a 19

Libre: 5 a 16  
20 a 49

# Asignación contigua

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	...								
								...	49

Asignado: P1 → 0 a 4

P3 → 17 a 19

P4 → 5 a 9

Libre: 10 a 16

20 a 49

# Asignación contigua

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	...								
								...	49

Asignado: P1 → 0 a 4

P3 → 17 a 19

P4 → 5 a 9

Libre: 10 a 16

20 a 49

A) P5 necesita 35 bloques → Los tengo pero no contiguos

# Asignación contigua

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	...								
								...	49

Asignado: P1 → 0 a 4  
P3 → 17 a 19  
P4 → 5 a 9

Libre: 10 a 16  
20 a 49

B) Finaliza P3 → Consolidó

# Asignación contigua

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	...								
								...	49

Asignado: P1 → 0 a 4

P3 → 17 a 19

P4 → 5 a 9

Libre: 10 a 16

17 a 19

20 a 49

Entonces → Libre: 10 a 49

¿Cómo funcionan  
los algoritmos de  
planificación?

¿Cómo funcionan  
los accesos a  
memoria?

¿Cuándo tengo que  
acceder a Swap y  
cuando no?

¿Qué significa...?

# Paciencia

## Lo vas a ver en la cursada

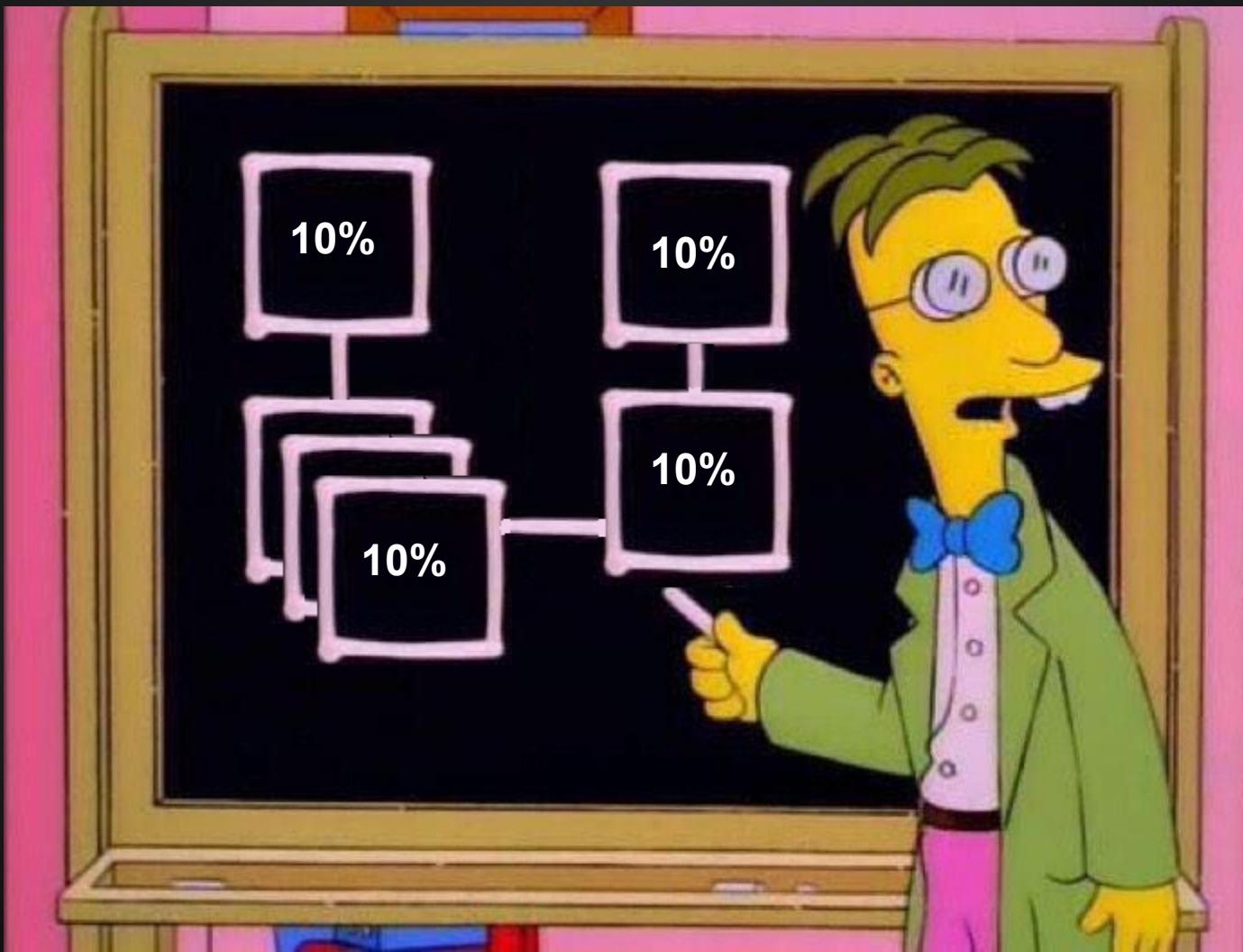


Pero si ya estás en tema y estás ansioso, acércate a preguntar.

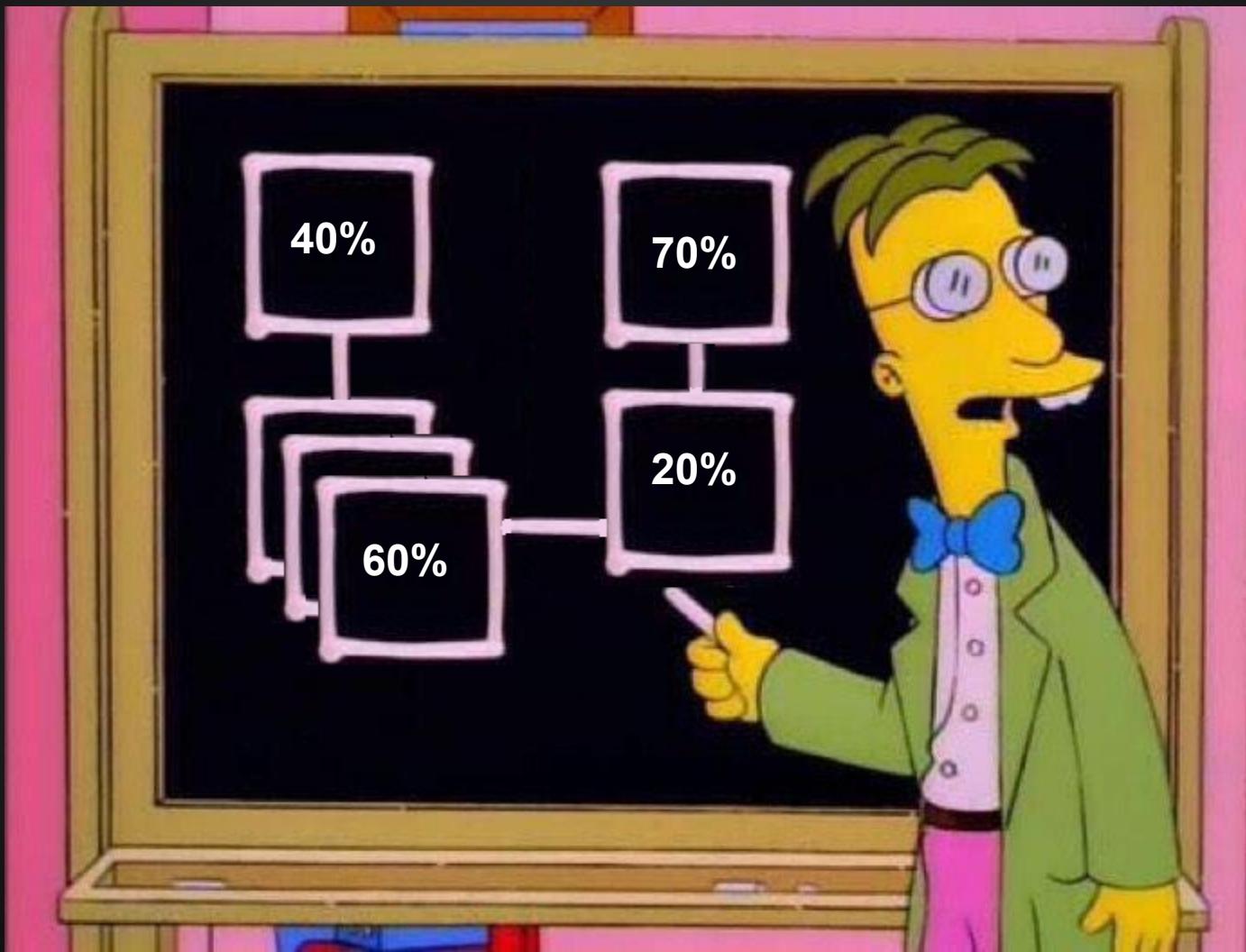
# ¿Preguntas?



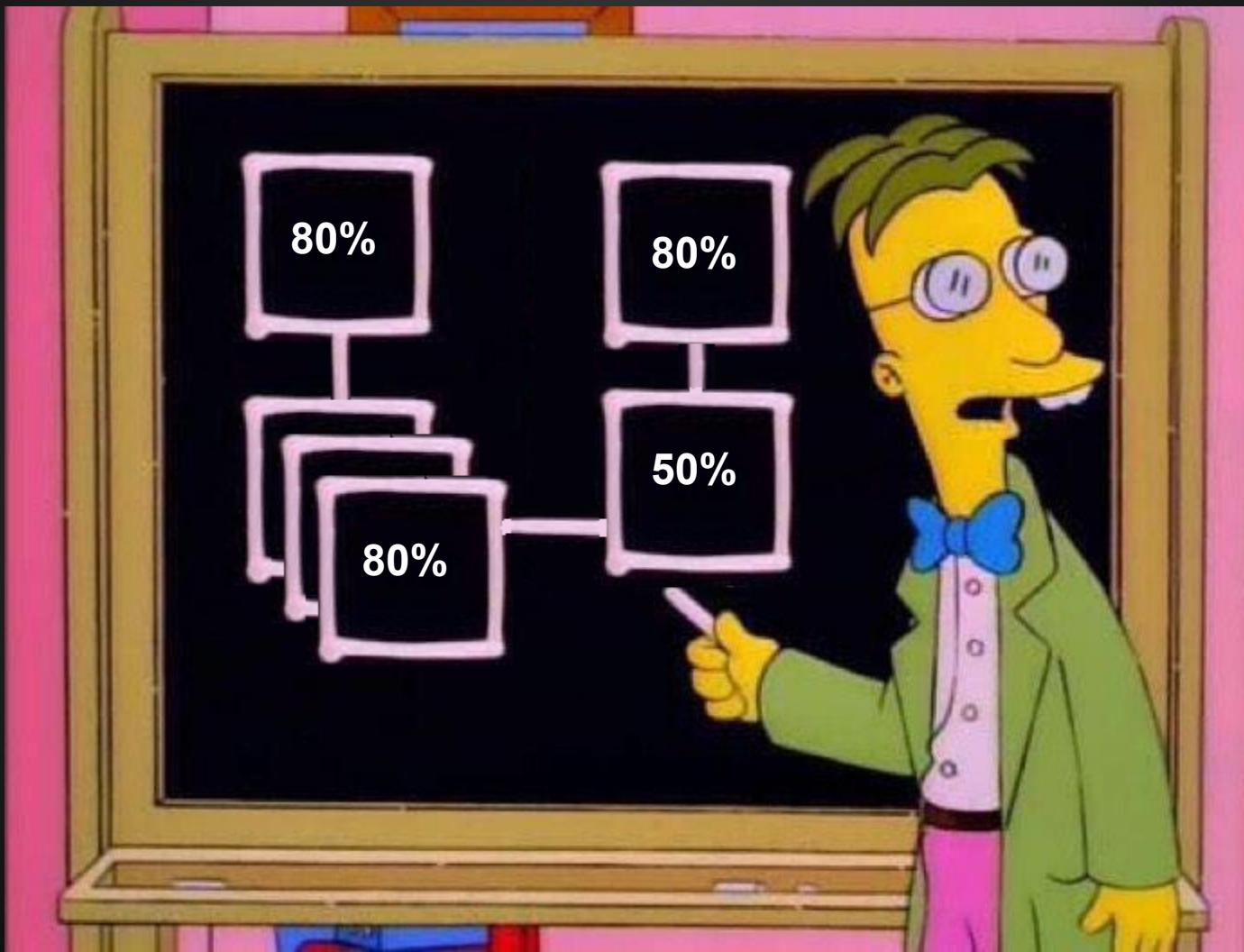
# Checkpoint 1 - 12 de Septiembre



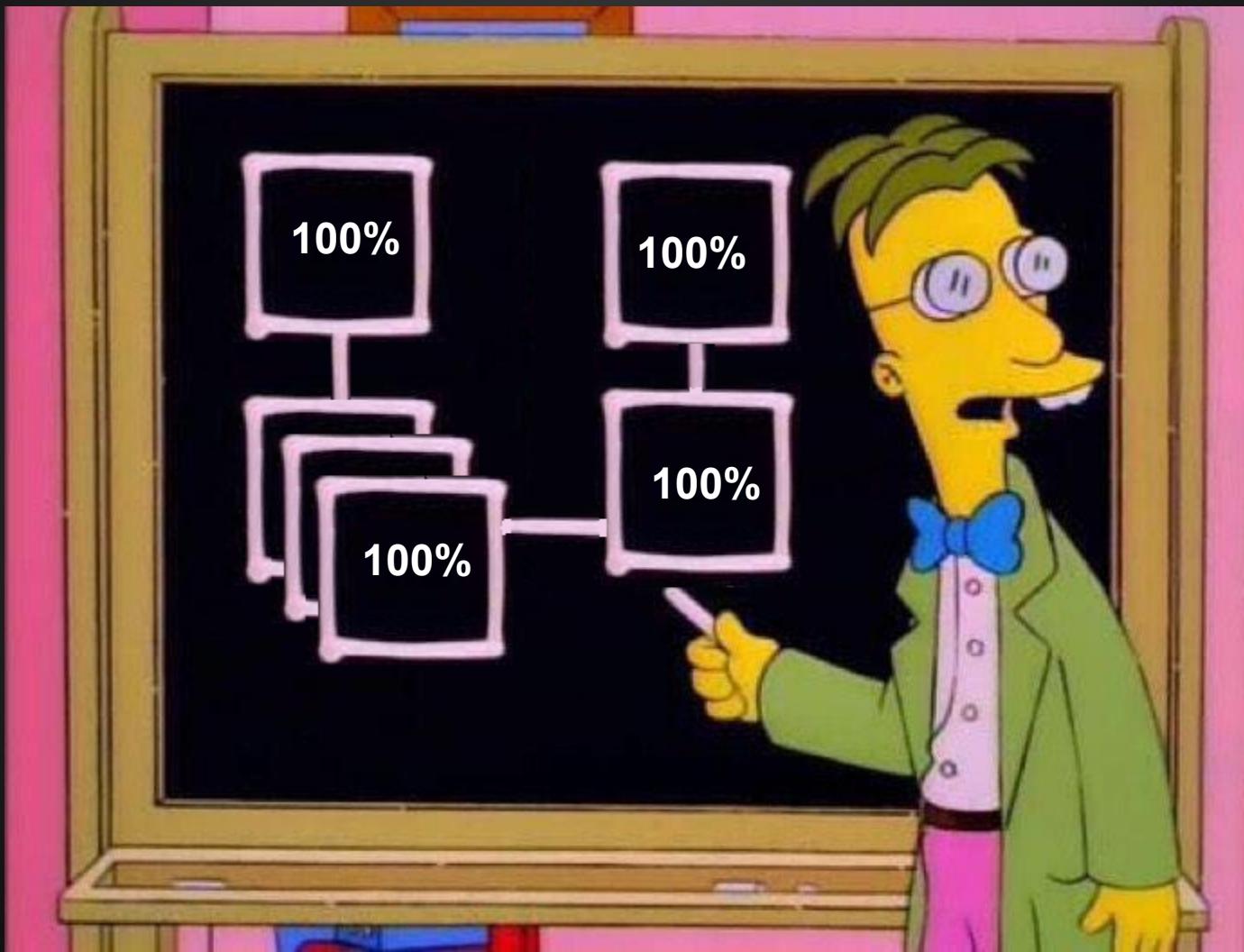
# Checkpoint 2 - 3 de Octubre



# Checkpoint 3 - 7 de Noviembre



# Entrega Final - 28 de Noviembre



# ¿Preguntas?

