

# SIMULACIONES EN MERCADOS DE VALORES

J A V I E R   H E R N Á N D E Z   N A V A R R O



## Preguntas

¿Qué es un mercado de valores?

¿Qué es paralelizar?

¿Cómo simular posibles  
escenarios del mercado de  
valores?

¿Cómo paralelizar simulaciones?

# ¿Qué es un Mercado de Valores?

- Espacio físico o virtual de compra y venta de instrumentos financieros:
  - Acciones
  - Bonos
  - Fondos de inversión
- Constantes cambios en los precios, incluso en unidades muy pequeñas de tiempo.
- Es un Movimiento Browniano Geométrico.

Comprar

INICIO > AAPL · NASDAQ

## Apple

213,07 \$ ↑14,78 % +27,43 Año hasta la fecha

Después del cierre: 214,20 \$ (↑0,53 %) +1,13

Cerrados: 12 jun, 19:59:56 UTC-4 · USD · NASDAQ · Renuncia de responsabilidad

1 día

5 D

1 mes

6 M

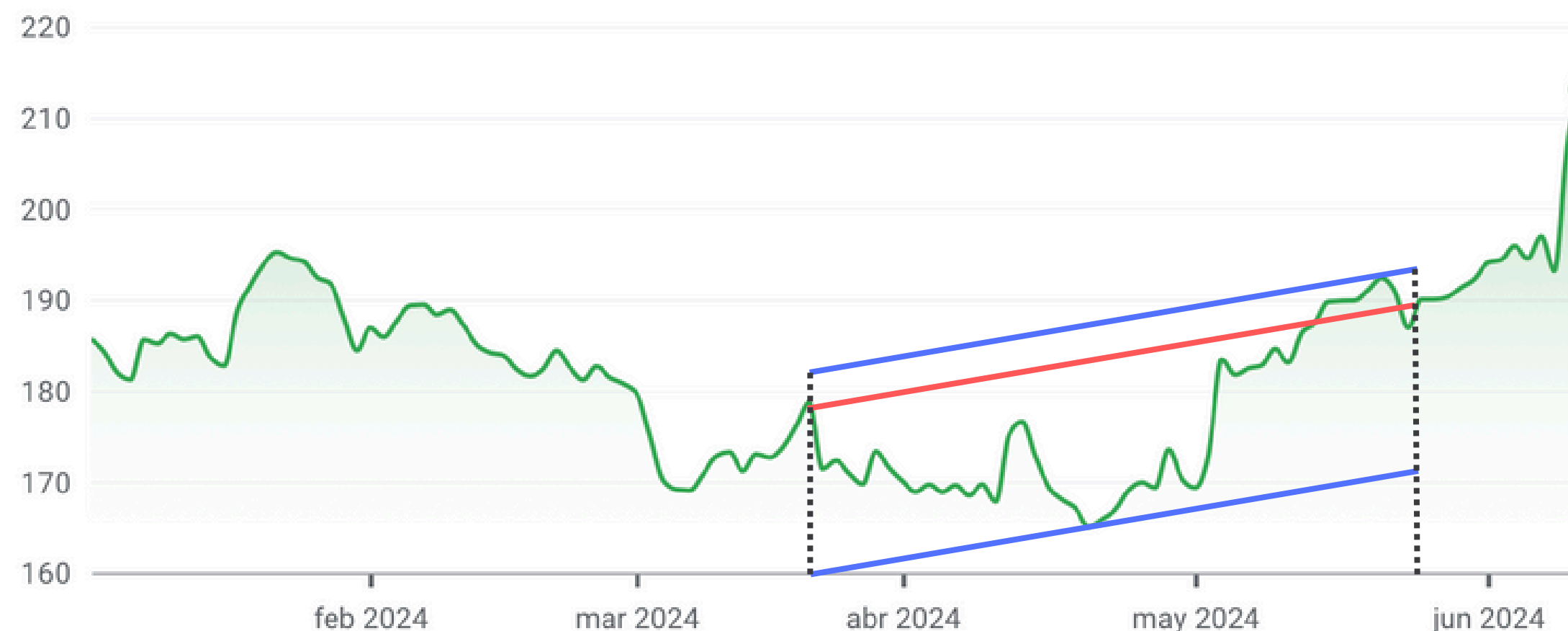
YTD

1 año

5 años

MÁX.

Eventos clave >



Tomado de: Google Finances

# Movimiento Browniano Geométrico

---

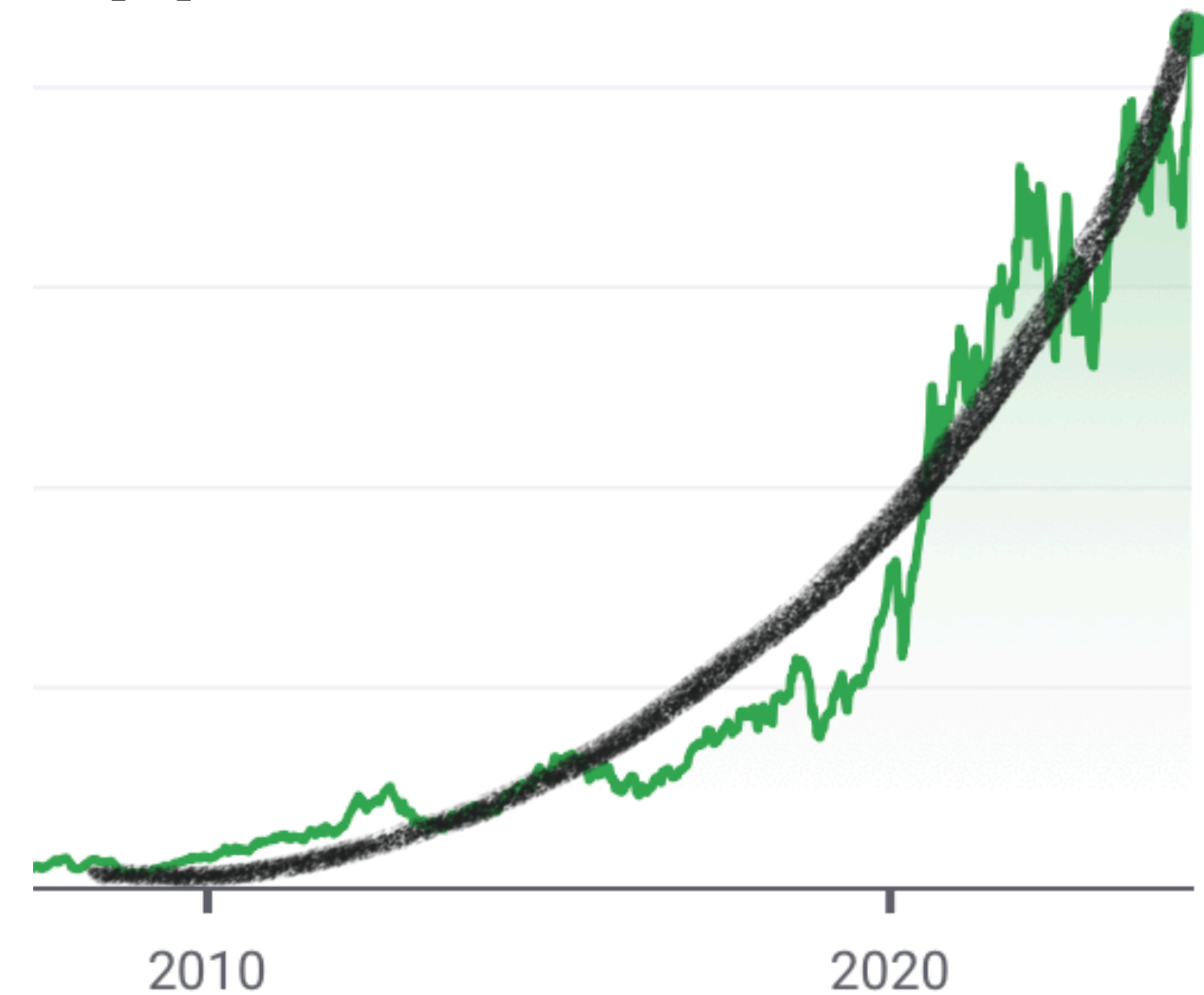
$$S_t = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t\right), \quad t \geq 0$$

- Donde  $W_t$  es un movimiento browniano estándar.

Aplicaciones:

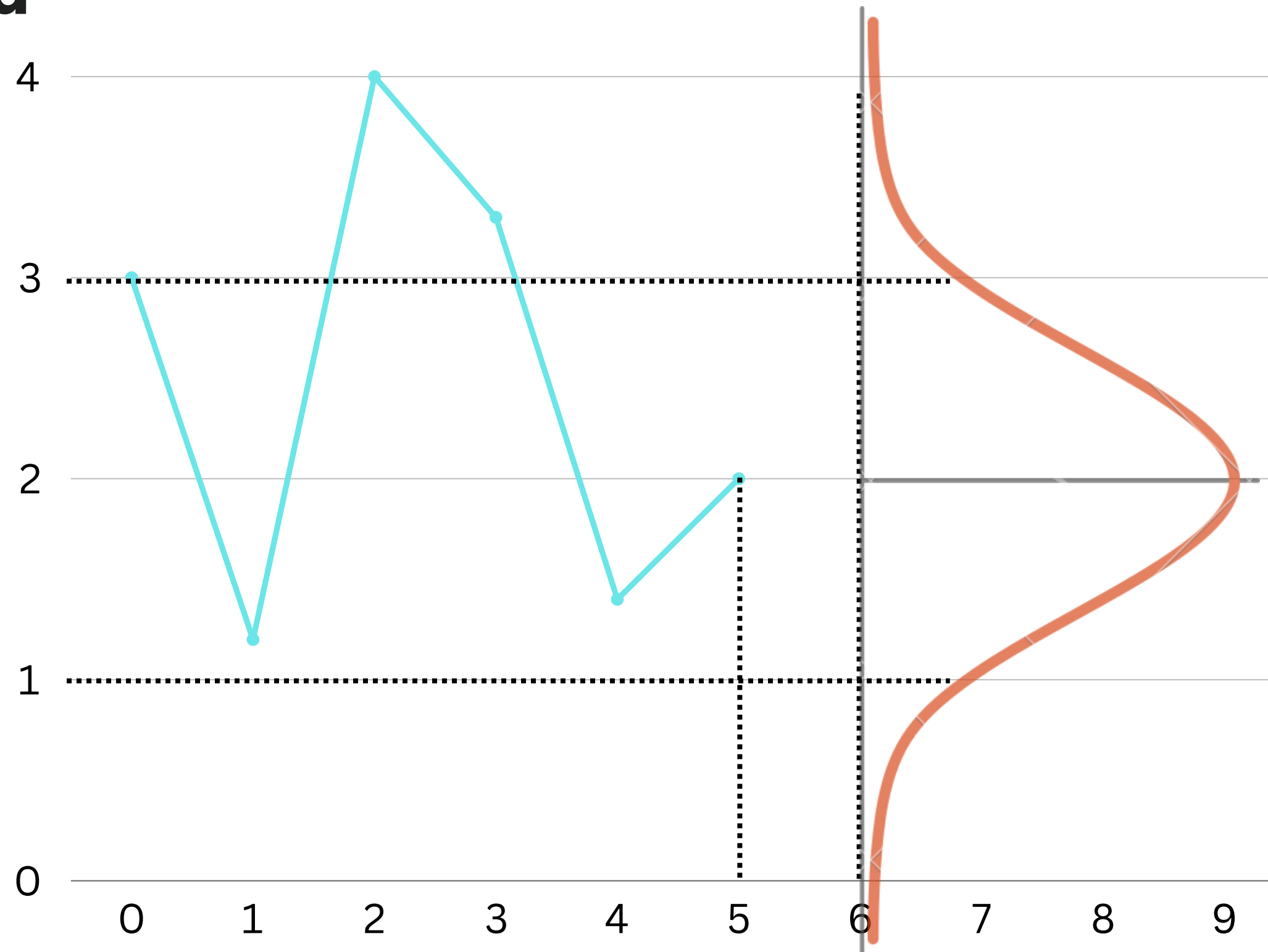
- Simulación de Montecarlo.
- Modelo Black-Scholes.
- Value at Risk. (VaR)

## Apple



Tomado de: Google Finances

# Idea intuitiva



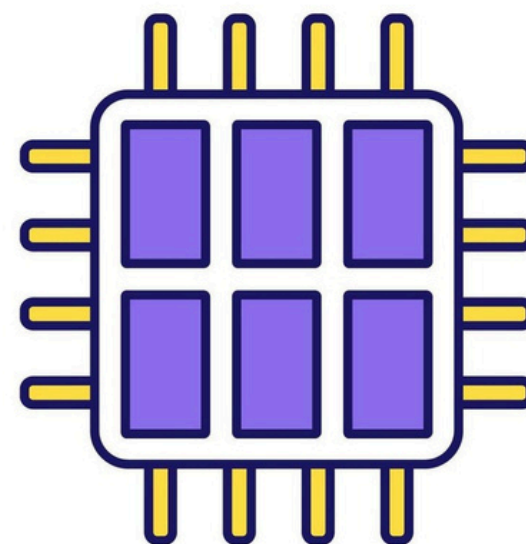
Fuente: elaboración propia.

# Conceptos importantes

---

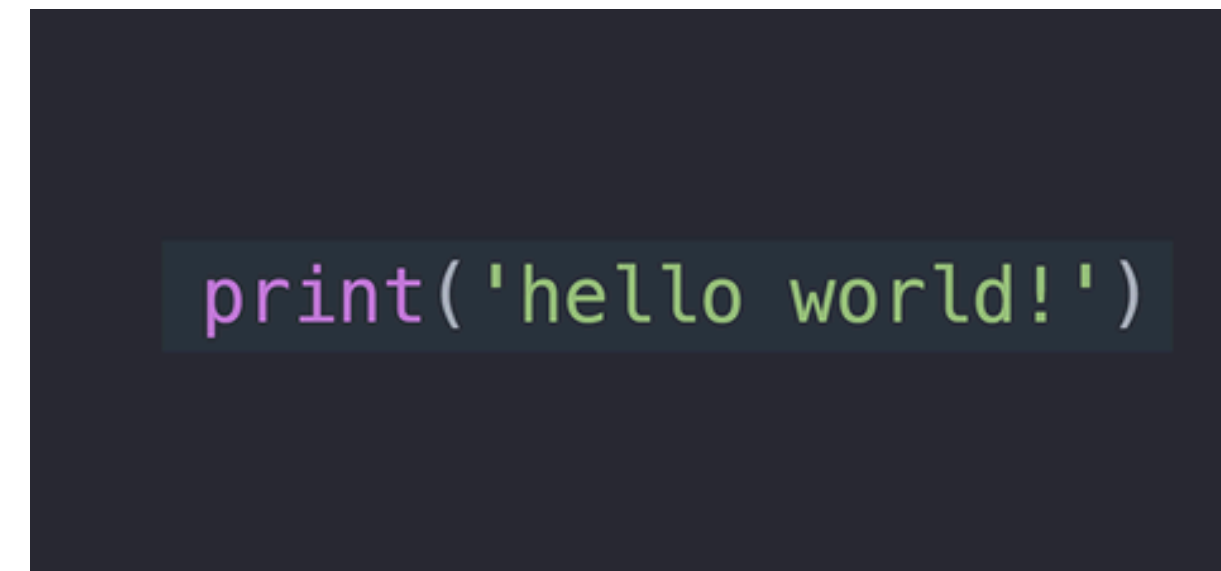


CPU



NÚCLEOS

Paralelizar



PROCESOS

# Procesos estructurados

---

Es la forma usual de programar, se caracteriza por ser:

- Selectiva.
- Iterativa.
- Secuencial.

RUN

```
1 # checking response.status_code (if you get 502, try removing the code)
2 if response.status_code != 200:
3     print(f"Status: {response.status_code} - Try removing the code")
4 else:
5     print(f"Status: {response.status_code}\n")
6
7 # using BeautifulSoup to parse the response object
8 soup = BeautifulSoup(response.content, "html.parser")
9
10 # finding Post images in the soup
11 images = soup.find_all("img", attrs={"alt": "Post image"})
12
13 # downloading images
14 for image in images:
15     # ...
```

Tomado de: <https://eiposgrados.com/wp-content/uploads/2023/01/python-vs-php-codigo-1024x546.jpg>

# Procesos estructurados

---

```
print('Programa que suma dos números.')  
  
num_1 = input('Ingrese el primer número: ')  
  
num_2 = input('Ingrese el segundo número: ')  
  
suma = num_1 + num_2  
  
print('La suma es de: ' + suma)
```

Fuente: elaboración propia.



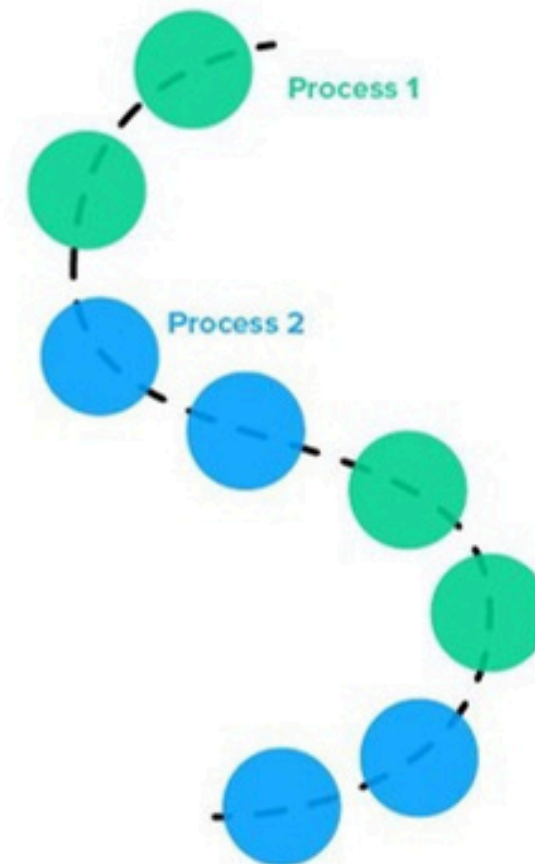
# ¿Cómo llevar el procesador al límite?

Existen principalmente dos maneras de ejecutar varias tareas o procesos a la vez:

- Procesos concurrentes
- Procesos paralelos

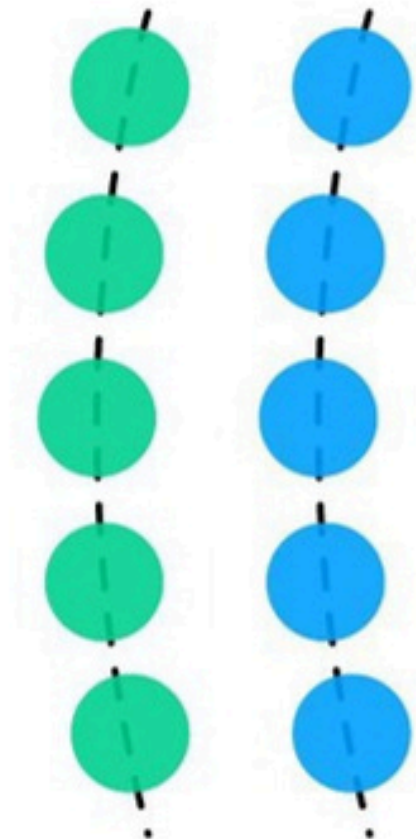
RUN

Concurrency



vs

Parallelism

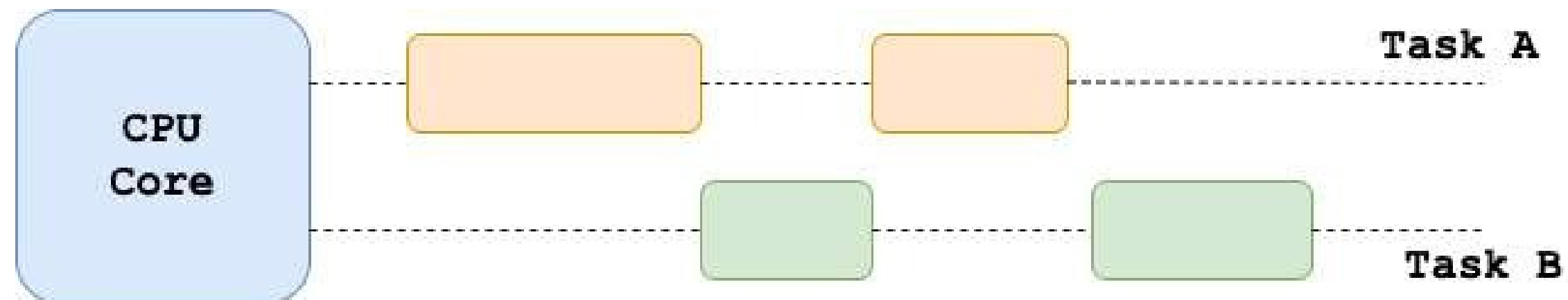


Tomado de: <https://eiposgrados.com/wp-content/uploads/2023/01/python-vs-php-codigo-1024x546.jpg>

# Procesos concurrentes

- Tienen la capacidad de ejecutar múltiples procesos al mismo tiempo en el mismo sistema informático de manera superpuesta.
- Importante: no se están usando más núcleos.

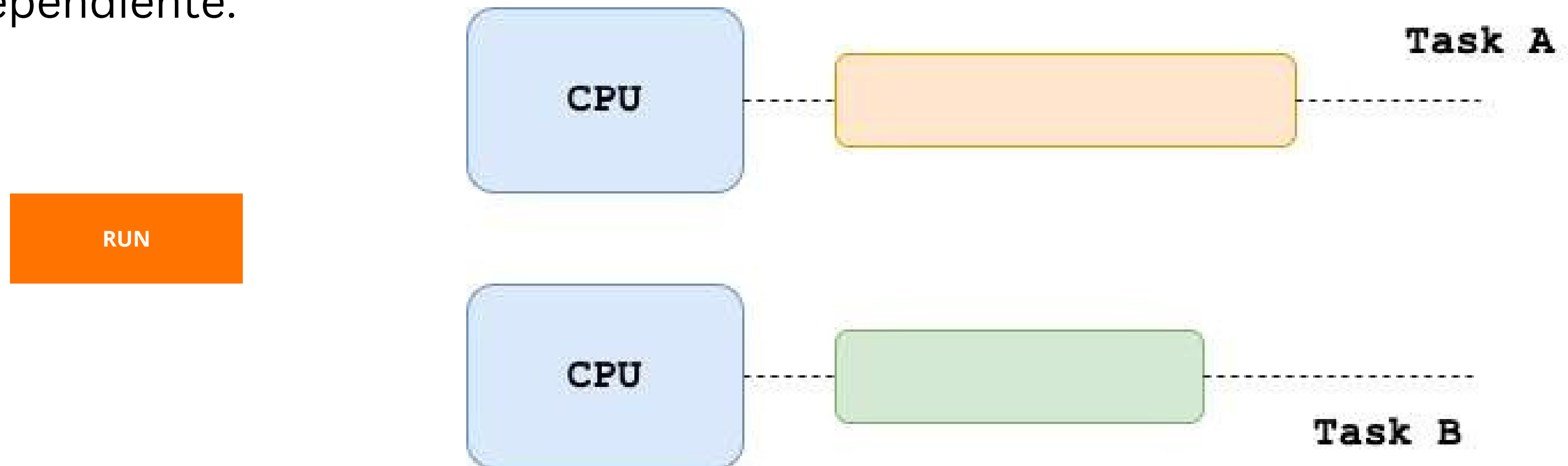
RUN



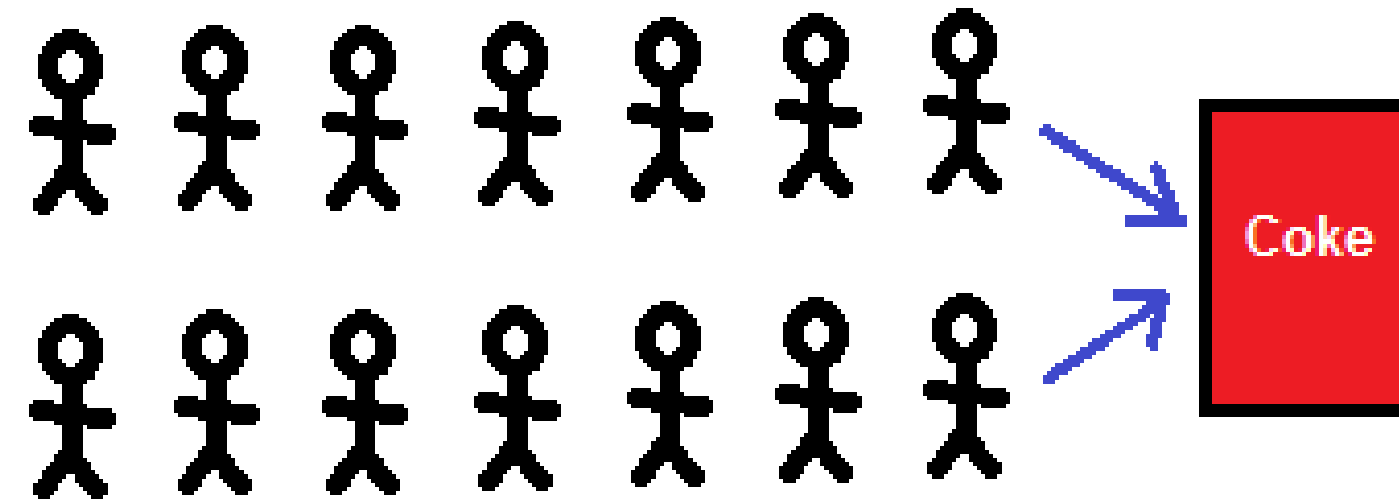
Tomado de: <https://eiposgrados.com/wp-content/uploads/2023/01/python-vs-php-codigo-1024x546.jpg>

# Procesos paralelos

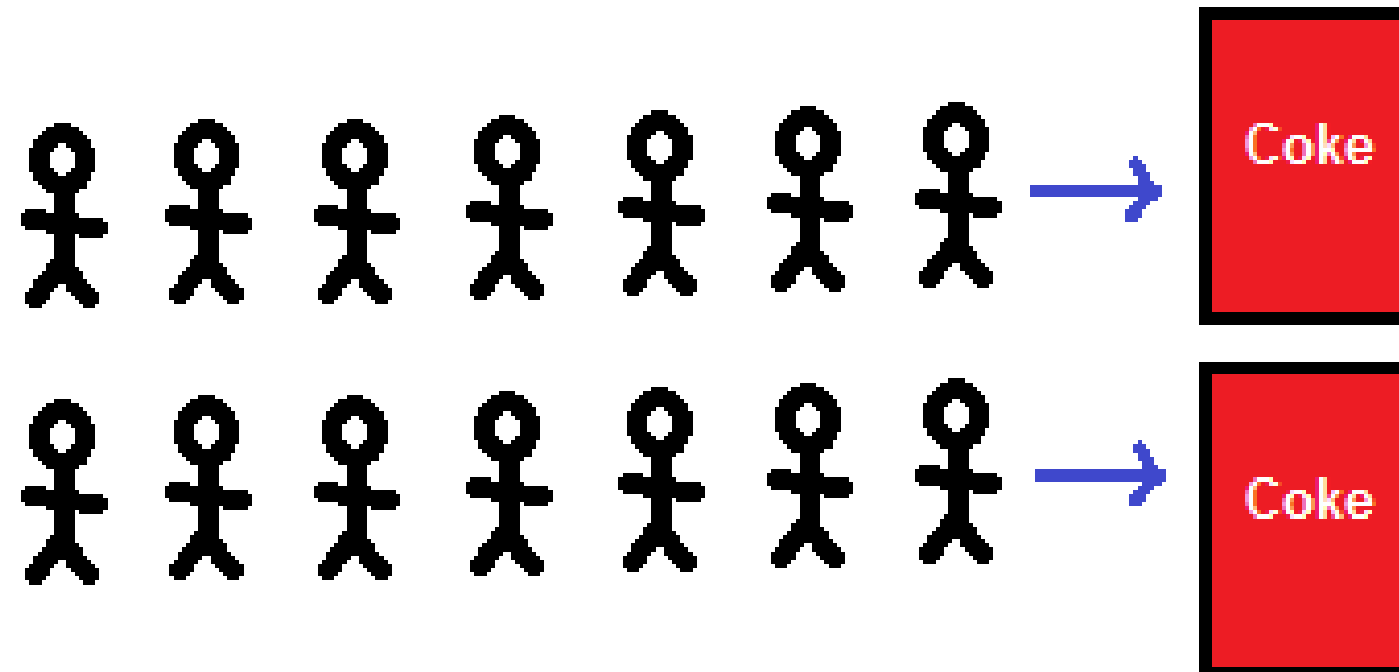
- Permiten a un programa ejecutar procesos simultáneamente usando múltiples unidades de procesamiento
- Cada núcleo está realizando los procesos de manera independiente.



# Procesos concurrentes vs procesos paralelos



Concurrent: 2 queues, 1 vending machine

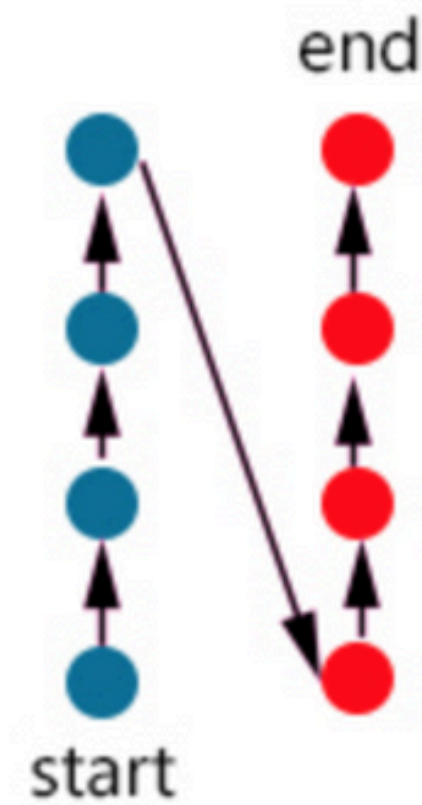


Parallel: 2 queues, 2 vending machines

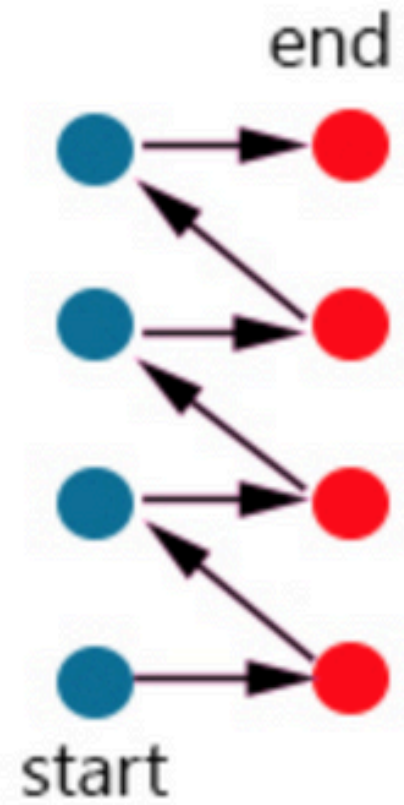
# En resumen

---

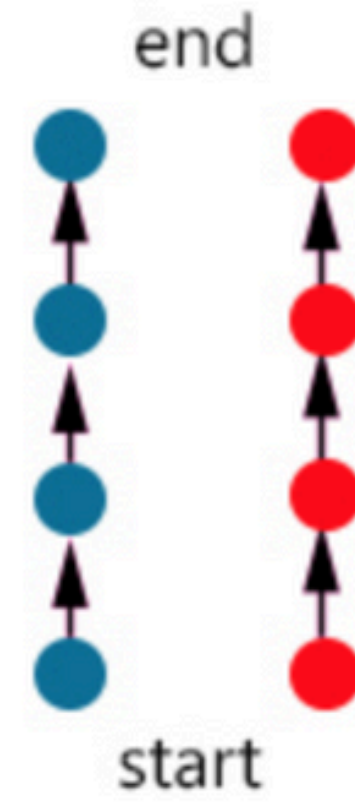
Sequential



Concurrent



Parallel



[Ir al código](#)

# CONCLUSIONES

# RECOMENDACIONES

N O P A R A L E L I C E N ( ? )

# ESPACIO PARA PREGUNTAS

M U C H A S   G R A C I A S