

Estudio de Ejecución Especulativa en Go

Sebastián Rodríguez

Javier Rivera

Italo Ciafaroni

Tarea 02 – Lenguaje de Programación

27 de octubre de 2025

1. Introducción

El objetivo de este estudio es cuantificar el impacto de la ejecución especulativa sobre dos tareas de cómputo intensivo provistas en el anexo de la consigna: una búsqueda con Proof-of-Work (PoW) y el conteo de números primos mediante división sucesiva. Se compara el desempeño de la estrategia especulativa frente a una ejecución secuencial tradicional, midiendo tiempos de respuesta y calculando el *speedup* relativo.

2. Configuración experimental

Todas las corridas se ejecutaron con los mismos parámetros de entrada:

- Dimensión de las matrices para la condición: $n = 400$.
- Dificultad del PoW (`input_A`): 5 ceros iniciales.
- Límite para la búsqueda de primos (`input_B`): 500 000.
- Umbral de decisión: 1 (garantiza que la rama ganadora sea siempre la correspondiente al PoW).
- Cantidad de corridas independientes por estrategia: 30.

El programa completo se ejecutó mediante:

```
go run . -n 400 -umbral 1 -runs 30 -difficulty 5 -pow-data casoA -primes-limit 500000
```

Las métricas detalladas de cada corrida se encuentran en el archivo `metricas_caseA.csv`.

3. Metodología

La implementación especulativa lanza de forma concurrente las ramas A (PoW) y B (primos) mientras el hilo principal evalúa la condición costosa basada en la traza del producto de matrices. Una vez conocida la rama ganadora, se cancela cooperativamente la perdedora.

Para la línea base secuencial se reutiliza la misma condición, pero se difiere la ejecución del cómputo vencedor hasta conocer el resultado, evitando cualquier trabajo especulativo.

Cada estrategia se ejecutó 30 veces con los parámetros anteriores. A partir del CSV se promediaron los tiempos totales de cada corrida y se calculó el *speedup* utilizando la definición clásica:

$$\text{Speedup} = \frac{T_{\text{secuencial}}}{T_{\text{especulativo}}}. \quad (1)$$

4. Resultados

La Tabla 1 resume los tiempos promedio observados y el *speedup* relativo.

Estrategia	Tiempo promedio (ms)	Speedup relativo
Especulativa	21,484	1,159
Secuencial	24,903	1,000

Cuadro 1: Promedio de tiempos totales y speedup obtenido a partir de `metricas_caseA.csv`.

La Figura 1 muestra la comparación visual entre las dos estrategias, incluyendo la evolución por corrida y el *speedup* individual. El gráfico se genera con `python3 plot_metrics.py metricas_caseA.csv comparacion_caseA.png`.

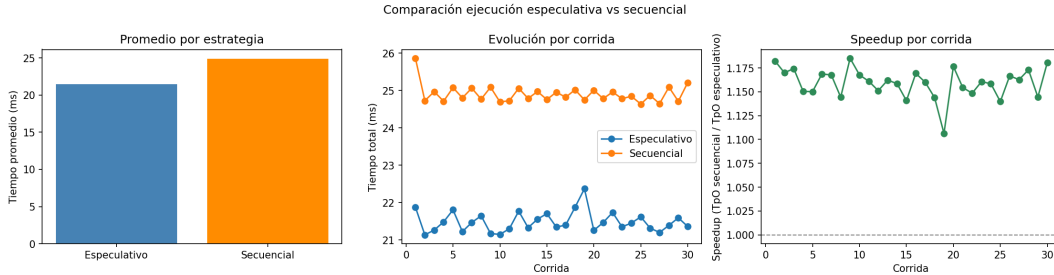


Figura 1: Comparación especulativo vs. secuencial (tiempos promedio, series por corrida y speedup).

5. Análisis

En todas las corridas observadas la rama especulativa B (primos) se canceló antes de completar trabajo significativo: el canal de cancelación se activa a los 5,5 ms en promedio, por lo que la mayor parte del costo recae en la rama ganadora (PoW). La evaluación de la condición también tarda alrededor de 5 ms, pero en la estrategia especulativa ocurre solapada con la ejecución del PoW.

En la versión secuencial, la condición y la rama ganadora se ejecutan en serie, acumulando aproximadamente 25 ms. La estrategia especulativa reduce el tiempo total a 21,484 ms, logrando un *speedup* de 1.159 (14% de mejora). La variabilidad entre corridas es baja y las curvas de la Figura 1 son consistentes, lo que evidencia que el beneficio proviene del solapamiento de trabajos y la cancelación temprana de B.

6. Conclusiones

Para el escenario analizado, la ejecución especulativa ofrece una mejora clara sobre la estrategia secuencial al solapar la evaluación de la condición con el cómputo de la rama ganadora y abortar

rápida mente la perdedora. El ahorro promedio de 3,4 ms por corrida (14 %) justifica el uso del patrón especulativo siempre que el coste de lanzar la rama adicional sea bajo en comparación con el trabajo cancelado.

Como trabajo futuro se sugiere repetir el estudio con otros umbrales y combinaciones de entradas para identificar escenarios donde la especulación no aporte ventajas o incluso empeore el rendimiento, así como extender el análisis a la desviación estándar y distribuciones de tiempos.