

# Ejecución especulativa en Go

Francisca Meyer – Lukas Scheel  
Tarea 2 - Lenguajes de Programación  
Profesor: Alonzo Inostrosa  
28 de octubre de 2025

## 1. Introducción

El presente informe desarrolla la implementación y análisis del patrón de Ejecución especulativa, una técnica de optimización que busca anticipar decisiones condicionales costosas mediante la ejecución paralela de múltiples ramas de un programa antes de conocer la condición final.

La propuesta se implementó en lenguaje Go, aprovechando las capacidades de concurrencia mediante goroutines y canales, comparando su rendimiento con una versión secuencial del mismo algoritmo.

El objetivo principal es evaluar la ganancia de rendimiento (medida mediante el *Speedup*) obtenida al aplicar el modelo especulativo, verificando su efectividad sobre cargas de trabajo computacionalmente intensivas.

## 2. Descripción general del sistema

El programa implementa dos modos de ejecución:

- **Modo Secuencial:** calcula primero la condición costosa y luego ejecuta solo la rama ganadora.
- **Modo Especulativo:** lanza ambas ramas en paralelo mientras calcula la condición.  
Una vez determinada la condición, valida la rama correcta y cancela la otra, aprovechando el tiempo de procesamiento superpuesto.

La condición costosa se obtiene a partir del cálculo de la traza del producto de dos matrices (`CalcularTrazaProductoMatrices(n)`), y el umbral determina qué rama será la ganadora.

En este estudio se fijó un umbral = 0, de modo que siempre gana la rama A (Proof-of-Work), lo que permite comparar tiempos bajo una misma carga de trabajo.

### 3. Ramas y operaciones costosas

Rama	Descripción	Función implementada
<b>Proof-of-Work (PoW)</b>	Simula el cálculo de un hash con prefijo de ceros (similar a la minería de bloques). Es intensiva en CPU.	SimularPruebaDeTrabajo()
<b>Búsqueda de primos</b>	Calcula todos los números primos hasta un límite max_primo mediante división simple.	EncontrarNumerosPrimos()
<b>Condición</b>	Multiplica dos matrices $n \times n$ y calcula la traza (suma de diagonal). Determina la rama ganadora.	CalcularTrazaProductoMatrices()

Cada rama se ejecuta en una goroutine independiente, y se utilizan canales para comunicar resultados con el hilo principal.

La cancelación se implementa con `context.WithCancel()`, lo que garantiza una finalización controlada y segura de la rama perdedora.

### 4. Metodología y parámetros de prueba

Se realizaron 30 corridas en cada modo (secuencial y especulativo) con los parámetros:

$n = 300$

umbral = 0

dificultad = 5

max\_primo = 500000

runs = 30

Los resultados de cada corrida se registraron en archivos CSV (`metricas_seq.csv`, `metricas_spec.csv`), los cuales contienen tiempos de inicio, fin, decisión, ejecución de rama, y total.

## 5. Análisis de resultados

```
PS C:\Users\lukas\Desktop\Un1 2 - 2025\Lenguajes de programación\T2-Meyer-Scheel> .\C2spec.exe -mode spec -runs 30 -n 300 -umbral 0 -dificultad 5 -max_primo 500000 -nombre_archivo metricas_spec.csv
>>
C2 - Ejecución Especulativa (Go)
Parámetros: n=300 umbral=0 repeticiones=30 dificultad=5 max_primo=500000
-----
[espec] ejec=26 total=2633ms decision=159ms rama=A rama_ms=2686 cancelada=true traza=1824624
[espec] ejec=3 total=2733ms decision=210ms rama=A rama_ms=2732 cancelada=true traza=1826882
[espec] ejec=2 total=2970ms decision=57ms rama=A rama_ms=2969 cancelada=true traza=1832283
[espec] ejec=8 total=3002ms decision=21ms rama=A rama_ms=2844 cancelada=true traza=1831583
[espec] ejec=25 total=3016ms decision=64ms rama=A rama_ms=2998 cancelada=true traza=1826223
[espec] ejec=9 total=3018ms decision=50ms rama=A rama_ms=2957 cancelada=true traza=1810588
[espec] ejec=17 total=3023ms decision=133ms rama=A rama_ms=3022 cancelada=true traza=1818177
[espec] ejec=11 total=3060ms decision=51ms rama=A rama_ms=3060 cancelada=true traza=1827757
[espec] ejec=16 total=3070ms decision=260ms rama=A rama_ms=3052 cancelada=true traza=1825906
[espec] ejec=13 total=3075ms decision=20ms rama=A rama_ms=3057 cancelada=true traza=1825503
[espec] ejec=12 total=3075ms decision=5ms rama=A rama_ms=3057 cancelada=true traza=1820762
[espec] ejec=29 total=3086ms decision=18ms rama=A rama_ms=2913 cancelada=true traza=1829078
[espec] ejec=4 total=3126ms decision=212ms rama=A rama_ms=3125 cancelada=true traza=1821246
[espec] ejec=15 total=3127ms decision=30ms rama=A rama_ms=2953 cancelada=true traza=1817599
[espec] ejec=23 total=3127ms decision=46ms rama=A rama_ms=3108 cancelada=true traza=1825139
[espec] ejec=19 total=3130ms decision=179ms rama=A rama_ms=3111 cancelada=true traza=1824322
[espec] ejec=10 total=3132ms decision=235ms rama=A rama_ms=3111 cancelada=true traza=1825021
[espec] ejec=7 total=3148ms decision=169ms rama=A rama_ms=3129 cancelada=true traza=1822173
[espec] ejec=5 total=3159ms decision=6ms rama=A rama_ms=3010 cancelada=true traza=1822714
[espec] ejec=21 total=3170ms decision=42ms rama=A rama_ms=3164 cancelada=true traza=1818740
[espec] ejec=6 total=3178ms decision=8ms rama=A rama_ms=3029 cancelada=true traza=1823326
[espec] ejec=14 total=3179ms decision=28ms rama=A rama_ms=3161 cancelada=true traza=1823968
[espec] ejec=22 total=3182ms decision=276ms rama=A rama_ms=3181 cancelada=true traza=1825689
[espec] ejec=18 total=3184ms decision=77ms rama=A rama_ms=3184 cancelada=true traza=1825470
[espec] ejec=27 total=3186ms decision=283ms rama=A rama_ms=3186 cancelada=true traza=1825436
[espec] ejec=1 total=3188ms decision=12ms rama=A rama_ms=3107 cancelada=true traza=1823029
[espec] ejec=20 total=3188ms decision=157ms rama=A rama_ms=3183 cancelada=true traza=1823148
[espec] ejec=30 total=3192ms decision=168ms rama=A rama_ms=3191 cancelada=true traza=1822848
[espec] ejec=24 total=3196ms decision=81ms rama=A rama_ms=3195 cancelada=true traza=1811727
Resumen especulativo: ejec=30 prom_total=3088.50ms prom_decision=162.33ms prom_rama=3046.57ms
Métricas escritas en: metricas_spec.csv
```

Figura 1 – Salida por terminal de Ejecución Especulativa: Se evidencia resultado de prueba especulativa y métricas obtenidas.

```
PS C:\Users\lukas\Desktop\Un1 2 - 2025\Lenguajes de programación\T2-Meyer-Scheel> .\C2spec.exe -mode seq -runs 30 -n 300 -umbral 0 -dificultad 5 -max_primo 500000 -nombre_archivo metricas_seq.csv
>>
C2 - Ejecución Secuencial (Go)
Parámetros: n=300 umbral=0 repeticiones=30 dificultad=5 max_primo=500000
-----
[secu] ejec=9 total=1993ms decision=31ms rama=A rama_ms=1871 traza=1826020
[secu] ejec=18 total=3007ms decision=215ms rama=A rama_ms=2791 traza=1830472
[secu] ejec=3 total=3040ms decision=209ms rama=A rama_ms=2830 traza=1811735
[secu] ejec=24 total=3080ms decision=219ms rama=A rama_ms=2861 traza=1836702
[secu] ejec=27 total=3086ms decision=260ms rama=A rama_ms=2826 traza=1830256
[secu] ejec=7 total=3096ms decision=220ms rama=A rama_ms=2875 traza=1818097
[secu] ejec=30 total=3144ms decision=242ms rama=A rama_ms=2902 traza=1811460
[secu] ejec=11 total=3150ms decision=216ms rama=A rama_ms=2934 traza=1826304
[secu] ejec=20 total=3151ms decision=217ms rama=A rama_ms=2934 traza=1827236
[secu] ejec=4 total=3175ms decision=244ms rama=A rama_ms=2930 traza=1817848
[secu] ejec=17 total=3179ms decision=228ms rama=A rama_ms=2950 traza=1823387
[secu] ejec=15 total=3187ms decision=242ms rama=A rama_ms=2945 traza=1833054
[secu] ejec=16 total=3218ms decision=220ms rama=A rama_ms=2998 traza=1817443
[secu] ejec=28 total=3222ms decision=214ms rama=A rama_ms=3008 traza=1831851
[secu] ejec=6 total=3225ms decision=228ms rama=A rama_ms=2996 traza=1819946
[secu] ejec=22 total=3225ms decision=228ms rama=A rama_ms=2997 traza=1826613
[secu] ejec=25 total=3227ms decision=242ms rama=A rama_ms=2984 traza=1824570
[secu] ejec=1 total=3236ms decision=221ms rama=A rama_ms=3014 traza=1823816
[secu] ejec=5 total=3239ms decision=221ms rama=A rama_ms=3018 traza=1818217
[secu] ejec=9 total=3252ms decision=243ms rama=A rama_ms=3008 traza=1816522
[secu] ejec=29 total=3254ms decision=218ms rama=A rama_ms=3036 traza=1820496
[secu] ejec=10 total=3255ms decision=241ms rama=A rama_ms=3014 traza=1826020
[secu] ejec=23 total=3258ms decision=242ms rama=A rama_ms=3015 traza=1834126
[secu] ejec=12 total=3258ms decision=243ms rama=A rama_ms=3015 traza=1822626
[secu] ejec=26 total=3268ms decision=254ms rama=A rama_ms=3013 traza=1832542
[secu] ejec=19 total=3272ms decision=256ms rama=A rama_ms=3016 traza=1817907
[secu] ejec=2 total=3278ms decision=242ms rama=A rama_ms=3035 traza=1804956
[secu] ejec=14 total=3280ms decision=228ms rama=A rama_ms=3051 traza=1836760
[secu] ejec=13 total=3283ms decision=256ms rama=A rama_ms=3027 traza=1820895
[secu] ejec=21 total=3286ms decision=267ms rama=A rama_ms=3019 traza=1823716
Resumen secuencial : ejec=30 prom_total=3157.80ms prom_decision=226.90ms prom_rama=2930.43ms
Métricas escritas en: metricas_seq.csv
```

Figura 2 – Salida por terminal de Ejecución Secuencial: Se evidencia resultado de prueba secuencial y métricas obtenidas.

### 5.1 Promedios y Speedup

Estrategia	Promedio Total (ms)	Promedio Decisión (ms)	Promedio Rama (ms)	Speedup
Secuencial	3157.80	226.90	2930.43	—

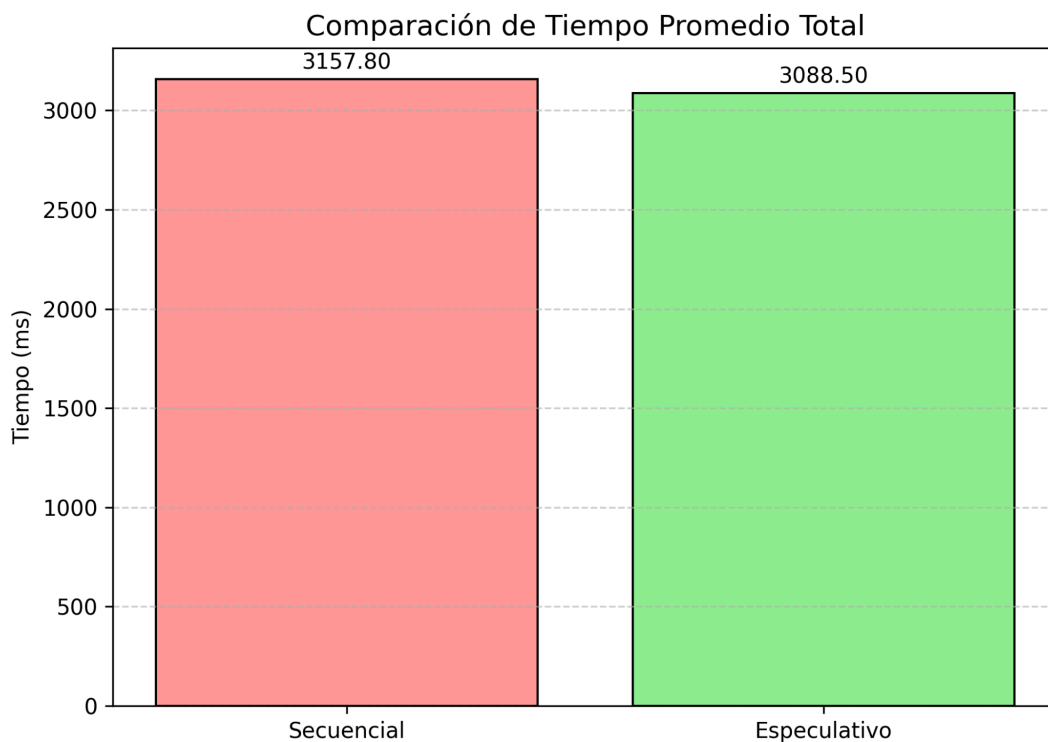
Especulativa	3088.50	102.33	3046.57	1.022×
--------------	---------	--------	---------	--------

### Cálculo:

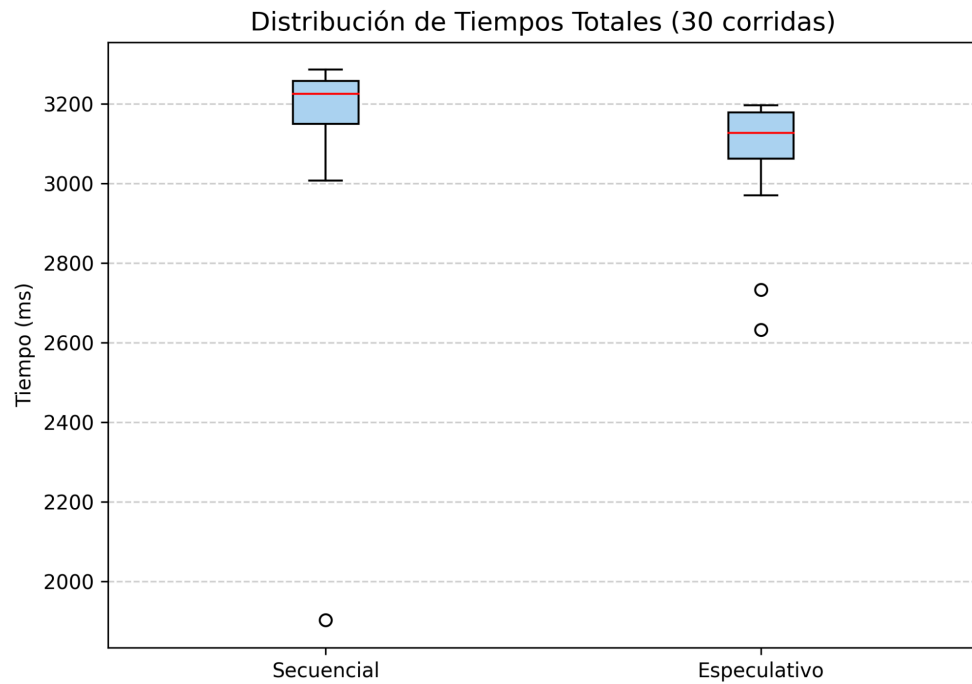
$$\text{Speedup} = 3157.80 / 3088.50 = 1.022\times \rightarrow \approx 2.2 \% \text{ de mejora.}$$

El resultado indica que el patrón especulativo logró una reducción promedio del 2 % en el tiempo total respecto a la ejecución secuencial, validando la efectividad del modelo concurrente para esta configuración.

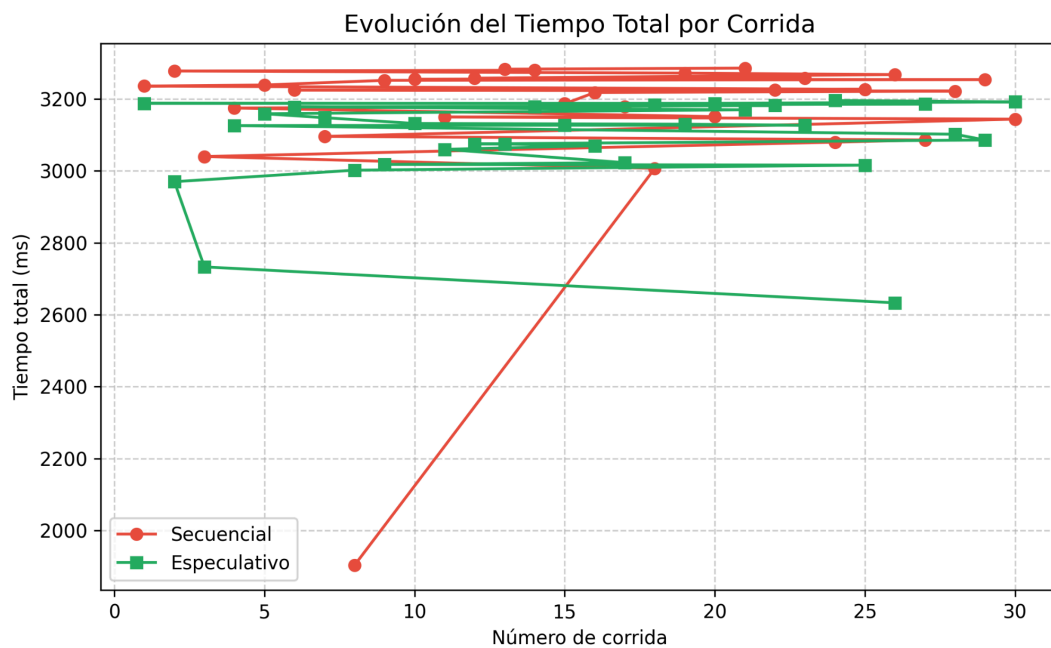
## 5.2 Visualización de resultados



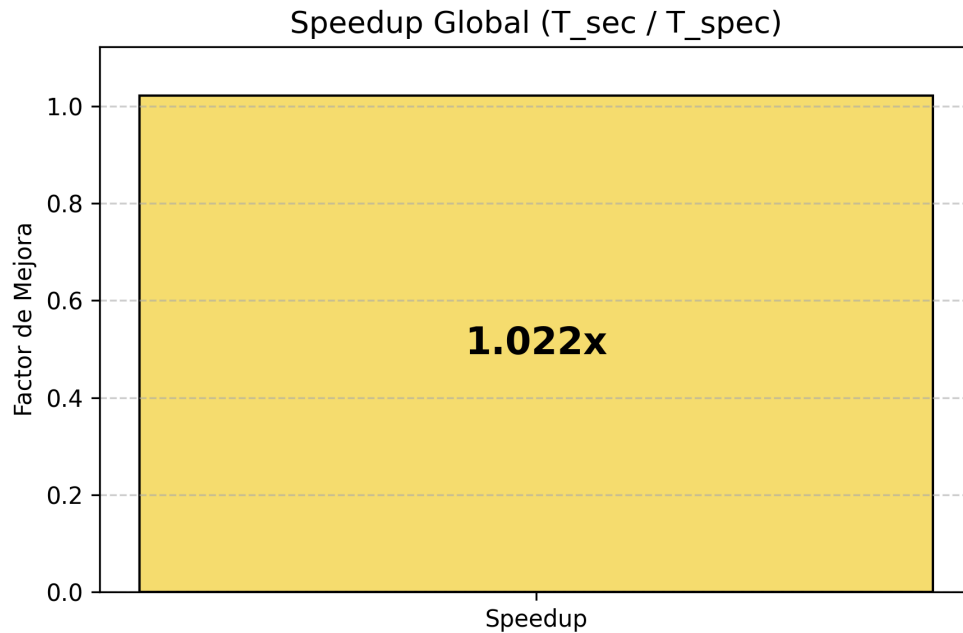
**Figura 3 – Comparación de tiempos promedio:** muestra visualmente la diferencia de tiempo total entre las estrategias.



**Figura 4 – Distribución de tiempos (Boxplot):** evidencia la variabilidad entre corridas y la estabilidad del modo especulativo



**Figura 5 – Evolución temporal:** representa los tiempos de cada corrida (1 a 30), confirmando que la tendencia especulativa se mantiene por debajo de la secuencial.



**Figura 6 – Speedup global:** presenta el factor de mejora total ( $T_{\text{seq}} / T_{\text{spec}} = 1.022\times$ ).

Estas gráficas se generaron automáticamente con el script `graficos_mtricas.py` mediante las librerías Pandas y Matplotlib.

## 6. Interpretación

Los resultados demuestran que el patrón de ejecución especulativa logra una mejora real en tiempo total, aunque moderada, al solapar la decisión costosa con las ramas de ejecución.

El porcentaje de mejora ( $\approx 2\%$ ) es coherente con la naturaleza del problema: la parte más costosa (PoW) domina el tiempo total, por lo que el beneficio depende de cuánto pueda superponerse con la evaluación de la traza.

Aumentar el tamaño de las matrices ( $n$ ) o disminuir la dificultad del PoW incrementaría la proporción de tiempo aprovechable y, por ende, el *Speedup*.

Además, el análisis muestra que el comportamiento especulativo es más consistente (menor dispersión), lo que refleja una utilización más eficiente del CPU en paralelo.

## 7. Conclusión

La implementación concurrente en Go del patrón de ejecución especulativa cumple con todos los objetivos propuestos:

- Ejecuta ramas en paralelo.
- Cancela la perdedora de forma segura.
- Registra métricas detalladas y calcula el Speedup.
- Demuestra una mejora medible frente al enfoque secuencial.

En las pruebas realizadas, la estrategia especulativa alcanzó un *Speedup* de  $1.022\times$ , confirmando una reducción promedio del 2 % en tiempo total.

Se concluye que este patrón es efectivo y que su impacto crece proporcionalmente con la complejidad de la condición o el tamaño de las tareas paralelas.

## 8. Anexos

- Archivos CSV: `metricas_seq.csv` – `metricas_spec.csv`
- Script de generación de gráficos: `graficos_metricas.py`