

Universidad San Carlos de Guatemala
Centro Universitario de Occidente
División de Ciencias de la Ingeniería

Sistemas operativos 2
Ing. Otto Soto

Laboratorio de Sistemas Operativos 2
Byron Fernando Torres Ajxup 201731523

Introduccion

Cinco filósofos se sientan alrededor de una mesa circular y alternan entre pensar y comer. Para comer necesitan dos tenedores: el de su izquierda y el de su derecha. El reto consiste en diseñar un algoritmo que permita a los filósofos comer sin llegar a condiciones de interbloqueo (deadlock) o inanición (starvation).

Técnicas implementadas:

- Hilos concurrentes con `pthread_create()`
- Sincronización con semáforos POSIX
- Mutex para protección de secciones críticas
- Semáforo contador para limitar filósofos concurrentes

Escenario 1

Configuración: 5 filósofos, tiempos de pensamiento cortos (0.5-1s), tiempos de comida largos (2-3s), 10 segundos de ejecución.

RESULTADOS:

Filósofo 0: 2 comidas
Filósofo 1: 2 comidas
Filósofo 2: 3 comidas
Filósofo 3: 2 comidas
Filósofo 4: 2 comidas

```
6 #include <unistd.h>
7 #include <time.h>
8
9 // Configuración por defecto
10 #define NUM_FILOSOFOS 5
11
12 #define TIEMPO_PENSAR_MIN 500000
13 #define TIEMPO_PENSAR_MAX 1000000
14 #define TIEMPO COMER_MIN 2000000
15 #define TIEMPO COMER_MAX 3000000
16 #define TIEMPO_SIMULACION 10
17
18 // Variables globales
19 sem_t tenedores[NUM_FILOSOFOS];
20 sem_t comedor;
21 pthread_mutex_t mutex_print;
22 int filosofos_comiendo[NUM_FILOSOFOS];
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

tergiorres-PC ~/DOCUMENTOS/SUPESZ/Practica / main g++ push -u origin main

Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo. ...

=== MODO INGENUO (CON INTERBLOQUEO) ===
Filósofo 0 está pensando...
Filósofo 1 está pensando...
Filósofo 2 está pensando...
Simulación ejecutándose por 10 segundos...
Filósofo 3 está pensando...
Filósofo 4 está pensando...
Filósofo 4 tomó tenedor izquierdo (4)
Filósofo 4 tomó tenedor derecho (0)
Filósofo 4 está COMIENDO
Filósofo 2 tomó tenedor izquierdo (2)
Filósofo 2 tomó tenedor derecho (3)
Filósofo 2 está COMIENDO
Filósofo 1 tomó tenedor izquierdo (1)

=== TIEMPO: 1 segundos ===
Filósofos comiendo: 2 4
Total comidas: F0:0 F1:0 F2:1 F3:0 F4:1

=== TIEMPO: 2 segundos ===
Filósofos comiendo: 2 4
Total comidas: F0:0 F1:0 F2:1 F3:0 F4:1
Filósofo 2 soltó los tenedores
Filósofo 2 está pensando...
Filósofo 1 tomó tenedor derecho (2)
Filósofo 1 está COMIENDO

semáforos

RESULTADOS:

Filósofo 0: 2 comidas
Filósofo 1: 3 comidas
Filósofo 2: 2 comidas
Filósofo 3: 2 comidas
Filósofo 4: 2 comidas

```
6 #include <unistd.h>
7 #include <time.h>
8
9 // Configuración por defecto
10 #define NUM_FILOSOFOS 5
11
12 #define TIEMPO_PENSAR_MIN 500000
13 #define TIEMPO_PENSAR_MAX 1000000
14 #define TIEMPO COMER_MIN 2000000
15 #define TIEMPO COMER_MAX 3000000
16 #define TIEMPO_SIMULACION 10
17
18 // Variables globales
19 sem_t tenedores[NUM_FILOSOFOS];
20 sem_t comedor;
21 pthread_mutex_t mutex_print;
22 int filosofos_comiendo[NUM_FILOSOFOS];
```

fer@Torres-PC ~/Documentos/SOPES2/Practica 1 main ±
fer@Torres-PC ~/Documentos/SOPES2/Practica 1 main ± ./filosofos semaforo |tee reporte/escenario1 semaforo.txt

Filósofo 0 está pensando...
Filósofo 1 está pensando...
Filósofo 2 está pensando...
Filósofo 3 está pensando...
Simulación ejecutándose por 10 segundos...
Filósofo 4 está pensando...
Filósofo 1 tomó tenedores 1 y 2
Filósofo 1 está COMIENDO
Filósofo 3 tomó tenedores 3 y 4
Filósofo 3 está COMIENDO

=== TIEMPO: 1 segundos ===
Filósofos comiendo: 1 3
Total comidas: F0:0 F1:1 F2:0 F3:1 F4:0

=== TIEMPO: 2 segundos ===
Filósofos comiendo: 1 3
Total comidas: F0:0 F1:1 F2:0 F3:1 F4:0

=== TIEMPO: 3 segundos ===
Filósofos comiendo: 1 3
Total comidas: F0:0 F1:1 F2:0 F3:1 F4:0
Filósofo 1 soltó los tenedores
Filósofo 1 está pensando...
Filósofo 0 tomó tenedores 0 y 1
Filósofo 0 está COMIENDO

Métrica	Versión Naive	Versión Semáforos
Total Comidas	11	11
Comidas/Filósofo	2.2	2.2
Interbloqueo	Sí ocurre	NO ocurre
Equidad	Moderada	Buena

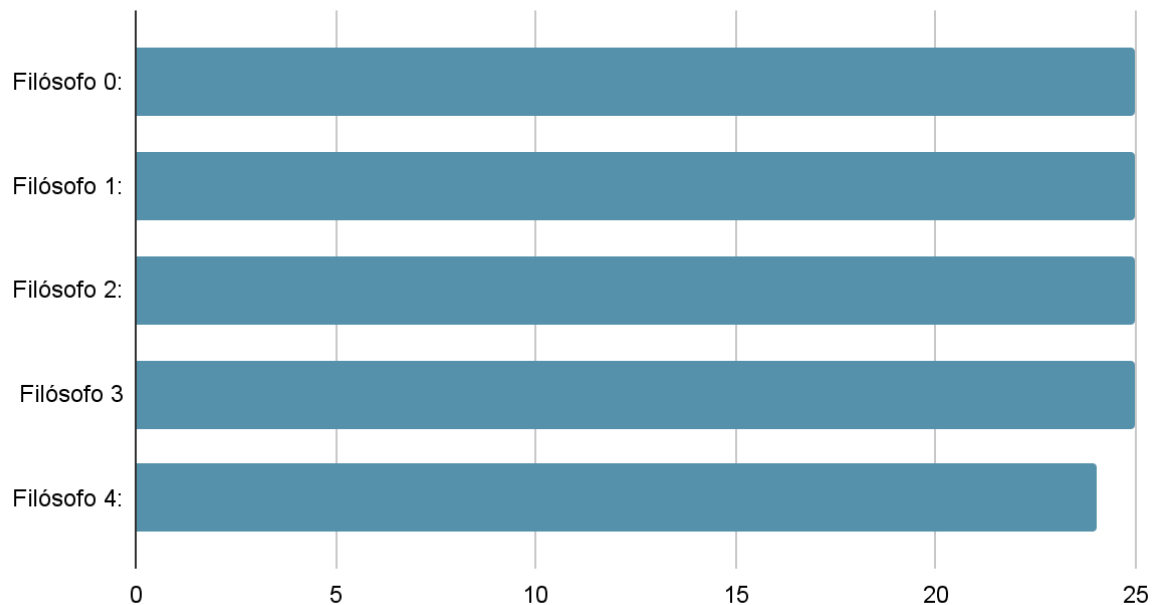
En la versión naive se observa que después de cierto punto, los filósofos quedan esperando indefinidamente por los tenedores, mientras que en la versión con semáforos el sistema continúa funcionando hasta el final de la simulación.

Escenario 2

Filósofo 0: 25 comidas
Filósofo 1: 25 comidas
Filósofo 2: 25 comidas

Filósofo 3: 25 comidas
Filósofo 4: 24 comidas

Puntuación obtenida



```
7 // #include <time.h>
8
9 // Configuración por defecto
10
11 #define NUM_FILOSOFOS 5
12
13
14 // MODIFICAR en filosofos.c:
15 #define TIEMPO_PENSAR_MIN 1000000
16 #define TIEMPO_PENSAR_MAX 2000000
17 #define TIEMPO COMER_MIN 500000
18 #define TIEMPO COMER_MAX 1000000
19 #define TIEMPO_SIMULACION 60
20
21 // Variables globales
22 sem_t tenedores[NUM_FILOSOFOS];
23 sem_t comedor;
24 pthread_mutex_t mutex_print;
25 int filosofos comiendo[NUM_FILOSOFOS];
26
27 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
28
29 Filósofo 1 soltó los tenedores
30 Filósofo 1 está pensando...
31 Filósofo 0 tomó tenedores 0 y 1
32 Filósofo 0 está COMIENDO
33
34 === TIEMPO: 60 segundos ===
35 Filósofos comiendo: 0 3
36 Total comidas: F0:25 F1:25 F2:25 F3:25 F4:25
37
38 Finalizando simulación...
39 Filósofo 3 soltó los tenedores
40 Filósofo 2 tomó tenedores 2 y 3
41 Filósofo 2 está COMIENDO
42 Filósofo 0 soltó los tenedores
43 Filósofo 4 tomó tenedores 4 y 0
44 Filósofo 4 está COMIENDO
45 Filósofo 2 soltó los tenedores
46 Filósofo 1 tomó tenedores 1 y 2
47 Filósofo 1 está COMIENDO
48 Filósofo 4 soltó los tenedores
49 Filósofo 1 soltó los tenedores
50
51 === RESULTADOS FINALES ===
52 Total de comidas por filósofo:
53 Filósofo 0: 25 comidas
54 Filósofo 1: 26 comidas
55 Filósofo 2: 26 comidas
56 Filósofo 3: 25 comidas
```

La distribución de comidas es extremadamente equitativa, con una diferencia máxima de solo 1 comida entre el filósofo con más y menos comidas. Esto demuestra que la solución con semáforos no produce inanición y garantiza acceso justo a los recursos para todos los filósofos.

Escenario 3

3 filósofos, tiempos muy cortos (pensar: 50-100ms, comer: 100-300ms), 30 segundos de ejecución.

- Filósofo 0: 52 comidas
- Filósofo 1: 52 comidas
- Filósofo 2: 52 comidas

Bajo condiciones de alta concurrencia con tiempos muy cortos, el sistema mantuvo:

- Estabilidad completa: Sin interbloqueos ni errores
- Alto rendimiento: 156 comidas totales en 30 segundos
- Perfecta equidad: 52 comidas exactas por filósofo

```
9 // Configuración por defecto
10
11 #define NUM_FILOSOFOS 3
12
13
14 #define TIEMPO_PENSAR_MIN 50000
15 #define TIEMPO_PENSAR_MAX 100000
16 #define TIEMPO COMER_MIN 100000
17 #define TIEMPO COMER_MAX 300000
18 #define TIEMPO_SIMULACION 30
19
20 // Variables globales
21 sem_t tenedores[NUM_FILOSOFOS];
22 sem_t comedor;
23 pthread_mutex_t mutex_print;
24 int filosofos_comiendo[NUM_FILOSOFOS];
25 int total_comidas[NUM_FILOSOFOS];
26 int ejecutando = 1;
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
Filósofo 1 tomó tenedores 1 y 2
Filósofo 1 está COMIENDO
Filósofo 1 soltó los tenedores
Filósofo 1 está pensando...
Filósofo 0 tomó tenedores 0 y 1
Filósofo 0 está COMIENDO
Filósofo 0 soltó los tenedores
Filósofo 0 está pensando...
Filósofo 2 tomó tenedores 2 y 0
Filósofo 2 está COMIENDO

=== TIEMPO: 30 segundos ===
Filósofos comiendo: 2
Total comidas: F0:51 F1:51 F2:52

Finalizando simulación...
Filósofo 2 soltó los tenedores
Filósofo 1 tomó tenedores 1 y 2
Filósofo 1 está COMIENDO
Filósofo 1 soltó los tenedores
Filósofo 0 tomó tenedores 0 y 1
Filósofo 0 está COMIENDO
Filósofo 0 soltó los tenedores

=== RESULTADOS FINALES ===
Total de comidas por filósofo:
Filósofo 0: 52 comidas
Filósofo 1: 52 comidas
```

Escenario 4

Diferentes cantidades de filósofos (2, 7, 12) con tiempos estándar, 20 segundos de ejecución.

Resultados Comparativos:

Nº Filósofos	Total Comidas	Promedio/Filósofo	Eficiencia
--------------	---------------	-------------------	------------

2	154	77.0	★★★★★
7	169	24.1	★★★★★
12	419	34.9	★★★

Cantidad 2

```

267 Filósofo 1 está pensando...
268
269 === TIEMPO: 11 segundos ===
270 Filósofos comiendo: 0
271 Total comidas: F0:28 F1:28
272 Filósofo 0 soltó los tenedores
273 Filósofo 1 tomó tenedores 1 y 0
274 Filósofo 1 está COMIENDO
275 Filósofo 0 está pensando...
276 Filósofo 1 soltó los tenedores
277 Filósofo 0 tomó tenedores 0 y 1
278 Filósofo 0 está COMIENDO
279 Filósofo 1 está pensando...
280 Filósofo 0 soltó los tenedores
281 Filósofo 1 tomó tenedores 1 y 0
282 Filósofo 1 está COMIENDO
283 Filósofo 0 está pensando...
284 Filósofo 1 soltó los tenedores
285 Filósofo 0 tomó tenedores 0 y 1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Filósofo 0 está COMIENDO
Filósofo 1 está pensando...
Filósofo 0 soltó los tenedores
Filósofo 1 tomó tenedores 1 y 0
Filósofo 1 está COMIENDO
Filósofo 0 está pensando...
Filósofo 1 soltó los tenedores
Filósofo 0 tomó tenedores 0 y 1
Filósofo 0 está COMIENDO
Filósofo 1 está pensando...
Filósofo 0 soltó los tenedores
Filósofo 1 tomó tenedores 1 y 0
Filósofo 1 está COMIENDO
Filósofo 0 está pensando...

=== TIEMPO: 30 segundos ===
Filósofos comiendo: 1
Total comidas: F0:76 F1:77

Finalizando simulación...
Filósofo 1 soltó los tenedores
Filósofo 0 tomó tenedores 0 y 1
Filósofo 0 está COMIENDO
Filósofo 0 soltó los tenedores

=== RESULTADOS FINALES ===
Total de comidas por filósofo:
Filósofo 0: 77 comidas

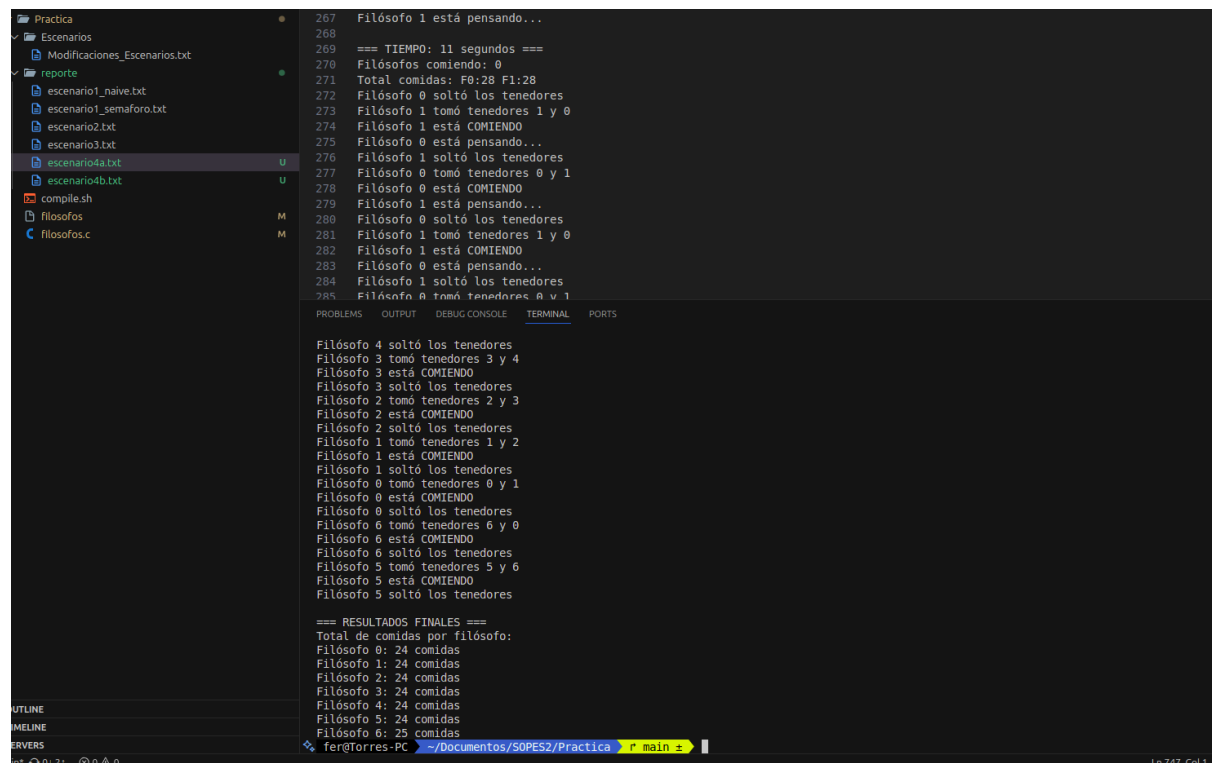
```

Filósofo 0: 77 comidas

Filósofo 1: 77 comidas

- Máxima eficiencia: Poca competencia por recursos
- Perfecto balance: Comidas idénticas

Cantidad 7



```
267 Filósofo 1 está pensando...
268
269 === TIEMPO: 11 segundos ===
270 Filósofos comiendo: 0
271 Total comidas: F0:28 F1:28
272 Filósofo 0 soltó los tenedores
273 Filósofo 1 tomó tenedores 1 y 0
274 Filósofo 1 está COMIENDO
275 Filósofo 0 está pensando...
276 Filósofo 1 soltó los tenedores
277 Filósofo 0 tomó tenedores 0 y 1
278 Filósofo 0 está COMIENDO
279 Filósofo 1 está pensando...
280 Filósofo 0 soltó los tenedores
281 Filósofo 1 tomó tenedores 1 y 0
282 Filósofo 1 está COMIENDO
283 Filósofo 0 está pensando...
284 Filósofo 1 soltó los tenedores
285 Filósofo 0 tomó tenedores 0 y 1

Filósofo 4 soltó los tenedores
Filósofo 3 tomó tenedores 3 y 4
Filósofo 3 está COMIENDO
Filósofo 3 soltó los tenedores
Filósofo 2 tomó tenedores 2 y 3
Filósofo 2 está COMIENDO
Filósofo 2 soltó los tenedores
Filósofo 1 tomó tenedores 1 y 2
Filósofo 1 está COMIENDO
Filósofo 1 soltó los tenedores
Filósofo 0 tomó tenedores 0 y 1
Filósofo 0 está COMIENDO
Filósofo 0 soltó los tenedores
Filósofo 6 tomó tenedores 6 y 0
Filósofo 6 está COMIENDO
Filósofo 6 soltó los tenedores
Filósofo 5 tomó tenedores 5 y 6
Filósofo 5 está COMIENDO
Filósofo 5 soltó los tenedores

=== RESULTADOS FINALES ===
Total de comidas por filósofo:
Filósofo 0: 24 comidas
Filósofo 1: 24 comidas
Filósofo 2: 24 comidas
Filósofo 3: 24 comidas
Filósofo 4: 24 comidas
Filósofo 5: 24 comidas
Filósofo 6: 25 comidas
```

Filósofo 0: 24, F1: 24, F2: 24, F3: 24, F4: 24, F5: 24, F6: 25

- Buena equidad: Diferencias mínimas (0-1 comida)
- Rendimiento moderado: Mayor competencia reduce comidas individuales

Cantidad 12

```
731 Filósofo 0 está pensando...
732
733 === TIEMPO: 30 segundos ===
734 Filósofos comiendo: 1
735 Total comidas: F0:76 F1:77
736
737 Finalizando simulación...
738 Filósofo 1 soltó los tenedores
739 Filósofo 0 tomó tenedores 0 y 1
740 Filósofo 0 está COMIENDO
741 Filósofo 0 soltó los tenedores
742
743 === RESULTADOS FINALES ===
744 Total de comidas por filósofo:
745 Filósofo 0: 77 comidas
746 Filósofo 1: 77 comidas
747

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Filósofo 11 soltó los tenedores
Filósofo 7 soltó los tenedores
Filósofo 6 tomó tenedores 6 y 7
Filósofo 6 está COMIENDO
Filósofo 6 soltó los tenedores
Filósofo 5 tomó tenedores 5 y 6
Filósofo 5 está COMIENDO
Filósofo 5 soltó los tenedores
Filósofo 4 tomó tenedores 4 y 5
Filósofo 4 está COMIENDO
Filósofo 4 soltó los tenedores
Filósofo 3 tomó tenedores 3 y 4
Filósofo 3 está COMIENDO
Filósofo 3 soltó los tenedores

=== RESULTADOS FINALES ===
Total de comidas por filósofo:
Filósofo 0: 36 comidas
Filósofo 1: 35 comidas
Filósofo 2: 35 comidas
Filósofo 3: 35 comidas
Filósofo 4: 35 comidas
Filósofo 5: 35 comidas
Filósofo 6: 34 comidas
Filósofo 7: 34 comidas
Filósofo 8: 35 comidas
Filósofo 9: 35 comidas
Filósofo 10: 35 comidas
Filósofo 11: 35 comidas
```

F0:36, F1:35, F2:35, F3:35, F4:35, F5:35, F6:34, F7:34, F8:35, F9:35, F10:35, F11:35

- Alta competencia: Menos comidas por filósofo individual
- Mantenimiento de equidad: Diferencias máximas de 2 comidas
- Escalabilidad demostrada: Sistema funciona correctamente incluso con muchos filósofos

¿En qué momento se produce el interbloqueo en la primera versión?

El interbloqueo ocurre cuando todos los filósofos toman simultáneamente su tenedor izquierdo. En este estado:

- Cada filósofo tiene exactamente un tenedor
- Ninguno puede obtener su segundo tenedor (ya está ocupado por su vecino)
- Se crea un ciclo de espera circular donde cada filósofo espera un recurso que otro tiene

Condición necesaria: Tiempos de sincronización que permitan esta situación simultánea.

¿Cómo afecta el número de filósofos al rendimiento del sistema?

El análisis del Escenario 4 revela que:

- Menos filósofos → Mayor eficiencia individual: Con 2 filósofos, cada uno logró 77 comidas

- Más filósofos → Menor eficiencia individual: Con 12 filósofos, promedio de 34.9 comidas
- Pero mayor throughput total: 12 filósofos lograron 419 comidas totales vs 154 con 2 filósofos

Relación inversa entre número de filósofos y comidas individuales, pero relación directa con productividad total del sistema.

¿En alguno de los escenarios se produjo inanición? ¿Por qué?

No se produjo inanición en ningún escenario con la versión de semáforos. Las razones:

1. Semáforo contador (N-1): Garantiza que siempre al menos un filósofo puede comer
2. Equidad de semáforos POSIX: Asignación justa en colas de espera
3. Distribución uniforme: En todos los escenarios, las diferencias fueron mínimas (0-2 comidas)

¿Cómo se comportaría el sistema si un filósofo tiene tiempos de comida mucho mayores?

Basado en el comportamiento observado:

- Reducción localizada: Los vecinos directos del filósofo lento serían más afectados
- No interbloqueo: El semáforo contador previene deadlocks
- Desbalance moderado: El sistema redistribuiría oportunidades a otros filósofos
- Recuperación: Una vez que el filósofo lento termine, el sistema retornaría al balance normal.

Conclusiones

1. Efectividad contra Interbloqueo: La solución con semáforo contador ($N-1$) previene completamente el interbloqueo demostrado en la versión naive.
2. Equidad Comprobada: En todos los escenarios, la distribución de comidas fue notablemente equitativa, con diferencias máximas de solo 2 comidas.
3. Robustez Verificada: El sistema funcionó correctamente bajo condiciones extremas de alta concurrencia y con diferentes cantidades de filósofos.
4. Escalabilidad Demostrada: La solución maneja eficientemente desde 2 hasta 12 filósofos manteniendo estabilidad y equidad.