

MAC0422 - Sistemas Operacionais

EP2 - Simulação Multithread da corrida Miss and Out



Nome: Matheus Silveira Feitosa

Considerações Gerais da Implementação

O problema consistiu em simular a modalidade de ciclismo Miss and Out, transformando cada thread em um competidor, com base nesta abordagem, cada ciclista irá competir pelo acesso a pista para poder atualizar sua própria posição.

Com isso, foram definidas duas abordagens, a ingênua e a “eficiente”, onde na primeira, apenas um ciclista podia acessar a pista inteira por vez, e na segunda, foi usada a abordagem descrita a seguir.

Implementação

(Abordagem “eficiente”)

Inicialmente, considerando que a construção do programa como um todo, separa a movimentação em duas partes, a primeira, onde tentamos mover o ciclista uma unidade pra frente e uma pra cima, e em seguida, movemos todos eles uma unidade pra baixo (rumo a faixa mais interna).

Seguindo esta idéia, considerando que, em geral, na segunda fase, apenas um ciclista pode descer por vez, pois este tem que esperar o ciclista abaixo liberar espaço, optou-se por, na abordagem eficiente, ao invés de usar um mutex na pista inteira, usar um mutex por posição, isto é, cada mutex protege exatamente 10 células correspondentes as 10 faixas verticais da pista.

Implementação

(Abordagem “eficiente”)

Com base neste modelo, a movimentação vertical foi implementada, permitindo que apenas um ciclista descesse e ao ser impedido, este verificava se o abaixo dele já havia descido, e então esperava. Repetimos este processo recursivamente para os ciclistas acima, com isso, no segundo turno, cada ciclista trava apenas a própria faixa.

Já a movimentação horizontal, foi implementada como um clássico problema dos filósofos famintos, onde cada thread trava duas colunas da pista de forma alternada, isto é, caso ele esteja numa posição par, então ele trava a sua coluna atual e a próxima, caso ímpar, fazemos o contrário, de forma a evitar deadlock quando todos tentassem pegar a posição seguinte e, similarmente, a escolha por mover sempre pra cima, foi feita por este mesmo motivo, uma vez que o ciclista deve esperar a posição seguinte estar livre para então poder avançar, impedindo o caso onde todos esperam uns aos outros indefinidamente.

Máquinas de teste

De forma a justificar e padronizar alguns dos resultados obtidos, abaixo estão dispostos as especificações da máquina em que foram executados os testes.

Especificações da máquina de testes	
Processador	Processador 11th Gen Intel(R) Core(TM) i5-11300H @ 3.10GHz, 3110 Mhz.
Número de Processadores	4 Núcleos, 8 Processadores Lógicos
Sistema Operacional	Debian GNU/Linux 12 (bookworm) 5.15.167.4-microsoft-standard-WSL2 #1
Memória RAM	Kingston Fury Impact, 8GB, 3200MHz

Resultados dos testes

Para verificarmos seu funcionamento, o algoritmo foi testado com 3 tamanhos diferentes de ciclistas e de pistas com todas as suas combinações. Abaixo estão os resultados com intervalo de confiança de 95%.

Abordagem ingênua			
Número de ciclistas	Tamanhos de pista (metros)		
	100 (Pequena)	200 (Média)	400 (Grande)
50 (Poucos)	Tempo: [5.276s; 5.779s] Memória [2365kB; 3327kB]	Tempo: [10.293s; 11.121s] Memória [2305kB; 3048kB]	Tempo: [21.586s; 22.929s] Memória [2374kB; 3343kB]
100 (Normal)	Tempo: [13.702s; 14.474s] Memória [3036kB ; 4131kB]	Tempo: [26.559s; 28.592s] Memória [2870kB; 3775kB]	Tempo: [52.955s; 56.495s] Memória [3235kB; 4443kB]
200 (Muitos)	Tempo: [32.036s; 33.934s] Memória [5083kB; 6325kB]	Tempo: [65.649s; 68.151s] Memória [5111kB; 6366kB]	Tempo: [136.084s; 142.870s] Memória [4793kB; 6022kB]

Abordagem “Eficiente”			
Número de ciclistas	Tamanhos de pista (metros)		
	100 (Pequena)	200 (Média)	400 (Grande)
50 (Poucos)	Tempo: [5.191s; 5.596s] Memória [2296kB; 3140kB]	Tempo: [9.947s; 10.990s] Memória [2619kB; 3801kB]	Tempo: [20.598s; 21.997s] Memória [2355kB; 3359kB]
100 (Normal)	Tempo: [13.191s; 14.042s] Memória [2902kB; 3805kB]	Tempo: [26.442s; 28.210s] Memória [2843kB; 3667kB]	Tempo: [52.916s; 56.390s] Memória [2975kB; 3962kB]
200 (Muitos)	Tempo: [31.646s; 33.536s] Memória [4821kB; 6059kB]	Tempo: [65.602s; 68.326s] Memória [5331kB; 6608kB]	Tempo: [136.096s; 141.570s] Memória [4501kB; 5560kB]

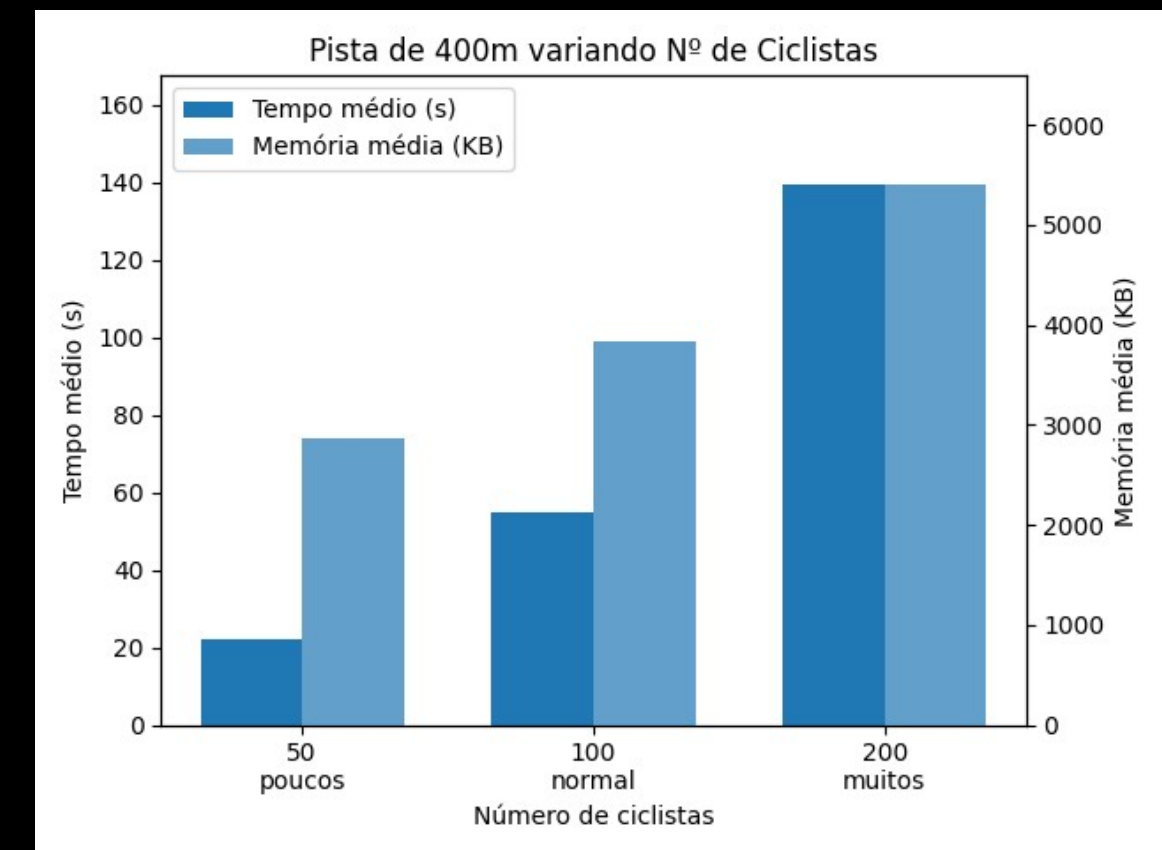
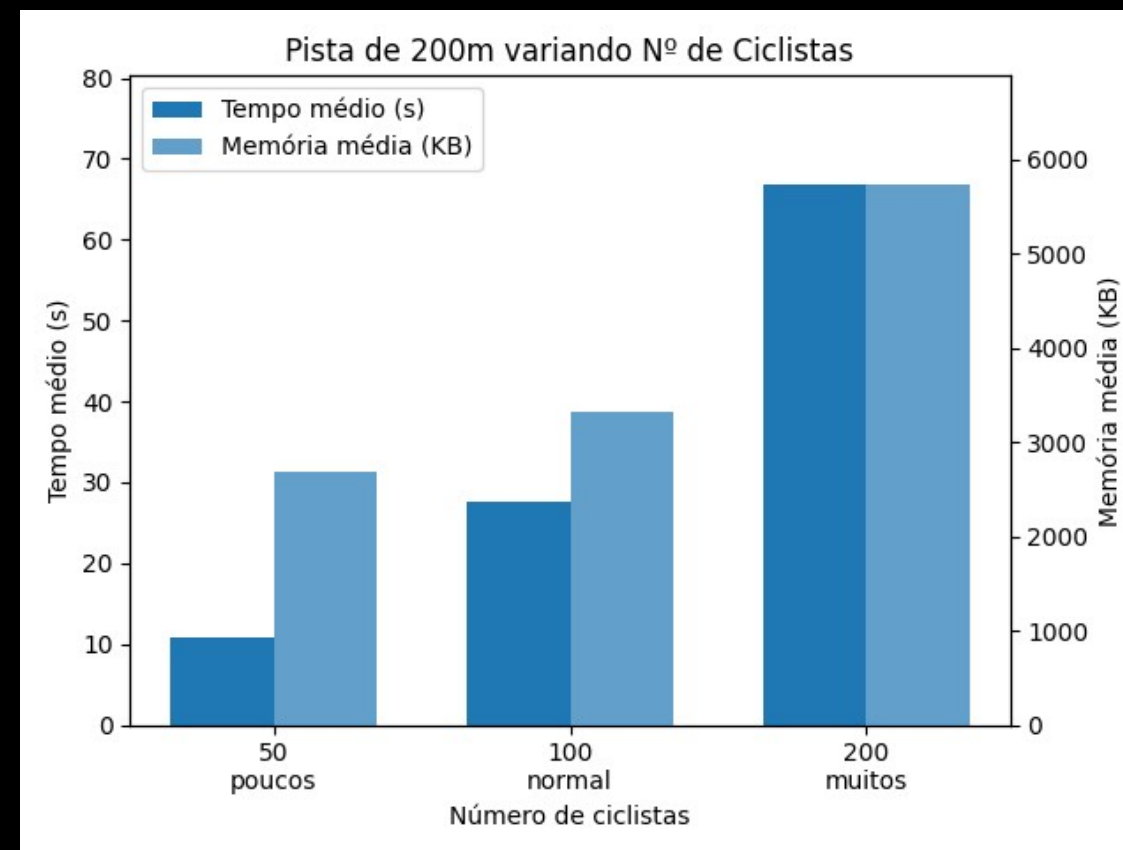
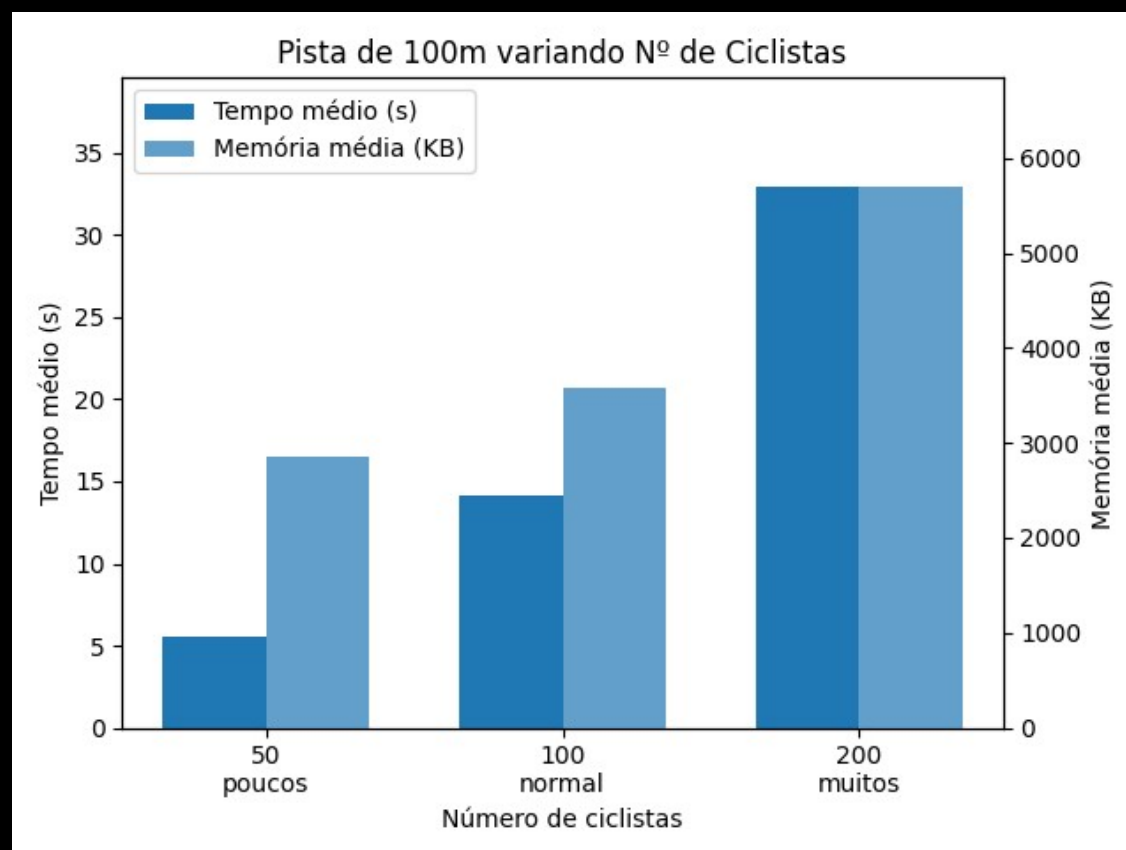
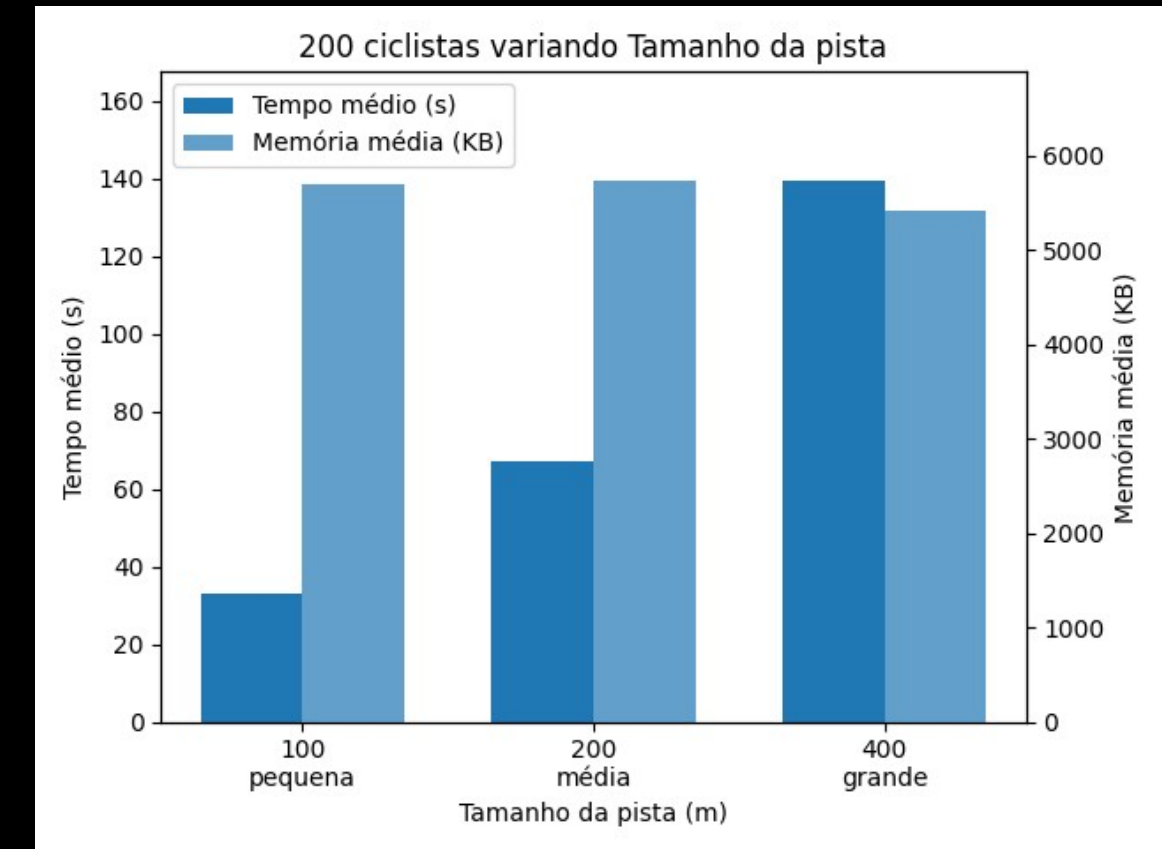
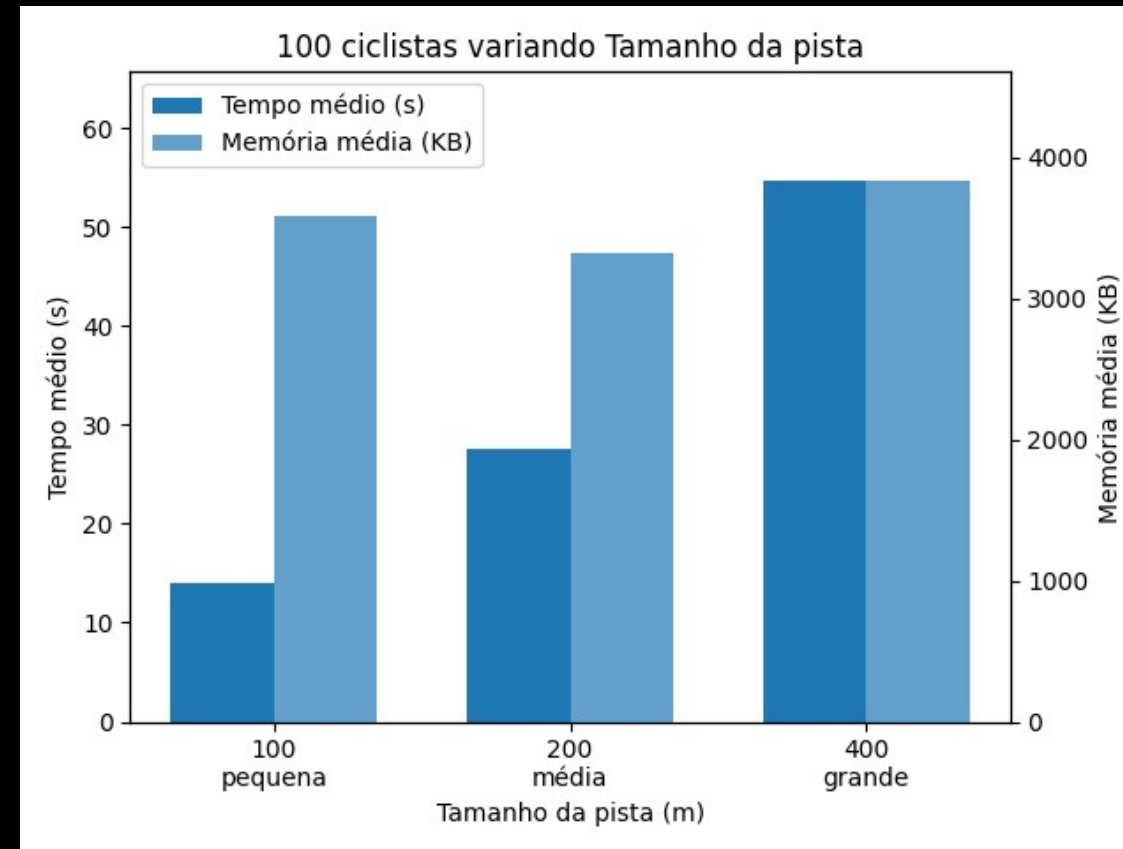
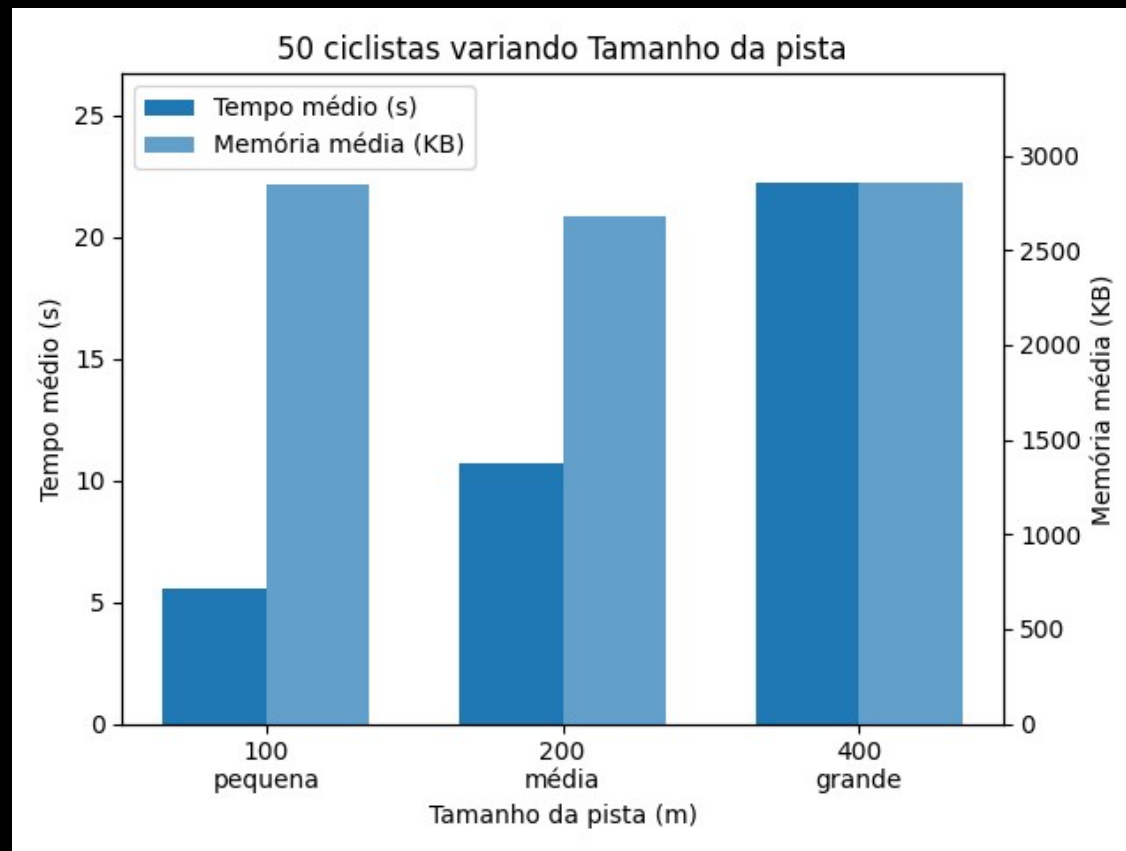
Resultados dos testes

E de forma a facilitar a comparação direta entre os dados, abaixo colocamos as médias simples de cada um dos testes.

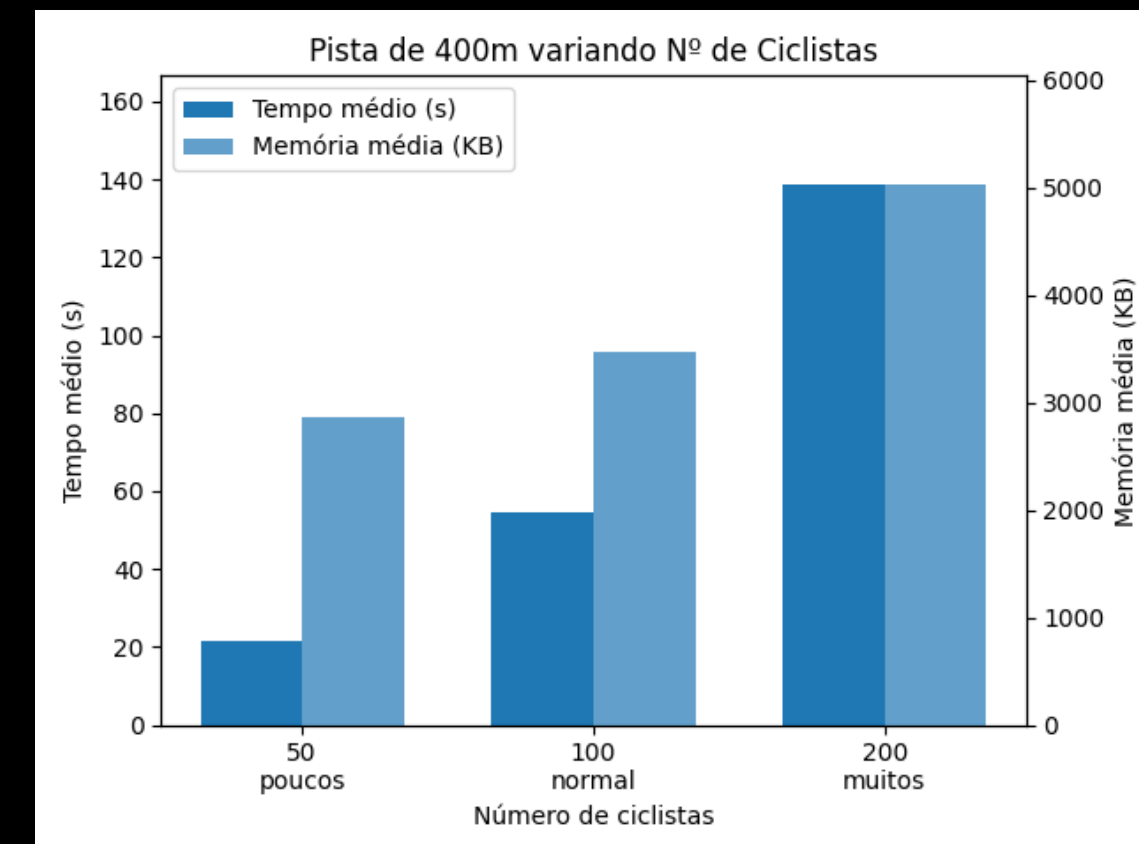
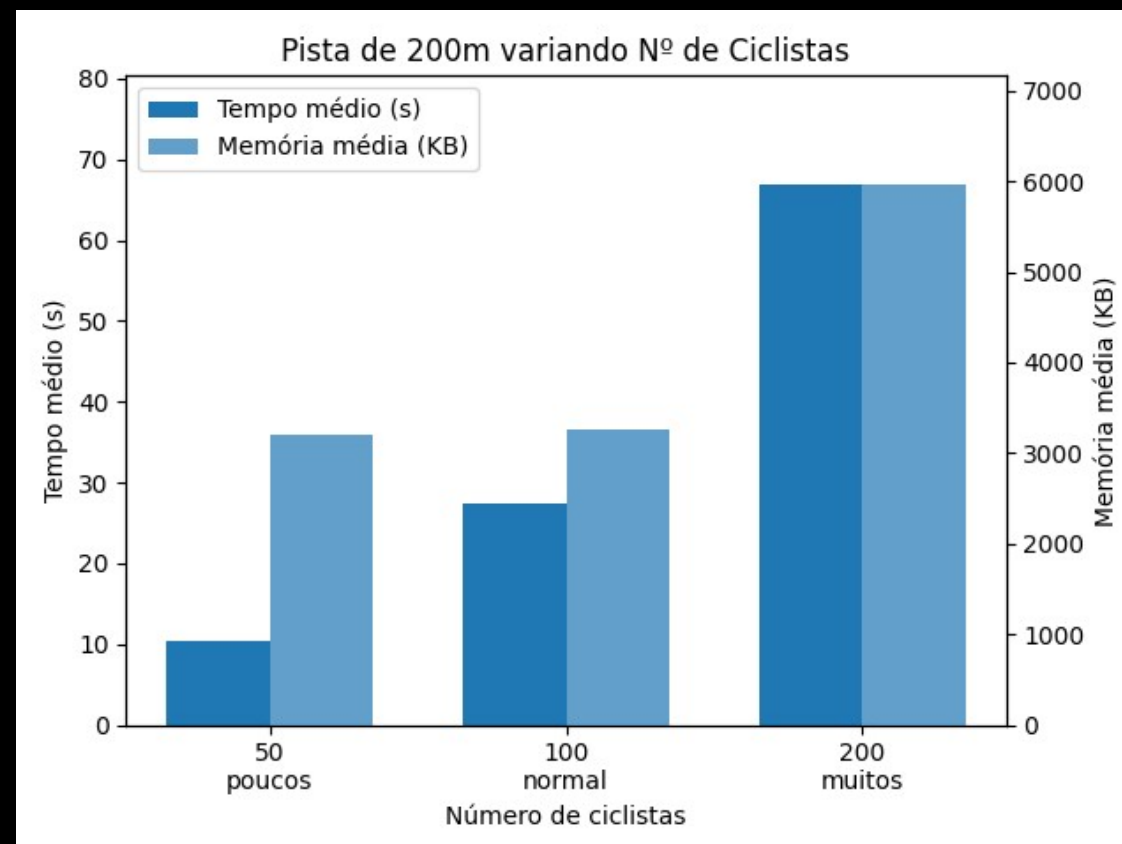
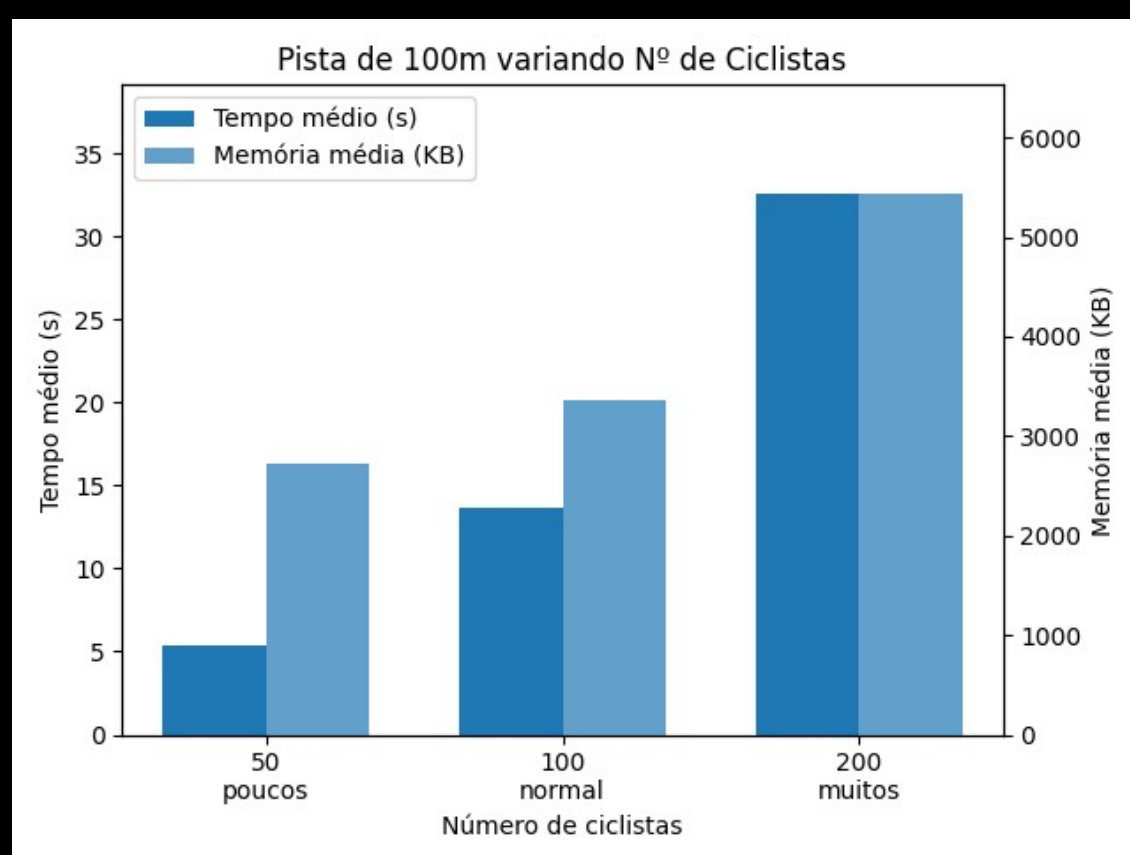
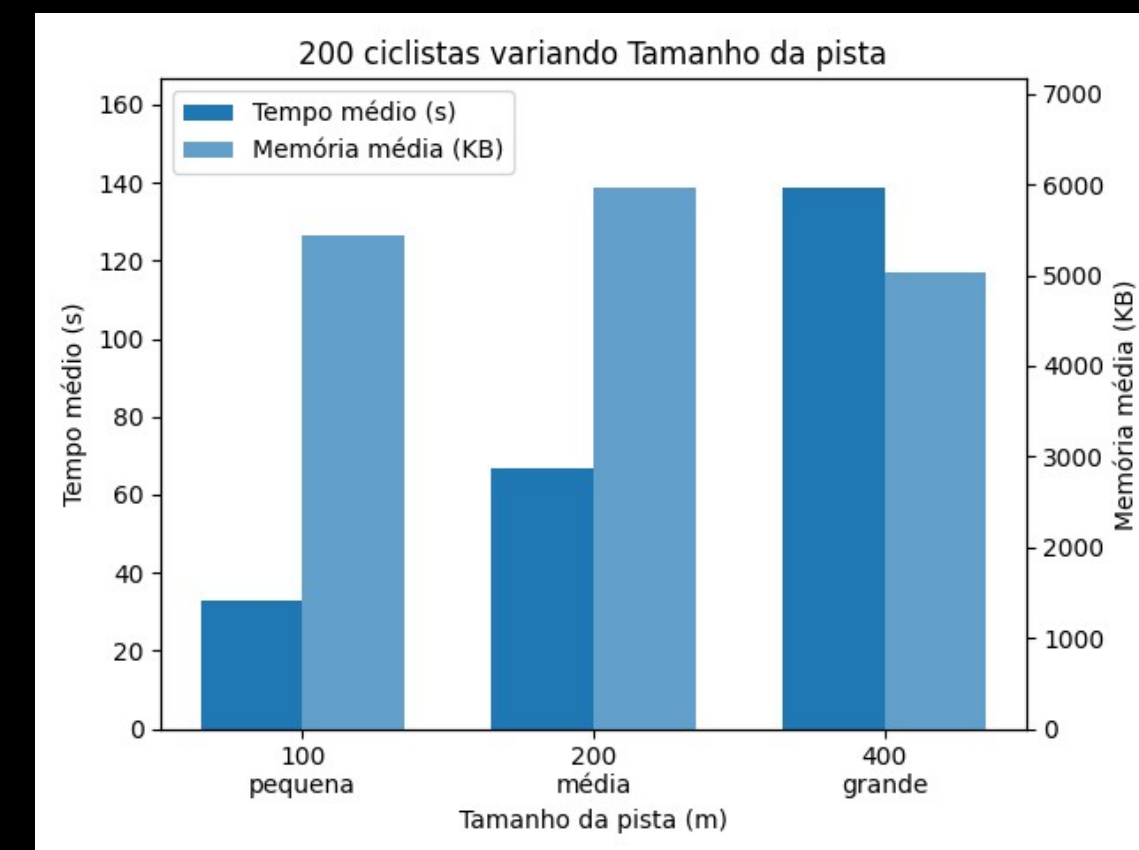
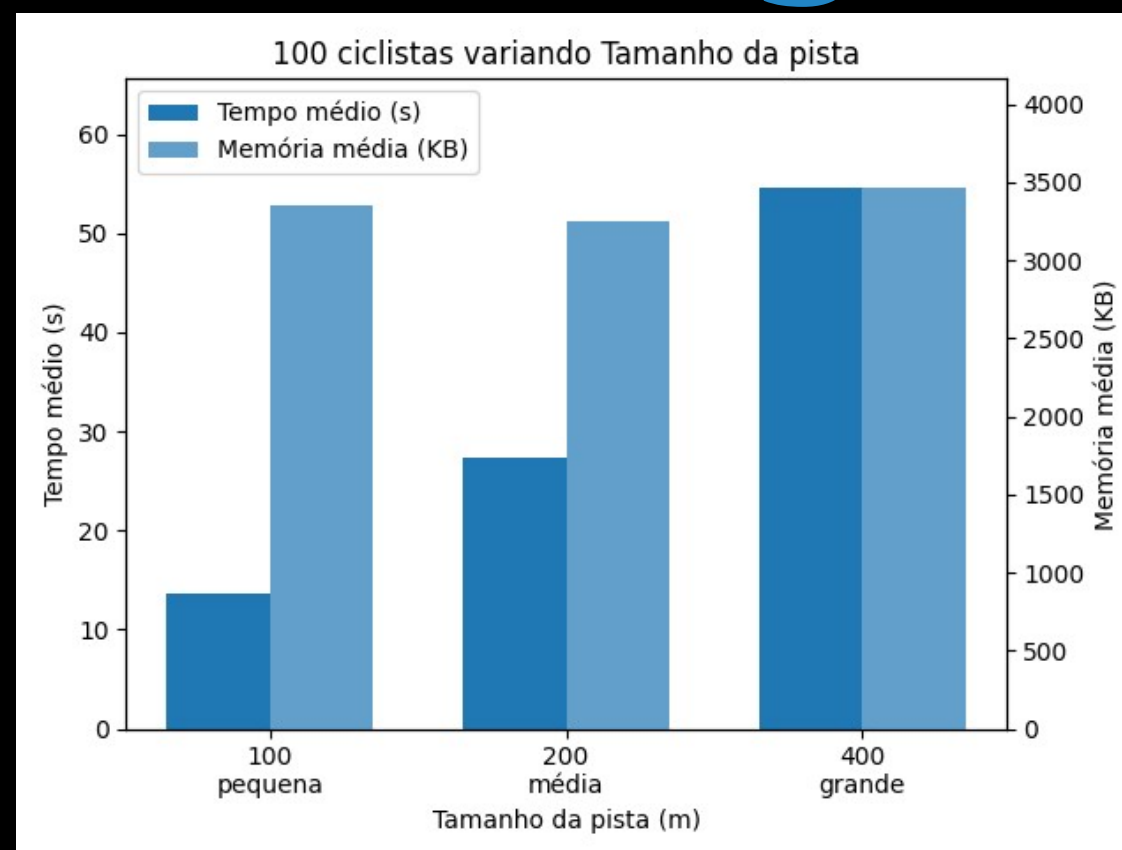
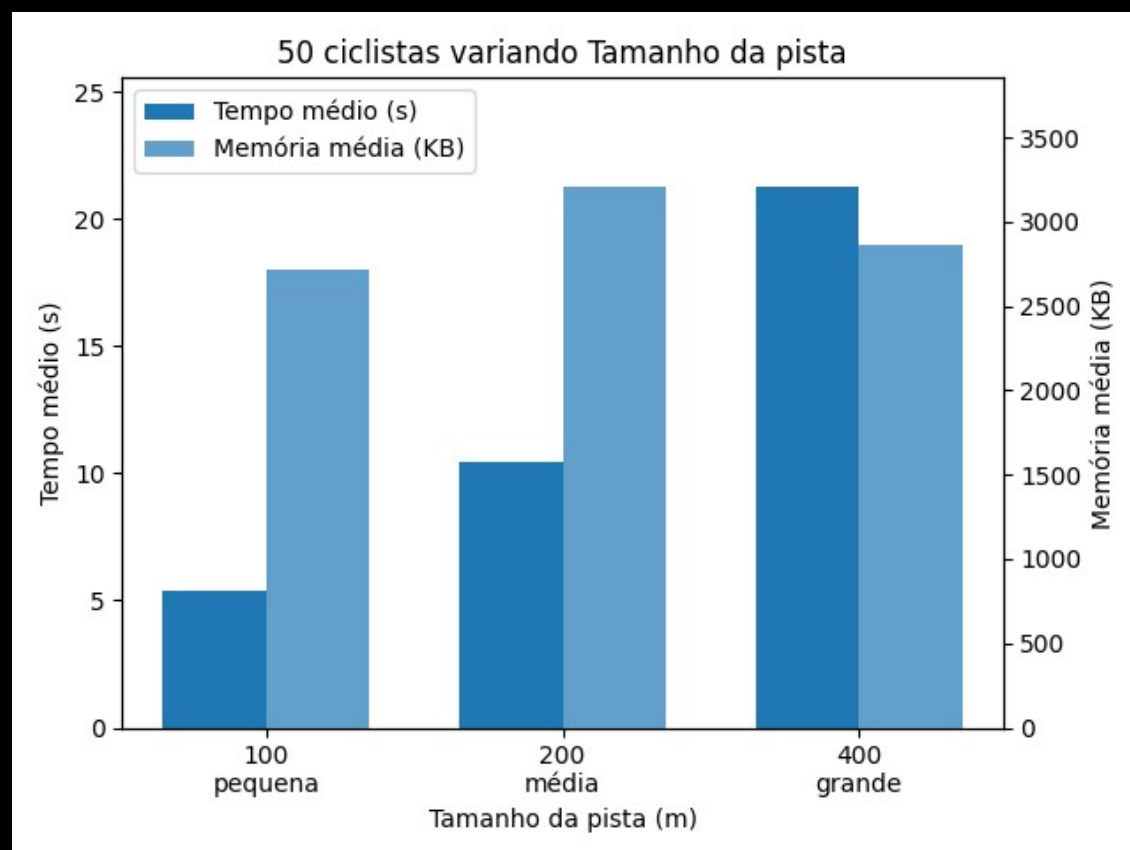
Abordagem ingênua			
Número de ciclistas	Tamanhos de pista (metros)		
	100 (Pequena)	200 (Média)	400 (Grande)
50 (Poucos)	Tempo: 5.527s Memória: 2846kB	Tempo: 10.707s Memória: 2677kB	Tempo: 22.257s Memória: 2859kB
100 (Normal)	Tempo: 14.088s Memória: 3584kB	Tempo: 27.575s Memória: 3323kB	Tempo: 54.725s Memória: 3839kB
200 (Muitos)	Tempo: 32.985s Memória 5704kB	Tempo: 66.900s Memória 5738kB	Tempo: 139.477s Memória 5407kB

Abordagem “Eficiente”			
Número de ciclistas	Tamanhos de pista (metros)		
	100 (Pequena)	200 (Média)	400 (Grande)
50 (Poucos)	Tempo: 5.394s Memória 2718kB	Tempo: 10.468s Memória 3210kB	Tempo: 21.297s Memória 2857kB
100 (Normal)	Tempo: 13.617s Memória 3354kB	Tempo: 27.326s Memória 3255kB	Tempo: 54.653s Memória 3468kB
200 (Muitos)	Tempo: 32.591s Memória 5440kB	Tempo: 66.964s Memória 5970kB	Tempo: 138.833s Memória 5031kB

Resultados (Abordagem Ingênua)



Resultados (Abordagem “Eficiente”)



Análise dos Resultados

Abordagem Ingênua x Abordagem “Eficiente” - Resultados inesperados observados

De maneira geral, as diferenças entre as duas abordagens, estão apenas na forma como o acesso a pista é feito, sendo outras funcionalidades, como espera pelo movimento do ciclista, ou então, o gerenciamento de voltas essencialmente idênticos.

Com isso, era esperado inicialmente duas coisas, primeiro, que a abordagem “eficiente” iria consumir mais memória (pelo uso de mais mutexes), e segundo, que será significativamente mais rápida do que a abordagem “ingênua”, (por permitir que mais de um ciclista se movimente por vez), entretanto, na prática, tal expectativa não foi cumprida.

- A começar pelo tempo de execução, foi observado que no geral, o método “eficiente” era de fato mais rápido, mas a diferença de tempos era de no máximo 1 segundo entre os dois, com destaque para o teste com a pista grande e poucos ciclistas, tal fenômeno entretanto, pode ser explicado tendo em vista que, devido a lotação a qual tais pistas foram submetidas, temos que mesmo na abordagem “eficiente”, não necessariamente todas as threads em execução, conseguiriam acessar a pista simultaneamente, de forma a aproximar os tempos de execução de ambas as abordagens.
- Com respeito a discrepância na quantidade de memória, podemos justificar tal observação, considerando que, a cada vez que uma thread é suspensa, a fila de threads suspensas aumenta, assim como a quantidade de informações que precisam ser guardadas, para que tais threads sejam acordadas, com isso, devido ao fato de o acesso exclusivo a pista inteira implicar num maior número de threads suspensas em um mesmo ponto, temos que a estrutura de dados que fará a gestão deste único ponto, também irá aumentar consideravelmente, influenciando no uso de memória.

Análise dos Resultados

Abordagem Ingênua x Abordagem “Eficiente” - Resultados esperados observados

Para além dos resultados esperados, já observados como por exemplo, o aumento mais significativo no uso de memória, conforme aumentamos o número de ciclistas frente ao tamanho da pista, ou então, o aumento no tempo de simulação com o aumento, seja do número de ciclistas, seja do tamanho da pista.

Ainda ocorreram, devido a explicação dos fenômenos inesperados anteriormente, uma melhor justificativa de outros resultados observados, como segue:

- Primeiramente, tivemos que, ao dobrarmos o número de ciclistas ou o tamanho da pista, observamos que, o tempo de simulação é dobrado, sugerindo que a complexidade da simulação é de ordem $O(dk)$, onde d é o tamanho da pista e k é o número de ciclistas, o que era esperado pois, conforme aumentamos o tamanho da pista, aumentamos linearmente a quantidade de movimentos que cada ciclista terá que fazer para completá-la, isto é, cada ciclista k , deverá fazer d movimentações, para completar uma volta.
- Outro ponto que também era esperado, e foi observado, é que, o aumento no uso de memória, era mais significativo conforme aumentávamos o número de ciclistas do que o tamanho da pista, o que era esperado, pois a maioria das estruturas é mais fortemente dependente do número de ciclistas, como por exemplo, o próprio sistema de controle de voltas, que são criadas 2 estruturas para cada ciclista, enquanto que a única estrutura dependente do tamanho da pista, é a própria pista e, para a abordagem “eficiente”, a quantidade de mutexes, o que é evidenciado especialmente nos gráficos, onde, para uma mesma quantidade de ciclistas, os usos de memória são praticamente os mesmos.