



# EP2 - Corrida

Ísis Ardisson Logullo 7577410

15/05/25

# Arquitetura

Com os parâmetros  $d$  e  $k$  cria-se um vetor de tamanho  $k$  de Ciclistas (struct), contendo:

- Posição que o ciclista se encontra na pista - linha x coluna ( $m$  e  $n$ ).
- Velocidade do ciclista (1 para 30km/h, 2 para 60km/h) ( $v$ ).
- Quantidade de voltas que o ciclista completou (volta).
- Quanto tempo ele pedalou (tempo).
- Indicadores: se ele quebrou (quebrado), se ele foi eliminado (desclassificado) e se ele executou todas as voltas até o final (terminou).

A classificação foi contada volta a volta, utilizando um vetor de pilhas de tamanho  $2 * k$  (quantidade máxima de voltas). Quando um ciclista completa uma volta, ele entra na pilha da volta. Ao completar uma volta de eliminação, achamos o primeiro da pilha que ainda não quebrou, não foi eliminado e está na linha de chegada para ser eliminado.

# Funcionalidades

A pista é construída por uma matriz de  $d$  colunas e  $k$  linhas. As posições dos ciclistas são atualizadas de 0 até  $d-1$  na matriz (esquerda para a direita) . O acesso a pista é feito através de uma matriz de mutex de mesmo tamanho da pista, um mutex para cada posição  $(i, j)$  da matriz.

A cada intervalo de tempo (60ms ou 20ms), as threads dos ciclistas executam operações e atualizam suas posições na matriz. Após isso, elas ficam presas em uma barreira e uma decisão de posição é feita.

A decisão de posição adiciona um intervalo de tempo, para depois permitir que os ciclistas pedalem (suficiente para o ciclista mover 1m). Ela também verifica quebras e eliminações.

Após restar somente um ciclista na pista a corrida acaba e a classificação final surge. É ordenado o vetor de ciclistas com base na quantidade de voltas percorridas e, caso houver empate, pelo tempo pedalado.

# Funcionalidades

Para realizar a ultrapassagem: se os ciclistas conseguirem avançar para frente, eles avançam. Caso contrário, eles verificam se existem duas posições consecutivas livres nas faixas mais externas as que ele está.

Se isso for possível, então ele avança. Senão, ele se move na mesma velocidade que o ciclista da frente, isto é, aguarda o ciclista da frente avançar.

Foram utilizadas barreiras do pthread.h e um vetor *continua*, de k posições, que indica se, após a decisão de posição, cada ciclista pode ou não continuar a corrida.

Além da matriz de mutex para o acesso à pista, foi utilizado um mutex para cada posição do vetor de pilhas (garante um acesso por vez de cada ciclista). Há um mutex que protege a função de aleatoriedade para não ocorrer um vício.

A primeira volta é a 0 e é calculado a probabilidade de um ciclista quebrar assim que ele chega em uma volta múltipla de 6, se ele quebrar deixa de aparecer nos próximos rankings. Após um ciclista completar todas as voltas que deveria pedalar, sua thread fica em execução.

# Resultados

Foram usados os tamanhos de pistas: pequeno 100m, médio 200m e grande 400m.

Foram usados o número de ciclistas: poucos 5, médio 10 e muitos 20.

Números maiores que isso demoravam muito tempo, deixando impossível testar várias vezes.

Foram gerados gráficos com intervalo de confiança de 95% para cada quantidade de ciclistas em todos os tamanhos de pista. Isso para cada tipo de acesso a pista: ingênua e efetiva.

Pequeno d = 100m :

---

k = 5

Ingênua - media: 104,29s	dp: 6,12	ic: 101,694s a 106,885s
Eficiente - media: 115,613s	dp: 7,455	ic: 112,451s a 118,775s

---

k= 10

Ingênua - media: 167,003s	dp: 12,561	ic: 161,676s a 172,330s
Eficiente - media: 166,462s	dp: 12,174	ic: 161,299s a 171,625s

---

k= 20

Ingênua - media: 301,419s	dp: 23,49	ic: 291,457s a 311,381s
Eficiente - media: 335,68s	dp: 33,049	ic: 321,664s a 349,696s

# Resultados

Médio d = 200m :

k = 5

Ingênua

media: 163,445s dp: 12,981

ic: 157,940s a 168,950s

Eficiente

media: 177,919s dp: 5,273

ic: 175,683s a 180,155s

k= 10

Ingênua

media: 358,551s dp: 53,276

ic: 335,956s a 381,146s

Eficiente

media: 317,55s dp: 49,815

ic: 296,423s a 338,677s

k= 20

Ingênua

media: 654,38s dp: 44,025

ic: 635,709s a 673,051s

Eficiente

media: 735,48s dp: 78,003

ic: 702,398s a 768,562s

Grande d = 400m :

k = 5

Ingênua

media: 318,636s dp: 32,95

ic: 304,662s a 332,610s

Eficiente

media: 396,691s dp: 33,235

ic: 382,596s a 410,786s

k= 10

Ingênua

media: 656,057s dp: 43,986

ic: 637,402s a 674,712s

Eficiente

media: 1004,057s dp: 222,782

ic: 909,573s a 1098,540s

k= 20

Ingênua

media: 2539,24s dp: 288,603

ic: 2416,841s a 2658,638s

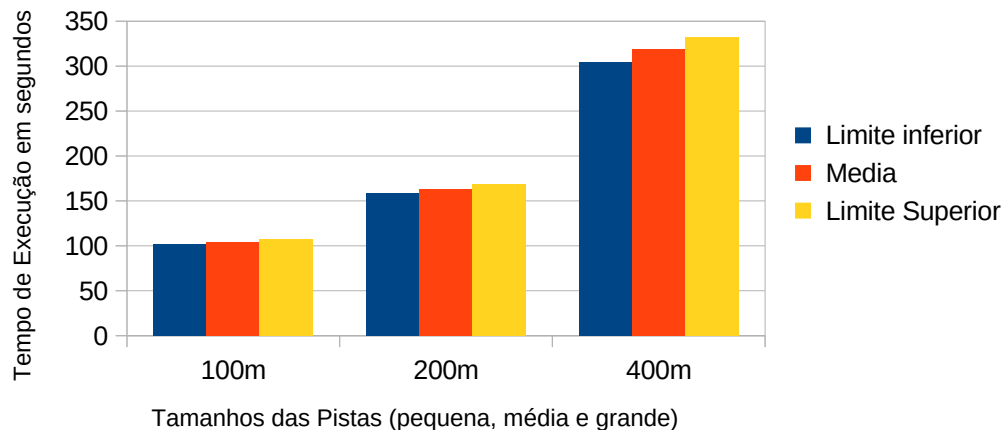
Eficiente

media: 1200,291s dp: 63,916

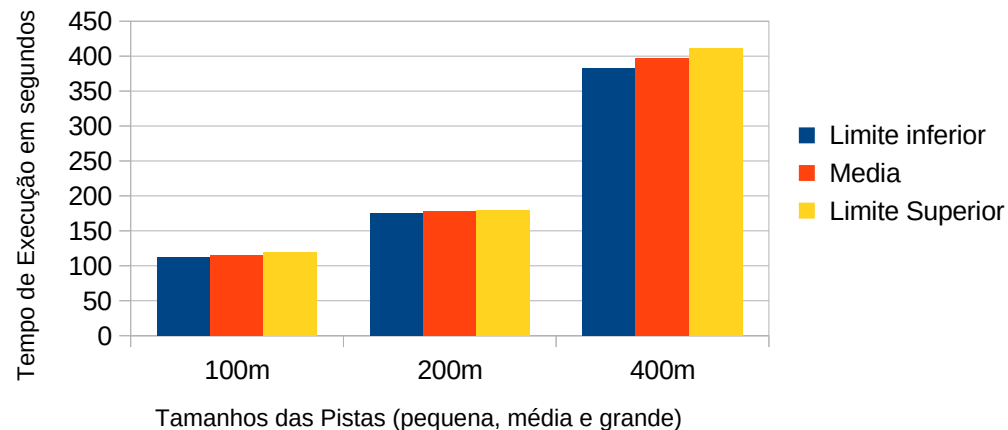
ic: 1173,184s a 1227,398s

# Gráficos: 5 Ciclistas

Tempo de Execução X Tamanho da Pista: 5 Ciclistas - Ingênuo

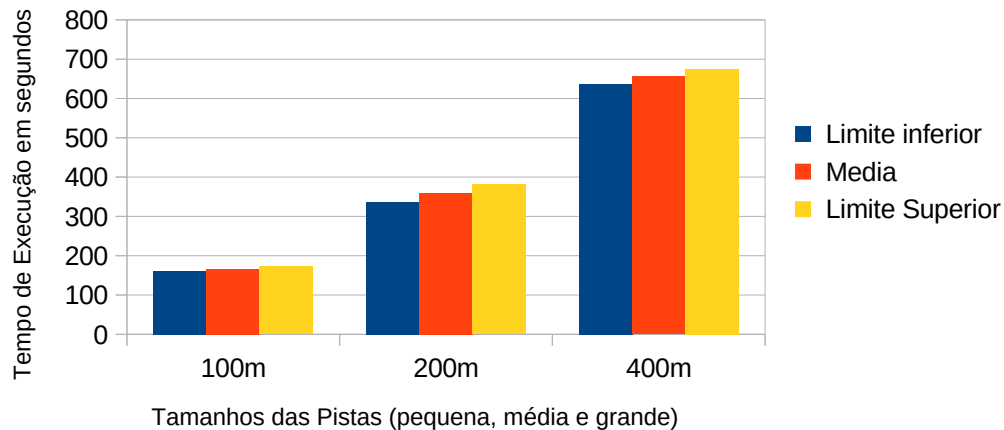


Tempo de Execução X Tamanho da Pista: 5 Ciclistas - Eficiente

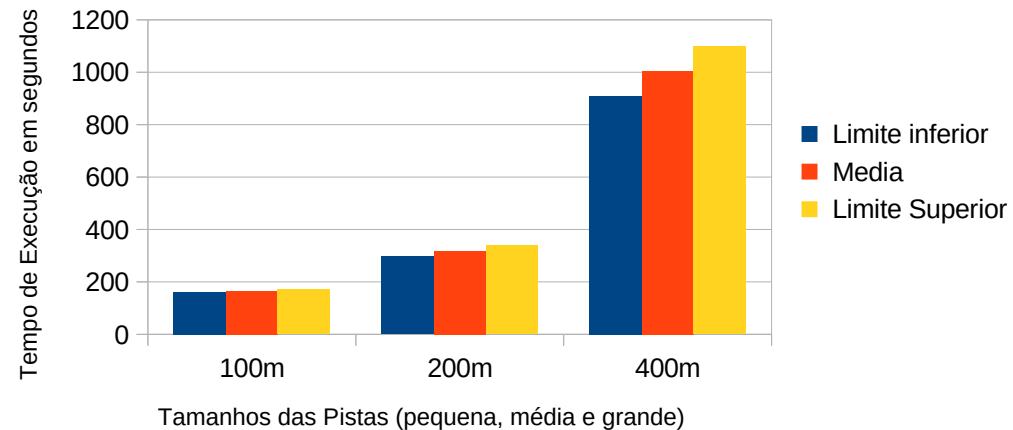


# Gráficos: 10 Ciclistas

Tempo de Execução X Tamanho da Pista: 10 Ciclistas  
- Ingênua



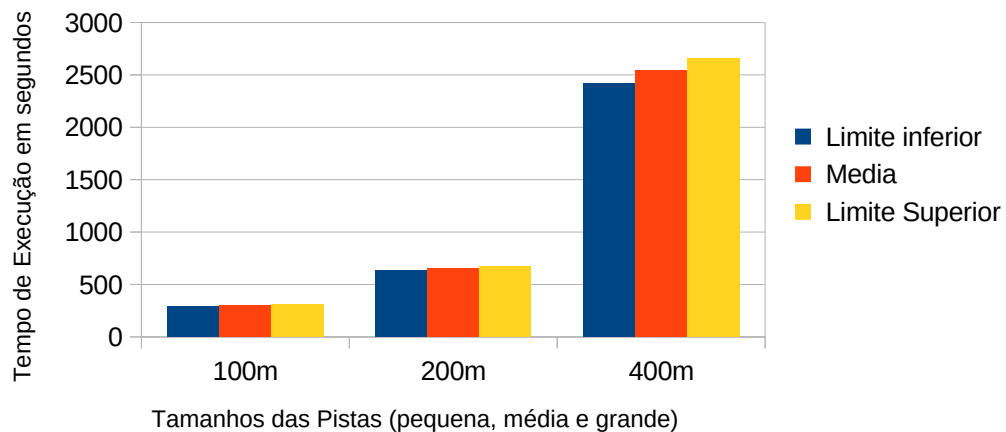
Tempo de Execução X Tamanho da Pista: 10 Ciclistas  
- Eficiente



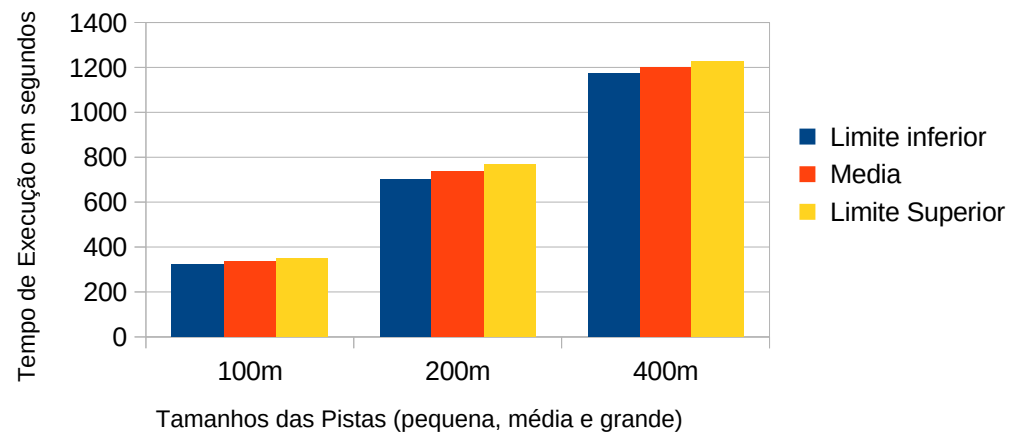


# Gráficos: 20 Ciclistas

Tempo de Execução X Tamanho da Pista: 20 Ciclistas  
- Ingênua



Tempo de Execução X Tamanho da Pista: 20 Ciclistas  
- Eficiente



# Conclusão

Os resultados dos testes foram os esperados, em sua maioria. Conforme aumentamos a quantidade de ciclistas, também aumenta-se o uso de memória. Isso ocorre porque a maioria das alocações depende do número de ciclistas.

No caso de diferentes tamanhos de pistas, a diferença de uso de memória foi mínima. Já que somente a matriz que representa a pista e a matriz de semáforos dependem do tamanho da pista. Assim, não houve tanta diferença para diferentes tamanhos de pista com a mesma quantidade de ciclistas.

Com relação ao tempo de execução: os testes mostraram que o tamanho da pista e a quantidade de ciclistas influenciam diretamente no tempo. Nos testes para diferentes quantidades de ciclistas, houve uma diferença maior do que nos testes para diferentes tamanhos de pistas. Isso ocorreu pois por conta das ordens de grandezas diferentes para pistas e ciclistas.

Como os testes com muitos ciclistas são 4 vezes maiores do que com pequenos ciclistas, é esperado que o programa rode 4 vezes mais lento, só que os valores obtidos foram menores que isso, devido aos ciclistas quebrados e às aleatoriedades.