

# MAC0422 - EP2

Gabriel de Russo e Carmo  
Victor Wichmann Raposo  
Outubro de 2016

# Sumário

- Visão geral
- Problemas e soluções
- Resultados dos experimentos

# Visão Geral

- Três módulos
  - ep2 - principal
  - corrida - manutenção da corrida
  - utilitario - auxílio (constantes, debug ...)

# Visão Geral

- $2n$  threads
- Sem coordenador
- Vetor pista e vetor de ciclistas
- 1 u.t. = 60 ms

# Problemas e Soluções

- Problema: Apenas dois ciclistas lado a lado
- Solução: Vetor pista tem duas dimensões (faixas)  
pista[d][2]

# Problemas e Soluções

- Problema: No início todos os ciclistas começam lado a lado
- Solução: Os ciclistas começam enfileirados no final da volta -1 (com exceção do primeiro)

# Problemas e Soluções

- Problema: Apenas dois ciclistas por metro na pista podem causar congestionamento dependendo da ordem de execução (discutido no PACA)
- Solução: Não permitir que o ciclista apressado se desloque

# Problemas e Soluções

- Problema: O tempo decorrido para cada ciclista deve ser o mesmo (sincronização de threads)
- Solução: Barreira de sincronização



# Barreira

- Variável de condição (`pthread_cond_t`)
- Contador de threads que alcançaram a barreira
- Threads que chegam esperam continue (`pthread_cond_wait`)
- Última a chegar dá continue (`pthread_cond_broadcast`)
- O contador é zerado com aritmética modular
- Threads que serão destruídas entram na barreira uma última vez

# Problemas e Soluções

- Problema: Velocidade de 30 km/h permite que ciclistas se desloquem apenas meio metro
- Solução: Flag que indica se o ciclista está no meio do metro, que é levada em conta na atualização da posição

# Problemas e Soluções

- Problema: Quebra aleatória restrita
- Solução: Vetor de flags para indicar se houve uma quebra na volta e contadores de ciclistas por equipe

# Problemas e Soluções

- Problema: Sorteio de velocidade restrito
- Solução: Flag para marcar se o ciclista restringe a velocidade dos demais atrás dele. Restringe apenas se lhe for sorteado 30 km/h.

# Problemas e Soluções

- Problema: Muitas operações necessitam a ordem relativa das equipes
- Solução: Vetor de permutação para cada equipe e posição na equipe para cada ciclista

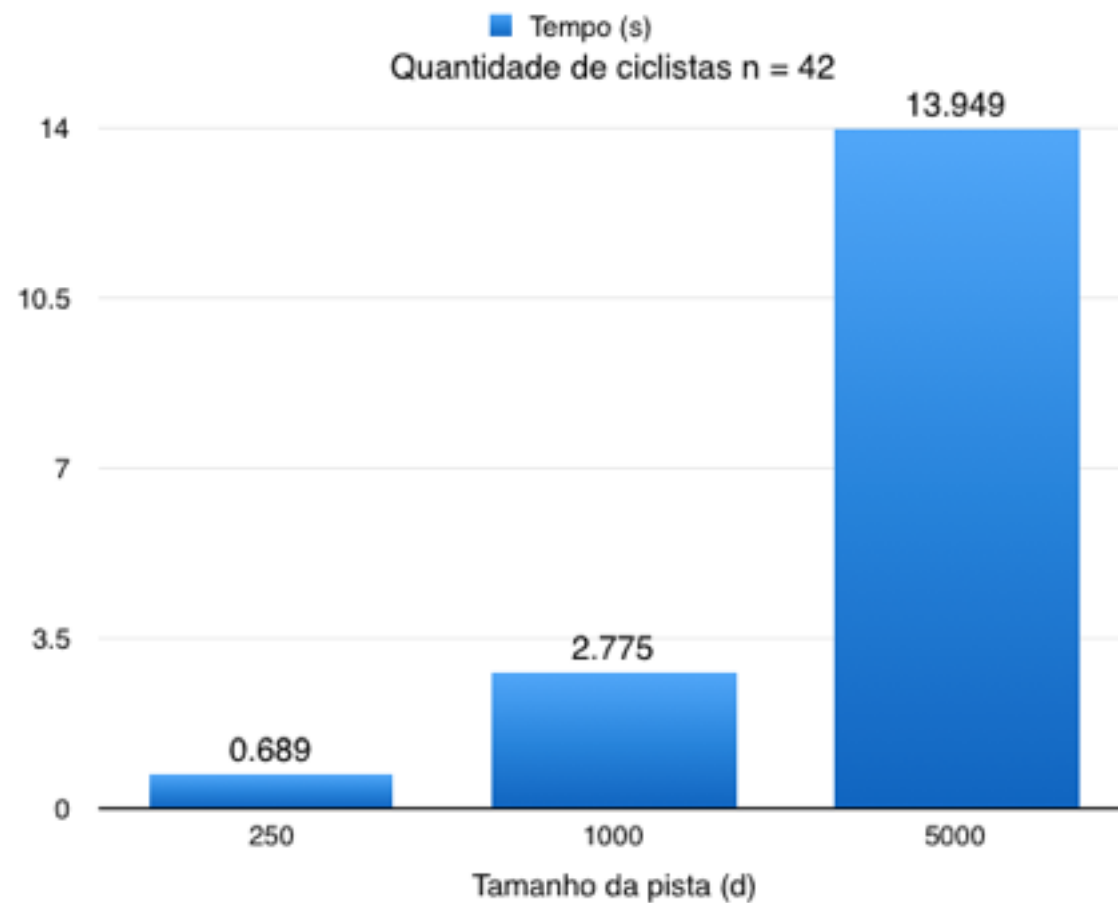
# Problemas e Soluções

- Problema: Ultrapassagens
- Solução: Ao atualizar a posição verificamos se houve ultrapassagem
  - Mesma equipe: Manter a ordem relativa, cuidado com retardatário
  - Equipes diferentes: Só importam os terceiros

# Resultado dos experimentos

- testes foram feitos num Ubuntu com i7 @ 3.4ghz
- 30 vezes para cada teste
- uso de memória foi medido pelo maximum resident set size
- fixamos **n** e variamos **d**
- fixamos **d** e variamos **n**

# n fixo



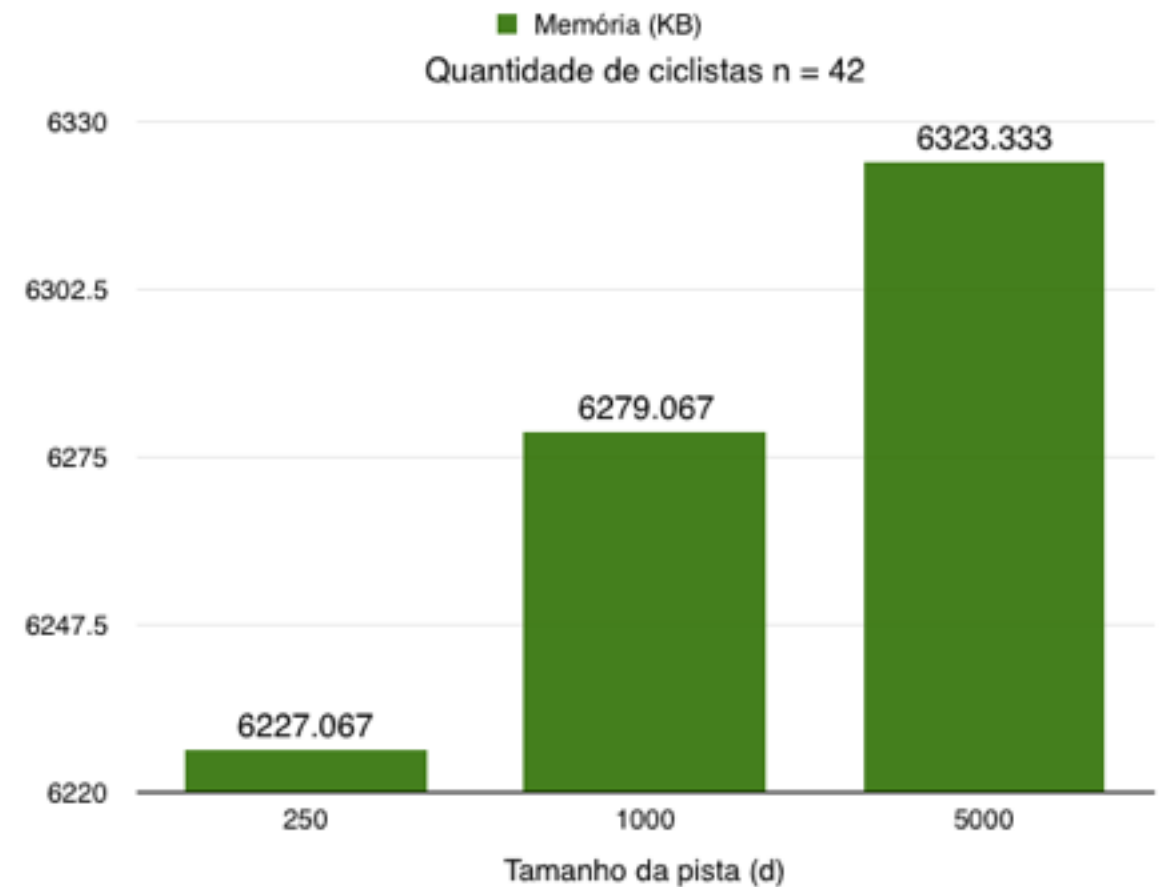
## Tempo médio

IC 0.95

250: 0.003

1000: 0.006

5000: 0.033



## Memória média

IC 0.95

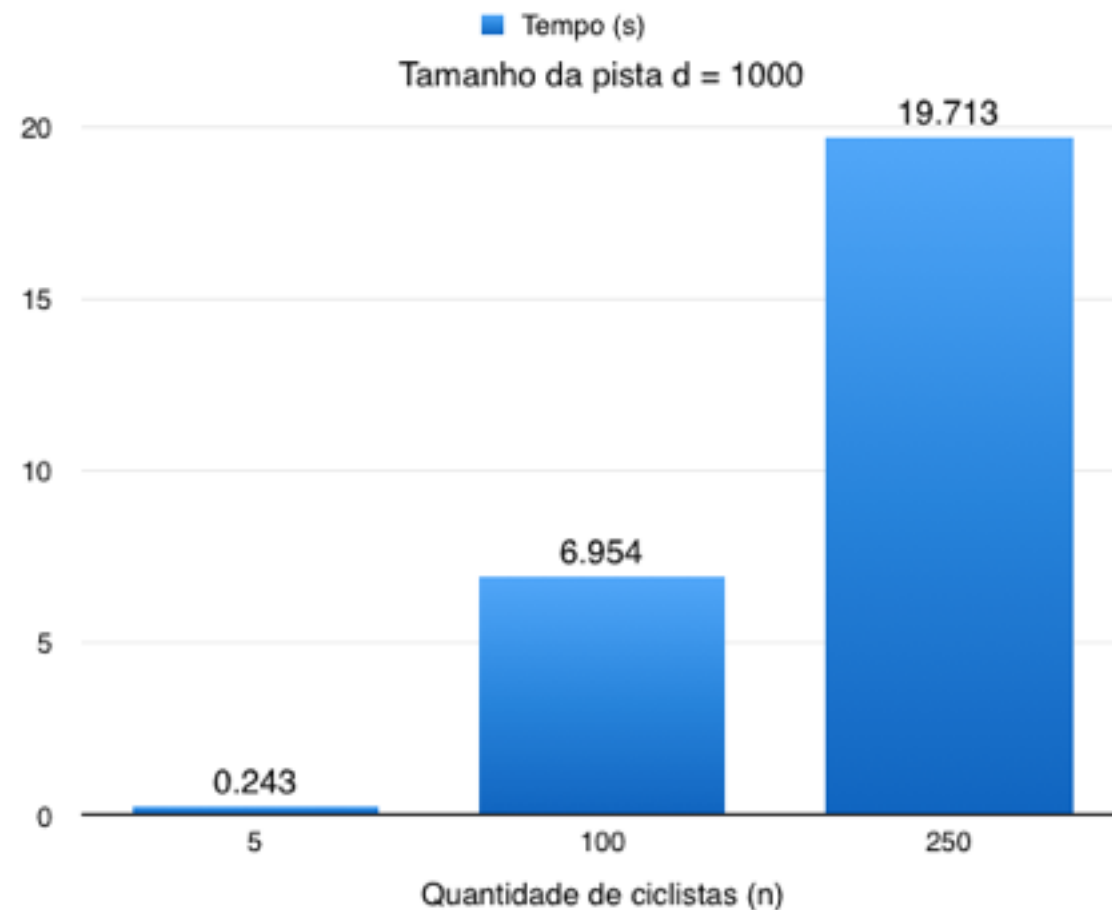
250: 34.41

1000: 33.40

5000: 39.86



# d fixo



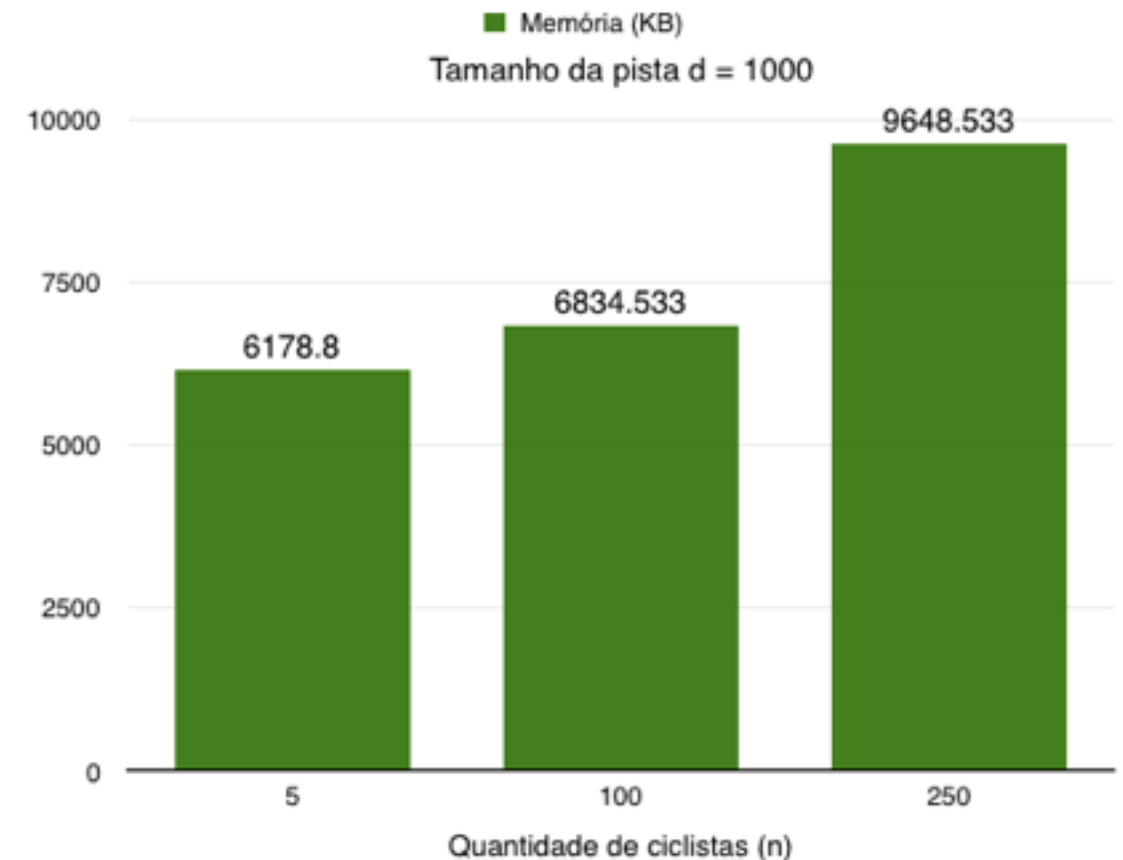
Tempo médio

IC 0.95

5: 0.002

100: 0.021

250: 0.129



Memória média

IC 0.95

5: 7.97

100: 209.88

250: 328.94