



# Shell e Escalonador - Ep1

21/04/25

Ísis Ardisson Logullo - 7577410

# Escalonador com prioridade

- Esse escalonador foi baseado no funcionamento do Round Robin com o acréscimo das prioridades nas interações do quantum. Um vetor de prioridades sempre é alterado e influencia nos acréscimos dos quanta utilizados.
- Utilizado fila com uma lista ligada.
- A detecção de um novo tempo é análogo ao SRTN, mas nesse é detectado a mudança de quantum, acrescentando um quantum no tempo atual.
- Sempre que há um elemento na estrutura de dados e há execução de um processo, há preempção após um quantum passar.
- A cada segundo a thread que executa o processo diminui o valor do dt e para verificar o valor no escalonador, foi utilizado um vetor de semáforos empty e um vetor de semáforo full. Assim, quando a thread está alterando o valor do dt, o escalonador fica esperando essa operação acontecer para assim tomar as decisões necessárias.
- Quantum utilizado de 0.5s.

# Computador A

Nome do modelo: 12th Gen Intel(R) Core(TM) i7-12700H

Trace Esperado:

1)First-Come First-Served

deadlines: media= 0,17 dp= 0,064 ic= 0,151 ate 0,189  
preempcoes: media= 9 dp= 0 ic= 9

2)Shortest Remaining Time Next

deadlines: media= 0,39 dp= 0,151 ic= 0,345 ate 0,435  
preempcoes: media= 12 dp= 0 ic= 12

3)Escalonamento com prioridade

deadlines: media= 0,64 dp= 0,045 ic= 0,627 ate 0,653  
preempcoes: media= 147 dp= 0 ic= 147

Trace Inesperado:

1)First-Come First-Served

deadlines: media= 0,38 dp= 0,06 ic= 0,362 ate 0,398  
preempcoes: media= 9 dp= 0 ic= 9

2)Shortest Remaining Time Next

deadlines: media= 0,36 dp= 0,265 ic= 0,28 ate 0,44  
preempcoes: media= 10 dp= 0 ic= 10

3)Escalonamento com prioridade

deadlines: media= 0,01 dp= 0,03 ic= 0,001 ate 0,019  
preempcoes: media= 39 dp= 0 ic= 39

# Computador B

Processador: Intel64 Family 6 Model 154 Stepping 3 GenuineIntel ~2300 Mhz

Trace Esperado:

1)First-Come First-Served

deadlines: media= 0,15 dp= 0,052 ic= 0,122 ate 0,164

preempcoes: media= 9 dp= 0 ic= 9

2)Shortest Remaining Time Next

deadlines: media= 0,37 dp= 0,143 ic= 0,321 ate 0,428

preempcoes: media= 12 dp= 0 ic= 12

3)Escalonamento com prioridade

deadlines: media= 0,61 dp= 0,041 ic= 0,599 ate 0,631

preempcoes: media= 147 dp= 0 ic= 147

Trace Inesperado:

1)First-Come First-Served

deadlines: media= 0,36 dp= 0,045 ic= 0,334 ate 0,371

preempcoes: media= 9 dp= 0 ic= 9

2)Shortest Remaining Time Next

deadlines: media= 0,32 dp= 0,121 ic= 0,26 ate 0,40

preempcoes: media= 10 dp= 0 ic= 10

3)Escalonamento com prioridade

deadlines: media= 0,007 dp= 0,05 ic= 0,009 ate 0,15

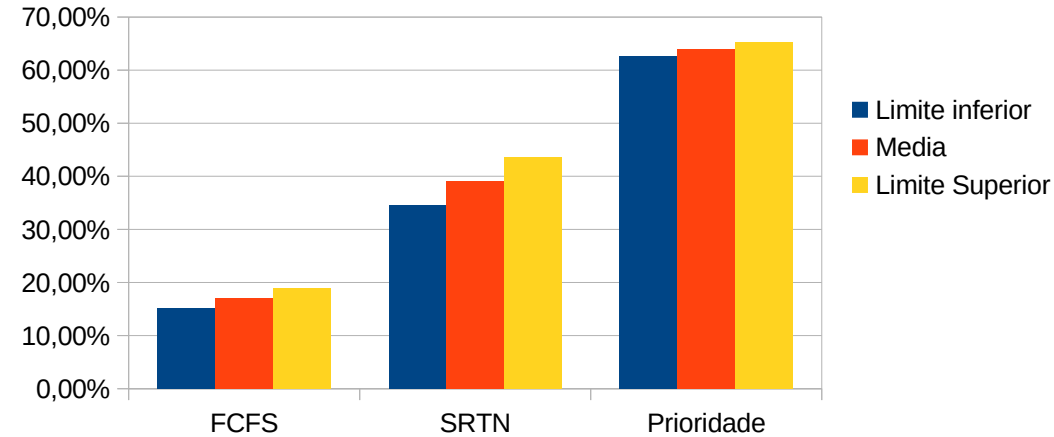
preempcoes: media= 39 dp= 0 ic= 39

# Gráficos Computador A

Trace de comportamento esperado

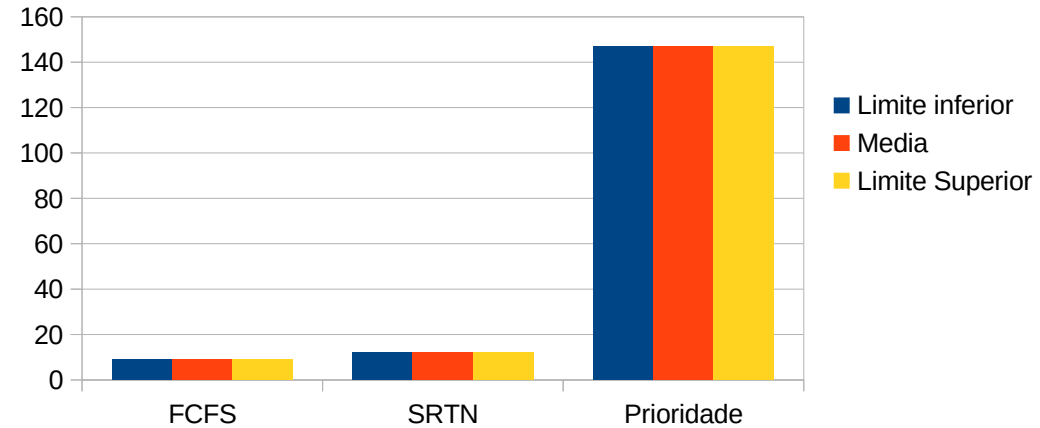
Deadlines - Esperados

Computador A



Preempções - Esperados

Computador A

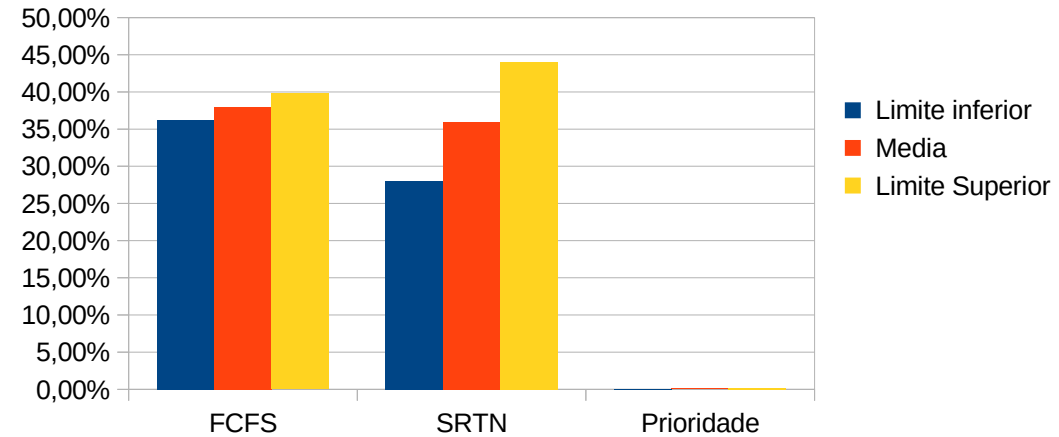


# Gráficos Computador A

Trace de comportamento inesperado

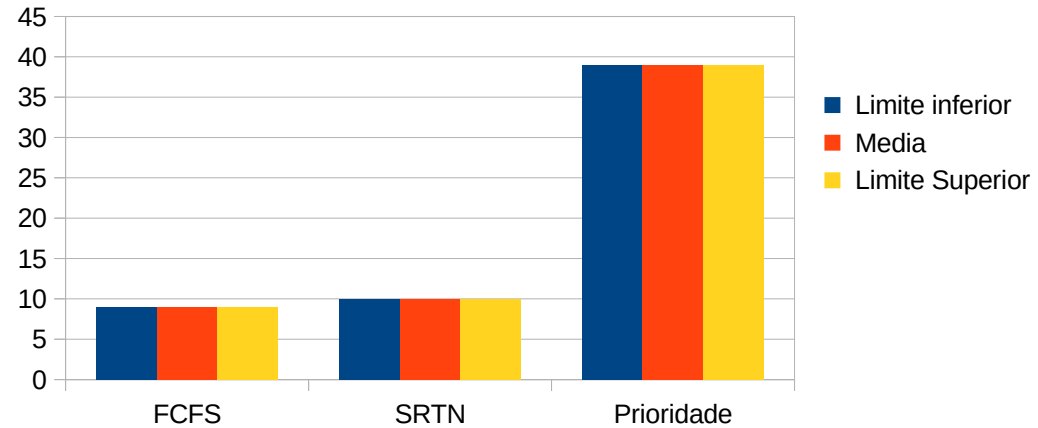
## Deadlines - Inesperados

Computador A



## Preempções - Inesperados

Computador A

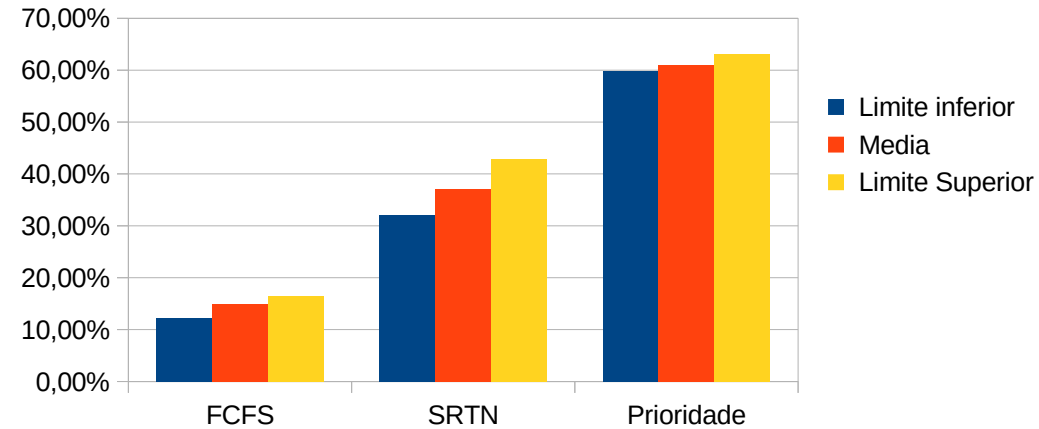


# Gráficos Computador B

Trace de comportamento esperado

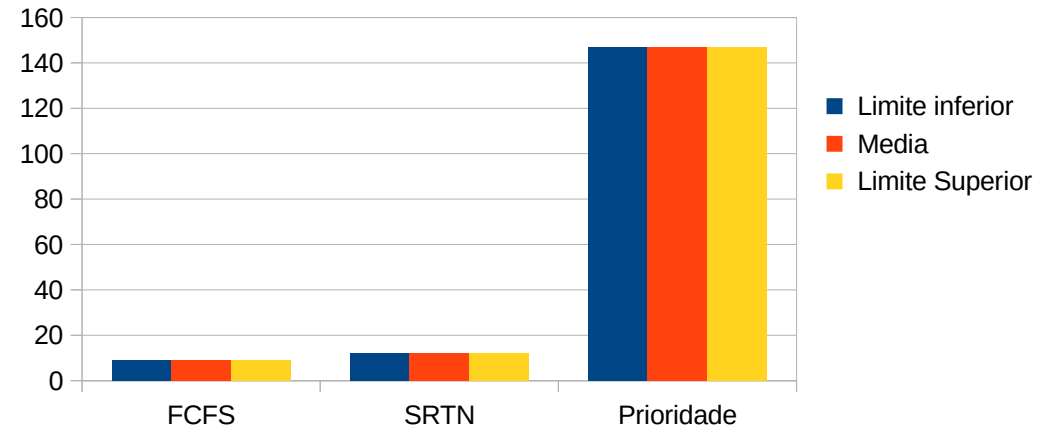
Deadlines - Esperados

Computador B



Preempções - Esperados

Computador B

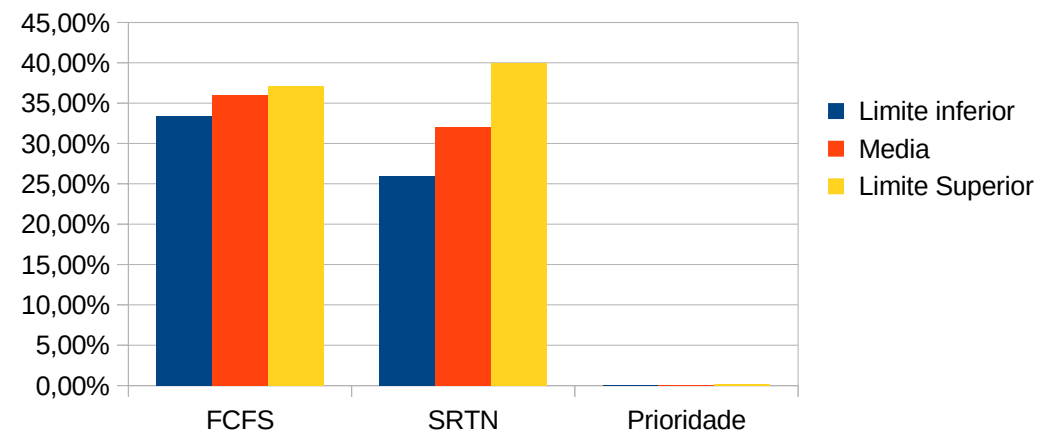


# Gráficos Computador B

Trace de comportamento inesperado

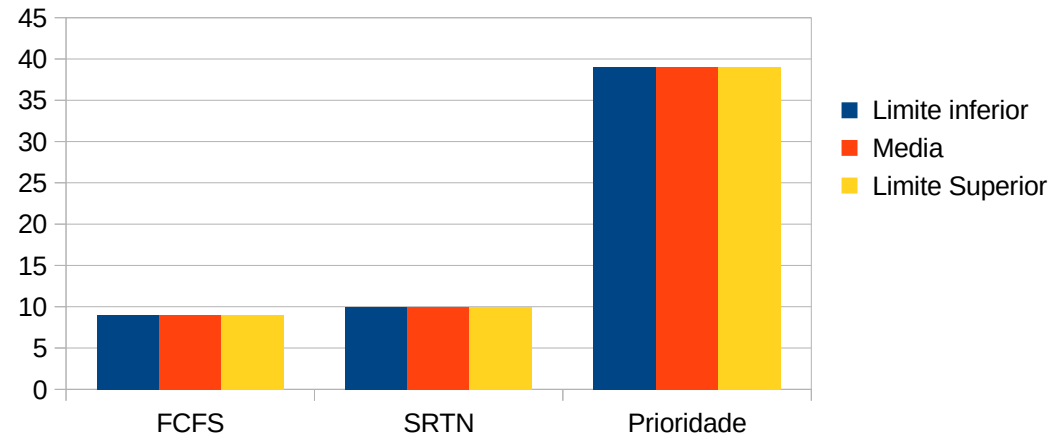
## Deadlines - Inesperados

Computador B



## Preempções - Inesperados

Computador B







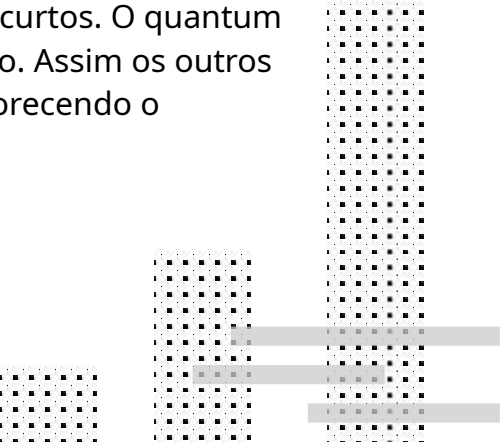
# Conclusões

Os algoritmos de escalonamento se comportaram de maneira quase determinística, com resultados bem semelhantes em todos os 30 testes para cada escalonador.

Quanto às mudanças de contexto, o Escalonador com Prioridade, pela sua característica de preempção a cada quantum, possui maior quantidade de mudanças de contexto que os outros dois para um mesmo input.

Escalonador com Prioridade se diferencia muito dos outros dois (FCFS e SRTN), por conta da preempção por quanta. Os deadlines desses outros 2 são bem semelhantes.

O Escalonador de Prioridade se comporta mal quando existem muitos processos super curtos. O quantum faz com que ele fique postergando por mais tempo algo que poderia acabar de imediato. Assim os outros levam vantagem nesse caso, pois o SRTN busca executar o processo com menor dt, favorecendo o cumprimento da deadline.





# Conclusões

Quando se tem inputs gerados aleatoriamente (mistura de grandes e pequenos), a quantidade média de mudanças de contexto do FCFS e do SRTN ficam bem próximas. Isso acaba afetando um pouco a característica do SRTN de executar o processo com menor dt. Se o tempo de execução de um processo tivesse um intervalo maior de valores possíveis, haveria maior probabilidade de ocorrer preempção pelo SRTN, aumentando a quantidade de mudanças de contexto.

