

MAC0422

Exercício-programa 2

Bárbara de Castro Fernandes - 7577351

Taís Aparecida Pereira Pinheiro - 7580421

Problema

- Corrida de ciclistas por pontos
- Pista com d metros de comprimento
- n ciclistas
- v voltas

Pista

- Matriz de inteiros de tamanho $10 \times d$
- Cada posição representa 1 metro de pista
- Só é possível haver um ciclista por posição
- Cada posição possui o *id* do ciclista que está nela ou *-1*, se estiver vazia

Ciclista

- Uma *struct* com campos:

```
typedef struct ciclista {  
  int id;           ← identificador do ciclista  
  int v;            ← velocidade atual  
  int vAntes;       ← velocidade na iteração anterior  
  int colocacao;    ← posição  
  int volta;        ← volta atual do ciclista  
  int ativo;        ← caso o ciclista tenha saído da corrida ou não  
  int pontuacao;    ← pontuação acumulada  
  int livre;        ← se a velocidade do ciclista foi determinada por outro  
  double tempoVoltaAtual; ← tempo gasto na simulação da última volta  
  double chegada;   ← tempo em todo o percurso  
  pthread_t thread; ← thread representando o ciclista  
} Ciclista;
```

Semáforos e barreiras de sincronização

- Semáforos
 - semVel: controla a velocidade determinada pelos ciclistas
 - semPrint: semáforo para controlar a impressão das informações de saída
 - Sem **local: controla o acesso a cada posição da pista
- Barreiras de sincronização
 - iniVolta: sincroniza o início de uma iteração
 - fimVolta: sincroniza o término de uma iteração

Decisões de projeto - parte 1

- Definição sobre o início de cada volta:

Cada ciclista define a sua velocidade aleatoriamente e se necessário (no caso da velocidade ser 30km/h) define também a velocidade dos ciclistas que estão na mesma linha.

- A volta em si:

Cada ciclista realiza a sua volta respeitando o tempo de entrada na pista e avançando de posição apenas quando possível, mesmo que a sua velocidade possibilite mais rapidez. Ao fim de cada volta é captado o tempo gasto na volta em questão.

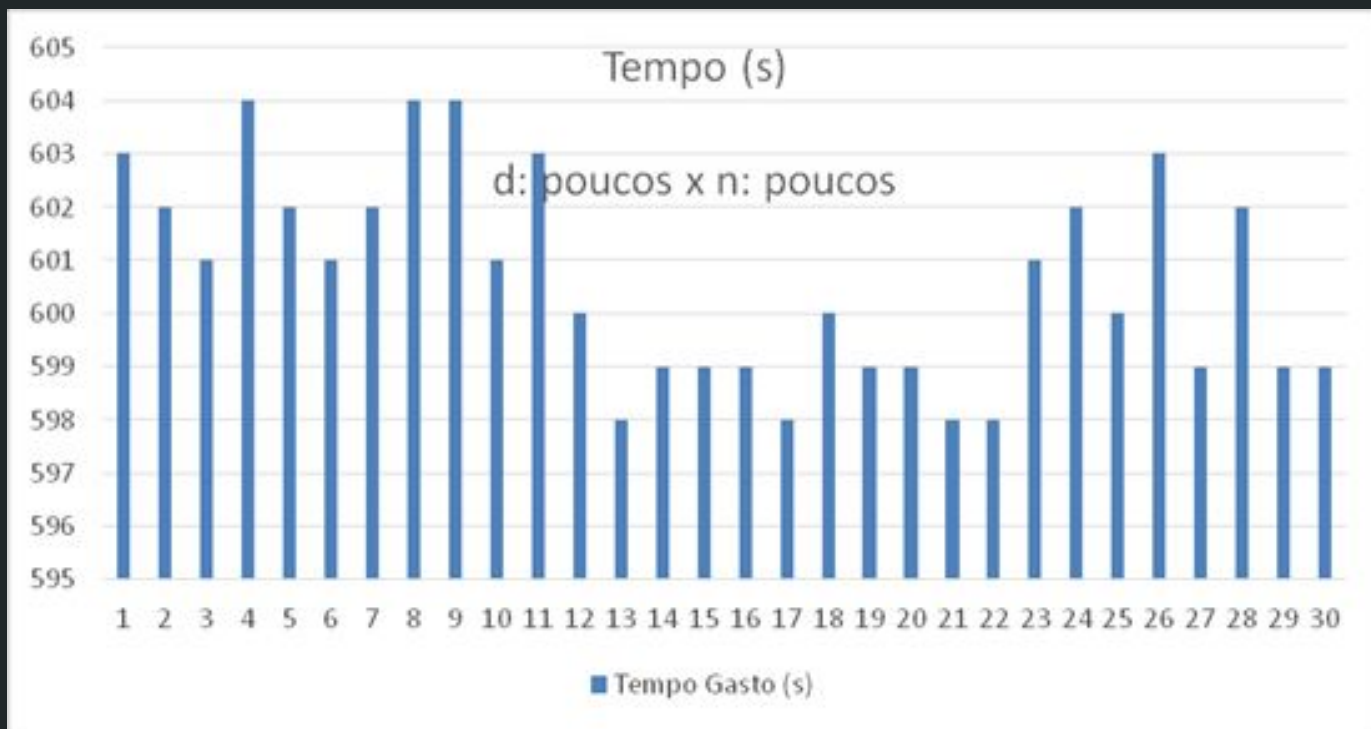
- Análise da volta completada:

Organiza o vetor de acordo com quem gastou menos tempo para completar a volta, jogando os ciclistas quebrados pro fim do vetor. Ao fim de cada volta é captado o tempo gasto na volta em questão.

Saída

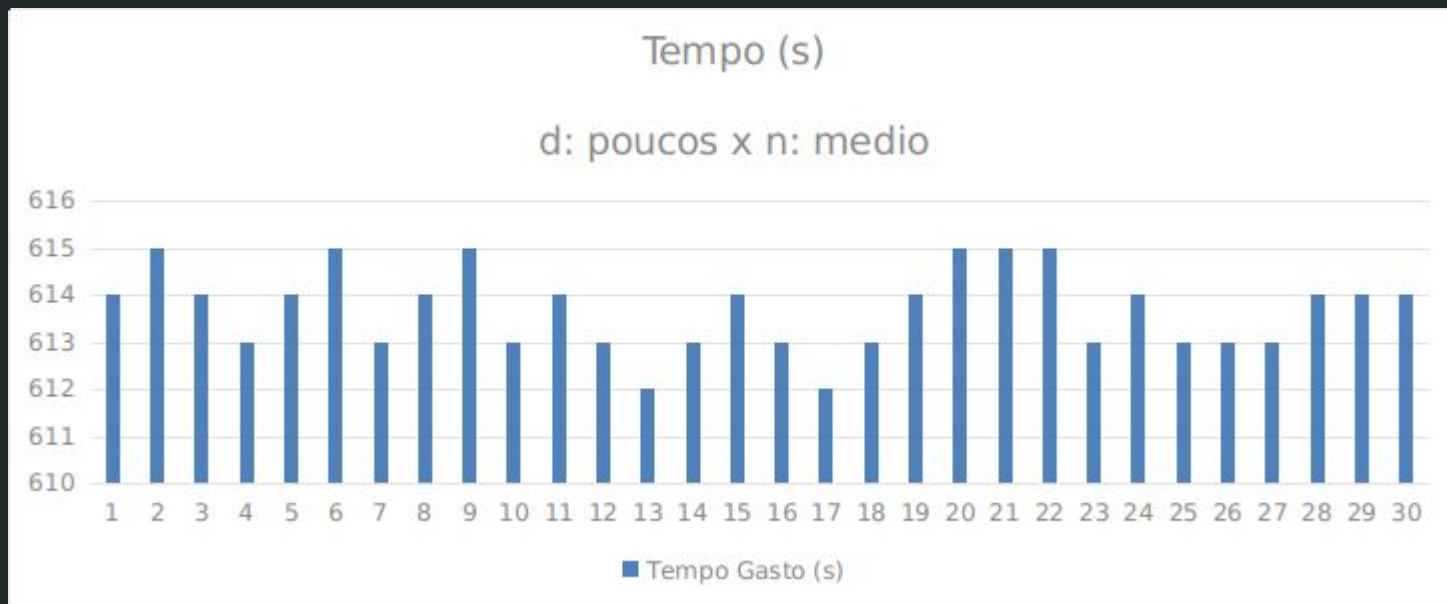
- Informar cada volta completa e as posições dos ciclistas
- Em voltas múltiplas de 10, as pontuações acumuladas em ordem decrescente
- No final da simulação, a pontuação final dos ciclistas e o instante em que cruzaram a linha de chegada, identificando os ciclistas que quebraram e em que volta isso aconteceu
- Opção *debug*
 - Informa a cada 60ms (ou 20ms, nas duas últimas voltas e caso alguém pedale a 90km/h), o *status* de cada posição na pista, incluindo o identificador do ciclista naquela posição ou a informação de que não há nenhum ciclista ali

Resultados d:poucos e n:poucos



Média: 600,6333 e IC(95%) : 0,608326

Resultados d:poucos e n:medio



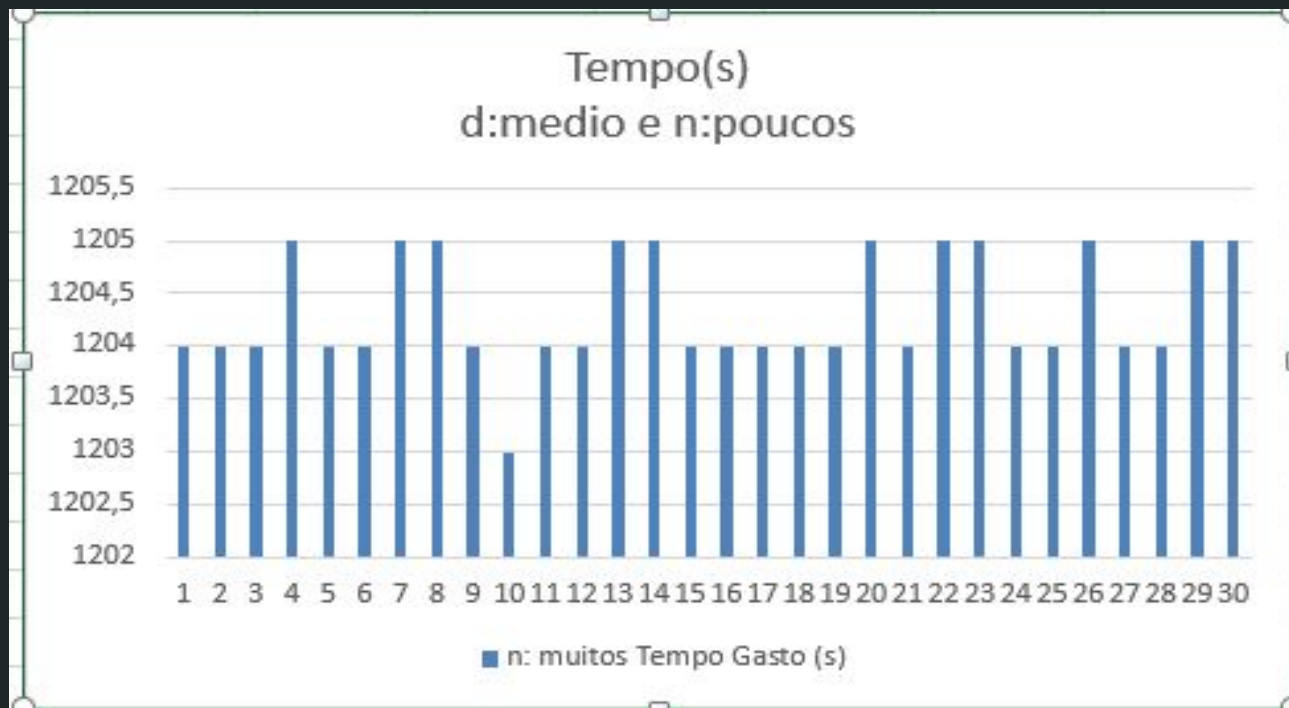
Média: 613,7 e IC(95%) : 0,264801

Resultados d:poucos e n:muitos



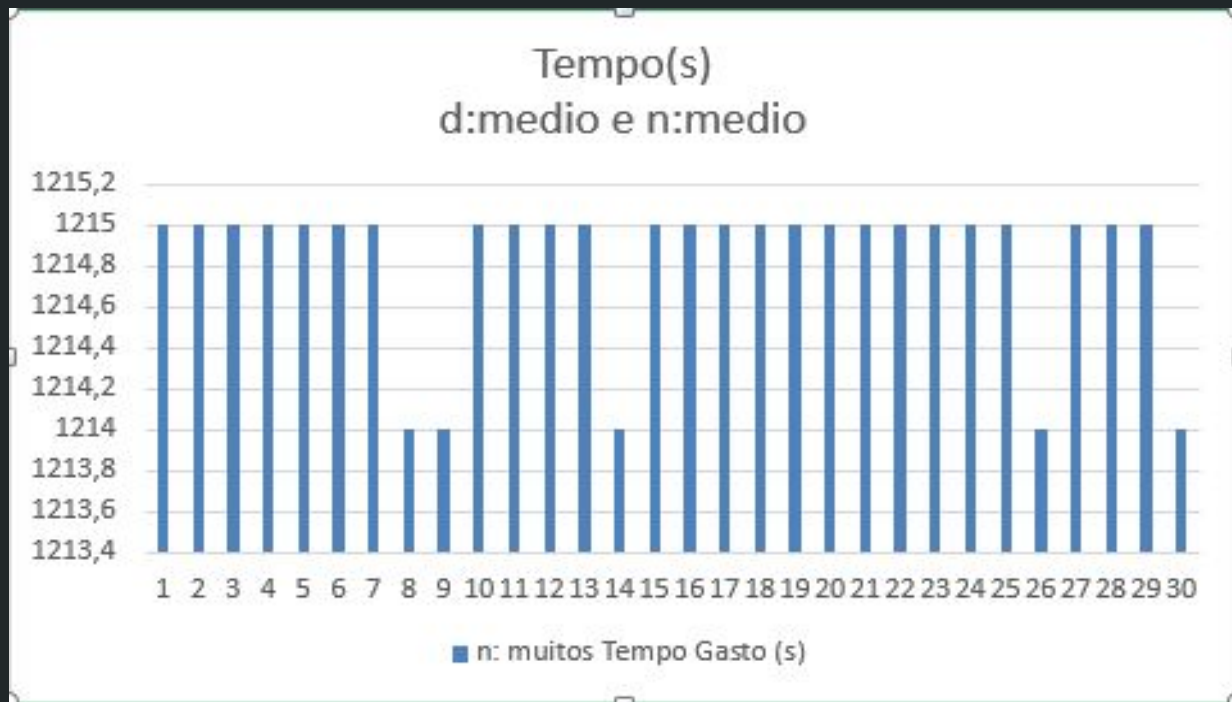
Média: 642,3333 e IC(95%) : 0,182895

Resultados d:medio e n:poucos



Média: 1204,333 e IC(95%) : 0,174943

Resultados d:medio e n:medio



Média: 1214,833 e IC(95%) : 0,0994

Resultados d:medio e n:muitos



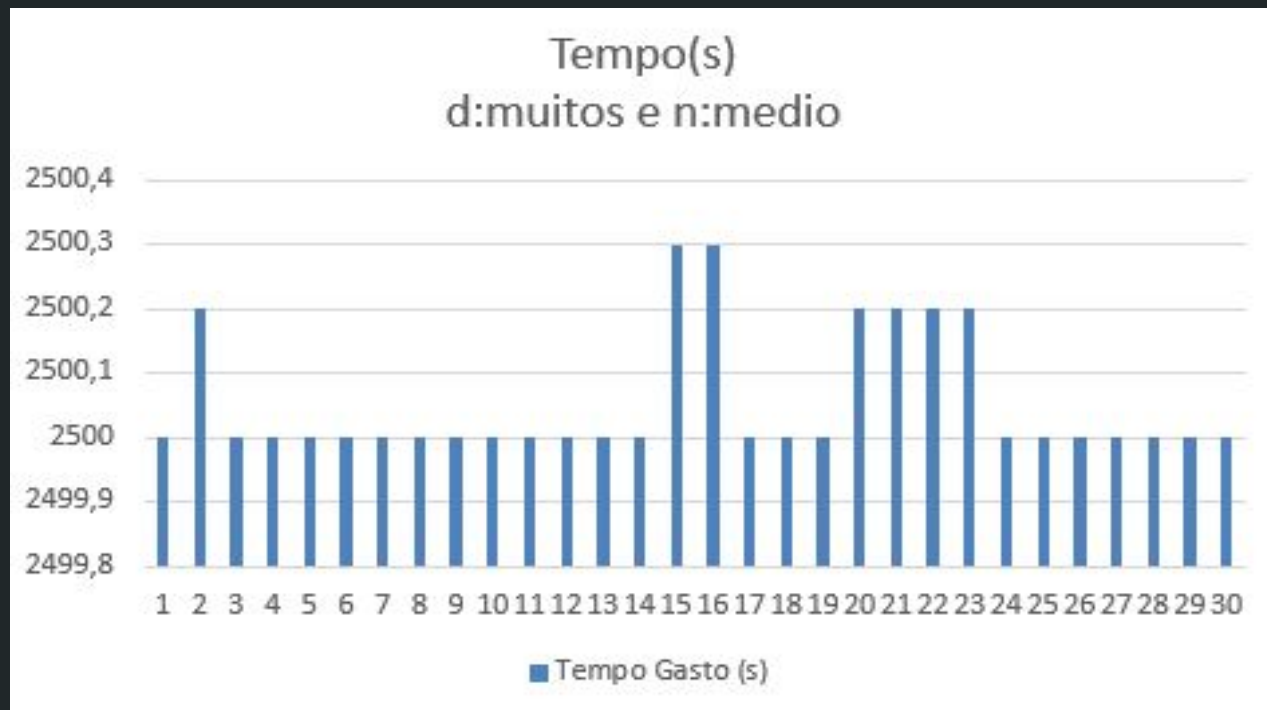
Média: 1242,167 e IC(95%) :0,0994

Resultados d: muitos e n: poucos



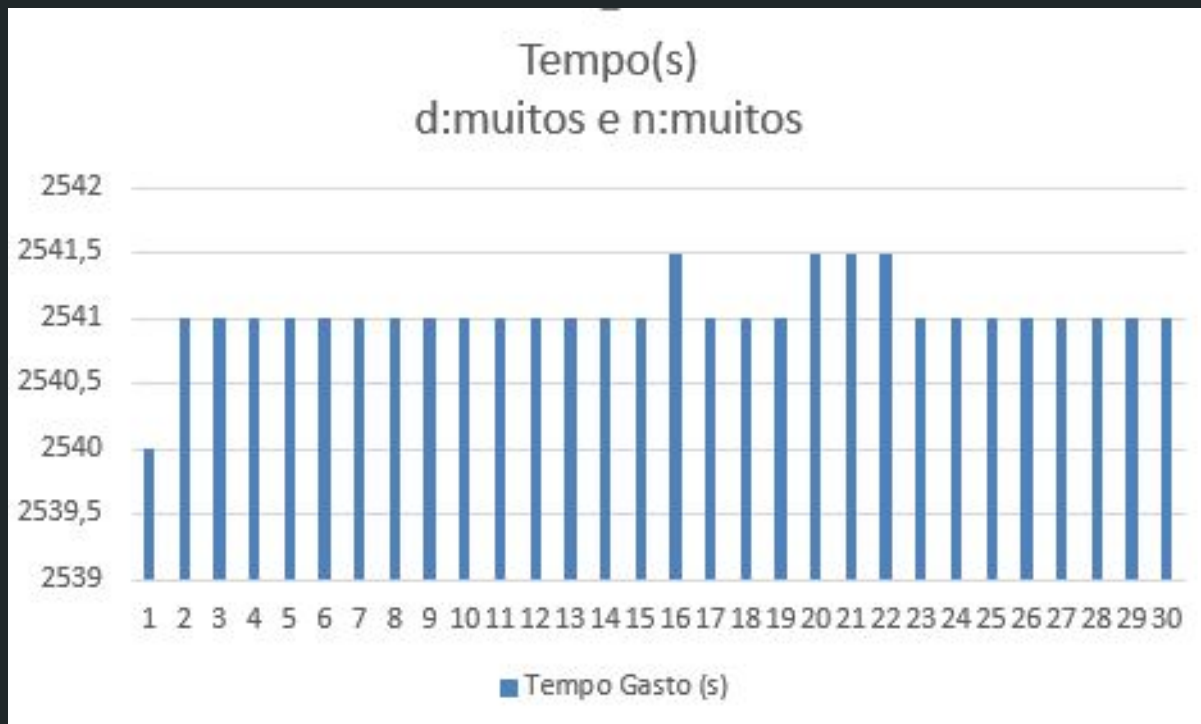
Média: 2406,533 e IC(95%) :0,033398

Resultados d:muitos e n:medio



Média: 2500,053 e IC(95%) :0,029263

Resultados d:muitos e n:muitos



Média: 2541,033 e IC(95%) :0,044531

Ufa!!!

Os testes refletem o esperado, especialmente quando o tamanho da pista aumenta muito. Como os ciclistas respeitam a hora de entrar na pista e percorrem de acordo com a velocidade estabelecida, o tempo deve sim aumentar consideravelmente.