

Aula Prática 8 – Estrutura de Dados

Iago da Silva Rodrigues Alves – 2022035881

Tópico 1 - Selecionar as estruturas de dados a serem monitorizadas

O Vetor é a estrutura de dados principal que armazena os elementos a serem ordenados. Nesse caso os elementos escolhidos foram números inteiros.

Tópico 2 - Selecionar as funções a serem instrumentadas

- **Função insertionSort:** Implementa o algoritmo de ordenação do tipo Insertion Sort. É instrumentada para monitorar o acesso aos elementos do vetor durante a ordenação, registrando os acessos de escrita.
- **Função gerarVetorAleatorio:** Gera um vetor aleatório que será utilizado para a ordenação. É instrumentada para monitorar o acesso aos elementos do vetor antes da ordenação, registrando um acesso de leitura ao vetor.
- **Função preencherZeros:** Inicializa o vetor com elementos nulos.

Tópico 3 - Definir as fases de monitoramento

1. **Fase pré-ordenamento:** Monitorar o acesso aos elementos do vetor antes da chamada da função insertionSort. Isso ajudará a capturar o padrão de acesso à memória antes da ordenação. Nesta fase, é registrado um acesso de leitura ao vetor, registrando o tamanho do elemento e o tamanho total do vetor.
2. **Fase de ordenamento:** Monitorar o acesso aos elementos do vetor durante a execução da função insertionSort. Isso permitirá analisar o padrão de acesso à memória e a localidade de referência durante a ordenação. Nesta fase, são registrados os acessos de escrita aos elementos do vetor, registrando o deslocamento e o tamanho do elemento.
3. **Fase pós-ordenamento:** Monitorar o acesso aos elementos do vetor após a ordenação, na função main. Isso ajudará a identificar o padrão de acesso à memória após a ordenação. Nesta fase, é registrado um acesso de escrita ao vetor completo, registrando o deslocamento inicial e o tamanho total do vetor.

Tópico 4 – Instrumentar o código

No exemplo escolhido abaixo a instrumentação permite registrar os acessos de escrita realizados durante o processo de ordenação, fornecendo informações sobre os padrões de acesso à memória e a localidade de referência durante a ordenação

```

void insertionSort(int arr[], int n) {
    for (int i = 1; i < n; i++) {
        int key = arr[i];
        int j = i - 1;

        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            ESCRIVEMEMLOG((j + 1) * sizeof(int), sizeof(int), 3);
            j = j - 1;
        }
        arr[j + 1] = key;
        ESCRIVEMEMLOG((j + 1) * sizeof(int), sizeof(int), 3); /
    }
}

```

Tópico 5 - Definir o plano de experimentos

A ideia inicialmente foi testar o programa instrumentado para um vetor aleatório de 100 elementos. Após isso eu tentaria identificar quais elementos do vetor foram mais acessados, identificar padrões de escrita e leitura, avaliar a eficiência do algoritmo de ordenação em relação à localidade de referência.

Foram definidos 3 identificadores fictícios:

- 0 - Para indicar a operação de escrita quando geramos o vetor, nulo
- 1 - Para indicar a operação de escrita dos elementos do vetor antes da ordenação
- 2 - Para indicar a operação de escrita no vetor durante a ordenação

Tópico 6 - Executar os experimentos

Exemplo de saída pós-instrumentação

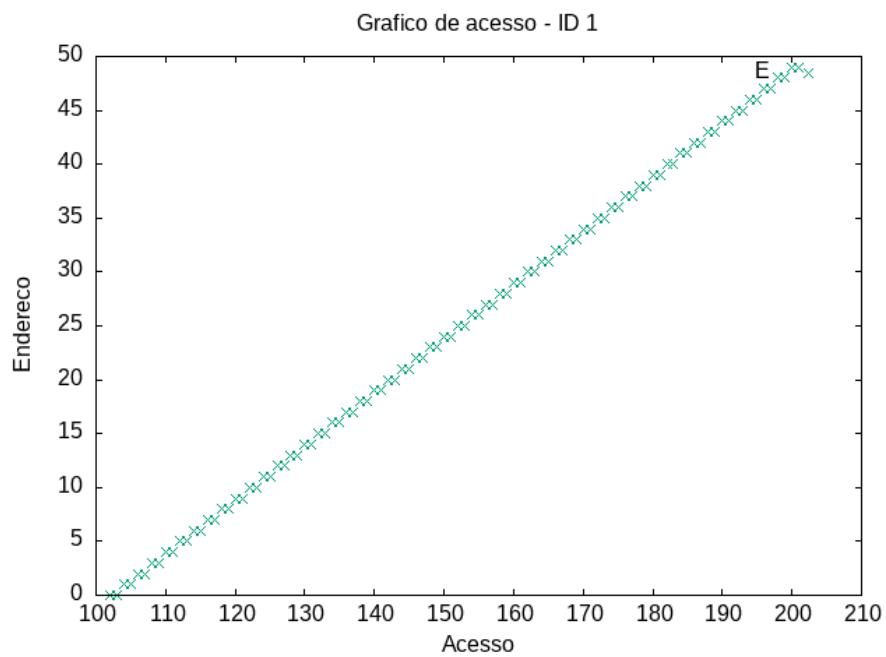
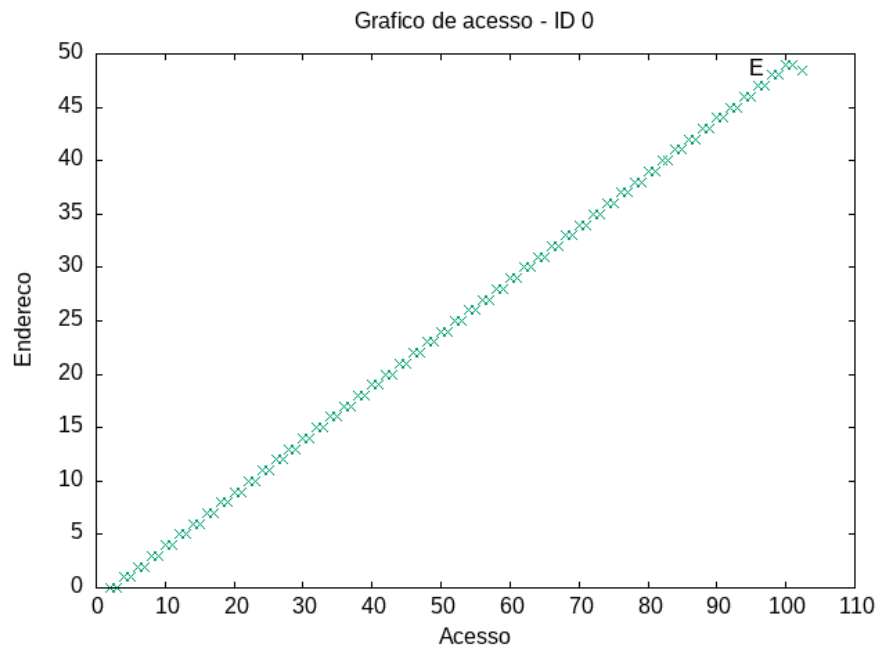
```

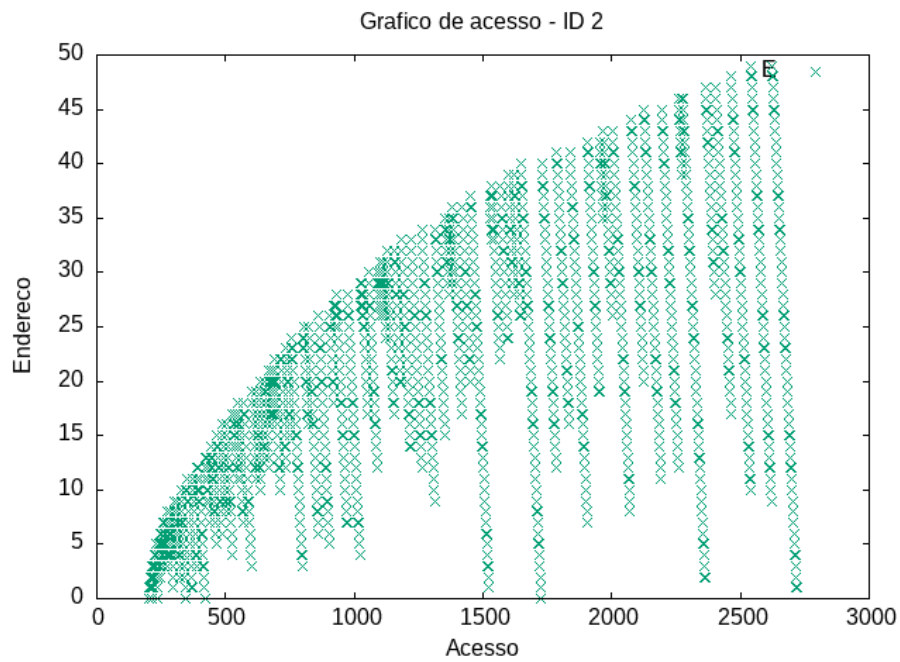
I 1 5810.420891869
L 0 2 1 0.000020870 4 400
E 0 3 2 0.000025280 0 400
E 0 4 3 0.000025950 4 4
E 0 5 3 0.000026560 0 4
E 0 6 3 0.000027090 8 4
...
E 0 2588 3 0.001407530 64 4
E 0 2589 3 0.001407770 60 4
E 0 2590 3 0.001408010 56 4
E 0 2591 3 0.001408250 52 4
E 0 2592 3 0.001408490 48 4
E 0 2593 3 0.001408730 44 4
F 2594 5810.422301009 0.001409140

```

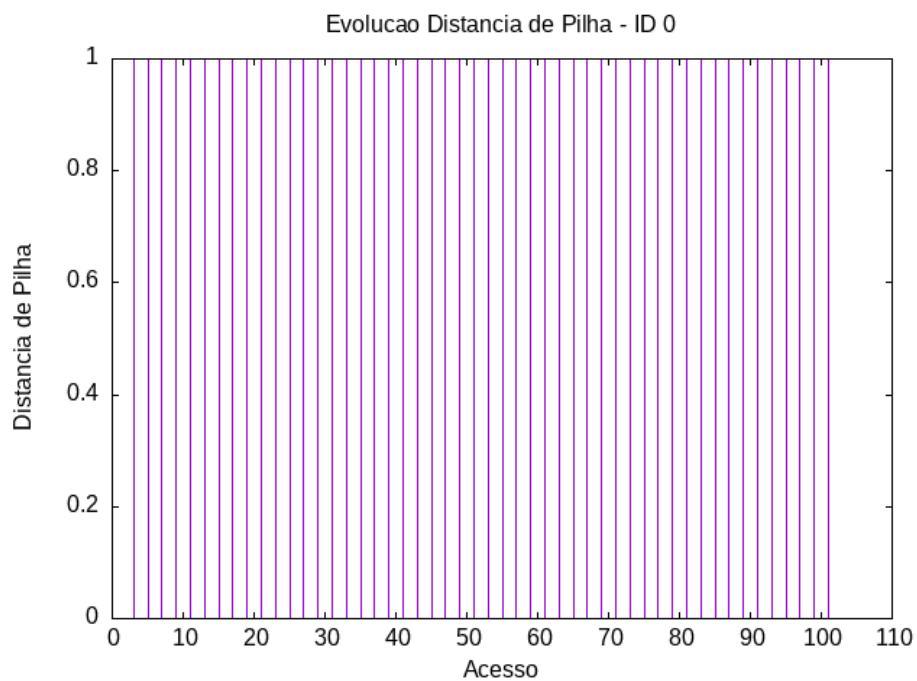
Tópico 7 – Gerar as visualizações

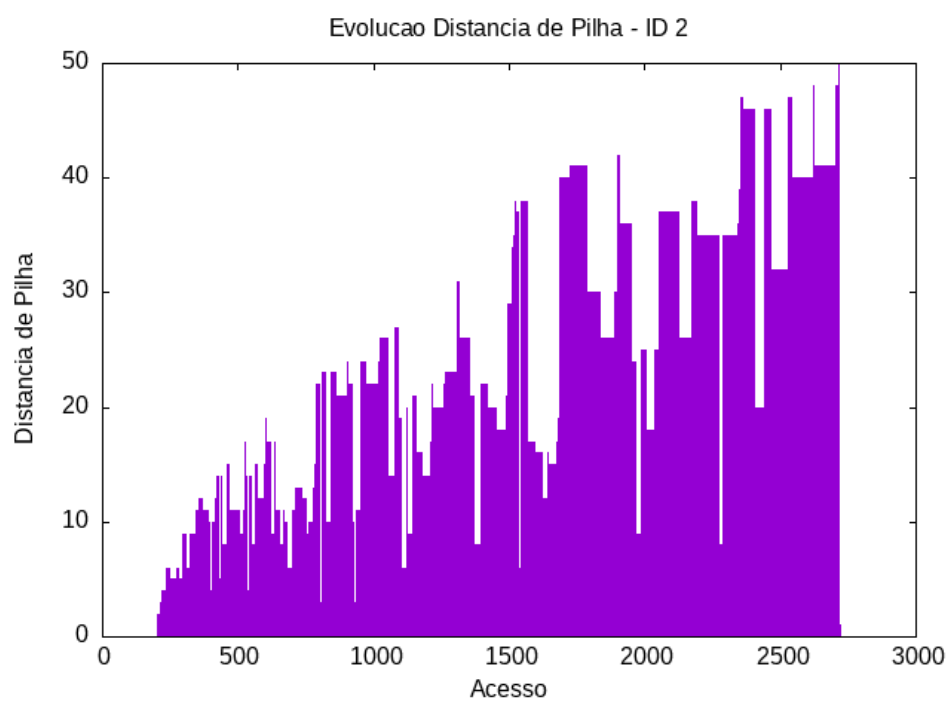
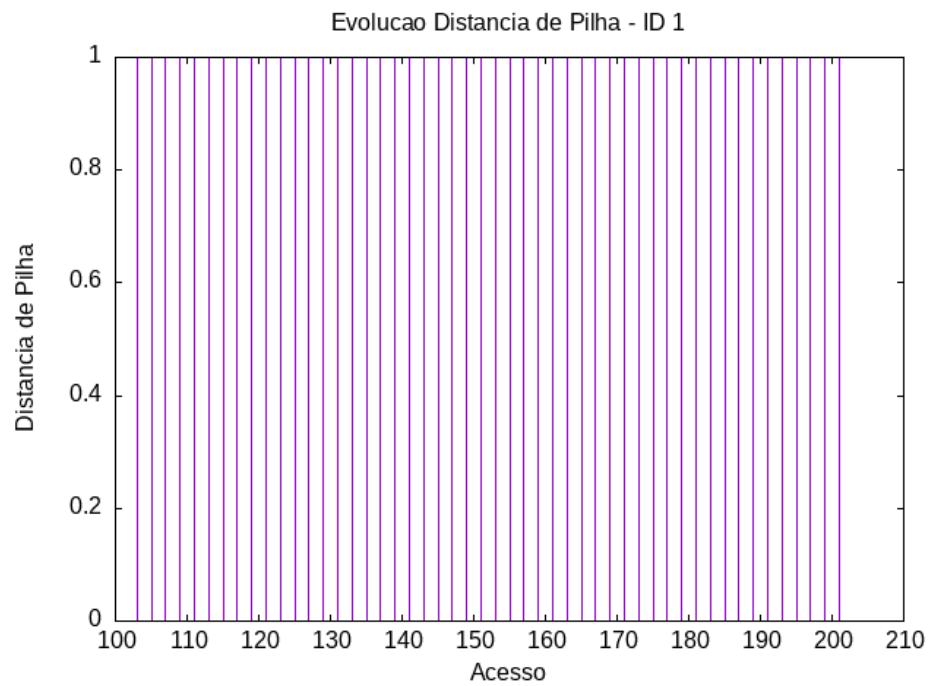
Gráficos de Acesso



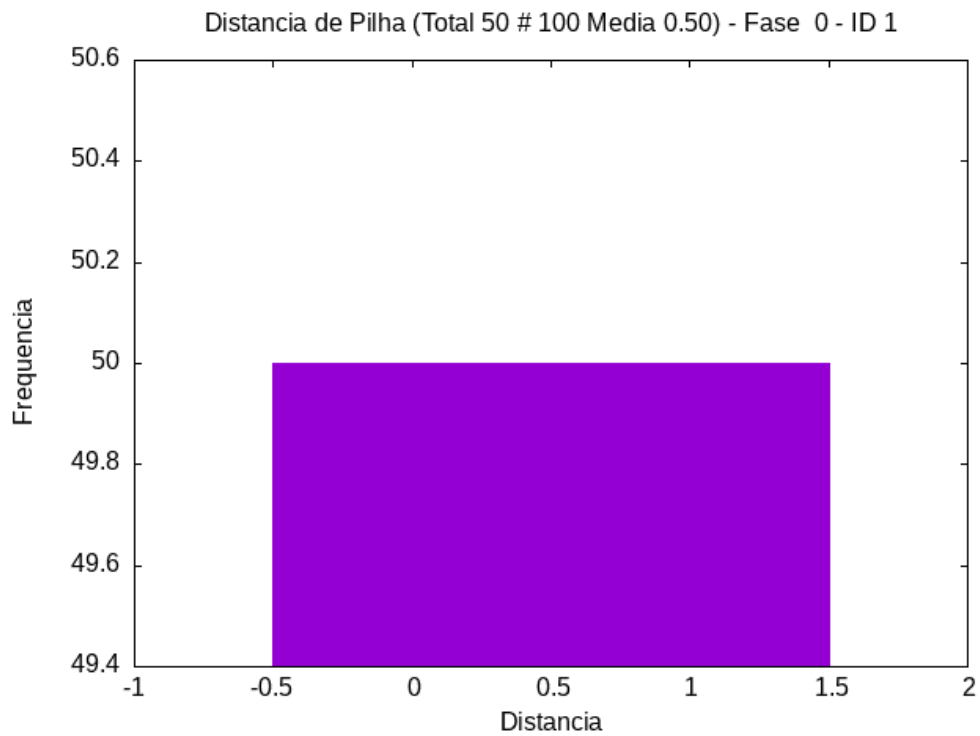
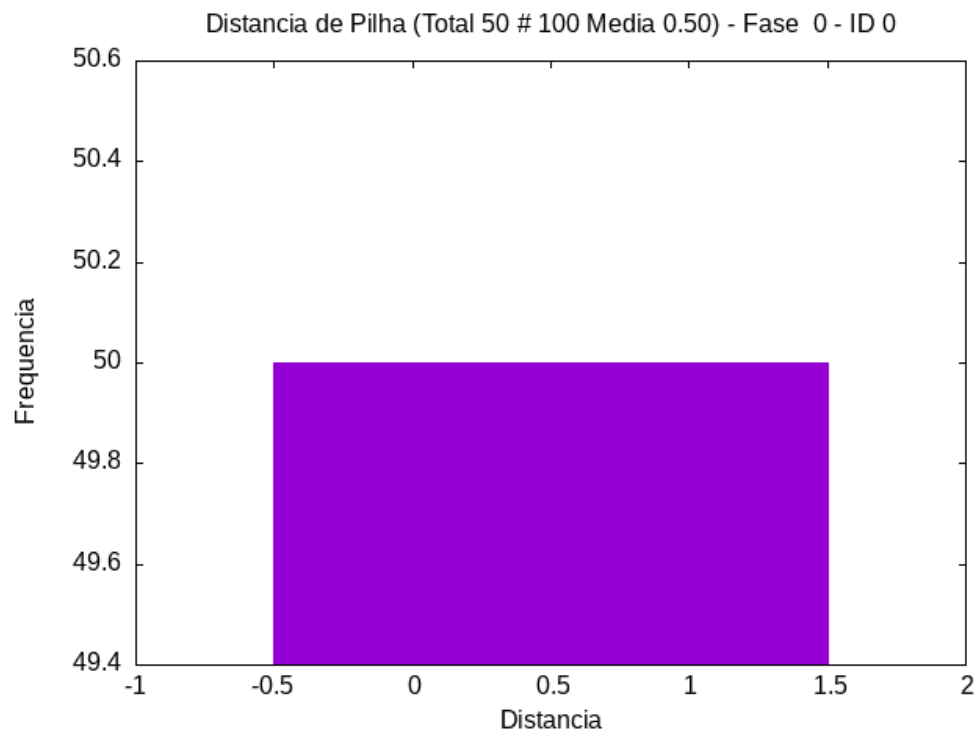


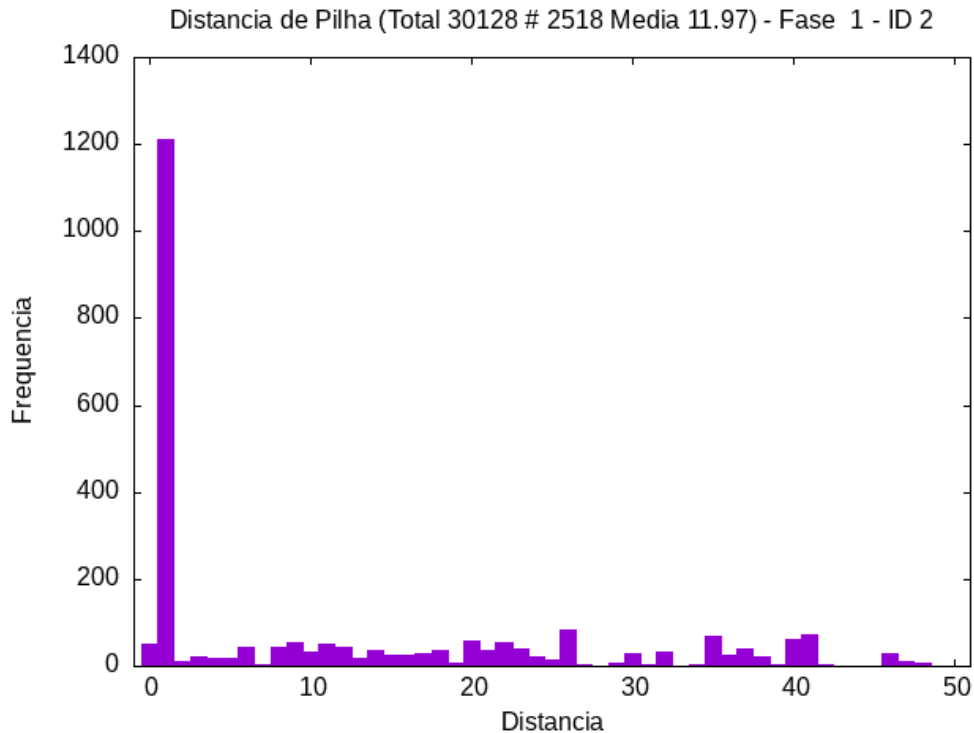
Gráficos Evolução Distância de Pilha





Gráficos Distância de Pilha





Tópico 8 - Analisar os resultados e visualizações

Acessos de memória: Para os identificadores 0 e 1 percebemos a presença de uma reta diagonal, que indica que a memória é acessada sequencialmente. Dessa forma, o valor do identificador aumenta linearmente à medida que os elementos do vetor são preenchidos. Lembrando que esses identificadores dizem respeito ao momento que o vetor é preenchido com zeros e ao momento que ele é preenchido com valores aleatórios. É válido destacar também que esses dois gráficos se completam uma vez que o acesso do ID 0 termina no 100 e o acesso do ID 1 começa no 100. Agora para o identificador 2 temos um comportamento completamente diferente. Esse ID, que diz respeito ao momento de ordenação do vetor, não possui um acesso sequencial e também não é nada uniforme. Desse modo, nota-se que certas posições de memória são acessadas com mais frequência do que outras e a amplitude do eixo x é bem superior ao ID's anteriores.

Evolução Distância de Pilha: Novamente temos um comportamento semelhante entre os dois primeiros identificadores. Nesse caso percebe-se a presença de retas verticais paralelas e em alguns espaços pequenos traços entre essas retas. Creio que isso possa indicar um comportamento de alocação e desalocação de memória consistente e controlado onde não há mudanças significativas na alocação ou desalocação de memória na pilha relacionadas a esses identificadores. Os pequenos traços entre as retas verticais podem representar pequenas variações na distância de pilha em pontos específicos. Agora para o ID 2 temos um comportamento diferente, em que a distância de pilha está variando de maneira mais significativa ao longo do tempo.

Distância de Pilha: Ao analisar a distância de pilha para os identificadores 0, 1 e 2, podemos observar diferentes comportamentos. Para os identificadores 0 e 1, a distância de pilha é mantida em intervalos específicos, indicando estabilidade nessas operações. Além disso, a frequência dessas operações permanece constante ao longo do tempo. No entanto, para o identificador 2, notamos uma variação significativa na frequência de alocação e desalocação de memória em relação à distância de pilha. Observa-se que a frequência é mais alta em distâncias de pilha próximas a 0, enquanto ocorre em menor frequência para distâncias maiores que 1 e menores que 50.