

**ALGORITMOS E ESTRUTURAS DE DADOS 2**

**TRABALHO PRÁTICO 1**

**Listas**

Autor(es): Gabriel dos Santos Silva

Prof. Jefferson de Oliveira Balduino

Serra-ES

Abril/2021

Sumário

[1 oBJETIVO 2](#_Toc66311967)

[2 FUNCIONAMENTO GERAL 2](#_Toc66311968)

[3 IMPLEMENTAÇÃO 3](#_Toc66311969)

[4 CÓDIGO-FONTE 3](#_Toc66311970)

# oBJETIVO

O Presente projeto tem por objetivo a implementação de um “Mundo de Blocos” desenvolvido na linguagem Java. Este Mundo de Blocos possuirá uma série de comandos que serão detalhados abaixo, além dos comandos seu funcionamento se dará com um arquivo de entrada em formato específico e será gerado um arquivo também de saída com o resultado das operações realizadas dentro do código.

**Comandos:**

* ***move a onto b*:** move o bloco a para cima do bloco b retornando eventuais blocos que já estiverem sobre a ou b para as suas posições originais.
* ***move a over b:*** Coloca o bloco a no topo do monte onde está o bloco b retornando eventuais blocos que já estiverem sobre a às suas posições originais.
* ***pile a onto b:*** coloca o bloco a juntamente com todos os blocos que estiverem sobre ele

em cima do bloco b, retornando eventuais blocos que já estiverem sobre b as suas posições originais.

* ***pile a over b:*** coloca o bloco a juntamente com todos os blocos que estiverem sobre ele

sobre o monte que contém o bloco b.

**Modelo dos arquivos:**

****

# FUNCIONAMENTO GERAL

O Mundo de Blocos deverá ser manipulado de acordo com os comandos explicitados no “Objetivo”, os comandos serão lidos do arquivo de entrada e interpretados pelo código. Inicialmente os blocos encontram-se em suas posições de origem. Depois de ler o arquivo e fazer o parse dos dados contidos, será feito uma busca por palavras chaves na linha que está sendo lida, encontrado a palavra, será feito a chamada para a função responsável por realizar aquela ação, por ex.: Encontrou-se o conjunto “move” e “onto” na linha, então identificamos que a função a ser chamada é a “move onto”, esta função receberá como parâmetro o bloco e a sua nova posição e realizará a ação. O código fará isso até chegar na linha contendo “quit” onde encerrará.

# IMPLEMENTAÇÃO

**Main**:

No arquivo principal “TrabalhosBlocos.java” temos a parte inicial do programa onde é feita a leitura do arquivo (Figura 1), além de conter também a primeira parte da lógica, uma cadeia de if’s para identificação de qual método devo chamar (Figura 2).

**Nós:**

No arquivo “Nodes.java” temos a implementação dos ponteiros identificando o bloco e apontando para o próximo bloco (célula ou nó) (Figura 3), utilizasse também de getters e setters.

**Estrutura de Dados:**

No arquivo “EstruturaDados.java” criasse a Lista Simplesmente Encadeada (LSE) com o início, fim e seus elementos. Temos também o método “pegaNodes” que retorna a célula solicitada (Figura 4). A criação do primeiro elemento da LSE e a adição de novos (Figura 5). A remoção de elementos da LSE (Figura 6).

**Bloco:**

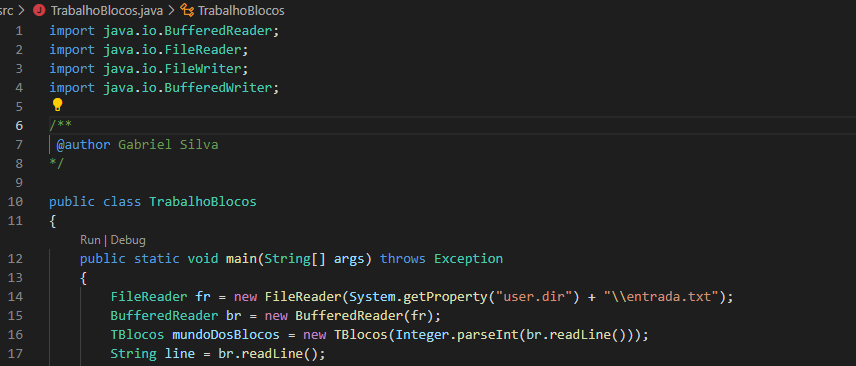
No arquivo “Bloco.java” temos o construtor do bloco e os getters e setters para ‘identificar’ o bloco (Figura 7).

**Manipulação do Mundo de Blocos:**

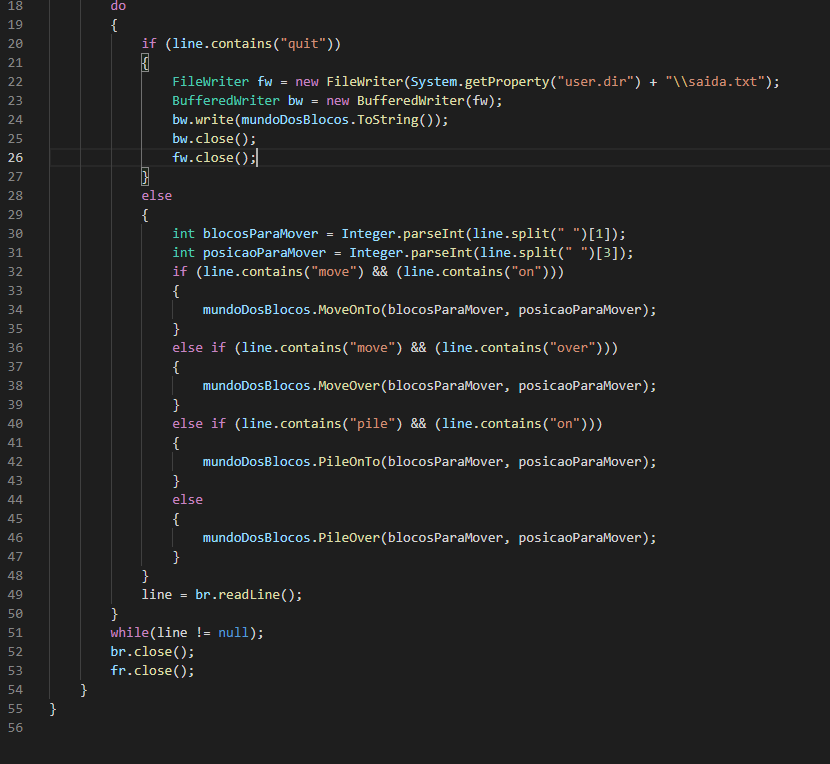
“TBlocos.java” neste arquivo que contém os métodos (citados no objetivo) para manipulação do Mundo de Blocos. O construtor cria o vetor de LSEs com base no total de blocos informados no arquivo de texto de entrada e o FOR irá adicionar em cada posição do vetor o bloco na LSE (Figura 8). Faz-se então a realocação dos blocos de acordo, após isso temos a lógica por trás dos movimentos realizados pelos blocos (Figura 9)(Figura 10) e também a criação do arquivo de saída (Figura 11).

# CÓDIGO-FONTE

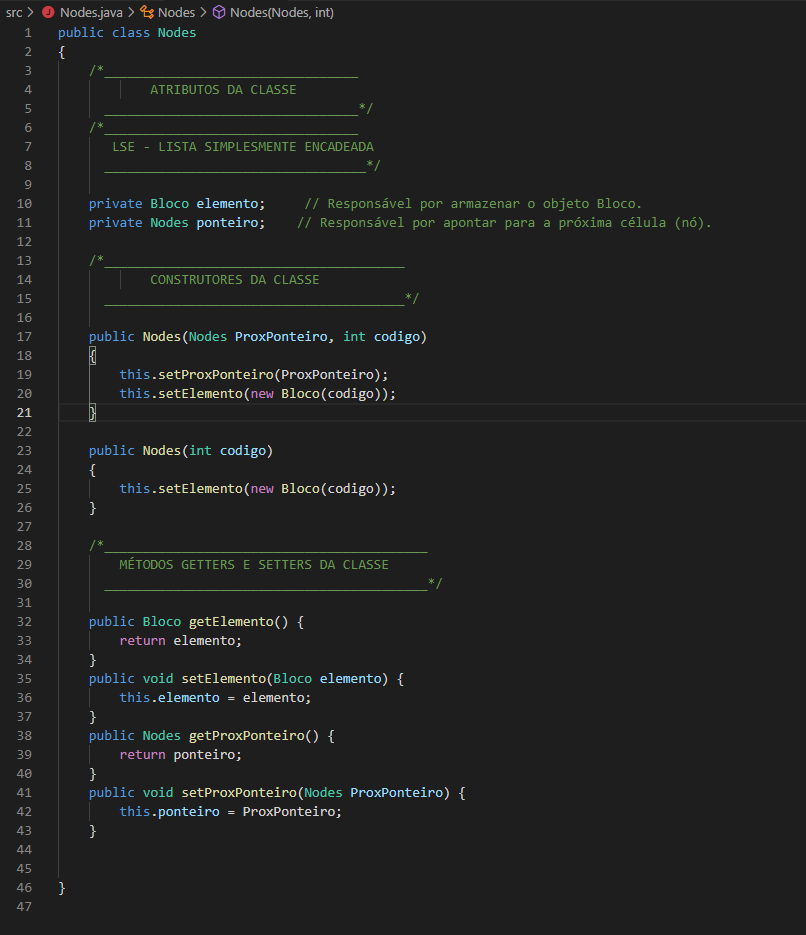
<https://github.com/bielusantos/AED2>



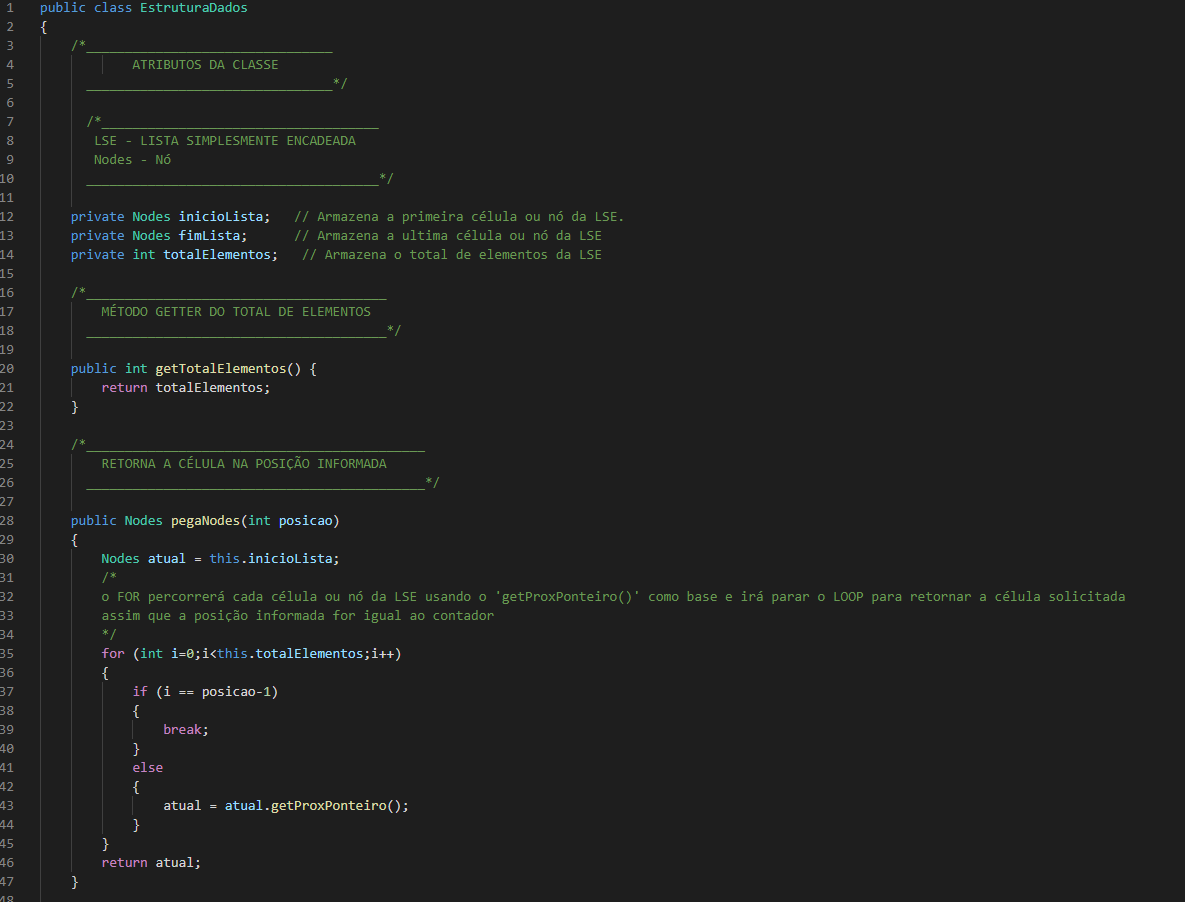
Figura



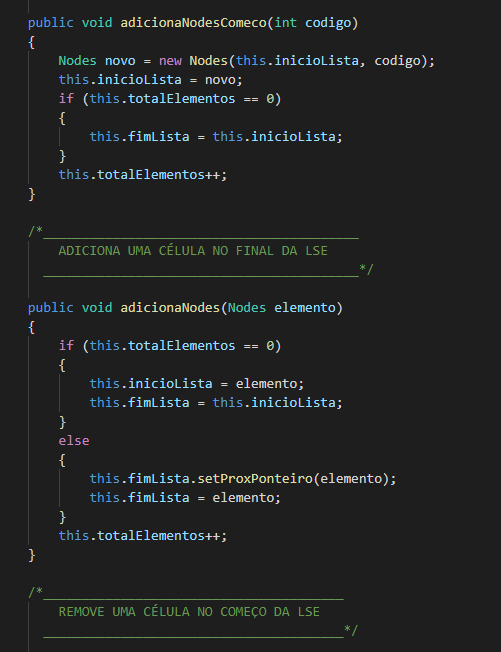
Figura



Figura



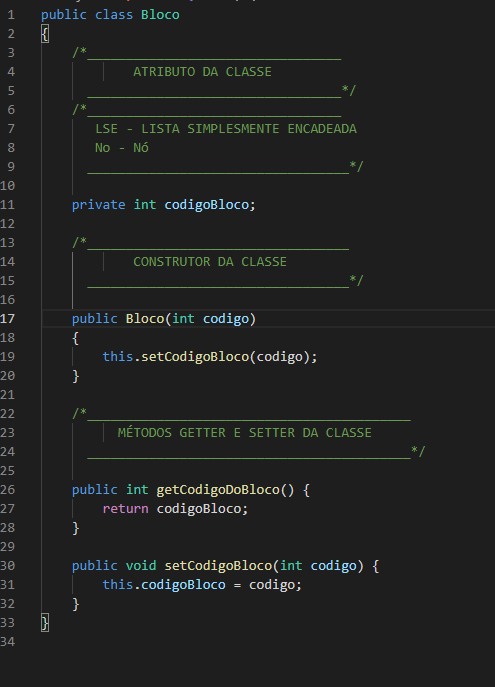
Figura



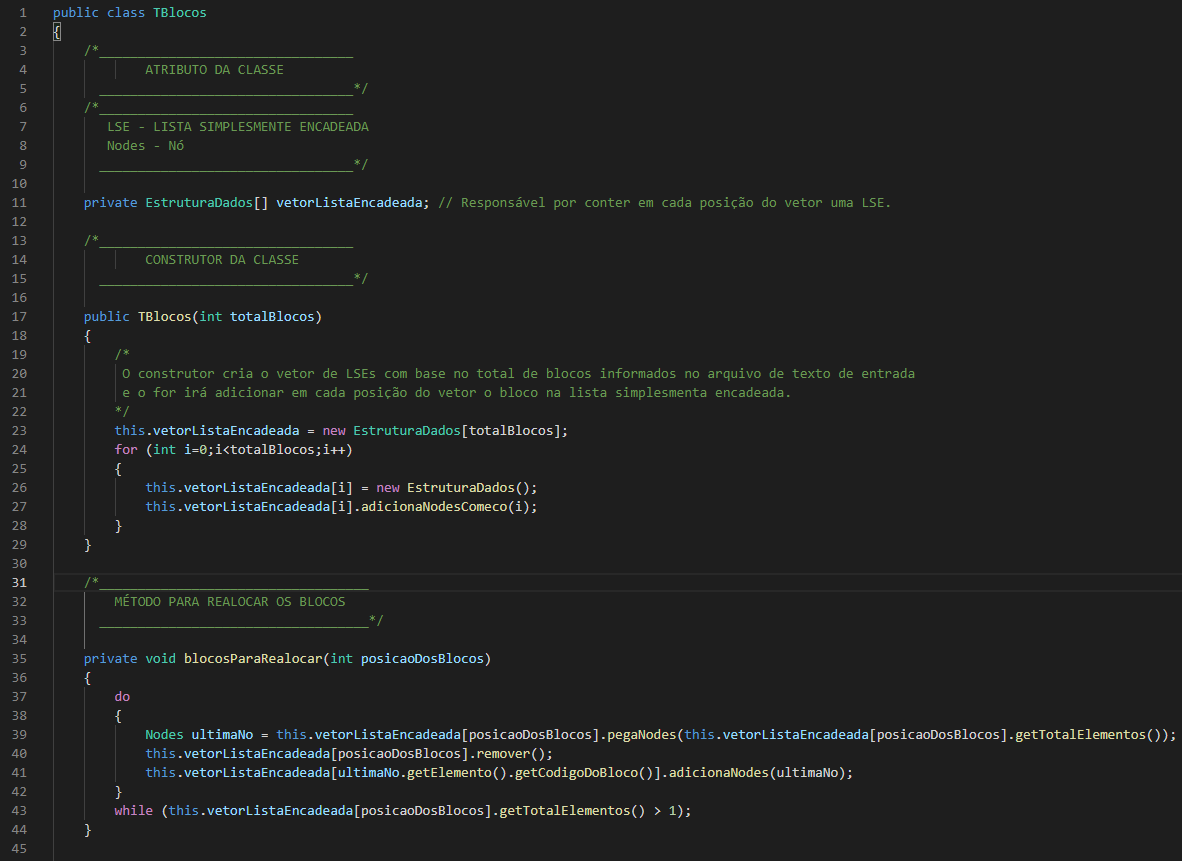
Figura



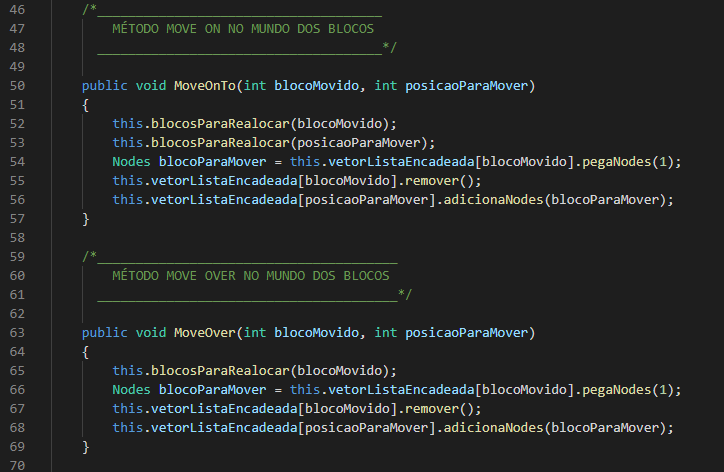
Figura



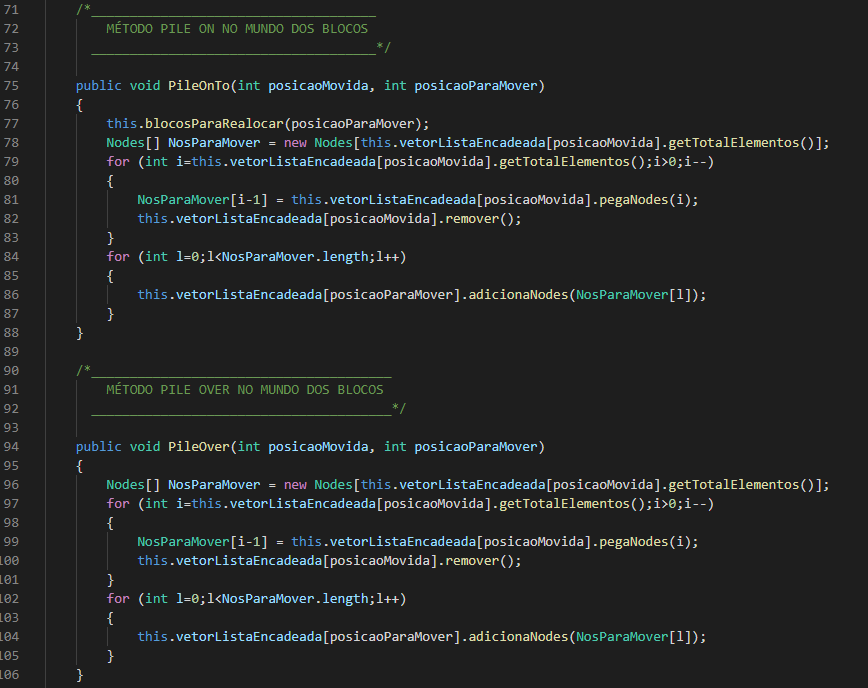
Figura



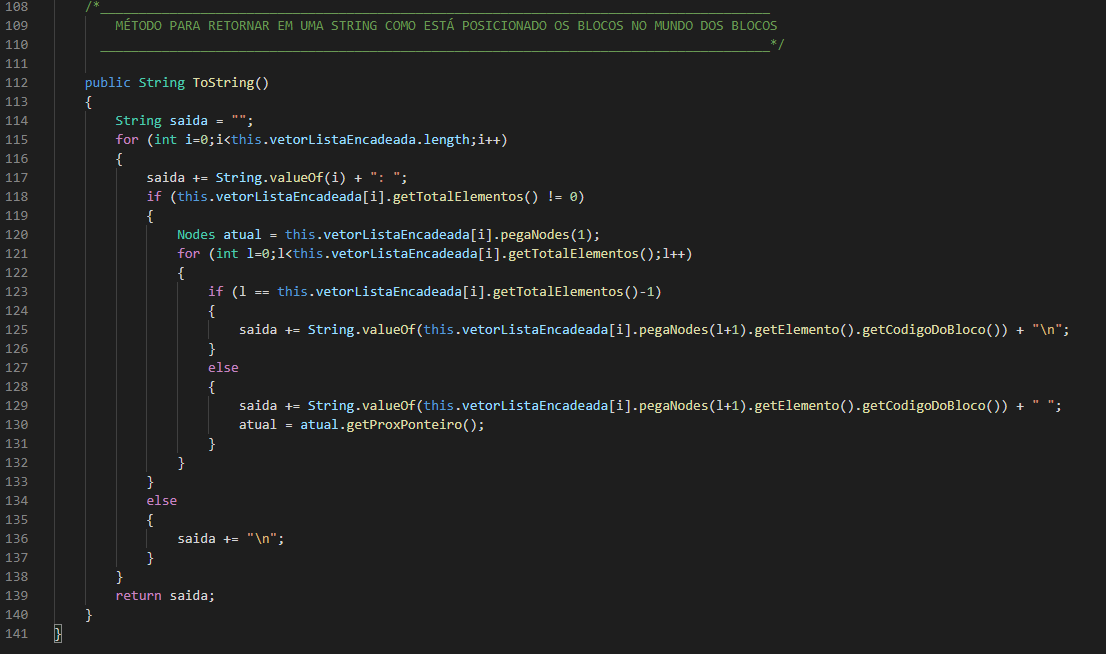
Figura



Figura



Figura



Figura