

# Estrutura de Dados I

## Lista duplamente encadeada

Matheus Gabriel

<2024-10-31 qui>

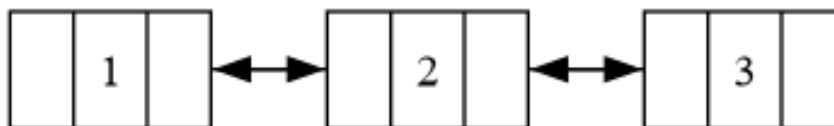
## Contents

<b>1</b>	<b>Introdução à lista duplamente encadeada</b>	<b>1</b>
1.1	Estrutura Visual com Graphviz . . . . .	1
1.2	Estrutura Visual Circular com Graphviz . . . . .	2
1.3	Estrutura Visual da Lista Duplamente Encadeada Circular com Graphviz . . . . .	2
1.4	Estrutura . . . . .	2
1.4.1	Importante, estilo pergunta de prova . . . . .	3

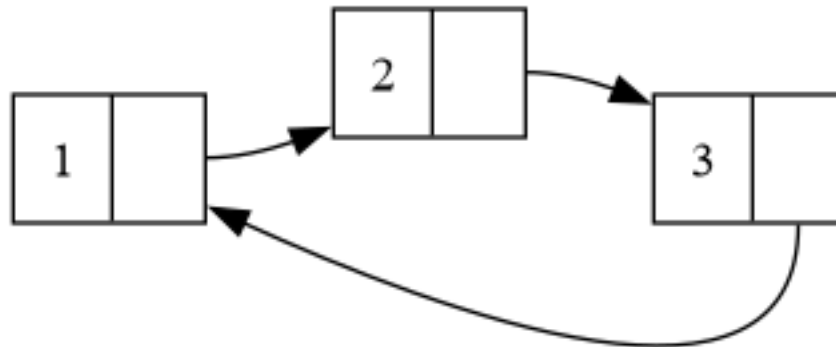
## 1 Introdução à lista duplamente encadeada

Na lista duplamente encadeada (diferente da simplesmente encadeada) o nó possui um endereço tanto pra trás dele quanto para frente, e além do *head* que aponta para o primeiro elemento, também temos obrigatoriamente o *tail* que sempre aponta para o último elemento.

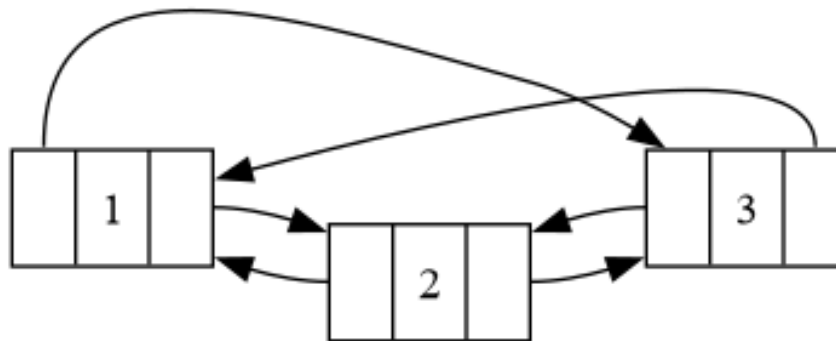
### 1.1 Estrutura Visual com Graphviz



## 1.2 Estrutura Visual Circular com Graphviz



## 1.3 Estrutura Visual da Lista Duplamente Encadeada Circular com Graphviz



A imagem ta meio errada mas a ideia permanece

## 1.4 Estrutura

No caso, a estrutura seria:

1. **previous**
2. **conteúdo**
3. **next**

Além disso, opcionalmente é possível ter um cursor **current** que armazena o nó atual sendo verificado.

#### 1.4.1 Importante, estilo pergunta de prova

Os TADs da lista circular e da lista [circular] duplamente encadeada são similares ao TAD da lista simplesmente encadeada.

Se eu alterar uma lista simplesmente encadeada por uma lista circular duplamente encadeada (por exemplo), o funcionamento não deveria quebrar, embora a abordagem interna mude um pouco (por motivos óbvios né, são tipos de lista diferentes).

Outra coisa, importante saber o que o `static` faz