

Resolução de Problemas Estruturados em Computação

Semana 1 - Introdução a ED e Arrays

Apresentação

Humberto Ferreira da Luz Junior

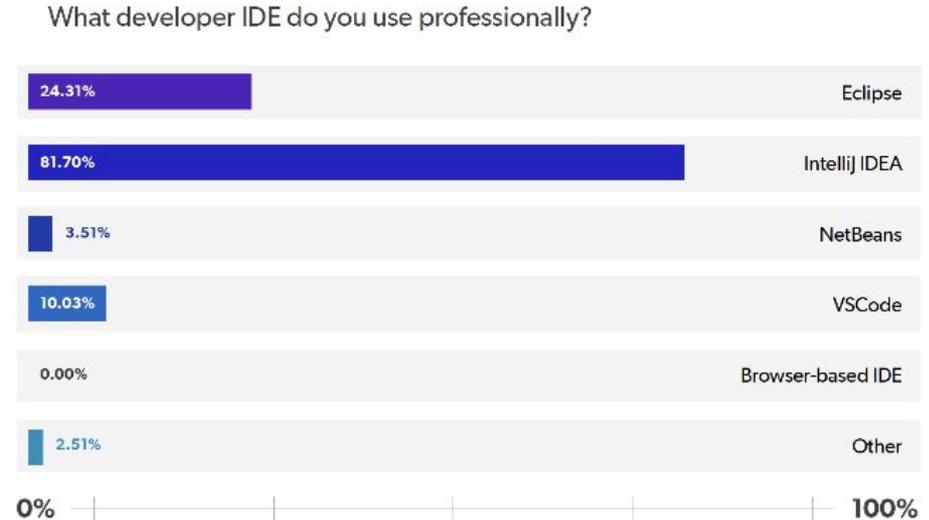
- Bacharel e mestre em Ciência da Computação pela UEL
- Especialista em Gestão Pública
- Analista senior na Assessoria de Tecnologia de Sistemas da UEL
- Diretor de desenvolvimento de sistemas desde 03/2020
- Fundador da Fábrica de Software da ATI/UEL
- Experiência de mais de 10 anos em Java EE, Linux e servidores de suporte ao desenvolvimento

Linguagem de programação: Java

- Mais fácil de ler e escrever do que C/C++
- Não tem ponteiros e alocação manual de memória
- Mais segura e menos suscetível a erros
- Orientada a objetos

Ambiente de Desenvolvimento (IDE)

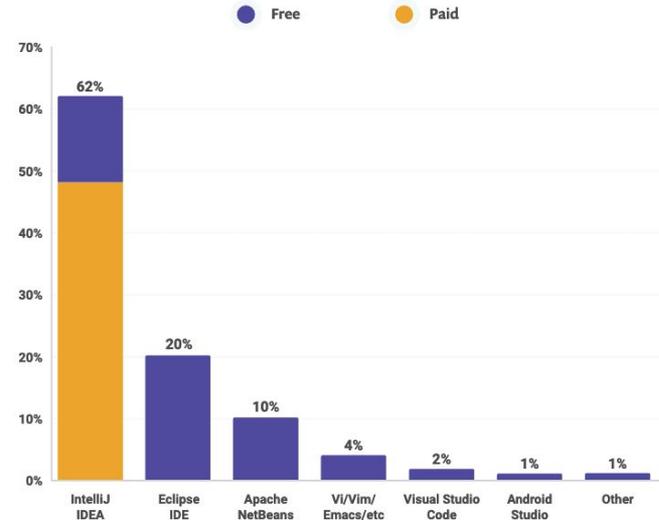
- [IntelliJ Community Edition \(recomendada\)](#)
 - [Aplicação de estudantes para a licença Ultimate](#)
- Eclipse
- Netbeans
- Visual Studio Code



Fonte: [JRebel Java Technology Report](#)

Ambiente de Desenvolvimento (IDE)

- [IntelliJ Community Edition \(recomendada\)](#)
 - [Aplicação de estudantes para a licença Ultimate](#)
- Eclipse
- Netbeans
- Visual Studio Code



Fonte: [JVM Ecosystem Report](#)

Ambiente de Desenvolvimento (IDE)

Projetos criados sempre utilizando o Maven:

- É uma ferramenta para gestão de projetos Java (dependência, construção, alteração de código, etc)
- Garante interoperabilidade entre os ambientes de desenvolvimento
 - Usado por padrão no Netbeans

Instalação da JDK

- <https://adoptopenjdk.net/>
- OpenJDK 11 ou 15

Github / Git

- [Download e instalação do Git](#)
- Disponibilização dos exemplos desenvolvidos
 - <https://github.com/hfluz>
- Entrega de projetos das atividades somativas e de frequência
 - Envio dos links dos repositórios pelo Blackboard

Formato das aulas

1. Explicação sobre uma estrutura de dados ou algoritmo
2. Apresentação de um problema para a turma resolver
3. Resolução gradual do problema ao longo da aula

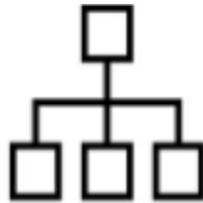
O que são estruturas de dados?

As estruturas de dados representam a forma como os dados são armazenados

- Listas lineares: Pilha, fila, lista encadeada.
- Listas não lineares: Árvores e grafos



Lista



Árvore

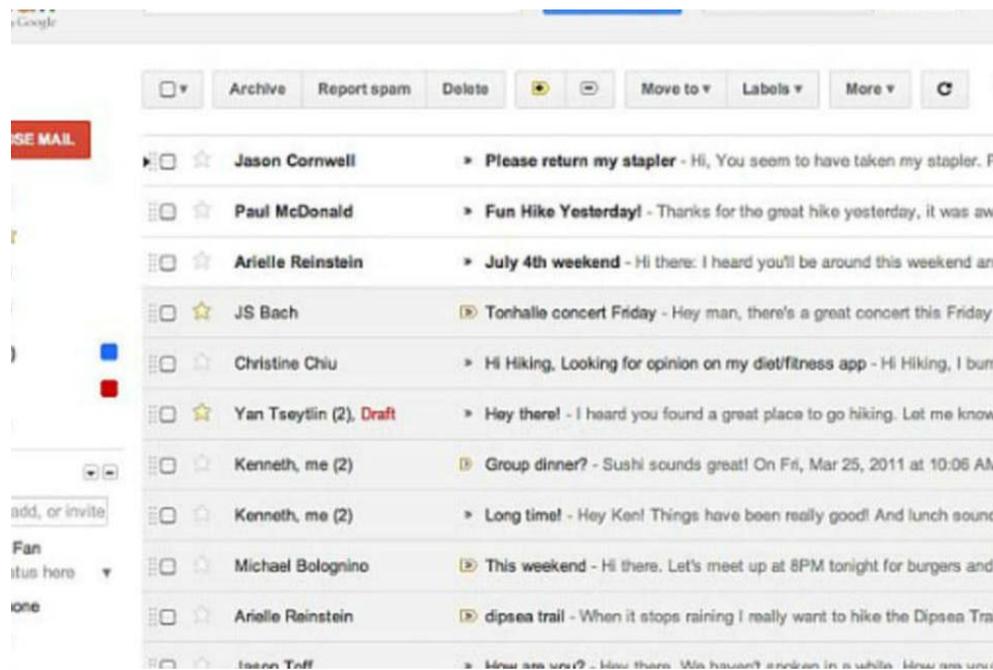
Algoritmos aplicados a ED

Os algoritmos são usados para manipular e acessar essas estruturas de dados

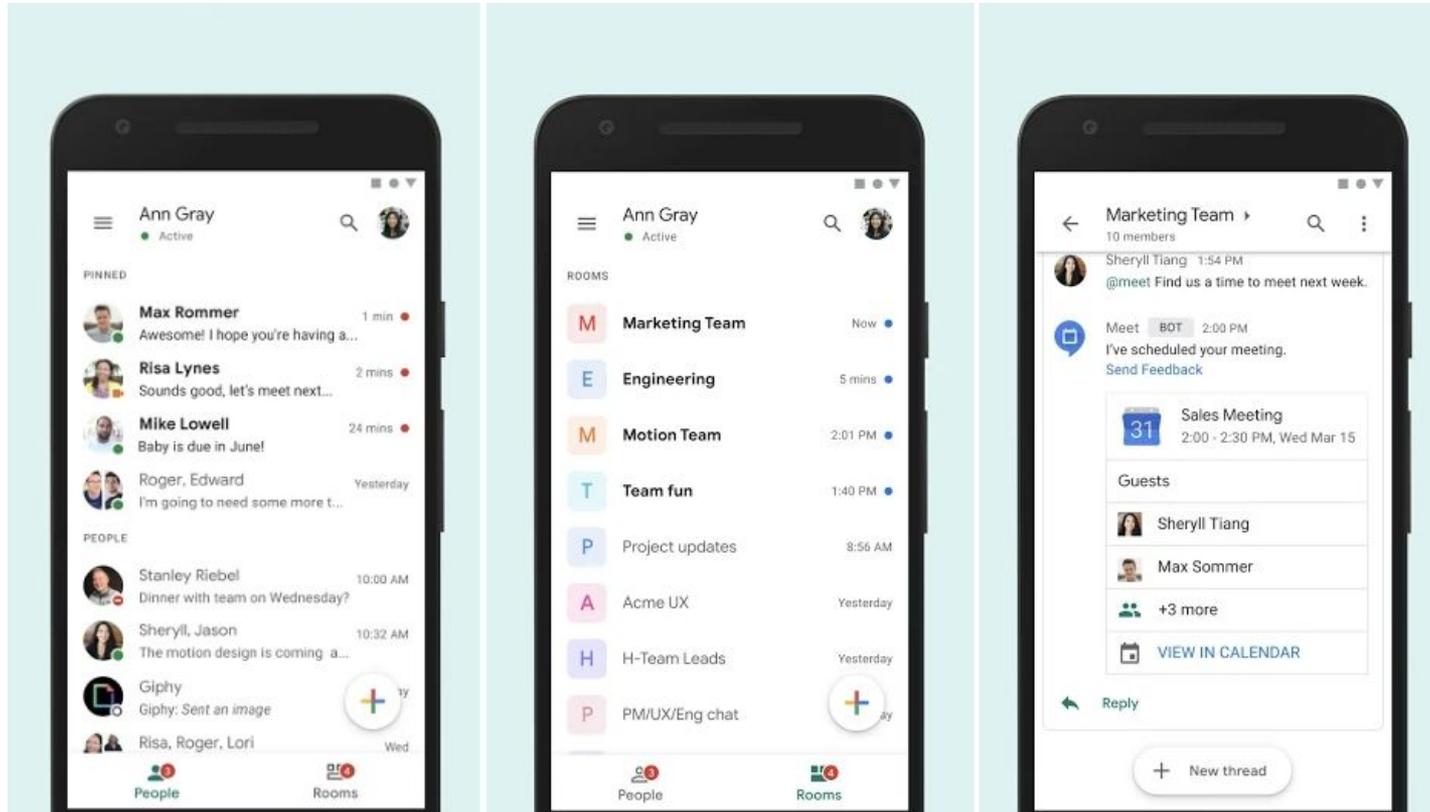
- Busca, ordenação, inclusão, exclusão, etc.



Por que estudar estruturas de dados e algoritmos?



Por que estudar estruturas de dados e algoritmos?



Por que estudar estruturas de dados e algoritmos?

- Disciplinas matriculadas na graduação
- Histórico no seu navegador de internet
- Cadastro de clientes
- Pedidos de compra em um e-commerce
- Imagens e vídeos

Por que estudar estruturas de dados e algoritmos?

Compras em um e-commerce

- E se você tiver milhões de pedidos?
 - Como fazer a busca de um pedido de forma eficiente?
 - Como você vai exibir os pedidos? Ordenados? E quando for necessário inserir um novo elemento?

Por que estudar estruturas de dados e algoritmos?

- A estruturas de dados podem ser usadas como uma ferramenta pelo programador

Por que estudar estruturas de dados e algoritmos?

- Processos seletivos em grandes empresas
- Competições de programação
- Resolução de problemas
- Prática de algoritmos para as outras disciplinas do curso
- Estímulo intelectual
- Diversão

Por que estudar estruturas de dados e algoritmos?

Toda estrutura de dados e algoritmo tem seus pontos fortes e suas fraquezas

Estrutura de dados	Vantagens	Desvantagens
Vetor	Rápida inserção, acesso rápido se você conhece o índice	Busca e exclusão lentas, tamanho fixo
Vetor ordenado	Busca mais rápida do que um vetor desordenado	Inclusão e exclusão lentas, tamanho fixo
Pilha	Acesso rápido aos últimos dados incluídos	Acesso lento aos outros dados
Fila	Acesso rápido aos primeiros dados incluídos	Acesso lento aos outros dados
Lista encadeada	Rápida inserção e exclusão	Busca lenta

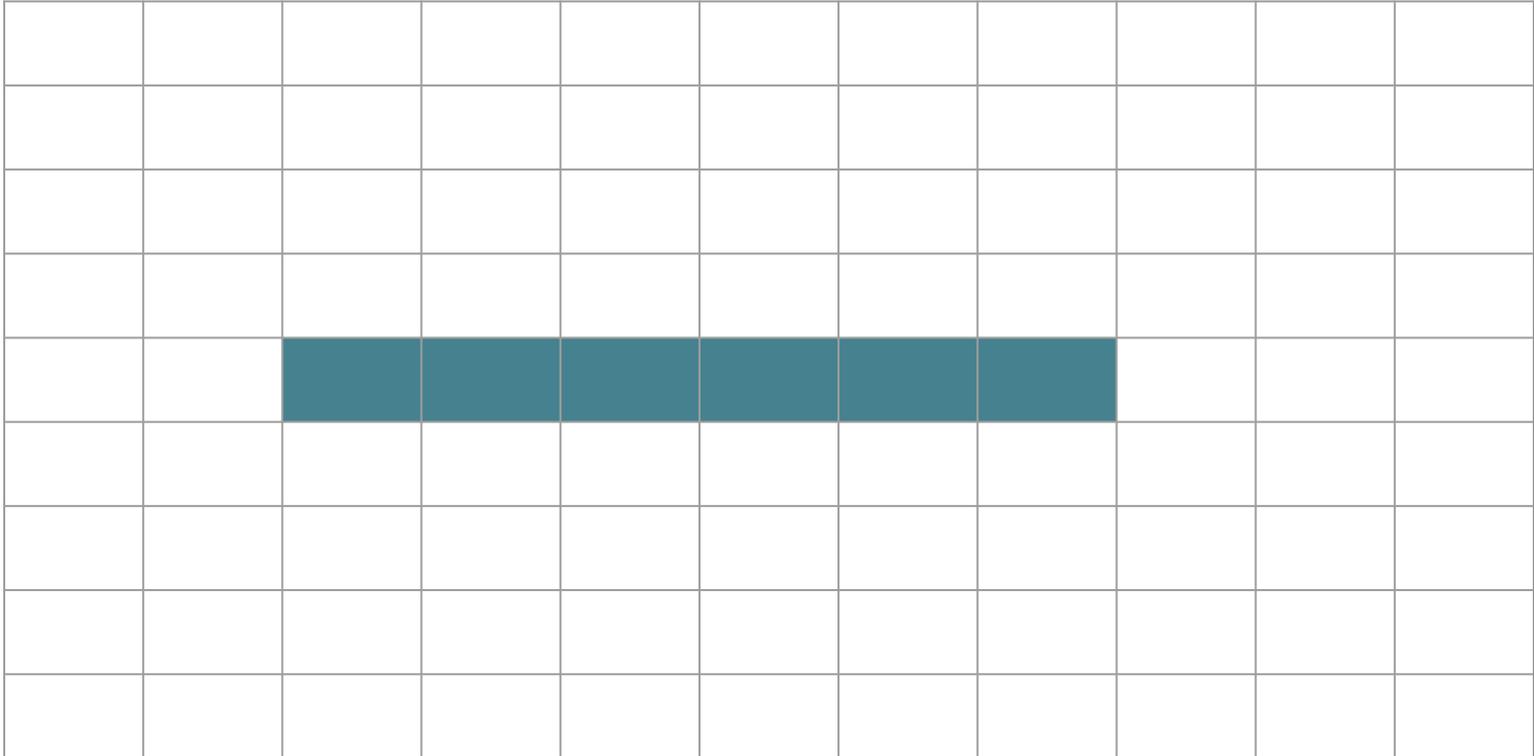
Arrays / Arranjos / Vetores

```
int[] intArray = new int[100]; //declaração e inicialização
int arrayLength = intArray.length; // obtém tamanho do array
int temp = intArray[3]; // obtém o 4º elemento do array
intArray[7] = 66; // insere o valor 66 na 8ª posição do
array
```

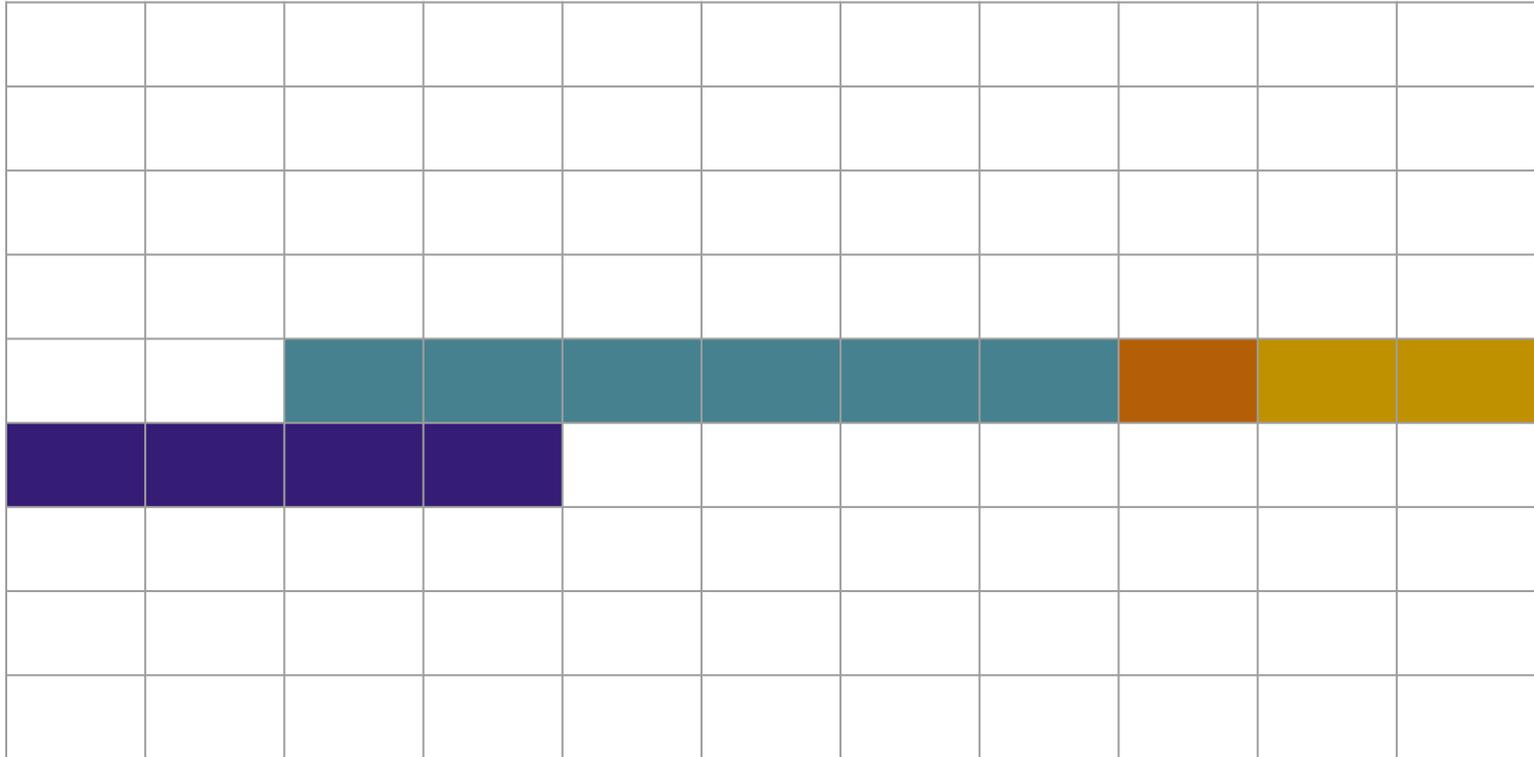
Inicialização de array primitivo:

```
int[] intArray = { 0, 3, 6, 9, 12, 15, 18, 21, 24, 27 };
```

Arrays - Alocação contígua de memória



Arrays - Alocação contígua de memória



Painel de Pontuação de um jogo

- São armazenadas as n maiores pontuações do jogo
- Para cada registro são salvos o nome do jogador e sua pontuação
- A pontuação é armazenada em ordem decrescente
- Podemos incluir uma nova pontuação e imprimir o painel com todas pontuações registradas

Diagrama de classes



Adição de um novo registro no painel

Registros		
Índice	Nome	Pontuação
0	Ricardo	79
1	Lúcia	57
2	Erick	50
3	Maria	42
4		

Vinícius	55
----------	----

Adição de um novo registro no painel

Registros		
Índice	Nome	Pontuação
0	Ricardo	79
1	Lúcia	57
2	Erick	50
3	Maria	42
4	Maria	42

Vinícius	55
----------	----

Adição de um novo registro no painel

Registros		
Índice	Nome	Pontuação
0	Ricardo	79
1	Lúcia	57
2	Erick	50
3	Erick	50
4	Maria	42

Vinícius	55
----------	----

Adição de um novo registro no painel

Registros		
Índice	Nome	Pontuação
0	Ricardo	79
1	Lúcia	57
2	Vinícius	55
3	Erick	50
4	Maria	42

Adição de um novo registro no painel

Registros		
Índice	Nome	Pontuação
0	Ricardo	79
1	Lúcia	57
2	Vinícius	55
3	Erick	50
4	Maria	42

João	43
------	----

Adição de um novo registro no painel

Registros		
Índice	Nome	Pontuação
0	Ricardo	79
1	Lúcia	57
2	Vinícius	55
3	Erick	50
4	Maria	42

João	43
------	----

Adição de um novo registro no painel

Registros		
Índice	Nome	Pontuação
0	Ricardo	79
1	Lúcia	57
2	Vinícius	55
3	Erick	50
4	João	43

Adição de um novo registro no painel

Registros		
Índice	Nome	Pontuação
0	Ricardo	79
1	Lúcia	57
2	Vinícius	55
3	Erick	50
4	Maria	42

Rafael	35
--------	----