

Roteiro do curso de Estrutura de Dados:

- **Aula – 1 Introdução ao Curso e Conceitos Básicos**
 - [Etapa 1] - Apresentação
 - Minha Formação
 - Onde trabalho
 - Com quê trabalho
 - O que me motiva estudar sempre
 - O que é estrutura de dados
 - A importância do assunto na tecnologia
 - A importância do assunto no dia-a-dia
 - Apresentação da ementa do curso
 - [Etapa 2] - Conceito de Memória
 - Tipos de memória do computador
 - Ideia geral do funcionamento da memória ram
 - Alocação de memória (Ideia geral)
 - [Etapa 3] - Instalação e Configuração do Ambiente
 - Instalação do Java JDK 11
 - Instalação da IDE IntelliJ
 - Rodar o primeiro programa
 - [Etapa 4] - Atribuição e Referência em Java
 - Diferença de atribuição de Valor e por referência
 - Como o Java trabalha com tipos primitivo em memória
 - Como o Java trabalho com objetos em memória
 - [Etapa 5] - Conceito de Nó e Encadeamento de Nó
 - Explicação teoria de um nó
 - Implementação da classe nó com Java
 - Encadeamento de vários nós
 - [Etapa 6] - Generics em Java
 - O que é?
 - Como utilizar
 - Vantagens da utilização
 - [Etapa 7] - Refatoração da Classe No
 - Codificação

- **Aula – 2 Pilhas [LIFO: Last In, First Out]**

- [Etapa 1] - Explicação teórica do conceito de Pilhas
- [Etapa 2] - Implementação da classe Pilha (Métodos: Top, Push e Pop)
- [Etapa 3] - Utilização da Classe Pilha

- **Aula – 3 Filas [FIFO: First in, First Out]**

- [Etapa 1] - Explicação teórica do conceito de Filas
- [Etapa 2] - Implementação da classe Fila (Métodos: Enqueue, Dequeue)
- [Etapa 3] - Utilização da Classe Fila
- [Etapa 4] - Refatoração para embutir o nó
- [Etapa 5] - Refatoração para utilização de *generics*

- **Aula – 4 Listas Encadeadas**

- [Etapa 1] - Explicação teórica do conceito de Listas Encadeadas
- [Etapa 2] - Implementação da classe No
- [Etapa 3] - Implementação da classe ListaEncadeada (isEmpty e size())
- [Etapa 4] - Implementação da classe ListaEncadeada (add())
- [Etapa 5] - Implementação da classe ListaEncadeada (getNo())
- [Etapa 6] - Implementação da classe ListaEncadeada (get())
- [Etapa 7] - Implementação da classe ListaEncadeada (remove())
- [Etapa 8] - Implementação da classe ListaEncadeada (toString())
- [Etapa 9] - Utilização da Classe ListaEncadeada

- **Aula – 5 Listas Duplamente Encadeadas**

- [Etapa 1] - Explicação teórica do conceito de Listas Encadeadas
- [Etapa 2] - Implementação da classe NoDuplo
- [Etapa 3] - Implementação da classe ListaDuplamenteEncadeada (size())
- [Etapa 4] - Implementação da classe ListaDuplamenteEncadeada (getNo(), get())
- [Etapa 5] - Implementação da classe ListaDuplamenteEncadeada (add())
- [Etapa 6] - Implementação da classe ListaDuplamenteEncadeada (add(index))
- [Etapa 7] - Implementação da classe ListaDuplamenteEncadeada (remove())
- [Etapa 8] - Implementação da classe ListaDuplamenteEncadeada (toString())
- [Etapa 9] - Utilização da Classe ListaDuplamenteEncadeada

- **Aula – 6 Listas Circulares**

- [Etapa 1] - Explicação teórica do conceito de Listas Circulares

- [Etapa 2] - Implementação da classe No
- [Etapa 3] - Implementação da classe ListaCircular
- [Etapa 4] - Utilização da Classe ListaCircular
- **Aula – 7 Árvores**
 - [Etapa 1] - Explicação teórica do conceito de Árvores
 - [Etapa 2] - Explicação teórica de Árvore de Busca Binária
 - [Etapa 3] - A interface Comparable
 - [Etapa 4] - Explicando operações básicas em árvores de busca binária
 - [Etapa 5] - Explicando atravessamento em Árvores Binárias
 - [Etapa 6] - Implementação da classe NoArvore (Nó Árvore)
 - [Etapa 7] - Implementação da classe ArvoreBinaria
 - [Etapa 8] - Utilização da classe ArvoreBinaria
- **Aula – 8 Principais Implementações das estruturas de dados do Java**
 - [Etapa 1] - Conceitos de equals e hashCode
 - [Etapa 2] - Stack (Pilha)
 - [Etapa 3] - Queue (Fila, LinkedList)
 - [Etapa 4] - List (ArrayList)
 - [Etapa 5] - Set (HashSet)
 - [Etapa 6] - Map (HashMap)
 - [Etapa 7] - Conclusão