

TUTORIAL: ESTRUTURA DE DADOS - ÁRVORES

MÓDULO 2

Olá, caro aluno.

Este é o material de apoio para estrutura de dados **Árvores**. Então agora você já aprendeu sobre listas, pilhas, filas. Existe uma outra estrutura chamada árvore, que é aplicada em algumas situações de dados, de programação e muitas vezes em situações organizacionais. Então pensei que seria interessante a gente citar aqui. Já tive experiências em criando certas Árvores como relatórios, não diretamente como codificação.

Entretanto, como nesse módulo também estamos falando sobre análise de sistemas, achei interessante inserirmos esse tema. Então, quando falamos de Árvores, estamos falando do quê? Precisamos de certos **Nós**. Mas quais **Nós**? Temos **Nós**, galhos, chamamos de folhas também.

Existem vários conceitos, mas os **Nós** vão se lugar uns aos outros, existe uma certa hierarquia. Então, por exemplo, o que eu posso fazer aqui? Eu vou utilizar o Paint rapidamente. Então, temos certos **Nós**.

Deixa eu colocar aqui uma situação melhor. Então, temos certos **Nós**. Esse **Nó** podemos chamar de **Nó pai**.

Teremos outros **Nós**. E **Nó** filho, que esse **Nó** filho vai ter outros **Nós** filhos e assim vai indo. Nossos dados podem ser estruturados dessa forma, dependendo da necessidade.

A partir do momento que estamos falando de uma estrutura organizacional, como o nosso código vai ser exemplo, como o seu material está de exemplo, podemos trabalhar com um sistema de estrutura de dados de **Árvore**. Então, **Árvore** vai ter aqui uma relação. Então, a **Árvore** X vai ter a relação com o Y, com o Z. Y vai ter com A, B e C e assim subsequentemente, ok? Como eu disse, essa é uma situação mais conceitual.

Não vou adentrar muito aqui na situação de código, porque realmente as mais importantes são as listas, pilhas e filas quando estamos falando de desenvolvimento. Mas é interessante esse bônus aqui, é interessante estar agregado ao nosso conhecimento. Como eu citei no material, um profissional de excelência no mercado necessita ter esses diferenciais, saber de listas é um ponto básico para um desenvolvedor.

Saber de pilhas e filas eu diria já que é um pequeno plus. Saber como funcionam **Árvores** pode ter certeza que é um diferencial, a pessoa que está te entrevistando ou a pessoa que te conhece, um manager, um superior, ele já vai entender que você tem uma visão de um analista, uma visão maior do que simplesmente de codificação. Não estou, de forma alguma, falando de quem conhece apenas a codificação, mas quanto mais conhecimentos agregarmos melhor profissional seremos para o mercado.

Então, sem mais delongas, vamos utilizar, por ser uma situação bem complexa para criar funções, nós vamos utilizar o anytree. Então, dentro desse módulo, vamos importar Node and rendertree. Esse Node será importante para criarmos os **Nós** e o rendertree, o Render, como eu já disse, ele vai renderizar a **Árvore**, vai criar, vai plotar, vai imprimir a nossa Árvore. Então, inicialmente, como temos lá no nosso material, se você quiser ter aí do seu lado, aberto, temos lá no material.

Importante aqui, uma curiosidade. Talvez você tenha visto meus outros vídeos, que eu consigo comentar todos e, ao mesmo tempo, descomentar. Então, você pode utilizar junto as teclas CTRL, K, C, ele vai comentar e CTRL, K, U, ele vai descomentar.

Ou você insere diretamente a cerquilha, a hashtag, em cada linha. Então, o que ele vai fazer aqui, pessoal? O **Nó** principal vai ser CEO, como você tem no seu material. Então, vai criar CEO, recebe a variável CEO, recebeu um Node, um Node que vai ser com o nome, o texto CEO dentro.

Lembre-se, isso aqui é uma variável, isso aqui é simplesmente o texto. Esse texto não sabe que isso aqui é uma variável. Então é simplesmente um texto, eu coloquei qualquer coisa aqui, isso aqui não importa. Então, segundamente nós criamos quem? Criamos a segunda parte da hierarquia. Então a gerente recebe um **nó** filho, com texto gerente, sendo que quem que é o pai? O pai vai ser CEO.

Então, na primeira linha, nós teremos CEO, na segunda já teremos gerente. Então, vai ter uma linha. Até interessante, eu vou fazer aqui por etapa. Vou trocar C e eu vou renderizar CEO gerente. Então, vamos lá, na linha raiz, no nó raiz, na linha zero, vai ter o quê? CEO, e na segunda linha vai ter o quê? Gerente que tem a relação com CEO como filho, o gerente vai ser o filho do CEO, ok? Pra isso, pra renderizar, pra imprimir, temos essa função aqui, ele faz um For, um looping de repetição, isso eu vou explicar melhor no módulo 3, porque iremos trabalhar com do, while, for, que são tipos de repetições.

Então, ele vai fazer essa impressão, vamos testar, o que tem que aparecer? Tem que aparecer CEO conectado com o gerente como seu filho. Vou executar, olha lá que bacana, CEO filho gerente, perfeito? Então, vamos também fazer isso com o diretor.

Diretor recebe node diretor, com o pai que vai ser CEO. Então, gerente e diretor estarão no mesmo nível executando gerente e diretor, ok? Agora, vamos colocar o quê? Vamos colocar que está embaixo de gerente. Colocamos supervisor, empregado 1, empregado 2. Então, com seus nomes, o parent gerente.

Diretor vai ter filho? Não, diretor não vai ter filho, O diretor não vai ter o **nó** filho. Então, se executarmos, o que nós teremos? CEO, gerente, supervisor, empregado 1 e 2, filhos do gerente, e diretor, sem nenhum child, sem nenhum nó raiz. Mais um nível aqui que nós temos, estagiário, Que vai ser funcionário, E hierarquicamente vai respeitar o empregado 1. Novamente, executamos, e olha lá que bacana.

Nível 0, CEO, gerente respeitando o CEO, diretor respeitando o CEO, supervisor, empregado 1 e empregado 2 respeitando o gerente, estagiário respeitando o empregado 1, ao gerente, ao CEO. Perceba, essa é uma estrutura de uma árvore. Então, muitas vezes, nós vamos utilizar esse conceito na parte da análise de sistemas.

Então, é importantíssimo que você aprenda isso, que você crie esse mindset de estruturação de dados utilizando Árvores, ok? Para que a gente possa continuar aqui, o que eu vou fazer? Eu vou fazer como que seja possível inserir um novo filho. Então, para inserir um novo filho é muito básico, eu criei aqui uma nova variável, novo funcionário, recebe node, novo funcionário, e vamos fazer

que esse novo funcionário seja abaixo hierarquicamente de diretor, certo? Então, vamos fazer isso.

Eu vou colocar aqui para imprimir, executo, olha que bacana. Agora o diretor tem um novo funcionário. Marlon, agora eu gostaria de demitir alguém. Eu tenho o poder, sou CEO, estou codificando aqui, eu quero demitir alguém. Então, para isso, você vai pegar a variável que você criou aqui em cima, Colocar parent none, tá? Então, none quer dizer o quê? Que ele não tem mais, ele não faz parte mais dessa árvore.

Se ele não faz parte mais dessa árvore, então, ele vai aparecer? Não, ele foi demitido. Então, a partir do momento que eu demitir o empregado 1, você também não vê mais o estagiário. Olha que interessante, o estagiário era abaixo do empregado 1. Então, a partir do momento que eu tirei esse nó, todos os **Nós** filhos também foram removidos.

Eu não vou me alongar mais aqui porque realmente a ideia de estudar a árvore é realmente um bônus. Entretanto, você está totalmente livre para fazer outros testes, criar novas formas, criar vários níveis, criar novos filhos e tudo mais que você queira testar

Obrigado, caro aluno.

Até já!