

TUTORIAL: ESTRUTURA DE DADOS - PILHAS

MÓDULO 2

Olá, aluno!

Esse é o material de apoio para a estrutura de dados **Filas**. Então, o conceito de **Fila** é um pouco mais fácil quando nós colocamos uma analogia para o conceito de mundo real.

Basicamente, se você chega em uma **Fila** de banco, você chegou na **Fila**, não tem ninguém, quem vai ser o primeiro a ser atendido? Você. Então, se você é o 1 e você foi atendido, a **Fila** fica vazia. Se você estava chegando no banco, uma pessoa chegou um pouco mais rápido que você, então a pessoa 1 vai entrar na **Fila**, ou seja, first in, primeiro a entrar, você entrou na **Fila** também, mas você vai ser o 2.

Então, o primeiro que entrou foi a pessoa 1. Quem que é a primeira pessoa a sair na **Fila**? É a pessoa 1. Então, first in, first out. Primeiro a entrar, primeiro a sair. Esse é o conceito de **Fila**.

Fila de mercado, **Fila** de banco. Na tecnologia da informação, podemos falar em **Fila** de processos. Então, logicamente, não podemos ser simplesmente superficiais e simplistas. Mas os processos, a partir do momento que um deles inicia, dentro do nosso sistema operacional, ele tem certas prioridades. Então, se ele tem uma prioridade igual a todos, se ele for o primeiro, ele tem que ser o primeiro a ser executado e a ser finalizado. A não ser que aconteça uma situação, por exemplo, do acionamento de um antivírus, o antivírus tem uma prioridade maior.

Então, o processo do antivírus acabará pulando, mas será um caso de exceção. Mas vamos fazer aqui uma situação planificada. Todos têm a mesma prioridade, se todos têm a mesma prioridade, então o primeiro que entrou é o primeiro que vai sair.

Em uma fila real, os idosos, por exemplo, poderiam pular a fila. Mas se todos tem menos de 65 anos, não somos PCD, não estamos gestantes, com criança de colo, ou qualquer outra situação de acessibilidade, acaba que todos estão com a mesma priorização. Então, se você chegou na **Fila**, está em segundo, você será o segundo a ser atendido e a ir para a sua casa.

No conceito de programação é a mesma situação. Então vamos lá. Inicialmente, você importará o módulo Queue, posteriormente você criará uma variável chamada fila Soveteria, que receberá queue.queue. Esse é o método que tem dentro desse módulo, é a situação que essa biblioteca trabalha e entenda que várias bibliotecas de Python podem trabalhar de certas formas.

Por isso é importante sempre ler a documentação. Você pode procurar tanto no Google quanto dentro aqui do Workstations. Então, por exemplo, essa queue é dentro desse PyLance.

Você pode procurar em PyLance, é o que você vai achar. Você vai ver aqui o resource, marketplace, license, features, tudo que tem aqui dentro você pode verificar. Então, fica aqui uma curiosidade para vocês explorarem, ok?

Então criamos fila_soveteria = queue.queue (). Nessa **Fila**, então o que nós fizemos? Deixa eu fazer o seguinte.

Coloquei aqui, fila_soveteria, usei o comando put. Então, put vai ser João. O João vai ser o primeiro da **Fila**. Coloquei na **Fila** também Maria e depois a Márcia.

O João é o primeiro, Maria é o segundo, Márcia é a terceira. Certo? Então, vamos lá. Vamos mostrar quem que é o primeiro da **Fila**.

Vou colocar aqui, prox_client = fila_soveteria.Queue[0]. Então, veja que essa biblioteca trabalha diferentemente da de pilhas.

Se você não consegue acessar diretamente, você tem que acessar a variável queue, e aí o índice. Então, lembre-se que o índice sempre se inicia em zero, tá? De listas, de pilhas e tudo mais. Então, se você quer o primeiro item, utiliza o índice zero. Quem que vai ser o próximo cliente aqui? Opa, falha técnica. Então, ele realmente absorveu aqui a variável, mas faltou o quê? Faltou imprimir. Então, vamos imprimir.

Imprimir próximo cliente na **Fila**. Então, absorveu a variável prox_client, recebeu o cliente, e aí nós fizemos um print com o texto próximo cliente na **Fila**, qualquer concatenando com o valor da variável prox_client. Agora sim.

Novamente, eu pedi para os nossos colegas não editarem isso, porque realmente esses erros podem acontecer. E é muito interessante. Na nossa vida, nossa carreira de programação, às vezes acabamos esquecendo básicos devido às milhares de linhas de código.

Aqui eu fiz a parte de atribuir um valor para a variável prox_client. Desculpe. Entretanto, faltou imprimir. É interessante deixar essa situação aqui para o nosso aluno. Ok. Então, João. Quem foi o primeiro? O primeiro foi João. Então, está certo. Marlon, eu quero saber quem que é o segundo. Se você quer saber quem que é o segundo, você pode fazer isso aqui. Em vez de zero, você coloca aqui. Você faz o seguinte.

Você pode imprimir aqui diretamente. Print, segundo da **Fila**. Pegamos essa variável inteira e colocamos o segundo índice da Fila. O segundo índice é um. Lembre-se, o primeiro índice é sempre zero. O segundo vai ser um.

Então, o segundo da Fila é fila_soveteria.queue[1]. O que nós teremos? João e Maria. Perfeito. Você pode também, nos seus exercícios, nesses testes que você está fazendo, executar um, dois, três. E se eu testar aqui? Eu tenho zero, um e dois. E Marlon, se eu executar aqui o índice três? Que vai ser o quarto da **Fila**.

O que vai acontecer? Deque index out of range. Então, por quê? Porque não tem quarto item, não tem índice três. Ele vai acusar index out of range.

Então, não existe. Não tem, está fora do range de índices. Ok? Perfeito.

Então, agora eu vou tirar esse aqui. Vou adicionar uma outra instância. Vou tirar esse aqui. Ele imprimiu o João de volta. Beleza. Agora, pessoal, nós queremos desenfileirar.

Então, para isso, nós temos o comando... É o inverso, não é? Não tem o put. E nós temos para quem? Então, o retrieve, que é o get. Então, o get nós vamos tirar a primeira pessoa da **Fila**. A primeira pessoa da **Fila** é João. Certo? Se eu tirar João, quem vai se tornar a primeira pessoa da **Fila**? Vamos executar. E o que ele vai imprimir aqui? Ele vai imprimir, inicialmente, João, que é a primeira parte.

Então, não tinha... Eu coloco aqui. João entrou na **Fila**. Maria entrou na **Fila**.

Marcia entrou na **Fila**. Aí, ele mostra quem é o primeiro da **Fila**. E mostra... Desculpa.

Ele atribui quem é o primeiro da **Fila** e mostra essa variável aqui. Nessa linha, cliente_atendida = fila_sorveteria.get, ele desenfileira o primeiro da **Fila**.

Eu posso desenfileirar a Maria? Não. Então, essa é a ideia. Todos estão com a mesma prioridade.

É obrigatório que o primeiro que entrou é o primeiro a sair a partir do momento que eu estou utilizando o conceito de estrutura de dados específico em **Filas**. Se eu uso o get, ele vai sair o primeiro e não tem como tirar outro. Isso é mandatório.

Isso é obrigatório. Então, a partir do momento que eu tirei o João da **Fila**, eu vou lá prox_cliente = fila_sorveteria.queue[0]. Quem que é o índice zero agora? É Maria. Por quê? Porque o João já saiu da **Fila**.

E ele imprime quem? Maria. Então, aluno, acredito que você esteja já ficando mais maduro nessa estrutura de dados, nessa situação de estrutura de dados. Já aprendeu sobre listas, pilhas e **Filas**.

Então, pode saber que isso é muito importante para a organização de dados. Isso também, não apenas diretamente no SQL, mas quando você trabalhar, quando você estiver na parte de back-end, não de data, mas back-end dentro, por exemplo, do Python ou outras linguagens como C Sharp, Java, você tem que fazer essa manipulação.

Realmente, como eu disse no material, algumas coisas são um pouco maçantes, até pejorativamente acabam se falando chatas, mas são extremamente cruciais. Isso é um profissional de mercado de muitos anos que está falando para vocês, não querendo ser arrogante, de forma alguma.

É extremamente crucial entender essas estruturas. Isso te facilita na forma de codificar, te ajuda a codificar, te ajuda a ter pensamentos de análise pré-conceituais, até de levantamento de requisitos. Então, você consegue até criar esse mindset quando você vai gerar até documentos.

Isso vai te ajudar. Você vai perceber que tudo isso que nós estamos fazendo aqui vai te ajudar não simplesmente para codificar, mas até na sua vida pessoal e profissional. Ok? Obrigado.

Até já.