



Residência
em Software

Módulo Programação JAVA (Avançado)

MÊS 01

INSTITUIÇÃO EXECUTORA



UESC

COORDENADORA



APOIO

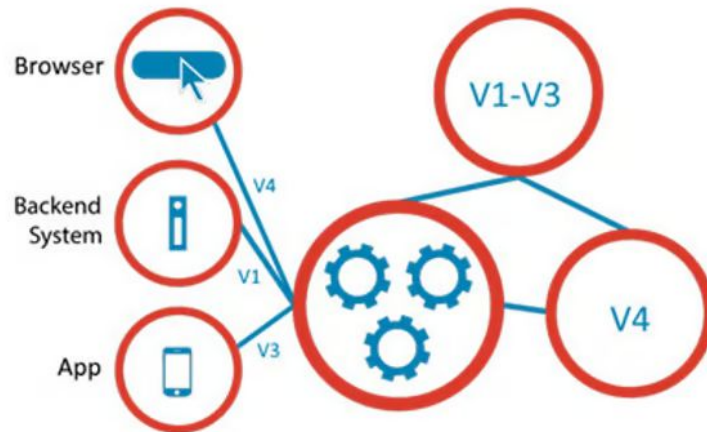
MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Paginação, Filtragem e Classificação com **Spring Boot**

Paginação de dados

Uma das coisas mais importantes para tornar um serviço WEB amigável é o **tempo de resposta**, e a **paginação** se torna essencial por esse motivo.



Paginação, Filtragem e Classificação com **Spring Boot**

Paginação de dados

Quando você tem um **conjunto de dados 'grande'**, você não desejaria enviar os dados para o cliente de uma só vez. A menos que seja uma **funcionalidade de exportação** específica, você não deveria enviar de volta todos os dados para o usuário.

Paginação, Filtragem e Classificação com **Spring Boot**

Paginação de dados

Imagine que você tem **100 mil linhas** em sua tabela e cada linha tem 10 colunas. Se o usuário está visualizando **apenas 20 linhas** na interface do usuário (nosso cliente) em um momento específico, por que enviar as **100 mil linhas**?

Paginação, Filtragem e Classificação com **Spring Boot**

Paginação, filtragem e classificação de dados

Em vez disso, você deveria enviar apenas um **subconjunto dos dados**.
Ao mesmo tempo, não queremos restringir o usuário de ver todos os dados.

Se o usuário quiser **adicionar um filtro** ou **classificar alguns dados** dessas 100 mil linhas, ele deveria conseguir fazê-lo.

Paginação, Filtragem e Classificação com **Spring Boot**

Paginação, filtragem e classificação de dados

De uma perspectiva de front-end, se o usuário clicar na próxima página, ele fará outra chamada para o back-end para obter o próximo conjunto de dados.

Paginação, Filtragem e Classificação com **Spring Boot**

Paginação, filtragem e classificação de dados

Ex: Lista de Livros na Amazon. Então, como podemos mostrar esses livros em uma página da web?

Há duas maneiras de mostrar esses livros em uma página da Web:

1. Mostrar todos os livros de uma só vez em uma única página da web, o que poderia custar mais memória e para uma Amazon como banco de dados de livros, isso não deveria ser feito.
2. Dividir conjuntos do total de livros por determinado tamanho e mostrar o conjunto selecionado em uma página da web. Além disso, aqui o usuário tem a oportunidade de selecionar qualquer conjunto que ele precisa ver, como abrir uma página específica em um livro.

Paginação, Filtragem e Classificação com **Spring Boot**

Paginação, filtragem e classificação de dados

Nesse contexto, podemos determinar Classificação como o conjunto paginado deve ser classificado. Suponhamos que você precise classificar toda a lista de livros pelo **Número ISBN** (*International Standard Book Numbering*), lá você pode usar classificar com Spring Boot e obter resposta ordenada e paginada.

Paginação, Filtragem e Classificação com **Spring Boot**

Paginação, filtragem e classificação de dados

A **Filtragem** é um processo utilizado para refinar ou limitar o conjunto de dados de acordo com critérios específicos estabelecidos pelo usuário. Por exemplo, consideremos uma situação onde temos uma vasta lista de livros e desejamos selecionar apenas aqueles que correspondem a um determinado **nome de autor** ou **título** específico.

A filtragem permite que apliquemos esse critério específico para isolar e obter apenas os dados relevantes do conjunto total. Dessa forma, o usuário final pode facilmente encontrar os livros de interesse sem precisar revisar toda a lista.

Paginação, Filtragem e Classificação com **Spring Boot**

Paginação, filtragem e classificação de dados

A **Classificação**, também conhecida como ordenação, é o processo de organização dos dados em uma sequência ou em categorias baseadas em um, ou mais critérios pré-determinados. Esse arranjo é feito para facilitar a análise, a visualização ou a apresentação dos dados de maneira mais eficiente e intuitiva.

Por exemplo, suponha que temos a mesma lista extensa de livros mencionada anteriormente. Além de querer filtrar os livros por nome de autor, podemos também querer ordená-los de acordo com diferentes critérios, como data de publicação, título, ou mesmo o preço do livro.

Paginação, Filtragem e Classificação com **Spring Boot**

Paginação, filtragem e classificação de dados

A classificação pode ser feita de forma **ascendente** ou **descendente**. No caso de uma **ordenação ascendente** por data de publicação, os livros **mais antigos** apareceriam primeiro na lista, enquanto numa **ordenação descendente**, os **mais recentes** estariam no topo da lista. Isso permite que o usuário final visualize os livros conforme a ordem de relevância ou preferência pessoal especificada, tornando a busca e a análise dos dados mais direcionadas e práticas.

Paginação, Filtragem e Classificação com **Spring Boot**

Exemplo de Uso

Usaremos como exemplo o Pageable no Spring Data Core, Spring Data JPA em uma API REST, implementando a paginação com classificação no Spring Boot.

Paginação, Filtragem e Classificação com **Spring Boot**

Adicionando Paginação na API

Etapas 1: Repository

- <https://gist.github.com/rog3r/1323778c986b34b1d20b932a64788d5c>

Etapas 3: Service

- <https://gist.github.com/rog3r/f601e385eec937ab5642366bf38e4ae4>

Etapas 4: ControllerV1

- <https://gist.github.com/rog3r/68a2f120b74b6bbb72e5b07572f93d1e>



Residência
em Software



Contato

rogerio.jesus@cepedi.org.br

<https://moodle.residenciatic18.cepedi.org.br/>