**Faculdade de Tecnologia de Mogi das Cruzes**

Documento de detalhes do funcionamento do protótipo para a matéria de Engenharia de Software 2.

Membros: Alessandra Cruz, Gabriel Oliveira, Juan Sanchez, Rafael Cordeiro

1. **Intervenção**

Este protótipo foi criado com a intenção de demonstrar o funcionamento das tecnologias escolhidas para o projeto. Para tal, foi feito uma demonstração de página web com filtro de registros feitos com o processamento no frontend. Tais filtros como Strings e Faixas numéricas de números inteiros e reais.

* 1. **Caso de Uso**

O protótipo traz um caso de uso conveniente ao projeto. Uma pessoa que deseja pesquisar informações sobre pessoas, podendo filtrar a lista com informações, como nome, cargo, faixa etária e faixa salarial.

Para o portal da transparência da câmara, são preferíveis alguns desses filtros que o protótipo traz.

1. **Tecnologias utilizadas**

O protótipo utiliza as seguintes tecnologias:

Front-end:

* HTML 5;
* CSS 3;
* Javascript (jQuery + Ajax).

Back-end:

* Java web (JDBC MySQL + Servlets);
* Banco de Dados MySQL;
* Servidores Apache Tomcat e Wampserver.

1. **Funcionamento**

O protótipo é funciona em uma comunicação Cliente – Servidor, aonde o cliente envia uma solicitação vazia ao servidor, por meio de requisição dinâmica (AJAX), ao carregar a página, o qual puxa os dados do banco de dados, os agrupa em uma estrutura JSON e a retorna ao cliente, como resposta.

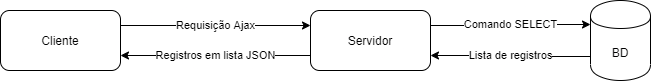
****

Imagem mostra fluxo de informações do sistema

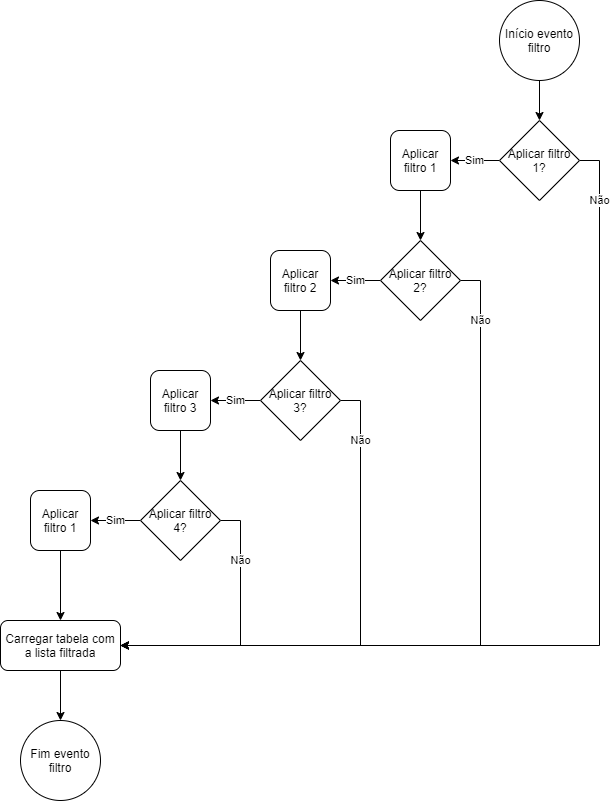
**3.1 O algoritmo de filtros**

O algoritmo de filtros é implementado nos scripts do lado do cliente e funcionam de forma incremental ou em cascata, aonde a filtragem dos dados é feita conforme os filtros são adicionados.

O lado do cliente recebe os dados em um vetor de objetos, que serve de backup para carregar um outro vetor de objetos que, por sua vez, recebe os registros provenientes do filtro, carregando as células da tabela da página.

Cada um dos 4 filtros possui uma flag, dizendo se está sendo utilizado. Toda vez que é inserido um filtro, seja adicionando ou apagando um caractere do nome, ou aplicando filtro de faixa numérica, um evento, que realiza o processo de filtragem é disparado e, após a filtragem, os dados do vetor da lista filtrada são inseridos na tabela.

Os filtros podem ser retirados ao apagar os valores dos campos de inserção da página, ou clicando no botão “Limpar Filtro”, que chama o mesmo evento de filtro, porém com todas as flags falsas. Fazendo a lista ser exibida do jeito que ela veio.

Fluxograma que mostra como funciona o procedimento de filtragem

* 1. **Requisições dinâmicas com AJAX e jQuery**

Um dos desafios foi encontrar um meio de programar requisições que não necessitassem recarregar a página. Tanto por uma questão de praticidade, como por uma questão de redução do trânsito de dados. Para tal, utilizamos o AJAX aliado com o encurtamento de código proporcionado pelo jQuery.

A requisição é realizada quando se abre a página que, ao receber, exibe na tela os resultados.

Para fins de depuração, na página há um botão “Refresh” que, ao ser clicado, realiza mais uma requisição ao servidor. Se o banco de dados for atualizado antes da requisição ser feita, a nova tabela será recebida pelo cliente, sem que seja ao menos necessário atualizar a página.

1. **Processamento no servidor**

No servidor, há uma divisão entre camada de controle e modelo, representado por uma classe Servlet (Controller) como controle e pelas classes de comunicação com o banco de dados JDBC (QueryExecuter) e montagem de resposta (DataEncapsulator) como modelo.

* 1. **Modelo e Controle**

A classe Controller, como camada de controle, recebe a requisição e chama o método catchData() da classe DataEncapsulator, da camada de modelo, que interage com as classes que implementam a conexão com o banco de dados MySQL, conseguindo assim os objetos que possuem os dados e os metadados necessários para montar a lista de objetos JSON.

O pacote do servidor também conta com uma classe JSON que estrutura os dados com o padrão chave-valor. Durante a implementação do método catchData(), um vetor de objetos da classe JSON é criado para receber todos os dados e transformá-los em uma string com o padrão JSON. Por meio de algumas iterações, os dados são inseridos no vetor JSON e são convertidos para uma string no padrão lista JSON, aonde são retornados para a chamada, na classe Controller.

A classe Controller então, recebe o retorno da função catchData em um objeto que realiza a resposta e prepara o padrão de resposta em um objeto que manipula comunicação HTTP.

* 1. **Banco de Dados**

O banco de dados foi desenvolvido de forma bem simples para poder representar uma fonte de onde os dados viriam.

Visando a segurança da informação e evitando a redundância de dados, o banco de dados do projeto será mais elaborado, atendendo os requisitos do cliente.

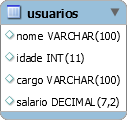


Diagrama da tabela implementada no banco de dados

1. **Como executar o protótipo**

Recomenda-se o uso do Netbeans para realizar a execução do protótipo mais facilmente, visto que esta IDE possui as ferramentas necessárias e foi a IDE utilizada para desenvolver esse protótipo.

Será necessário instalar o servidor Apache Tomcat para executar o trânsito cliente-servidor, além de um servidor para a comunicação dos programas em Java com o banco de dados MySQL. É recomendável o Wampserver, pois foi o utilizado durante o desenvolvimento.

Também será necessário instalar um banco de dados MySQL a partir da versão 8. O driver JDBC já está incluso no projeto.

Se der preferência por executar com o Netbeans, será necessário instalar uma extensão no navegador de conexão com o Netbeans, se preferir não utilizar o navegador padrão da IDE.

De preferência, use um ambiente Windows 10.

**IMPORTANTE:** Talvez a porta de conexão que o MySQL estiver usando não seja 3307. Para tal, verifique a mesma. Caso for diferente, vá ao código fonte da classe MySQLConnection, localizada em java\br\com\DBConnection\ no diretório src do projeto. No código, vá à linha 15 e mude a string que é inserida na variável “url” substituindo a porta para a utilizada pelo seu banco.