

Gerenciamento de Usuário

A implementação deve ser disponibilizado via github ou bitbucket.

- O desafio foi disponibilizado pelo GitHub através da conta brenodangelo e repositório desafio

Deve utilizar a linguagem Java para o backend e para o frontend tenha total liberdade para usar o framework que mais se sentir confortável.

- A aplicação foi desenvolvida em Java e o framework para frontend escolhido foi o PrimeFaces 8.

De preferência utilizar uma arquitetura MVC para organizar o front e o back.

- Foi usado o padrão JavaBean + DAO.

Disponibilize um arquivo que contenha instruções de como executar a aplicação.

- As instruções foram disponibilizadas no arquivo README do GitHub.

É necessário persistir os dados (gravar em banco de dados ou em arquivo), podendo ou não utilizar um framework (Hibernate por exemplo).

- Os dados são persistidos em um banco de dados PostgreSQL através do Hibernate. É possível também a configuração de outros bancos de dados suportados pela framework.

Implementar um sistema de gerenciamento de usuários, com seus respectivos perfis e os seus cargos desempenhados dentro da empresa.

- O sistema foi implementado para permitir o cadastramento dos itens e a relação entre eles através de uma interface WEB.
- A interface WEB segue um layout de cabeçalho com menu horizontal + conteúdo + rodapé e apesar de não estar descrito nos requisitos, abre uma sessão para o operador, já que há a possibilidade (e grande probabilidade) de uma tela de login ou integração com autenticação de terceiros.

Requisitos:

1. O sistema deve ter um gerenciamento dos cargos, cadastrando apenas o nome que é obrigatório e único de cada cargo. Devem haver as opções de editar um cargo, listar os cargos em ordem alfabética (opcional).
2. O sistema deve ter um gerenciamento de perfis de usuário, cadastrando apenas o nome que é obrigatório e único de cada perfil. Devem conter as opções de editar um perfil, listar os perfis em ordem alfabética e remover um perfil caso o mesmo não tenha nenhum usuário vinculado.
3. O sistema deve cadastrar os usuários com os seguintes campos e validações: nome (obrigatório), cpf (obrigatório único), dataNascimento, sexo (M ou F), cargo (obrigatório) e uma lista de perfis de usuário. Ao salvar o sistema deve armazenar também a data de cadastro do usuário. Deve haver a opção de editar um usuário seguindo as mesmas regras de validação.
4. O usuário deve compartilhar atributos de pessoas, ou seja deve implementar uma classe pessoa que seja abstrata, e contém os seguintes atributos: nome, cpf, dataNascimento e sexo.

- Foram feitas telas separadas para cadastrar, listar e editar os itens, onde o cadastro e a listagem são acessíveis pelo menu e a edição e exclusão pela listagem.
- Dentre as opções de ordenação, a escolhida foi o próprio navegador através do componente DataTable, configurado para ordenar os campos escolhidos e assim transferir os custos da ordenação para o computador do usuário. Ainda sim, todas as entidades implementam a interface Comparable para ordenação via backend.
- As validações dos campos foram escritas no próprio objeto, permitindo assim a validação através do Bean durante as ações (ou mesmo dinamicamente por Ajax) da tela.
- Os aplicativos usados para desenvolvimento e execução da aplicação foram:
 - IDE Eclipse 4.19.0
 - Apache TomCat 9
 - PostgreSQL 13
 - Framework Hibernate 3
 - Framework Primefaces 8 (JSF 2.2)
- Como não havia sugestão ou requisito descrito quanto ao gerenciamento de dependências, todos os arquivos JAR necessários para executar a aplicação estão dentro do próprio projeto.