Projeto GAMA

Cadastro de Usuários

Desafio a Vaga de Tech Lead

Edital 2021/25

André Luiz Braga, D.Sc

# Definição

A requisição do cliente consiste na criação de um sistema de cadastro de usuários e seus respectivos perfis e cargos. Os cargos e os perfis devem poder ser gerenciados com as funcionalidades de edição e listagem. Os usuários devem poder ser cadastrados ou editados, armazenando também a data de registro do cadastro. Os usuários devem ter obrigatoriamente Nome, CPF e cargo e opcionalmente, Data de Nascimento e Sexo.

# Escopo

Este projeto irá assumir as seguintes premissas, por razão de tempo para o desenvolvimento do protótipo:

* A interface com o usuário será simples, assim como o banco de dados, desta forma, utilizando apenas tipos de dados básicos de texto e numéricos. Entradas de dados do tipo data, por exemplo, poderão usar componentes sofisticados de entrada, como pequenos calendários em uma versão posterior
* Da mesma forma, CPFs não serão checados por validade oficial
* Utilizaremos o banco de dados JavaDB incorporado ao JDK e a ferramenta NetBeans, contudo, a utilização de um padrão de Objetos de Acesso de Dados (DAO) consistente provê o necessário isolamento e facilidade de incorporação de outros bancos, como o MySQL, em próximas versões.
* A interface com o usuáro, ou o “Front-End” (View do Modelo MVC) será feita em JAVA Swing em uma aplicação DESKTOP, uma vez que a construção de uma aplicação web requer mais tempo e requisitos de configuração e infraestrutura do cliente, o qual nesse ponto são inclusive desconhecidas.
* As funções de “Alteração” de Cargo e Perfil não estarão ativas, dado que tais entidades não possuem atributos a serem alterados, contudo, as interfaces com o usuário e o esqueleto principal estarão disponíveis para tai extensão rápida se necessário.
* Apesar de não requerido, o Usuário, Cargo e Perfil terão funções de “Listar” disponíveis para facilitar o processo de edição
* Funções de busca não foram requeridas, portanto não serão expostas, a não ser que necessárias internamente.
* Visando possibilitar a incorporação de diversos front-ends em versões futuras, o sistema terá um componente de interface entre o front end, ou seja, as interfaces com o usuário do modo Desktop em Swing, e o backend construído puramente em JAVA por requisição do cliente. Em futuras versões, como por exemplo, uma versão WEB JEE, esta camada poderá ser substituída por servlets ou frameworks como Struts, JSF, etc
* Decidimos utilizar a persistência através de puro JDBC e suas APIs JTA e JPA, dado que, para um banco de dados tão pequeno e muito poucas chances de modificação, os processos de configuração de frameworks de ORM poderia tomar mais tempo do que a implementação direta. Consideramos que a utilização de um padrão de Objetos de Acesso de Dados consistente (DAOs) já atende a uma boa estrutura de acesso ao banco e permite fácil incorporação de outros frameworks de persistência
* A Classe “Pessoa”, requisitada pelo usuário como sendo classe abstrata não vai ser considerada entidade do modelo de dados, uma vez que decidimos usar o modelo relacional puro, que não contempla herança, e não o modelo Objeto-Relacional, desta forma, permitindo que o sistema possa ser implementado em bancos de dados mais simples que não suportem o modelo Objeto-Relacional.

# Arquitetura

## Modelo de Componentes



## Modelo de Classes

### Controlador Principal

### Modelo - Classes de Dados e DAOs

### Objetos de Visualização

### Objetos de interface Front-BackEnd

# Interface com o Usuário

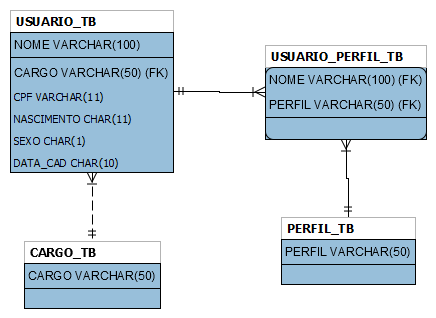
# Modelo de Dados

## Modelo Conceitual

O Sistema se constitui de um cadastro para usuários, os quais possuem alguns atributos básicos e um relacionamento com duas entidades simples, Cargo e Perfil. Tais entidades, que de outro modo seriam atributos do Usuário, foram destacadas como entidades, dado a sua necessidade de funções de edição e provavelmente, em futuras versões, possuírem elas mesmas atributos, como por exemplo, salário do cargo. O sistema descreve a possibilidade de um usuário poder ter mais de um perfil, e desta forma, formando uma relação M:N a ser modelada com uma relação (tabela) além das entidades descritas:

* **Entidade Pessoa** – Requisitada dentro da modelagem de classes como uma entidade pai abstrata e de outro modo não sendo indicada como tal, nesse caso, indicamos como entidade se olharmos do ponto de vista de um modelo de dados Objeto-Relaciona. Contudo, dado a simplicidade do modelo e a premissa de um banco de dados que não suporte o modelo Objeto-Relacional, não iremos levar essa entidade ao modelo lógico
* **Entidade Usuário** – Uma herança da entidade pessoa e irá conter os campos obrigatórios de Nome, CPF e cargo. Sendo este último, um relacionamento com uma outra entidade, uma vez que através de requisitos do usuário, ele poderá vir a ter mais atributos.
* **Entidade Cargo** – Apesar de ter apenas um atributo, sua própria chave, deve permitir edição, sendo, portanto, uma entidade a parte e relacionada de 1:N com os usuários (todo usuário deve ter um cargo e mais de um usuário pode ter o mesmo cargo)
* **Entidade Perfil** – Um usuário pode ter mais de um perfil e tal perfil pode ser atribuído a mais de um usuário, desta forma, teremos uma relação M:N, o qual vai levar a criação de uma tabela intermediária.

## Modelo Lógico/Fisico



# Ferramentas

* Netbeans
* JavaDB
* Git / GitHub

# Operação e Recursos