

Sistema de Apoio Para Localização de Pessoas Desaparecidas

BAN-II

Professor: **Jeferson Souza**

Alunos: **André Eduardo Pacheco Dias**

Bruno Werner

Wilton Jaciel Loch

1. Introdução

Este relatório tem como objetivo descrever a modelagem e implementação construída para o trabalho final da disciplina de Banco de Dados 2, que consiste em um sistema integrado com um banco de dados para auxiliar na localização de pessoas desaparecidas. Será apresentada e justificada a escolha da modelagem realizada, assim como os detalhes da implementação.

2. Visão Geral

O sistema foi modelado com o sistema gerenciador de banco de dados Postgresql. A implementação foi realizada com a linguagem de programação Java, utilizando o padrão de fachada e objeto de acesso a memória (DAO).

3. Modelagem dos dados

Existem seis unidades no sistema: Usuarios, Usuario_Tipos, Pessoas_Desaparecidas, Apelidos, Localizacoes e Denuncias. Todas as entidades possuem como chave primária uma coluna id, do tipo serial ou inteiro.

A entidade Usuarios tem como função armazenar os dados de um usuário que pode acessar o sistema. Guarda os dados de login e senha - utilizados para realizar a conexão no SALPD - nome, e tipo - papel que um usuário pode assumir no sistema.

Usuario_Tipos são os papéis que um usuário pode assumir. Essa entidade possui um nome e uma descrição. Existem cinco papéis pré-definidos que não podem ser alterados - a

aplicação foi construída com base no id de cada papel definido na população inicial do banco de dados (esta é a única entidade que tem como tipo de identificar um inteiro, invés de serial, uma vez que esse id não pode ser alterado) . Os papéis são: Administrador, Gestor, Agente, Informante e Anônimo. Mais detalhes das permissões de cada papel serão apresentados mais à frente.

As pessoas desaparecidas são armazenadas na entidade Pessoas_Desaparecidas, que registra informações da pessoa desaparecida como nome, RG, CPF, assim como a referência para o id usuário que inseriu esta pessoa no sistema e para o usuário quem fez a última atualização nesta mesma pessoa e referência para o id da última localização em que essa pessoa foi encontrada.

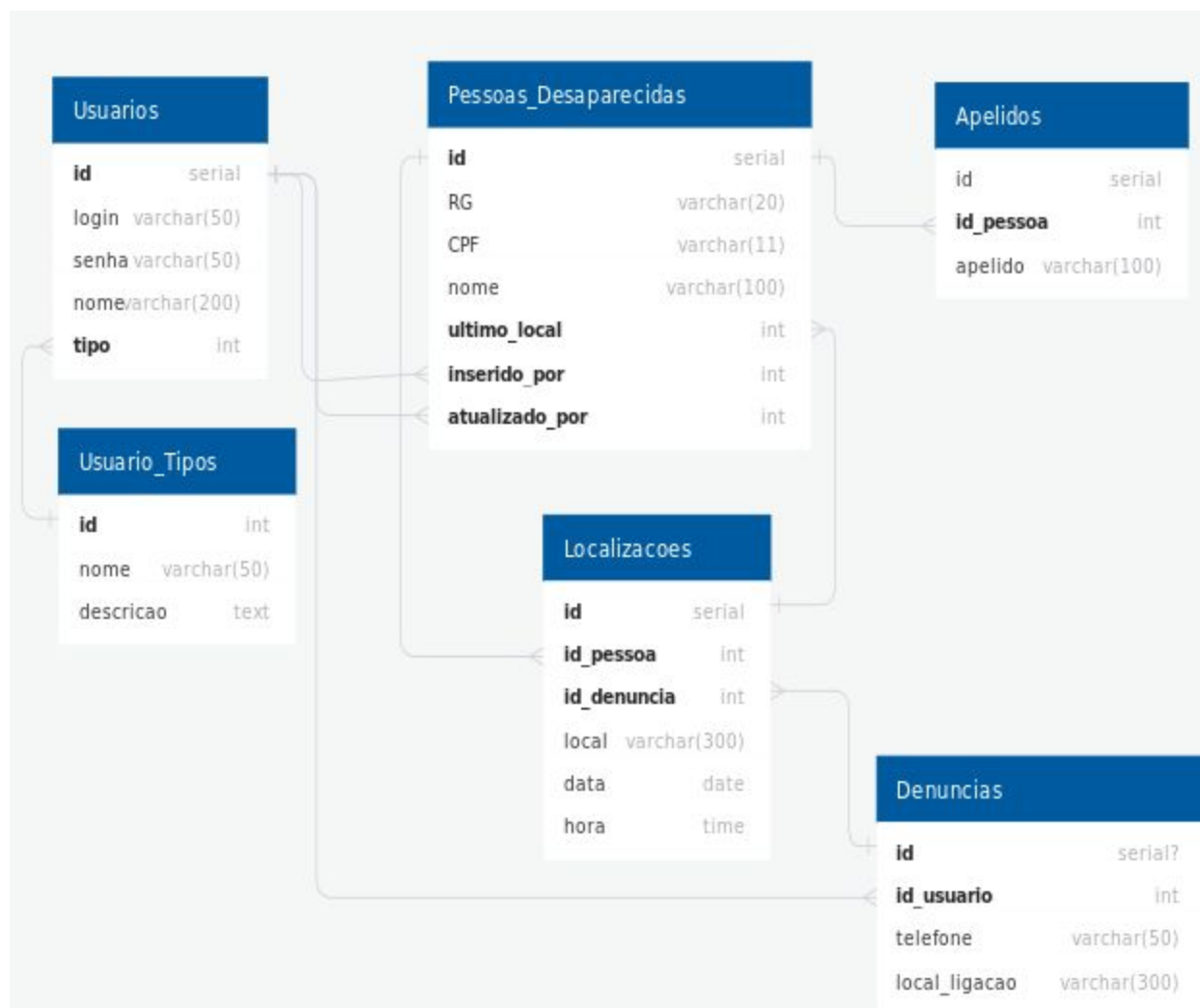
A entidade Denuncias possui a referência para o identificador do usuário que realizou a denúncia de uma pessoa desaparecida, o telefone utilizado para registrar a denúncia e a localização do denunciante obtida com base no telefone utilizado (o rastreo de uma localização não foi implementado).

Por fim, a entidade Localizacoes guarda a referência para o id da pessoa desaparecida existente na base de dados, outra para o id da denúncia que guarda a informação do usuário denunciante. Além disso, guarda o local, a data e a hora em que a pessoa desaparecida foi avistada.

Os detalhes das relações entre as entidades e os tipos dos atributos são descritos no diagrama abaixo.

Cardinalidade das relações entre as entidades:

Usuario	1 .. 1	Usuario_Tipos
Usuario	1 .. n	Pessoas_Desaparecidas
Usuario	1 .. n	Denuncias
Pessoas_Desaparecidas	1 .. n	Apelidos
Pessoas_Desaparecidas	1 .. n	Localizacoes
Localizacoes	1 .. 1	Denuncias



4. Implementação

A implementação é composta por sete pacotes: conexao, app, modelo, dao, negocio, util, visoes.

O pacote **conexao** possui apenas a classe Conexao, responsável por realizar a conexão com a base de dados para executar as consultas e transações. Toda consulta e transação realiza uma conexão nova com o banco de dados, e a encerra após o término ou falha. Nesta classe existem quatro atributos chaves para realizar uma conexão: DRIVER, URL, USUARIO e SENHA. O atributo USUÁRIO e SENHA são referentes ao nome do usuário e da senha utilizados para efetuar login no SGBD escolhido. DRIVER refere-se à localização do pacote do

driver compatível com a versão do Java e do SGBD utilizado. URL corresponde à url em qual a conexão será estabelecida, na implementação do sistema foi utilizado a seguinte url: *jdbc:postgresql://pellefant.db.elephantsql.com:5432/fhoodbzu*. A url tem o formato servidor:porta/usuario. O servidor utilizado foi o ElephantSQL, empresa que oferece serviços de banco de dados online em diversas linguagens diferentes. Como o código foi escrito de maneira genérica, caso seja necessário utilizar outra linguagem de SGBD ou hospedar o banco de dados em outro servidor, basta uma simples alteração nos atributos dessa classe e a adição de uma biblioteca compatível.

No pacote **modelo** existem as classes referentes às tabelas que armazenam os dados. Instâncias das classes deste pacote são utilizadas para carregar dados do servidor da base de dados para memória principal do hardware da aplicação ou atualizá-los.

Em **util** temos duas classes que são instanciadas múltiplas vezes em diversos pacotes, Operacao e Toolbox. No Toolbox existem métodos utilizados na implementação da interface do usuário com o sistema, como *limpaTela()* e *aguarda()*. A classe Operacao é responsável por todas as transações e consultas no sistema. Possui o atributo sucesso para indicar o sucesso da operação, mensagem para exibir as possíveis falhas para o usuário e dado para retornar informações necessárias em certas operações, como o retorno de uma consulta ou o id de uma inserção.

App é pacote que efetua a comunicação do usuário com o sistema: lê os comandos de entrada e exibe as mensagens de saída. Internamente, o pacote app tem acesso apenas ao pacote **Negocio**, que possui a classe NegocioFacade faz o intermédio entre a interface com o usuário e o acesso à base de dados e aplica as regras de negócio, como verificação da validade das letras de um nome ou dos números de um cpf. Para executar finalmente as transações na base de dados, o pacote DAO (data access object) realiza essa função: suas classes recebem o objeto a ser inserido, prepara a declaração na linguagem SQL, estabelece uma conexão com o SGBD e executa a transação ou consulta. Vale notar que para cada ação do sistema, uma instância da classe Operacao é realizada, e a referência para esse objeto instanciado é carregada do pacote App ao pacote DAO, armazenando as mensagens de possíveis falha ou o sucesso da ação realizada.

5. Permissões

As permissões do sistema foram implementadas apenas no lado da aplicação. Para cada tipo de usuário existe um menu associado à ele, com as opções que ele pode escolher. Algumas funções verificam o tipo de usuário que está logado no sistema e se adequa de acordo à ele, como a remoção de um outro usuário, que precisa ser de um tipo inferior.

Um usuário com papel administrador possui acesso irrestrito ao sistema. Um gestor pode cadastrar, listar e remover usuários com papel de agente e informante; cadastrar, listar e remover pessoas desaparecidas; cadastrar, listar e remover denúncias. Um agente têm as mesmas permissões, com a diferença que gerencia apenas usuários do tipo informante. Um informante pode realizar denúncias e listar as pessoas desaparecidas, assim como um usuário anônimo, porém suas informações não ficam salvas na denúncia.

6. Considerações finais

A modelagem e implementação do sistema, ao nosso ver, atendeu os requisitos exigidos na descrição do projeto. Como a aplicação é executada no console do sistema operacional, sem auxílio de uma interface gráfica, algumas funções do sistema não ficaram claras ou eficientes (na visão de um possível usuário real utilizando o sistema) o suficiente. Por exemplo, para remover um usuário ou denúncia, é necessário saber o id da tupla a ser removida e, para descobri-lo, é preciso listar todas as tuplas da entidade desejada que, por sua vez, não é tão legível quanto poderia ser. Entretanto, o sistema é funcional: com exceção da aquisição da localização de um denunciante, que precisaria ser feito uma análise mais detalhada para uma implementação funcional, o resultado final pode ser utilizado na prática, já que existe um controle de usuários, pessoas desaparecidas e denúncias assim como requisitado.