

Instituto de Ciências Matemáticas e de Computação
Departamento de Sistemas de Computação
Universidade de São Paulo
Av. Trabalhador São-carlense, 400, Caixa Postal 668 - CEP 13560-970 - São Carlos/SP
Fone (16) 3373-9700

SSC0103-1 - Programação Orientada a Objetos

RELATÓRIO DE CONSTRUÇÃO DO APLICATIVO DESKTOP PARA ADOÇÃO DE ANIMAIS - ADOPET

Alunos:

Mateus Prado Santos - 10851707
Matheus Lopes Rigato - 10260462
Matheus Tomieiro de Oliveira - 10734630
Victor Vieira Custodio Reis - 10734686

Professor: Dr. Márcio Delamaro
PAE: Claudinei Jr e Misael Jr

MAIO/2019

Resumo

Durante a construção do aplicativo, foram tomadas decisões importantes acerca de facilidade e comodidade para o usuário se relacionar com o programa. O resultado foi um programa esteticamente amigável, com praticidade de uso e com funções essenciais na hora de adotar um pet, tais como informações claras do anunciante, telas simples para escolha de pets, e principalmente, um chat para comunicação entre anunciante e usuário.

1.0) Aplicação

Área: Causa animal.

Projeto: Implementação de aplicativo com plataforma de auxílio na doação de animais.

2.0) Decisões iniciais e Métodos

Desde o princípio, decidiu-se que o projeto seria Desktop pela maior segurança dos membros da equipe em desenvolver para a plataforma utilizando a linguagem Java. Além disso, o grupo seria dividido por determinadas funções no projeto onde seu desenvolvimento coletivo seria integrado e supervisionado utilizando-se a ferramenta Github.

A divisão de funções ficou da seguinte forma:

- Um integrante do grupo ficou responsável pelo gerenciamento do banco de dados, onde a ferramenta escolhida pelo membro foi o MySQL por uma pequena experiência já adquirida.
- Outro integrante ficou responsável por gerenciar a frontend utilizando JavaFX pela vontade de aprender a utilizar a ferramenta que tem a premissa de facilitar a confecção de telas.
- Os outros dois integrantes da equipe ficaram responsáveis por fazer as classes de backend e, posteriormente, entrar no desenvolvimento da frontend em conjunto com o outro membro.

É importante ressaltar, porém, que essa divisão de grupos foi se desfazendo logo após o meio do projeto onde foi decidido pelo grupo que todos os membros deveriam desenvolver de modo fullstack ajudando uns aos outros. Assim, ao final do trabalho, cada integrante adquiriu conhecimento em cada uma das partes do programa.

3.0) Descrição e resultados

Como resultado das considerações iniciais, obtemos no projeto grande versatilidade devido ao uso do JavaFX e do MySQL. Com essas ferramentas foi possível criar um design agradável aos usuários, no caso do FX, como o armazenamento de imagens, mensagens, dados pessoais, e interações entre usuários com imprescindível uso do MySQL. Fora a diferença gritante na facilidade de desenvolver um design mais complexo, como o chat por exemplo, que foi todo construído do zero usando a FX e MySQL.

O projeto foi desenvolvido, de forma sucinta, da seguinte maneira:

Fora da aplicação em Java um banco de dados local, feito em MySQL, quem possui 4 Tables que guardam e retornam os dados recebidos do programa; são elas: clientes, pets, chat e mensagens. Especificamente nas três primeiras temos um id único que diferencia cada row como única, muito útil na manutenção de usuários, pets e chats parecidos mas não iguais.

Primeiramente, há classes indispensáveis para uso na classe principal; são elas: Pet, Usuário e Pair, sendo as duas primeiras, classes que instanciam um animal e um usuário, com seus respectivos atributos, e a última uma classe que implementa um tipo Genérico que organiza um Integer e uma String em um par.

Juntamente com elas, há uma classe chamada BDConexaoClass que controla todas as interações com o banco de dados MySQL. Nela, são criadas diversas Queries que procuram, modificam, inserem e removem dados do banco de dados. Sem ela, o controle dessa forma de armazenamento de dados seria caótica se feita por qualquer uma das classes superiores.

Há, por fim, uma classe chamada Gui que controla todas as outras classes e estende a classe Application(da JavaFX); Ela, chama todos os métodos, instancia objetos das classes citadas e controla basicamente todo o programa. Nela são instanciados os elementos visuais da interface gráfica, e seu controle nas alterações que fazem nos objetos de controle instanciados e no banco de dados.

De forma geral, pode-se observar ao entrar no programa um tela de login que busca o registro de um usuário no banco de dados e autoriza seu acesso. Há uma tela de cadastro para novos usuários. Já dentro do hall, há vários animais instanciados na tela na forma de um Vector<Pet>. Ao avançar as páginas a Gui solicita outro Vector para o banco de dados, sempre mantendo o mínimo de memória de objetos instanciados em uso. Pode se clicar nos pets e adquirir informações dele e de seu dono. A partir de qualquer pet pode ser clicado (contanto que não sejam os próprios anunciados pelo mesmo usuário) um botão de chat, que inicia uma conversa entre o anunciante e o interessado. Pode-se navegar pelas abas usando RadioButtons presentes na parte superior da tela de animais disponíveis, chat e uma terceira tela chamada Por que adotar. Essa ultima mostra informações e dados sobre o abandono de animais e motiva o usuário sobre a importância da adoção para contribuir no combate ao problema de abandono de animais.

5.0) Funcionalidades desenvolvidas

Temos como funcionalidades desenvolvidas:

- a) Sistema de cadastro de usuários para possibilitar a adoção, assim como o anúncio de animais;
- b) Visualização de fotos e informações sobre os animais disponíveis para adoção;
- c) Chat que possibilita contato entre o usuário que deseja a adoção e o usuário que a promove, bem como o contato com ONGs;
- d) Janela informativa do porquê a adoção é importante.

5.1) Funcionalidades não desenvolvidas

A única funcionalidade proposta no relatório inicial que não foi desenvolvida foi:

- a) Caixa de pesquisa com uso de filtros para melhor se adequar às possibilidades de cada usuário.

Isto aconteceu tanto por falta de tempo quanto por uma dificuldade em utilizar processos mais complexos em banco de dados onde nenhum dos membros possui conhecimentos avançados.