Programa: Controle de tickets de usuários
Autor: Guilherme Pedroso Matheos RA 22.117.020-2

Considerações importantes

- IDE utilizada: Netbeans with Java JDK 8.2.

- **importante** Antes de executar o programa pela primeira vez, verificar se o arquivo **dados.sav** está na **pasta principal** junto com o executável **.jar**. (caso o arquivo não seja encontrado o programa exibe uma mensagem “erro de i/o”).

Obs. para executar o programa dentro da IDE, o arquivo *dados.sav* deve estar na pasta /ProjetoLab.

- A página do administrador está em branco (não deu tempo de terminar..), o resto está funcionando.

Dados existentes no banco de dados para realizar login:

- Nome
- RA (numérico)
- Senha

Usuário
Guilherme
111
1234

Pedro
222
1234

Walter
444
1234

Super Usuário (suporte)
Lucas
444
1234

Administrador
Admin
0
qwerty

Funcionamento do programa

Página de login

- Na página de login, o funcionário insere suas informações de login - nome, RA (apenas numérico) e senha.

- O programa verifica se quem fez login é usuário, superusuário ou administrador e exibe a página correspondente.

Página do usuário

É possível realizar 3 operações:

- Criar ticket: caso nenhum ticket tenha sido aberto, o usuário pode abrir um ticket (não nulo).

- Ler ticket aberto: é possível ler, editar ou deletar (fechar) um ticket aberto.

- Ler resposta: aqui o usuário pode ler a resposta do suporte (superusuário), quando este responder seu ticket.

Página do super usuário

- O superusuário ou atendente de suporte, pode ver uma lista com todos os usuários que possuem ticket aberto.

- É possível selecionar um usuário com ticket aberto e enviar uma resposta contendo a solução do problema que o usuário enfrenta.

- O superusuário pode ver ou editar uma resposta de ticket.

Página do administrador

- É possível cadastrar, editar ou deletar usuários e superusuários.

Botão sair e salvar

- Ao fechar o programa com esse botão, tudo é salvo no arquivo *dados.sav*, para ser lido quando o programa iniciar novamente.

- Assim é possível fazer login como usuário, criar um ticket, sair do programa, abrir o programa, fazer login como superusuário e responder esse ticket.

Explicação das pastas (package)

Controller: contém o ArrayList dos Funcionários e a programação dos eventos de cada elemento da interface (botões, caixas de texto etc).

Model: define o Funcionário e as classes que herdam de funcionário.

View: contém a própria interface gráfica e o método main()

Conceitos de orientação a objetos e explicação das classes

Classe *Funcionário*

- É uma classe abstrata que define um modelo para as classes que herdam de *Funcionário* - *Usuário*, *SuperUsuário* e *Administrador* - polimorfismo.

- Contém o método abstrato *getCard()* que será sobrescrito (override) pelo mesmo método nas classes que herdam de Funcionário. Esse método retorna uma string que é utilizada para mostrar a tela correta dependendo do tipo de Funcionário que fez login.

Classe *Dados*

- Contém um ArrayList estático que guarda todos os funcionários, uma variável index que guarda o índice do funcionário que fez login, e o método *setDados()* que preenche o ArrayList para criação do arquivo *dados.sav* pela primeira vez.

- Os métodos *save()* e *read()* são utilizados para salvar o ArrayList em um arquivo *dados.sav* quando o usuário sair do programa pelo botão *Salvar e Sair*, e para ler os dados do arquivo e guardar no ArrayList ao iniciar o programa.

Classes *Controller*, *LoginController*, *UserController*, *SuperUserController*, *AdminController*

- Herdam da classe *Dados*, para terem acesso ao ArrayList dos funcionários.

- Contém a programação dos eventos de cada elemento da interface (botão de login, botão de enviar, caixas de texto etc).

- Para acessar funções específicas de cada tipo de funcionário utiliza-se downcasting (exemplo função deleta() na classe UserController).