

Plano de teste

App aula0902

ÍNDICE

1. Introdução
 - 1.1. Propósito do sistema
 - 1.2. Referências/documentações do projeto
2. Abordagem de testes
 - 2.1. Categorização dos requisitos funcionais e não funcionais
3. Casos de teste

1. INTRODUÇÃO

Software produzido ao decorrer das aulas de programação de aplicativos.

Consiste em um cadastro manual e login entre usuário e administrador (sem integração com algum banco de dados).

O sistema permitirá que o usuário faça um cadastro e em seguida podendo logar-se como usuário comum ou admin tendo este último, a possibilidade de pesquisar usuários, tal como um CRUD(Create, Read, Update and Delete).

1.2 Referências/Documentações do projeto

A tabela abaixo identifica a elaboração deste plano de teste.

Documento	Irá fazer parte do projeto?	Observações
Especificação de requisitos	Sim	Define as funcionalidades que o sistema deve possuir
Especificação de casos de uso	Sim	Descreve detalhadamente as funcionalidades do sistema
Caso de teste	Sim	Descreve as condições de entradas e saídas do sistema
Especificação do caso de teste	Sim	Define o conjunto de casos de testes a serem testados

2. Abordagem de testes

2.1 Categorização dos requisitos funcionais e não funcionais

Requisito funcional	Requisito não funcional
O sistema deve permitir o cadastro de pessoas físicas.	O sistema deve ser desenvolvido na linguagem de programação java.
O sistema deve permitir o login referente ao usuário cadastrado.	O sistema não utiliza banco de dados.
O sistema deve permitir o cadastro/login de um usuário comum.	O sistema deve ser mobile.
O sistema deve permitir o cadastro/login de um usuário administrador.	O sistema deve ter um design responsivo.
O sistema deve redirecionar o usuário comum para a tela comum.	O sistema deve ser desenvolvido na plataforma Android Studio.
O sistema deve redirecionar o usuário administrador para a tela admin.	O sistema deve possuir níveis de acesso para diferenciar usuários comuns de usuários administrativos, que acessarão áreas restritas do sistema.
Caso o usuário não esteja cadastrado, o sistema deve direcioná-lo para a tela de cadastro.	
Caso o usuário já esteja cadastrado, o sistema deve direcioná-lo para a tela de login.	
O sistema deve permitir ao usuário administrador que o mesmo pesquise os usuários cadastrados.	

2.2 Detalhamento da abordagem de teste

Tipo de teste:	Funcional
Onde:	Classe Main, método clicar()
Objetivo do teste:	Ao clicar no botão, um toast com a mensagem de bem-vindo deverá aparecer

Tipo de teste:	Funcional
Onde:	Classe Main, método login()
Objetivo do teste:	Ao clicar no botão login, o usuário deverá inserir o nome e senha. Caso o usuário for comum, o sistema deve direcioná-lo à tela comum e, caso o usuário for administrador o sistema deve direcioná-lo à tela admin. Caso o usuário esteja cadastrado a mensagem de bem-vindo deve aparecer caso o contrário, a mensagem "login ou senha incorreta" deverá aparecer.

Tipo de teste:	Funcional
Onde:	Classe Main, método criarUsuarios()
Objetivo do teste:	O método criarUsuarios() deve criar os "users test" e em seguida adicioná-los à arrayList.

Tipo de teste:	Funcional
Onde:	Classe Main, método cadastro

Objetivo do teste:	O método cadastro() deve chamar a intent do cadastro e iniciá-la, em seguida, a mesma deve receber a lista de users criados anteriormente na main.
---------------------------	--

Tipo de teste:	Funcional
Onde:	Classe Cadastro, método cadastrar()
Objetivo do teste:	O método cadastrar() deve realizar .getText() do novo login, senha e confirmação de senha e em seguida, iniciar uma condição de que, se a senha for igual a confirmação, deve adicionar o usuário à ArrayList e caso o contrário, deve aparecer o toast de que “as senhas tem que ser iguais”.

Tipo de teste:	Funcional
Onde:	Classe Usuários, métodos construtores
Objetivo do teste:	Deve criar os métodos construtores para o objeto Usuário.

Tipo de teste:	Funcional
Onde:	Classe Comum
Objetivo do teste:	Deve direcionar o usuário comum para a activity comum.

Tipo de teste:	Funcional
Onde:	Classe admin
Objetivo do teste:	Deve iniciar a activity admin.

Tipo de teste:	Funcional
Onde:	Classe admin
Objetivo do teste:	Deve remover a barra roxa.

Tipo de teste:	Funcional
Onde:	Classe admin, método buscar()
Objetivo do teste:	Deve disponibilizar o OnClick do login e resultado com o findViewById();

Tipo de teste:	Funcional
Onde:	Classe admin, método buscar()
Objetivo do teste:	Deve criar a String com a mensagem: "login "+login+" não encontrado";

Tipo de teste:	Funcional
Onde:	Classe admin, método buscar()

Objetivo do teste:	<p>Deve conter um for para a listaUsuarios que, se o login for .equals ao login já existente a mensagem: "login"+"encontrado com sucesso!" deve aparecer caso o contrário, um .SetText da mensagem de "login "+login+" não encontrado" deverá aparecer</p>
---------------------------	--

3. Casos de teste

Ação	Resultado esperado
Clicar() String mensagem armazena o campo.getText() com a mensagem "olá, "+campo.getText()+" seja bem-vindo".	Toast com a String mensagem;

Ação	Resultado esperado
Logar() Armazena o login e senha	Se o usuário e senha forem admin, direciona para a tela admin. Se o usuário e senha forem comum, direciona para a tela comum e os envia a mensagem "bem-vindo". Se o usuário e senha estiverem incorretos, a mensagem: "Login ou senha incorreta" aparece.

Ação	Resultado esperado
criarUsuarios()	Cria os "user test" e adiciona no ArrayList

Ação	Resultado esperado
cadastro() chama a intent do cadastro e a inicia, depois recebe a lista de users da criada anteriormente na main.	Inicia a activity cadastro.

Ação	Resultado esperado
cadastrar() Cria as Strings login, senha e conf.	se a senha for igual a conf cria o usuário e adiciona na lista se não

	aparece, o toast "as senhas tem que ser iguais" aparece.
--	--

Ação	Resultado esperado
buscar() busca o login com o .getText() for pra listaUsuarios(se o login for .equals ao login) resultado.setText(mensagem)	Mensagem: "login"+"encontrado com sucesso!". Se o login não for .equals: "login "+login+" não encontrado";