SENAI CIMATEC - CIMATEC - CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

GABRIEL BARRETO FERRAZ FRAGA E

MARCIO DOS SANTOS DE SOUZA

DESCRIÇÃO DO BACK END DO PROJETO ANDROID STUDIO

SALVADOR - BA

2024

GABRIEL BARRETO E

MARCIO DOS SANTOS DE SOUZA

DESCRIÇÃO DO BACK END DO PROJETO ANDROID STUDIO

Projeto apresentado como requisito

parcial para obtenção do grau de técnico em

Desenvolvimento de sistemas, pelo

Senai Cimatec.

Orientador: Prof.Leandro Santos da Cruz.

SALVADOR - BA

2024

DESCRIÇÃO DA CLASSE “ConnectionDb”:

Essa classe é utilizada especificamente para conectar a aplicação com o banco de dados firebase, ela conta com:

- **A linha ‘private static firebaseAuth auth’:** esta linha do código é responsável por declarar uma variável estática ‘auth’ do tipo FirebaseAuth. A palavra static significa que a variável pertence a classe connetcionDb, ao invés de pertencer a uma instância específica da classe.

- **Método Fireautenticacao:** o método Fireautenricacao() é um método estático que retorna uma instância do FirebaseAuth. Ele é acessado diretamente na classe connetcionDb sem a necessidade de criar uma instância da classe.

- Verificação: dentro do método Fireautenricacao() é criada uma verificação para saber se o auth é nulo, utilizando o if(auth == null). Isso é feito para garantir que seja criada apenas uma instância do FirebaseAuth durante toda a execução do programa. Se a variável auth for nula, entra na condição if e chama o método FirebaseAuth.getInstance() para obter uma nova instância do FirebaseAuth.

Por fim, o método retorna a instância do FirebaseAuth armazenada na variável auth.

DESCRIÇÃO DA CLASSE “MainActivity”:

- Método OnCreate: o método OnCreate() é chamado quando a atividade é criada, lá é definido o layout através do método `setContentView(R.layout.activity_main)`. Além disso, a principal função desse método, é inicializado a instância FirebaseAuth utilizando o método `Fireautenticacao()` da classe `ConnectionDb` e atribuído-a à variável `autenticacao`. Por último, o método initialize() é utilizado para inicializar os elementos da interface gráfica.

- Método initialize: o método initialize() é responsável por encontrar, através do id, as caixas de texto de login e senha e atribuí-las às variáveis editLogin e editPassword respectivamente.

- Método cadastrarConta: o método cadastrarConta(View view) é chamado quando o botão de cadastro é clicado, chamando o método trocar tela(2) e direcionando o usuário para tela de cadastro.

- Método entrar: o método entrar(View view) é chamado quando o usuário clicar no botão de Login. Ele obtém os textos das caixas de texto de login e senha, criando uma instância Usuario com esses valores e chamando o método 'signInWithEmailAndPassword' do objeto autenticacao para realizar o login utilizando o Firrebase Authentication. Se o login for bem sucedido, o usuário será direcionado para tela de produtos e, caso contrário, será emitido uma mensagem de erro.

- **Método recuperarConta:** o método recuperarConta(View view) é chamado quando o botão de recuperação de conta é clicado. Ele chama o método trocarTela(1) para trocar para a tela de recuperação de conta.

- Método trocarTela: o método trocarTela(Integer whichScreen) é responsável por trocar de tela com base no parâmetro whichScreen, definindo através do switch qual tela será aberta.

DESCRIÇÃO DA CLASSE "Tela_Cadastro":

A classe Tela_Cadastro estende a classe AppCompatActivity e é responsável por cadastrar novos usuários no banco de dados, permitindo-os logar posteriormente no aplicativo. Está classe conta com:

- Objetos instanciados: os objetos editEmail, editPhone, editPassword e editCorfim são instâncias da classe EditText e representam os campos de entrada de email, telefone, senha e confirmação de senha.

- Objeto autenticacao: o objeto autenticacao é uma instância da classe FirebaseAuth e é inicializado com a instância obtida do método Fireautenricacao() da classe connetcionDb. Essa parte do código é criada para que possa ser possível utilizar o banco de dados para autenticação dos usuários

- Método onCreate: o método onCreate() é chamado quando a atividade é criada, com o objetivo de configurar o layout da tela de cadastro utilizando o método setContentView(). Neste mesmo método é inicializado o objeto autenticacao chamando o método Fireautenricacao da classe connetcionDb. Além disso, é chamado o método initialize() para inicializar os campos de entrada. O initialize() após ler os valores dos campos, é responsável por atribuir-as as variáveis correspondentes.

- Método cadastrar: o método cadastrar(View view) é chamado quando o usuário clica no botão de cadastro. Este método obtém os valores inseridos no campo de entrada, conferindo se está preenchido e se a senha e a confirmação de senha são iguais. Se todas as condições forem atendidas, um novo objeto Usuario é criado com os dados fornecidos. O método createUserWithEmailAndPassword() do objeto autenticacao é chamado para criar um novo Usuario no Firebase Authentication. Dependendo do resultado da criação, uma mensagem é exibida de êxito ou erro.

- Método voltarLogin: o método voltarLogin(View view) é chamado quando o botão de voltar para o login é clicado. Ele chama o método trocaTela() para iniciar uma nova atividade,que é a tela principal de login.

- Método trocarTela: o método trocaTela() cria uma nova instância da classe `Intent` para iniciar a atividade MainActivity, que é a tela principal de login. Em seguida, chama o método startActivity() para iniciar a nova atividade.

DESCRIÇÃO DA CLASSE “Usuario”:

A classe Usuario tem apenas como objetivo representar um usuário do sistema, armazenando suas informações de Email, telefone e senha. Essa classe conta com apenas a criação dos três atributos, os métodos getters e setters, o construtor e a criação do método toString().

DESCRIÇÃO DA CLASSE “Recuperar_Senha”:

A classe Recuperar_Senha estende a classe AppCompatActivity e é responsável por dar a opção ao usuário de recuperar senha. Esta classe conta com as seguintes funcionalidades**:**

- Objeto autenticacao: o objeto autenticacao é uma instância da classe FirebaseAuth e é inicializado com a instância obtida do método Fireautenticacao() da classe ConnectionDb. Isso permite que o usuário use o Firebase Authentication para autenticar usuários.

- Objeto editTextEmail: o objeto editTextEmail é uma instância de EditText que representa o campo de entrada do Email para recuperação de senha.

- Método onCreate: O método onCreate() é chamado quando a atividade é criada. Ele configura o layout da tela de recuperação de senha usando o método setContentView(). Em seguida, inicializa o objeto autenticacao chamando o método Fireautenticacao() da classe ConnectionDb e chama o método initialize() para inicializar o campo de entrada de email.

- Método initialize(): O método initialize() encontra o campo de entrada de email no layout usando seu ID e atribui-o à variável editTextEmail.

- Método recuperarSenha: O método recuperarSenha(View view) é chamado quando o usuário clica no botão de recuperação de senha. Ele obtém o valor inserido no campo de entrada de email e verifica se não está vazio. Se o email não estiver vazio, o método sendPasswordResetEmail() do objeto autenticacao é chamado para enviar um email de recuperação de senha para o endereço de email fornecido. Dependendo do resultado do envio do email, uma mensagem de sucesso ou erro é exibida.

- Método VoltarLogin: O método VoltarLogin(View view) é chamado quando o botão de voltar para o login é clicado. Ele chama o método trocaTela() para iniciar uma nova atividade, que é a tela principal de login.

- Método trocarTela: o método trocaTela() cria uma nova instância da classe Intent para iniciar a atividade MainActivity, que é a tela principal de login. Em seguida, chama o método startActivity() para iniciar a nova atividade.

DESCRIÇÃO DA CLASSE “RecycleViewAdapter:

A classe `RecycleViewAdapter` estende a classe `RecyclerView.Adapter` e é usada para fornecer os dados e o layout para um `RecyclerView` exibir uma lista de produtos. Dentre as partes mais importantes dessa classe podemos citar:

- Variáveis de instância: a classe possui duas variáveis de instância: produtoList, que é uma lista de objetos da classe Produto, e context, que representa o contexto da aplicação.

- Construtor RecycleViewAdapter: O construtor RecycleViewAdapter é usado para passar a lista de produtos e o contexto para o adaptador.

- Método onCreateViewHolder: O método onCreateViewHolder é chamado quando uma nova instância de MyViewHolder é criada. Ele infla o layout do item de produto a partir do arquivo de layout R.layout.item_products e retorna uma nova instância de MyViewHolder que contém as referências aos elementos de interface do item.

- Método onBindViewHolder: o método onBindViewHolder é chamado para preencher os dados do produto em um MyViewHolder específico. Ele obtém o produto na posição fornecida da lista de produtos e define o texto de txt_productDetails com a representação em string do produto. Além disso, ele usa a biblioteca Glide para carregar a imagem do produto da URL fornecida em iv_productImage.

- Método getItemCount: o método getItemCount retorna o número total de produtos na lista.

- Classe MyViewHolder: a classe interna MyViewHolder estende a classe RecyclerView.ViewHolder e é usada para representar cada item de produto na lista. Ela possui duas variáveis de instância: iv_productImage, que é um ImageView para exibir a imagem do produto, e txt_productDetails, que é um TextView para exibir os detalhes do produto.

- Construtor MyViewHolder: O construtor MyViewHolder é usado para inicializar as referências aos elementos de interface do usuário do item de produto.

DESCRIÇÃO DA CLASSE “Produto”:

- Atributos: a classe Produto representa um produto da loja online e possui atributos como nome, descrição, preço, quantidade e Id.

- Getters e Setters: a classe possui métodos getters e setters para acessar e modificar os atributos do produto.

- Método toString: a classe também possui um método toString que retorna uma representação em string das informações do produto.

- Comparadores estáticos: é definido comparadores estáticos para auxiliar na ordenação da lista de produtos. Os comparadores são usados para ordenar os produtos por nome em ordem alfabética de A-Z, de Z-A e por preço em ordem crescente e decrescente.

- Construtor: o construtor da classe Produto é usado para inicializar os atributos do produto quando um novo objeto é criado.

POSSÍVEIS MELHORIAS PARA O CÓDIGO:

- **Integração dos produtos com banco de dados;**
- **Incremento de animações para o front end;**
- **Sistema de transação de produtos;**