



UNICEP - CENTRO UNIVERSITÁRIO CENTRAL PAULISTA

DANILO ALVES DA SILVA
SISTEMAS DE INFORMAÇÃO

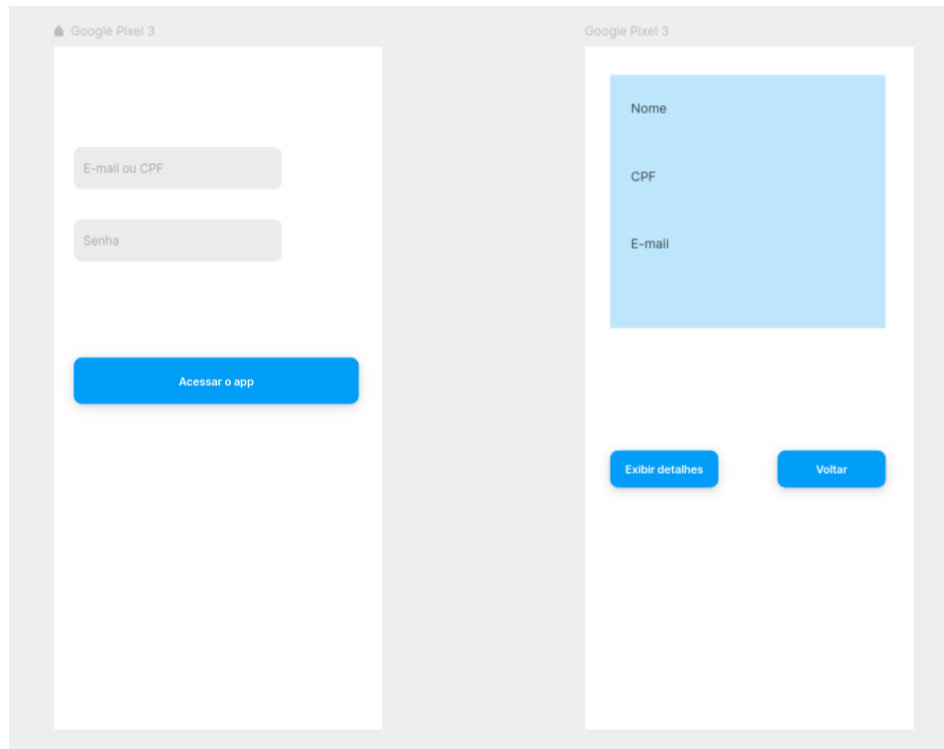
PROVA P1
CONSTRUÇÃO DE APP DE LOGIN

RIO CLARO

2020

1. Descrição

A avaliação consiste em um mini - projeto que deve apresentar um app de Login/Acesso ao sistema com as seguintes características:



- Desenvolver um app que tenha ao menos duas telas como fluxo principal - Login e exibição dos dados
- Utilizar os recursos vistos em aula - alinhamento via ConstraintLayout e carregamento de Activities
- Criar documentação em PDF com a descrição da atividade que foi realizada utilizando a ABNT.

2. Desenvolvimento

Foram criados dois layouts para as telas de Login, que é a principal e tela que ficara encarregada de carregar os dados:

The image displays two side-by-side wireframe layouts for mobile application screens. The left layout is a login screen featuring two input fields: 'E-mail ou CPF' and 'Senha', followed by a green button labeled 'ACESSAR APP'. The right layout is a registration or data entry screen featuring three input fields: 'Nome', 'E-mail', and 'CPF', followed by two green buttons labeled 'EXIBIR DADOS' and 'VOLTAR'. Both layouts are enclosed in a light gray border with dashed lines indicating the screen boundaries.

Obs: Tive problemas depois da primeira execução da aplicação, onde o emulador só apresentava uma tela preta, então não consegui testar todas funcionalidades, vou apresentar os protótipos das telas no modo de desenvolvimento e o código para chamada das telas.

Após o usuário inserir seu CPF ou E-mail e a senha de acesso, e executar a ação de clicar no botão “**Acessar App**” será feita uma validação desses dados, dentro de uma lista onde moquei alguns dados para que pudesse ser realizados alguns testes.

Criei uma classe chamada User e lá possui alguns atributos da classe e um método que retorna alguns usuários mocados.

Segue imagem abaixo:

```

public class User {
    public String Name;
    public String Email;
    public String Cpf;
    public String Password;

    User() {}

    User(String name, String email, String cpf, String password)
    {
        Name = name;
        Email = email;
        Cpf = cpf;
        Password = password;
    }

    public List<User> getUser(){
        List<User> users = new ArrayList<>();
        User danilo = new User( name: "Danilo", email: "dan.alves300@gmail.com", cpf: "25898713233", password: "nanadoludovigo01");
        User lais = new User( name: "Lais", email: "lais@uool.com.br", cpf: "12396478520", password: "lasanholia86");
        User peter = new User( name: "Peter", email: "parker_peter@outlook.com", cpf: "11977630425", password: "Miranha99");
        users.add(danilo);
        users.add(lais);
        users.add(peter);

        return users;
    }
}

```

```

public void login(View view){
    EditText editEmailOrCpf = findViewById(R.id.editTextTextEmailAddress);
    String cpfOrEmail = editEmailOrCpf.getText().toString();
    EditText editPassword = findViewById(R.id.editTextTextPassword);
    String password = editPassword.getText().toString();

    User user = new User();
    List<User> users = user.getUser();
    for (int i = 0; i < users.size(); i++){
        if((cpfOrEmail.equals(users.get(i).Cpf) || cpfOrEmail.equals(users.get(i).Email)) && password.equals(users.get(i).Password))
        {
            Intent intent = new Intent( packageContext: this, ShowDataActivity.class);
            startActivity(intent);
        }
        else{
            AlertDialog.Builder builder = new AlertDialog.Builder( context: this);
            builder.setTitle("Erro de login!");
            builder.setMessage("Dados de usuário inválidos!");
            builder.setPositiveButton( text: "OK", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface arg0, int arg1) {
                    Toast.makeText( context: MainActivity.this, text: "OK=" + arg1, Toast.LENGTH_SHORT).show();
                }
            });

            builder.create();
            builder.show();
        }
    }
}

```

Essa imagem logo acima, com o método **“login”**, é para ser feita a validação dos dados que foram introduzidos na primeira tela.

Ele recebe o valor dos dois inputs e utiliza dos seus valores para checar dentro da lista de usuários, se aquele usuário existe, caso sim, ele é enviado para a tela de exibição dos seus dados, caso não, é criado e exibida uma caixa de diálogo mostrando que os dados do usuário estão incorretos, mantendo – o na tela inicial.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.show_data);
    GetUser(intent);
}

private User GetUser(Intent intent){
    EditText editEmailOrCpf = findViewById(R.id.editTextTextEmailAddress);
    String cpfOrEmail = editEmailOrCpf.getText().toString();
    EditText editPassword = findViewById(R.id.editTextTextPassword);
    String password = editPassword.getText().toString();

    User u = new User();
    List<User> users = u.getUser();
    for(int i = 0; i < users.size(); i++){
        if((users.get(i).Email.equals(cpfOrEmail) || users.get(i).Cpf.equals(cpfOrEmail)) &&
            users.get(i).Password.equals(password)){
            u = users.get(i);
        }
    }

    return u;
}

public void backToHome(View view){
    Intent intent = new Intent( packageContext: this, MainActivity.class);
    startActivity(intent);
}

```

Já nesta imagem é mostrado a Activity da segunda dela, para ser exibidos os dados, essa função ficou incompleta e os dados não estão sendo exibidos.

Foi criado também o método backToHome, que através do clique no botão “**Voltar**”, o App retorna para a página inicial.

Link para repositório no GitHub, clique no ícone

