

**Centro Universitário Atenas**

Projeto Corrente do Bem

Desenvolvimento de aplicação mobile em React Native

**Curso:** Desenvolvimento de Sistemas – EAD – Sete Lagoas MG

**Semestre:** 01/2023

**Disciplina:**

**Aluno:** Filipe Miranda

**Matrícula:** E23-1-19773

**Título:** Documentação do Projeto Corrente do Bem - Formulário de login com e-mail e senha .

**Linguagem de Programação:** React Native

**Descrição:** Esta etapa do projeto é parte de uma aplicação móvel desenvolvida em React Native que permite o login atraves de e-mail e senha do usuários. O formulário Entrar" para ter acesso ao servidor.

**Visualização da tela criada:**

|  |
| --- |
|  |

**Funções Utilizadas:**



* **useState:** funcionalidade do React usada para construir interfaces de usuário, que permite criar e gerenciar estados em componentes funcionais. Neste projeto, são utilizados dois estados: "name" e "description", para armazenar os valores dos campos de entrada de texto .
* **Componentes do React Native:** São importados os componentes "View", "TextInput", "Text", "Image" e "TouchableOpacity" do React Native para criar a interface de usuário . Esses componentes são utilizados para criar as diferentes partes da tela, como a imagem do logo, o título, os campos de entrada de texto e o botão de entrar (acessar).

const PermissionForm = () => {

const [name, setName] = useState('');

const [description, setDescription] = useState('');

const handleNameChange = (text) => {

setName(text);

};

const handleDescriptionChange = (text) => {

setDescription(text);

};

const handleSubmit = () => {

// Aqui serÃ¡ para fazer a lÃ³gica para enviar os dados de cadastro para o servidor

console.log('Nome', name);

console.log('DescriÃ§Ã£o:', description);

};

* **handleNameChange:** É uma função que é chamada quando o valor do campo de entrada de texto do nome do login é alterado. Ela atualiza o estado "name" com o novo valor inserido.
* **handleDescriptionChange:** É uma função que é chamada quando o valor do campo de entrada de texto da descrição do e-mail é alterado. Ela atualiza o estado "description" com o novo valor inserido.
* **handleSubmit:** É uma função que é chamada quando o botão "Entrar" é pressionado. Neste momento, os dados do usuario são enviados para o servidor (lógica ainda não implementada) e são exibidos no console do desenvolvedor através dos comandos "console.log".

|  |  |
| --- | --- |
|  |  |

* **StyleSheet:** É um objeto do React Native que permite criar estilos para os componentes. Neste projeto, são utilizados estilos para definir a aparência do formulário login, como a cor de fundo, tamanho e estilo de fonte, margens, entre outros.

const PermissionForm = () => {

const [name, setName] = useState('');

const [description, setDescription] = useState('');

const handleNameChange = (text) => {

setName(text);

};

const handleDescriptionChange = (text) => {

setDescription(text);

};

const handleSubmit = () => {

// Aqui serÃ¡ para fazer a lÃ³gica para enviar os dados de cadastro para o servidor

console.log('Nome', name);

console.log('DescriÃ§Ã£o:', description);

};

return (

<View style={styles.container}>

<Image

style={styles.image}

source={require('./assets/download.png')}

/>

<Text style={styles.title}>Login da conta</Text>

<TextInput

style={styles.input}

Nesta parte do código, está sendo definido o retorno do componente **PermissionForm,** que é uma função que renderiza a interface de usuário do formulário de cadastro de permissões de usuários.

return (

<View style={styles.container}>

<Image

style={styles.image}

source={require('./assets/download.png')}

/>

<Text style={styles.title}>Login da conta</Text>

<TextInput

style={styles.input}

onChangeText={handleNameChange}

value={name}

placeholder="E-mail"

/>

<TextInput

style={styles.input}

onChangeText={handleDescriptionChange}

value={description}

placeholder="Senha"

/>

<TouchableOpacity style={styles.botao} onPress={handleSubmit} >

<Text style={styles.textbtn}>Entrar </Text>

</TouchableOpacity>

</View>

Detalhamento de cada elemento presente no retorno:

**<View style={styles.container}>:** componente que representa uma área de visualização retangular. Está sendo estilizado com a propriedade style utilizando o objeto de estilo styles.container, que é definido posteriormente no código utilizando o StyleSheet.create(). O estilo aplicado define a flexibilidade do componente (flex: 1), o alinhamento vertical e horizontal no centro (justifyContent: 'center' e alignItems: 'center') e o preenchimento interno de 16 unidades (padding: 16).

**<Image style={styles.image} source={require('./assets/download.png')} />:** componente do React Native que representa uma imagem. Ele está sendo estilizado com o objeto de estilo styles.image e recebe uma propriedade source que define a imagem a ser exibida. A imagem está sendo carregada usando o require('./assets/download.png'), que é um método do React Native para importar imagens locais no projeto. O caminho e o nome do arquivo da imagem estão sendo especificados como ./assets/download.png.

**<Text style={styles.title}></Text>:** componente do React Native que representa um texto. Ele está sendo estilizado com o objeto de estilo styles.title e exibe o texto " e-mail e senha".

**<TextInput**

**style={styles.input}**

**onChangeText={handleNameChange}**

**value={name}**

**placeholder="E-mail" />:**

e

**<TextInput**

**style={styles.input}**

**onChangeText={handleDescriptionChange}**

**value={description}**

**placeholder="Senha" />:**

Componente do React Native que representa um campo de entrada de texto. Ele está sendo estilizado com o objeto de estilo styles.input e recebe três propriedades:

* **onChangeText:** função de callback que será chamada quando o texto no campo for alterado, e está sendo associada à função handleNameChange ou handleDescriptionChange, conforme cada caso;
* **value:** que é o valor do texto exibido no campo, e está sendo definido como name ou description, para cada caso, que é o valor do estado name ou description definido utilizando a funcionalidade useState, inserida no início do código; e
* **placeholder:** que é um texto de dica exibido quando o campo está vazio, e está sendo definido como “Login da conta".

**<TouchableOpacity**

**style={styles.botao}**

**onPress={handleSubmit}>**

**</TouchableOpacity>:** Essa parte do código cria um botão "email" que pode ser pressionado pelo usuário. Ele está sendo estilizado com o objeto de estilo styles.botao e recebe uma propriedade onPress que define a função handleSubmit como a ação a ser executada quando o botão for pressionado.

**</View>:** É a tag de fechamento do componente <View>, indicando o fim do elemento de visualização retangular.