

| | | |
|--|--|-------------------------------------|
| CURSO: | Análise e Desenvolvimento de Sistemas – MOBILE | Avaliação Formadora (Competência 3) |
| Analisar, projetar e programar soluções em sistemas MOBILE | | |

Tipo de Atividade

- a) Trabalho Individual
- b) Valor: 15 pontos

Exercício de React Native : Sistema de cadastro Simples

Visão Geral do Projeto

Você criará um aplicativo com três telas principais:

1. **Tela de Login:** O ponto de entrada do aplicativo. O usuário poderá inserir suas credenciais (email e senha) para acessar a lista de usuários.
2. **Tela de Cadastro:** A partir da tela de Login, o usuário poderá navegar para um formulário de cadastro onde poderá criar uma nova conta (nome, email e senha).
3. **Tela de Lista de Usuários:** Uma tela protegida que só pode ser acessada após o login. Ela exibirá uma lista com os nomes e emails de todos os usuários cadastrados.

O foco deste exercício é a lógica de navegação e o gerenciamento dos dados dos usuários no *frontend*. **Não será necessário conectar a um banco de dados ou API real.** Os usuários serão salvos em uma lista (array) dentro do próprio aplicativo.

Objetivos de Aprendizagem

Ao final deste exercício, o aluno será capaz de:

- Estruturar um projeto React Native com múltiplas telas.
- Implementar navegação em pilha (Stack Navigation) usando a biblioteca React Navigation.
- Criar formulários controlados com os componentes TextInput e Button.
- Gerenciar o estado de formulários e listas utilizando o hook useState.
- Compartilhar dados entre diferentes telas (passando parâmetros ou utilizando Context API).
- Implementar lógica condicional de navegação (proteger uma tela contra acesso não autenticado).
- Utilizar o componente FlatList para renderizar listas de dados de forma otimizada.

Descrição Detalhada das Telas

1. Tela de Login (LoginScreen)

- **Componentes:**
 - Um título, como "Bem-vindo(a) de volta!".
 - Dois campos de entrada (TextInput): um para "Email" e outro para "Senha". A senha deve ter a visualização dos caracteres ocultada (secureTextEntry).
 - Um botão (Button ou TouchableOpacity) com o texto "Entrar".

- Um texto ou botão com o texto "Não tem uma conta? Cadastre-se" que, ao ser clicado, levará o usuário para a Tela de Cadastro.

Lógica:

- Ao clicar em "Entrar", o aplicativo deve verificar se existe um usuário na lista de cadastrados com o email e a senha fornecidos.
- Se as credenciais estiverem corretas, o usuário deve ser redirecionado para a ListaUsuariosScreen.
- Se as credenciais estiverem incorretas, exiba um alerta (Alert) com uma mensagem de erro (ex: "Email ou senha inválidos").

2. Tela de Cadastro (CadastroScreen)

• Componentes:

- Um título, como "Crie sua Conta".
- Três campos de entrada (TextInput): "Nome", "Email" e "Senha".
- Um botão com o texto "Cadastrar".

• Lógica:

- O usuário preenche os campos.
- Ao clicar em "Cadastrar", o aplicativo deve verificar se o email já não está em uso.
- Se o email for novo, um novo objeto de usuário (ex: { id: Date.now(), nome, email, senha }) deve ser adicionado à lista global de usuários.
- Após o cadastro bem-sucedido, o usuário deve ser redirecionado de volta para a LoginScreen para que possa fazer o login com a nova conta. Exiba um alerta informando que o cadastro foi realizado com sucesso.

3. Tela de Lista de Usuários (ListaUsuariosScreen)

• Componentes:

- Um título na barra de navegação, como "Usuários Cadastrados".
- Uma lista (FlatList) que exibe os usuários.
- Cada item da lista deve mostrar o **Nome** e o **Email** do usuário.

• Lógica:

- Esta tela deve receber a lista de usuários cadastrados e renderizá-la.
- O acesso a esta tela é condicional: só é possível chegar aqui após um login bem-sucedido.

Conceitos Fundamentais a serem aplicados

- **React Navigation:** Utilize o "Stack Navigator" para criar a estrutura de navegação em pilha, permitindo que o usuário avance (Login -> Cadastro) e retorne.
- **Gerenciamento de Estado (useState):**
 - Em cada tela de formulário, use estados para armazenar os valores dos TextInput.
 - Crie um estado "global" no componente principal (App.tsx) para armazenar o array de usuários cadastrados e passe-o para as telas que precisam dele.
- **Passagem de Dados:**
 - Passe a lista de usuários e a função para adicionar novos usuários como *props* para as telas de Login e Cadastro.
 - Passe a lista de usuários para a tela de Lista de Usuários.
- **Renderização de Listas (FlatList):** Use a FlatList na ListaUsuariosScreen por ser uma forma eficiente de renderizar listas, especialmente as que podem crescer em tamanho.

Desafios Adicionais (Opcionais)

1. **Validação de Formulário:** Adicione validações mais robustas, como verificar se o email tem um formato válido ou se a senha tem um número mínimo de caracteres. Desabilite o botão de "Cadastrar" ou "Entrar" enquanto os campos não estiverem preenchidos.

2. **Context API:** Em vez de passar os dados dos usuários via *props* por várias camadas (prop drilling), refatore o projeto para usar a Context API do React para criar um `UserContext` que proverá a lista de usuários e as funções de manipulação para qualquer componente que precisar.
3. **Logout:** Adicione um botão de "Sair" na `ListaUsuariosScreen` que, ao ser clicado, retorna o usuário para a `LoginScreen` e impede que ele volte para a lista sem fazer login novamente.
4. **Estilização Avançada:** Utilize `StyleSheet` para criar um design mais elaborado e consistente entre as telas. Personalize os componentes `TextInput` e crie botões customizados com `TouchableOpacity` e `Text`.
5. **Persistência de Dados:** Utilize o `AsyncStorage` para salvar a lista de usuários no dispositivo. Dessa forma, quando o aplicativo for fechado e reaberto, os usuários cadastrados não serão perdidos.