

Salvando os dados no Storage

- Instale os seguintes pacotes

yarn add @react-native-community/async-storage

- Inicie novamente a aplicação digitando

react-native run-android

Edite o arquivo src/pages/Main/index.js

Insira a linha 5

```
1 import React, { Component } from 'react'
2 import { Button } from 'react-native'
3
4 // importa a biblioteca do async-storage
5 import AsyncStorage from '@react-native-community/async-storage'
6 import api from '../../services/api'
7 import { Container, Form, Input, List, User, Avatar, Name, Bio } from './styles'
```

Insira da linha 15 até 26

```
src > pages > Main > JS index.js > Main > handleAddUser
9 export default class Main extends Component {
10   state = {
11     newUser: '',
12     users: [],
13   }
14
15   // cria o método componentDidMount() quando carregar o aplicativo
16   async componentDidMount() {
17     // recupera todos os users gravados no banco de dados local do smartphone
18     const users = await AsyncStorage.getItem('users');
19
20     // se existir usuários, altera o estado users para o conteúdo que
21     // está gravado no banco de dados e converte de JSON para
22     // objeto (JSON.parse)
23     if (users) {
24       this.setState({ users: JSON.parse(users) });
25     }
26   }
}
```

Insira da linha 27 até 40

```
27 // quando houver alteração no estado users
28 // _ é o propsState antes da atualização
29 // prevState - é o estado antes da atualização
30 componentDidUpdate(_, prevState) {
31   const { users } = this.state;
32   // se o estado anterior (prevState.users) for diferente do novo estado (users)
33   // significa que houve alteração e a informação deve ser gravada no banco de
34   // dados do smartphone
35   if (prevState.users !== users) {
36     // grava o novo array de users no banco de dados do smartphone convertendo-o
37     // para json (JSON.stringify)
38     AsyncStorage.setItem('users', JSON.stringify(users))
39   }
40 }
41
42 handleAddUser = async () => {
43   const { users, newUser } = this.state;
```