



Estácio

Estácio de Sá - Unidade São João de Meriti

Desenvolvimento Full Stack

Aluno: Victor de A. Costa

Missão Prática I Nível 1 | Mundo 4 - 2025.1 RPG0023 - Vamos criar um App!

A empresa "Meeting" busca criar um aplicativo móvel eficaz para o cadastro de fornecedores, com listas e imagens de alta qualidade, economizando recursos e proporcionando uma excelente experiência ao usuário. A escolha da tecnologia React Native é crucial para estabelecer uma presença sólida no mercado móvel.



---> CÓDIGOS UTILIZADOS

App.js

```
import React, { useState } from 'react';
import { Text, TextInput, View, StyleSheet, SafeAreaView, ScrollView }
from 'react-native';
import CadastroClientes from './CadastroCliente';
import BemVindo from './BemVindo';
import CadastroFornecedores from './CadastroFornecedores';
import LogoMeeting from './Logo';
import Footer from './Foot';

const App = () => {
  return (
    <SafeAreaView style={{ flex: 1 }}>
```

```

        <ScrollView>
            <BemVindo />
            <LogoMeeting />
            <CadastroClientes />
            <CadastroFornecedores />
            <Footer />
        </ScrollView>
    </SafeAreaView>
  );
};

const styles = StyleSheet.create({
  container: {
    padding: 10,
    alignItems: 'center',
  },
  input: {
    height: 40,
    borderColor: 'gray',
    borderWidth: 1,
    width: '100%',
    paddingHorizontal: 10,
    marginBottom: 10,
  },
  text: {
    padding: 10,
    fontSize: 42,
  },
});

export default App;

```

BemVindo.Js

```

import React, { useState, useEffect } from 'react';
import { Text, Image, TextInput, View, StyleSheet } from 'react-native';
import AsyncStorage from
  '@react-native-async-storage/async-storage/lib/module/AsyncStorage.js';

const BemVindo = () => {

```

```

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

useEffect(() => {
  const loadName = async () => {
    const savedName = await AsyncStorage.getItem('userName');
    if (savedName) {
      setName(savedName);
    }
  };
  loadName();
}, []);

useEffect(() => {
  if (name) {
    AsyncStorage.setItem('userName', name);
  }
}, [name]);

return (
  <View style={styles.container}>
    <TextInput
      style={styles.input}
      placeholder="Identifique-se"
      onChangeText={(text) => setName(text)}
      value={name}
    />
    <Text style={styles.text}>
      Seja Bem-Vindo ao Meeting{name ? `, ${name}!` : '!' }
    </Text>
  </View>

);
};

const styles = StyleSheet.create({
  container: {
    padding: 10,
    alignItems: 'center',
    backgroundColor: 'black',
    borderColor: 'lightblue',

```

```

    },
    text: {
      fontSize: 20,
      margin: 10,
      color: 'lightblue',
    },
    input: {
      height: 40,
      borderColor: 'lightblue',
      borderWidth: 2,
      width: '80%',
      paddingHorizontal: 10,
      borderRadius: 5,
      color: 'lightblue',
      textAlign: 'center',
    },
    tinyLogo: {
      margin: 10,
      alignItems: 'center',
      width: 60,
      height: 50,
    },
  },
));

export default BemVindo;

```

Logo.Js

```

import React from 'react';
import { View, Image, StyleSheet } from 'react-native';

const LogoMeeting = () => {
  return (
    <View style={styles.container}>
      <Image
        style={styles.logo}
        source={{
          uri: 'https://i.ibb.co/QvVcDQqJ/meeting.png',
        }}
      />
    </View>
  );
}

```

```

    );
  };

  const styles = StyleSheet.create({
    logo: {
      width: 1000,
      height: 300,
      resizeMode: 'contain',
    },
    container: {
      backgroundColor: 'black',
      alignItems: 'center',
      justifyContent: 'center',
      height: 300,
    },
  });

  export default LogoMeeting;

```

CadastroClientes.js

```

import React, { useState, useEffect } from 'react';
import { View, Text, TextInput, Button, FlatList, StyleSheet,
TouchableOpacity, Platform, Alert, Image } from 'react-native';
import AsyncStorage from
'@react-native-async-storage/async-storage/lib/module/AsyncStorage.js';

const CadastroClientes = () => {
  const [clientes, setClientes] = useState([]);
  const [nome, setNome] = useState('');
  const [endereco, setEndereco] = useState('');
  const [contato, setContato] = useState('');
  const [imagem, setImagem] = useState(null);
  const [editandoId, setEditandoId] = useState(null);

  useEffect(() => {

```

```

const carregarClientes = async () => {
  try {
    const dados = await AsyncStorage.getItem('clientes');
    if (dados) setClientes(JSON.parse(dados));
  } catch (error) {
    console.error('Erro ao carregar clientes:', error);
  }
};

carregarClientes();
}, []);

useEffect(() => {
  AsyncStorage.setItem('clientes', JSON.stringify(clientes));
}, [clientes]);

const selecionarImagem = (e) => {
  const file = e.target.files[0];
  if (file) {
    const reader = new FileReader();
    reader.onload = () => {
      setImagem(reader.result); // URI base64
    };
    reader.readAsDataURL(file);
  }
};

const adicionarOuEditarCliente = () => {
  if (!nome.trim() || !endereco.trim()) {
    Alert.alert('Preencha todos os campos!');
    return;
  }

  if (editandoId) {
    const atualizados = clientes.map((c) =>
      c.id === editandoId ? { ...c, nome, endereco, contato, imagem } :
c
    );
    setClientes(atualizados);
    setEditandoId(null);
  } else {

```

```

    const novo = { id: Date.now().toString(), nome, endereco, contato,
imagem };
    setClientes([...clientes, novo]);
  }

  setNome('');
  setEndereco('');
  setImagem(null);
};

const editarCliente = (cliente) => {
  setNome(cliente.nome);
  setEndereco(cliente.endereco);
  setImagem(cliente.imagem || null);
  setContato(cliente.contato || '');
  setEditandoId(cliente.id);
};

const excluirCliente = (id) => {
  if (Platform.OS === 'web') {
    if (window.confirm('Deseja realmente excluir este cliente?')) {
      const atualizados = clientes.filter(c => c.id !== id);
      setClientes(atualizados);
    }
  } else {
    Alert.alert('Confirmar', 'Deseja realmente excluir este cliente?', [
      { text: 'Cancelar', style: 'cancel' },
      {
        text: 'Excluir',
        style: 'destructive',
        onPress: () => {
          const atualizados = clientes.filter(c => c.id !== id);
          setClientes(atualizados);
        }
      }
    ],
    1);
  }
};

return (

```

```

<View style={styles.container}>
  <Text style={styles.titulo}>Cadastro de Clientes</Text>

  <TextInput
    style={styles.input}
    placeholder="Nome"
    value={nome}
    onChangeText={setNome}
  />

  <TextInput
    style={styles.input}
    placeholder="Endereço"
    value={endereço}
    onChangeText={setEndereço}
  />

  <TextInput
    style={styles.input}
    placeholder="Contato"
    value={contato}
    onChangeText={setContato}
  />

  <input type="file" accept="image/*" onChange={selecionarImagem} style={{
marginBottom: 10 }} />

  {imagem && <Image source={{ uri: imagem }}
style={styles.imagemPreview} />}

  <Button
    title={editandoId ? 'Salvar Edição' : 'Adicionar Cliente'}
    onPress={adicionarOuEditarCliente}
  />

  <FlatList
    data={clientes}

```



```

    keyExtractor={ (item) => item.id}
    renderItem={({ item }) => (
      <View style={styles.card}>
        {item.imagem && <Image source={{ uri: item.imagem }}
style={styles.imagemCard} />}
        <Text style={styles.textoCard}>👤 {item.nome}</Text>
        <Text>🏠 {item.endereco}</Text>
        <Text>☎️ {item.contato}</Text>

        <View style={styles.botoesCard}>
          <TouchableOpacity onPress={() => editarCliente(item)}
style={styles.botaoEditar}>
            <Text style={styles.textoBotaoCard}>Editar</Text>
          </TouchableOpacity>
          <TouchableOpacity onPress={() => excluirCliente(item.id)}
style={styles.botaoExcluir}>
            <Text style={styles.textoBotaoCard}>Excluir</Text>
          </TouchableOpacity>
        </View>
      </View>
    )}
  />
</View>
);
};

const styles = StyleSheet.create({
  container: { flex: 1, padding: 20, backgroundColor: '#f5f5f5' },
  titulo: { fontSize: 22, fontWeight: 'bold', marginBottom: 10 },
  input: { borderWidth: 1, padding: 10, marginBottom: 10, borderRadius: 5,
backgroundColor: '#fff' },
  botaoImagem: { backgroundColor: '#3498db', padding: 10, borderRadius: 5,
alignItems: 'center', marginBottom: 10 },
  textoBotao: { color: '#fff', fontWeight: 'bold' },
  imagemPreview: { width: 100, height: 100, borderRadius: 10,
marginBottom: 10 },
  card: { backgroundColor: '#fff', padding: 10, borderRadius: 10,
marginBottom: 10 },
  textoCard: { fontSize: 18, fontWeight: 'bold' },

```

```

    imagemCard: { width: 80, height: 80, borderRadius: 10, marginBottom: 5
},
    botoesCard: { flexDirection: 'row', justifyContent: 'space-between',
marginTop: 10 },
    botaoEditar: { backgroundColor: '#f1c40f', padding: 8, borderRadius: 5,
flex: 1, marginRight: 5, alignItems: 'center' },
    botaoExcluir: { backgroundColor: '#e74c3c', padding: 8, borderRadius: 5,
flex: 1, marginLeft: 5, alignItems: 'center' },
    textoBotaoCard: { color: '#fff', fontWeight: 'bold' },
  ));

export default CadastroClientes;

```

CadastroFornecedores.js

```

import React, { useState, useEffect } from 'react';
import { View, Text, TextInput, Button, FlatList, StyleSheet, Image,
Platform, TouchableOpacity, Alert } from 'react-native';
import AsyncStorage from
'@react-native-async-storage/async-storage/lib/module/AsyncStorage.js';
import { Select, MenuItem, InputLabel, FormControl } from '@mui/material';
import { DragDropContext, Droppable, Draggable, } from
'react-beautiful-dnd';

const CadastroFornecedores = () => {
  const [fornecedores, setFornecedores] = useState([]);
  const [nome, setNome] = useState('');
  const [endereco, setEndereco] = useState('');
  const [contato, setContato] = useState('');
  const [categoria, setCategoria] = useState('');
  const [imagem, setImagem] = useState(null);
  const [editandoId, setEditandoId] = useState(null);
  const [filtroCategoria, setFiltroCategoria] = useState('');

  useEffect(() => {
    const carregar = async () => {
      const dados = await AsyncStorage.getItem('fornecedores');
      if (dados) setFornecedores(JSON.parse(dados));
    };
  });

```

```

    carregar();
  }, []);

  useEffect(() => {
    AsyncStorage.setItem('fornecedores', JSON.stringify(fornecedores));
  }, [fornecedores]);

  const selecionarImagem = (e) => {
    const file = e.target.files[0];
    if (file) {
      const reader = new FileReader();
      reader.onload = () => {
        setImagem(reader.result);
      };
      reader.readAsDataURL(file);
    }
  };

  const adicionarFornecedor = () => {
    if (!nome || !endereco || !contato || !categoria) {
      Alert.alert('Preencha todos os campos!');
      return;
    }

    if (editandoId) {
      const atualizados = fornecedores.map((f) =>
        f.id === editandoId
          ? { ...f, nome, endereco, contato, categoria, imagem }
          : f
      );
      setFornecedores(atualizados);
      setEditandoId(null);
    } else {
      const novo = {
        id: Date.now().toString(),
        nome,
        endereco,
        contato,
        categoria,
        imagem,
      };
    }
  };

```

```

    };

    setFornecedores([...fornecedores, novo]);
  }

  setNome('');
  setEndereco('');
  setContato('');
  setCategoria('');
  setImagem(null);
};

const editarFornecedor = (fornecedor) => {
  setNome(fornecedor.nome);
  setEndereco(fornecedor.endereco);
  setContato(fornecedor.contato);
  setCategoria(fornecedor.categoria);
  setImagem(fornecedor.imagem || null);
  setEditandoId(fornecedor.id);
};

const excluirFornecedor = (id) => {
  if (Platform.OS === 'web') {
    if (window.confirm('Deseja realmente excluir este fornecedor?')) {
      const atualizados = fornecedores.filter(c => c.id !== id);
      setFornecedores(atualizados);
    }
  } else {
    Alert.alert('Confirmar', 'Deseja realmente excluir este
fornecedor?', [
      { text: 'Cancelar', style: 'cancel' },
      {
        text: 'Excluir',
        style: 'destructive',
        onPress: () => {
          const atualizados = fornecedores.filter(c => c.id !== id);
          setFornecedores(atualizados);
        }
      }
    ],
  );
  };
};

```

```

const onDragEnd = (result) => {
  if (!result.destination) return;
  const novaOrdem = Array.from(fornecedores);
  const [removido] = novaOrdem.splice(result.source.index, 1);
  novaOrdem.splice(result.destination.index, 0, removido);
  setFornecedores(novaOrdem);
};

const fornecedoresFiltrados =
  filtroCategoria === ''
    ? fornecedores
    : fornecedores.filter((f) => f.categoria === filtroCategoria);

return (
  <View style={styles.container}>
    <Text style={styles.titulo}>Cadastro de Fornecedores</Text>

    <TextInput
      style={styles.input}
      placeholder="Nome"
      value={nome}
      onChangeText={setNome}
    />
    <TextInput
      style={styles.input}
      placeholder="Endereço"
      value={endereco}
      onChangeText={setEndereco}
    />
    <TextInput
      style={styles.input}
      placeholder="Contato"
      value={contato}
      onChangeText={setContato}
    />
    <TextInput
      style={styles.input}
      placeholder="Categoria"
      value={categoria}

```

```

        onChangeText={setCategoria}
      />

      <input
        type="file"
        accept="image/*"
        onChange={selecionarImagem}
        style={{ marginBottom: 10 }}
      />
      {imagem && <Image source={{ uri: imagem }} style={styles.imagem} />}

    <Button
      title={editandoId ? 'Salvar Edição' : 'Adicionar Fornecedor'}
      onPress={adicionarFornecedor}
    />
  <View style={{ marginTop: 20 }} />
    <FormControl fullWidth style={{ marginBottom: 15 }}>
      <InputLabel id="filtro-categoria-label">Filtrar por
categoria</InputLabel>
      <Select
        labelId="filtro-categoria-label"
        value={filtroCategoria}
        label="Filtrar por categoria"
        onChange={(e) => setFiltroCategoria(e.target.value)}
      >
        <MenuItem value="">Todas</MenuItem>
        {[...new Set(fornecedores.map((f) => f.categoria))].
          .sort()
          .map((cat) => (
            <MenuItem key={cat} value={cat}>
              {cat}
            </MenuItem>
          ))}
      </Select>
    </FormControl>

    <br>
    </br>

    <DragDropContext onDragEnd={onDragEnd}>

```

```

<Draggable key={item.id} draggableId={item.id}
index={index}>
  { (provided) => (
    <View ref={provided.innerRef} {...provided.draggableProps}>
      {fornecedoresFiltrados.map((item, index) => (
        <Draggable key={item.id} draggableId={item.id}
index={index}>
          { (provided) => (
            <View
              ref={provided.innerRef}
              {...provided.draggableProps}
              {...provided.dragHandleProps}
              style={ [styles.card, provided.draggableProps.style] }
            >
              {item.imagem && (
                <Image
                  source={{ uri: item.imagem }}
                  style={styles.imagemCard}
                />
              )}
              <Text style={styles.textoCard}>📌 {item.nome}</Text>
              <Text>🏠 {item.endereco}</Text>
              <Text>☎ {item.contato}</Text>
              <Text>📦 {item.categoria}</Text>

              <View style={styles.botoesCard}>
                <TouchableOpacity
                  onPress={() => editarFornecedor(item)}
                  style={styles.botaoEditar}
                >
                  <Text
style={styles.textoBotaoCard}>Editar</Text>
                </TouchableOpacity>
                <TouchableOpacity
                  onPress={() => excluirFornecedor(item.id)}
                  style={styles.botaoExcluir}
                >
                  <Text
style={styles.textoBotaoCard}>Excluir</Text>
                </TouchableOpacity>
              </View>
            </Draggable>
          )}
        </View>
      )}
    </Draggable>
  )}
</View>

```

```

        </View>
      )}
    </Draggable>
  )})
  {provided.placeholder}
</View>
)}

</Droppable>
</DragDropContext>

</View>
);
};

const styles = StyleSheet.create({
  container: { flex: 1, padding: 20, backgroundColor: '#f5f5f5' },
  titulo: { fontSize: 22, fontWeight: 'bold', marginBottom: 10 },
  input: {
    borderWidth: 1,
    padding: 10,
    marginBottom: 10,
    borderRadius: 5,
    backgroundColor: '#fff',
  },
  imagem: { width: 100, height: 100, borderRadius: 10, marginBottom: 10 },
  card: {
    backgroundColor: '#fff',
    padding: 10,
    borderRadius: 10,
    marginBottom: 10,
  },
  textoCard: { fontSize: 18, fontWeight: 'bold' },
  imagemCard: { width: 80, height: 80, borderRadius: 10, marginBottom: 5 },
  botoesCard: {
    flexDirection: 'row',

```



```

      justifyContent: 'space-between',
      marginTop: 10,
    },
    botaoEditar: {
      backgroundColor: '#f1c40f',
      padding: 8,
      borderRadius: 5,
      flex: 1,
      marginRight: 5,
      alignItems: 'center',
    },
    botaoExcluir: {
      backgroundColor: '#e74c3c',
      padding: 8,
      borderRadius: 5,
      flex: 1,
      marginLeft: 5,
      alignItems: 'center',
    },
    textoBotaoCard: { color: 'fff', fontWeight: 'bold' },
  });

export default CadastroFornecedores;

```

Foot.js

```

import React from 'react';
import { Text, View, StyleSheet } from 'react-native';

const Footer = () => {
  return (
    <View style={styles.container}>
      <Text style={styles.text}>
        App Criado por Victor de A. Costa
      </Text>
    </View>
  );
};

```


```
const styles = StyleSheet.create({
  container: {
    padding: 10,
    alignItems: 'center',
    backgroundColor: 'black',
    borderColor: 'lightblue',
  },
  text: {
    fontSize: 10,
    margin: 10,
    color: 'lightblue',
  },
});

export default Footer;
```

Imagens

Victor|

Seja Bem-Vindo ao Meeting, Victor!



Cadastro de Clientes

Nome

Endereço

Contato

Escolher Arquivo

 | Nenhum arquivo escolhido

ADICIONAR CLIENTE

Cadastro de Fornecedores

Nome

Endereço

Contato

Categoria

Escolher Arquivo | logo_crt_2023.png

ADICIONAR FORNECEDOR

Filtrar por categoria

Design



 Createe

 Rio

 21

 Design

Editar

Excluir