



# Missão Prática | Nível 4 | Mundo 1

Luís Ricardo Nogueira

## Objetivo da Prática

- Configurar o ambiente de desenvolvimento React Native;
- Implementar a funcionalidade de entrada de texto em um componente React Native;
- Implementar um Componente de Lista Dinâmica (ScrollView);
- Implementar componentes React Native para exibir informações de forma dinâmica em listas;
- Empregar elementos visuais em um aplicativo React Native.

## 1º Procedimento | Criação do ambiente expo

- Instalei o Expo CLI com o seguinte comando:  
`npm install -g expo-cli`
- Dentro da pasta do projeto criada na minha área de trabalho iniciei o prompt e criei o projeto raiz com o seguinte comando:  
`expo init FornecedorApp`
- Adicionei também as seguintes dependências:  
`expo install expo-image-picker`  
`npm install react-native-web`

## 2º Procedimento | Construindo o código

Código usado no projeto:

```
import React, { useState } from 'react';
import { StyleSheet, View, Text, TextInput, Button, FlatList, Image,
TouchableOpacity } from 'react-native';
import * as ImagePicker from 'expo-image-picker';

const App = () => {
  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 adicionarFornecedor = () => {
    if (nome && endereco && contato && categoria) {
      setFornecedores([...fornecedores, { nome, endereco, contato,
categoria, imagem }]);
      setNome('');
      setEndereco('');
      setContato('');
      setCategoria('');
      setImagem(null);
    }
  };

  const verificarPermissao = async () => {
    const { status } = await
ImagePicker.requestMediaLibraryPermissionsAsync();
    return status === 'granted';
  };

  const selecionarImagem = async () => {
    const resultado = await ImagePicker.launchImageLibraryAsync({
      mediaTypes: ImagePicker.MediaTypeOptions.Images,
      allowsEditing: true,
      aspect: [4, 3],
      quality: 1,
    });

    if (!resultado.cancelled) {
      setImagem(resultado.uri);
    }
  };
}
```

```

    }
  };

  return (
    <View style={styles.container}>
      <Text style={styles.title}>Cadastro Empresa Meeting</Text>
      <TextInput placeholder="Nome" value={nome} onChangeText={setNome}
style={styles.input} />
      <TextInput placeholder="Endereço" value={endereco}
onChangeText={setEndereco} style={styles.input} />
      <TextInput placeholder="Contato" value={contato}
onChangeText={setContato} style={styles.input} />
      <TextInput placeholder="Categoria" value={categoria}
onChangeText={setCategoria} style={styles.input} />
      <Button title="Selecionar Imagem" onPress={selecionarImagem} />
      {imagem && <Image source={{ uri: imagem }} style={styles.image} />}
      <Button title="Adicionar Fornecedor" onPress={adicionarFornecedor}
/>

      <FlatList
        data={fornecedores}
        keyExtractor={(item, index) => index.toString()}
        renderItem={({ item }) => (
          <View style={styles.item}>
            {item.imagem && <Image source={{ uri: item.imagem }}
style={styles.image} />}
            <Text>Nome: {item.nome}</Text>
            <Text>Endereço: {item.endereco}</Text>
            <Text>Contato: {item.contato}</Text>
            <Text>Categoria: {item.categoria}</Text>
          </View>
        )}
      />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 22,
    backgroundColor: '#fff',
  },
  title: {
    fontSize: 23,
    marginBottom: 15,
  },
});

```

```
    padding: 24,  
  },  
  input: {  
    borderWidth: 1,  
    borderColor: '#ccc',  
    padding: 12,  
    marginBottom: 10,  
  },  
  image: {  
    width: 100,  
    height: 100,  
    marginBottom: 15,  
  },  
  item: {  
    padding: 10,  
    borderBottomWidth: 1,  
    borderBottomColor: '#ccc',  
  },  
});  
  
export default App;
```

## Conclusão

Utilizei o cenário Expo e compilei usando android debug pelo app do próprio expo.