



Estácio

FACULDADE ESTÁCIO

CÂMPUS VOLTA REDONDA – RJ

DESENVOLVIMENTO FULL STACK

DISCIPLINA – VAMOS CRIAR UM APP!

TURMA – 2023.2

SEMESTRE – 4

VOLTA REDONDA, NOVEMBRO 2024.

DESENVOLVIMENTO FULL STACK

DISCIPLINA – VAMOS CRIAR UM APP!

TURMA – 2023.2

SEMESTRE – 4

ALUNO – BRUNO SAMPAIO BASTOS

TUTOR – JHONATAN ALVES

GITHUB - <https://github.com/BrunoTI-Code?tab=repositories>

VOLTA REDONDA, NOVEMBRO 2024.

1 OBJETIVOS DA PRÁTICA:

- Configurar o ambiente de desenvolvimento React Native;
- Implementar a funcionalidade de entrada de texto em um componente React Native;I
- 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.

2 MISSÃO PRÁTICA | VAMOS CRIAR UM APP!

Nessa atividade revisaremos tudo o que utilizamos nas micro atividades anteriores. Além disso, veremos também como o React Native responde a eventos e qual lógica de manipulação será utilizada neste exercício.

2.1 CONTEXTUALIZAÇÃO

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. Nesta atividade, você aprenderá os princípios básicos do React Native.

3 REQUISITOS FUNCIONAIS:

3.1 CADASTRO DE FORNECEDORES:

O aplicativo deve permitir o cadastro de fornecedores, incluindo informações detalhadas, como nome, endereço, contato e categorias de produtos fornecidos. Essas informações serão exibidas utilizando componentes como <Text>, <TextInput>, e <Image>.

3.2 LISTAGEM DE FORNECEDORES:

Deve ser possível visualizar uma lista de fornecedores cadastrados, com opções de pesquisa e filtragem com base em critérios como categoria ou localização. A lista de fornecedores será exibida utilizando componentes como <Text> e <Image>.

4 ASSOCIAÇÃO DE IMAGENS:

O aplicativo deve permitir a associação de imagens aos perfis dos fornecedores. Os usuários devem poder fazer o upload de logotipos ou fotos relacionadas ao fornecedor, utilizando o componente <Image>.

5 EXPERIÊNCIA DE USUÁRIO INTUITIVA:

A interface do aplicativo deve ser intuitiva e fácil de usar, garantindo que os usuários possam navegar, adicionar e editar informações de forma eficiente. Isso será alcançado usando componentes como <Text>, <TextInput>, e <Image>.

6 ROTEIRO DE PRÁTICA

6.1 MATERIAL NECESSÁRIO PARA A PRÁTICA

-Editor de texto ou IDE sendo opções sugeridas: Notepad++, Nano Editor, VS Code;

Navegador Web:

Google Chrome, Firefox, MS Edge, Safari ou Opera.

- Procedimentos Vamos passar por alguns componentes que serão necessários para o desenvolvimento do App para a empresa Meeting

Criando um Componente Simples:

Abra o seu editor de código.

Crie um novo arquivo JavaScript (ou JSX) para o seu componente.

Copie e cole o código a seguir para criar um componente que exibe um nome dentro de um elemento <Text>:

```
import React from 'react';
import { Text } from 'react-native';
const Cat = () => {
  const name = 'Maru';
  return <Text>Hello, I am {name}!</Text>;
};
export default Cat;
```

Este código cria um componente React Native simples.

Adicionando Elementos ao Componente:

Copie e cole o código a seguir para adicionar um elemento <TextInput> e alinhá-lo com o <Text> dentro de um <View>:

```
import React from 'react';
import { Text, TextInput, View } from 'react-native';
const Cat = () => {
  return (
    <View>
      <Text>Hello, I am...</Text>
      <TextInput
        style={{{
          height: 40,
          borderColor: 'gray',
          borderWidth: 1,
        }}}
        defaultValue="Name me!"
      />
    </View>
  );
};
export default Cat;
```

Utilizando Props:

Copie e cole o código a seguir para criar um componente que aceita props (propriedades) e renderiza o nome de acordo com as props:

```
import React from 'react';
import { Text, View } from 'react-native';
const Cat = (props) => {
  return (
    <View>
      <Text>Hello, I am {props.name}!</Text>
    </View>
  );
};
const Cafe = () => {
  return (
    <View>
      <Cat name="Maru" />
      <Cat name="Jellylorum" />
      <Cat name="Spot" />
    </View>
  );
};
export default Cafe;
```

Agora, você aprendeu como passar props para componentes e personalizá-los.

Utilizando Imagens:

Copie e cole o código a seguir para criar um componente que utiliza a tag <Image> para exibir uma imagem:

```
import React from 'react';
import { Text, View, Image } from 'react-native';
const CatApp = () => {
  return (
    <View>
      <Image
        source={{ uri: 'https://reactnative.dev/docs/assets/p_cat1.png' }}
        style={{ width: 200, height: 200 }} />
      <Text>Hello, I am your cat!</Text>
    </View>
  );
};
export default CatApp;
```

Agora, você sabe como incorporar imagens em seu aplicativo React Native.

7 DESENVOLVENDO SEU APLICATIVO:

Após a conclusão das atividades e com base nos códigos e conhecimentos adquiridos, é hora de desenvolver o seu aplicativo, levando em consideração os requisitos apresentados na contextualização. Utilize as habilidades e os componentes do React Native que você aprendeu para criar um aplicativo eficaz que atenda às necessidades da empresa "Meeting". Certifique-se de incorporar os princípios do desenvolvimento React Native, como a criação de componentes, o uso de props e a exibição de imagens, para alcançar os objetivos definidos. Este é o momento de aplicar seus conhecimentos e criar uma presença sólida no mercado móvel para a empresa.

7.1 RESULTADOS ESPERADOS

Após seguir esses passos, você teve uma compreensão inicial dos princípios do React Native, incluindo como criar componentes, utilizar props e adicionar imagens. Estas são as bases para a construção de aplicativos móveis eficazes e de alta qualidade. Com base nos códigos acima você verá os seguintes componentes que serão base para a construção do seu aplicativo.

8 RESULTADOS DA EXECUÇÃO E DOS CÓDIGOS:

8.1 BOTAO

```
≡ Botao.jsx ×
1 import React, { Component } from 'react';
2 import { StyleSheet, Text, TouchableOpacity } from 'react-native';
3
4 class CustomButton extends Component {
5   render() {
6     const { title, onPress, color = "#0175ee" } = this.props;
7     return (
8       <TouchableOpacity style={[styles.buttonImg, { backgroundColor: color }]} onPress={onPress}>
9         <Text style={styles.buttonText}>{title}</Text>
10        </TouchableOpacity>
11      );
12    }
13  }
14
15 const styles = StyleSheet.create({
16   buttonImg: {
17     padding: 12,
18     borderRadius: 12,
19     marginBottom: 8,
20     alignItems: "center",
21   },
22   buttonText: {
23     color: "#FFF",
24     fontSize: 16,
25     fontWeight: "bold",
26   },
27 });
28
29 export default CustomButton;
30
```

8.2 CADASTRO

```
≡ Cadastro.jsx ×

1 import * as ImagePicker from 'expo-image-picker';
2 import React, { Component } from 'react';
3 import { Alert, Image, StyleSheet, TextInput, View } from 'react-native';
4 import CustomButton from './Botao';
5
6 class CadastroFornecedor extends Component {
7   state = {
8     nome: '',
9     endereco: '',
10    contato: '',
11    categorias: '',
12    imagem: null,
13  };
14
15  handleCadastro = () => {
16    const { nome, endereco, contato, categorias, imagem } = this.state;
17    if (!nome || !endereco || !contato || !categorias) {
18      this.showAlert("Erro", "Por favor, preencher todos os campos.");
19      return;
20    }
21
22    const novoFornecedor = { nome, endereco, contato, categorias, imagem };
23    this.props.onCadastroSubmit(novoFornecedor);
24    this.clearFields();
25    this.showAlert("Sucesso", "Fornecedor cadastrado com sucesso!");
26  };
27
28  pickImage = async () => {
29    let result = await ImagePicker.launchImageLibraryAsync({
30      mediaTypes: ImagePicker.MediaTypeOptions.Images,
31      allowsEditing: true,
32      aspect: [4, 3],
33      quality: 1,
34    });
35
36    if (!result.canceled) {
37      this.setState({
38        imagem:
39          result.assets && result.assets.length > 0
40            ? result.assets[0].uri
41            : null,
42      });
43  };
44
```

```
≡ Cadastro.jsx ×

40     ? result.assets[0].uri
41     : null,
42   );
43 }
44 };
45
46 clearFields = () => {
47   this.setState({
48     nome: '',
49     endereco: '',
50     contato: '',
51     categorias: '',
52     imagem: null,
53   });
54 };
55
56 showAlert = (title, message) => {
57   Alert.alert(title, message);
58 };
59
60 render() {
61   const { nome, endereco, contato, categorias, imagem } = this.state;
62   return (
63     <View style={styles.container}>
64       <TextInput
65         style={styles.input}
66         placeholder="Nome Completo"
67         value={nome}
68         onChangeText={(text) => this.setState({ nome: text })}
69       />
70       <TextInput
71         style={styles.input}
72         placeholder="Endereço"
73         value={endereco}
74         onChangeText={(text) => this.setState({ endereco: text })}
75       />
76       <TextInput
77         style={styles.input}
78         placeholder="Contato"
79         value={contato}
80         onChangeText={(text) => this.setState({ contato: text })}
81     />
```

```
  Cadastro.jsx ×
79      value={contato}
80      onChangeText={(text) => this.setState({ contato: text })}
81    />
82    <TextInput
83      style={styles.input}
84      placeholder="Categoria/Produto"
85      value={categorias}
86      onChangeText={(text) => this.setState({ categorias: text })}
87    />
88
89    {imagem && <Image source={{ uri: imagem }} style={styles.image} />}
90
91    <View style={styles.buttonContainer}>
92      <CustomButton title="Escolher Imagem" onPress={this.pickImage} />
93      <CustomButton
94        title="Cadastrar"
95        color="#03992e"
96        onPress={this.handleCadastro}
97      />
98    </View>
99  </View>
100 );
101 }
102 }
103
104 const styles = StyleSheet.create({
105   container: {
106     flex: 1,
107     justifyContent: "center",
108     alignItems: "center",
109     paddingHorizontal: 20,
110   },
111   input: {
112     width: 350,
113     fontSize: 16,
114     marginBottom: 16,
115     paddingHorizontal: 12,
116     paddingVertical: 10,
117     backgroundColor: "#ffffff",
118     shadowColor: "#000",
119     shadowOffset: {
120       width: 0
```

```
120     width: 0,
121     height: 2,
122   },
123   shadowOpacity: 0.25,
124   shadowRadius: 3.84,
125   elevation: 5,
126   borderRadius: 12,
127 },
128 image: {
129   width: 160,
130   height: 160,
131   marginTop: 12,
132   borderRadius: 100,
133   marginBottom: 12,
134 },
135 buttonContainer: {
136   flexDirection: "row",
137   justifyContent: "space-between",
138   width: 350,
139 },
140 });
141
142 export default CadastroFornecedor;
143
```

8.3 FORNECEDORES

```
≡ Fornecedores.jsx ×

1 import React, { useMemo, useState } from 'react';
2 import { Alert, Image, StyleSheet, Text, TextInput, TouchableOpacity, View } from 'react-native';
3
4 const useFiltrarFornecedores = (fornecedores, categoriaFiltro) => {
5   const fornecedoresFiltrados = useMemo(
6     () =>
7       fornecedores.filter(
8         (fornecedor) =>
9           categoriaFiltro === "" ||
10           fornecedor.categorias.includes(categoriaFiltro)
11       ),
12       [fornecedores, categoriaFiltro]
13   );
14   return fornecedoresFiltrados;
15 };
16
17 const ListaFornecedores = ({ fornecedores, onRemove }) => {
18   const [categoriaFiltro, setCategoriaFiltro] = useState("");
19
20   const fornecedoresFiltrados = useFiltrarFornecedores(
21     fornecedores,
22     categoriaFiltro
23   );
24
25   const handleCategoriaFiltroChange = (text) => {
26     setCategoriaFiltro(text);
27   };
28
29   const handleDelete = (fornecedor) => {
30     Alert.alert(
31       "Confirmar exclusão",
32       "Deseja realmente excluir este fornecedor?",
33       [
34         { text: "Cancelar", style: "cancel" },
35         { text: "Excluir", onPress: () => onRemove(fornecedor.id) },
36       ]
37     );
38   };
39
40   return (
41     <View style={styles.container}>
42       <TextInput
```

```
  Fornecedores.jsx ×
40  return (
41    <View style={styles.container}>
42      <TextInput|
43        style={styles.input}
44        placeholder="Pesquisar por Categoria"
45        value={categoriaFiltro}
46        onChangeText={handleCategoriaFiltroChange}
47      />
48
49      {fornecedoresFiltrados.map((fornecedor, index) => (
50        <View key={index} style={styles.fornecedorContainer}>
51          <Image
52            source={{ uri: fornecedor.imagem }}
53            style={styles.imagemFornecedor}
54          />
55          <View style={styles.infoContainer}>
56            <Text style={styles.nomeFornecedor}>{fornecedor.nome}</Text>
57            <Text style={styles.detalheFornecedor}>
58              Endereço: {fornecedor.endereco}
59            </Text>
60            <Text style={styles.detalheFornecedor}>
61              Contato: {fornecedor.contato}
62            </Text>
63            <Text style={styles.detalheFornecedor}>
64              Categoria: {fornecedor.categorias}
65            </Text>
66            <TouchableOpacity
67              style={styles.actionButton}
68              onPress={() => handleDelete(fornecedor)}
69            >
70              <Text style={styles.actionText}>Excluir</Text>
71            </TouchableOpacity>
72          </View>
73        </View>
74      )));
75    </View>
76  );
77};
78
79 const styles = StyleSheet.create({
80   container: {
81     flex: 1
```

≡ Fornecedores.jsx ×

```
79  const styles = StyleSheet.create({
80    container: {
81      flex: 1,
82      paddingHorizontal: 16,
83      backgroundColor: "#dae8e7",
84    },
85    input: {
86      fontSize: 16,
87      padding: 16,
88      marginBottom: 12,
89      borderWidth: 1,
90      borderColor: "#CCC",
91      borderRadius: 25,
92      backgroundColor: "#FFF",
93    },
94    fornecedorContainer: {
95      flexDirection: "row",
96      marginBottom: 16,
97      padding: 16,
98      backgroundColor: "#FFF",
99      borderWidth: 1,
100     borderColor: "#DDD",
101     borderRadius: 12,
102     shadowColor: "#000",
103     shadowOffset: {
104       width: 0,
105       height: 2,
106     },
107     shadowOpacity: 0.25,
108     shadowRadius: 3.84,
109     elevation: 5,
110     alignItems: "center",
111   },
112   infoContainer: {
113     flex: 1,
114     marginLeft: 12,
115   },
116   nomeFornecedor: {
117     fontSize: 20,
118     fontWeight: "bold",
119     color: "#333",
120   }
```

```
120 },  
121     detalheFornecedor: {  
122         fontSize: 16,  
123         color: "#666",  
124     },  
125     imagemFornecedor: {  
126         width: 100,  
127         height: 100,  
128         borderRadius: 50,  
129     },  
130     actionBar: {  
131         padding: 8,  
132         width: 70,  
133         marginTop: 8,  
134         backgroundColor: "#DDD",  
135         borderRadius: 10,  
136     },  
137     actionText: {  
138         color: "#f52525",  
139     },  
140 },  
141 );  
142  
143 export default ListaFornecedores;  
144
```

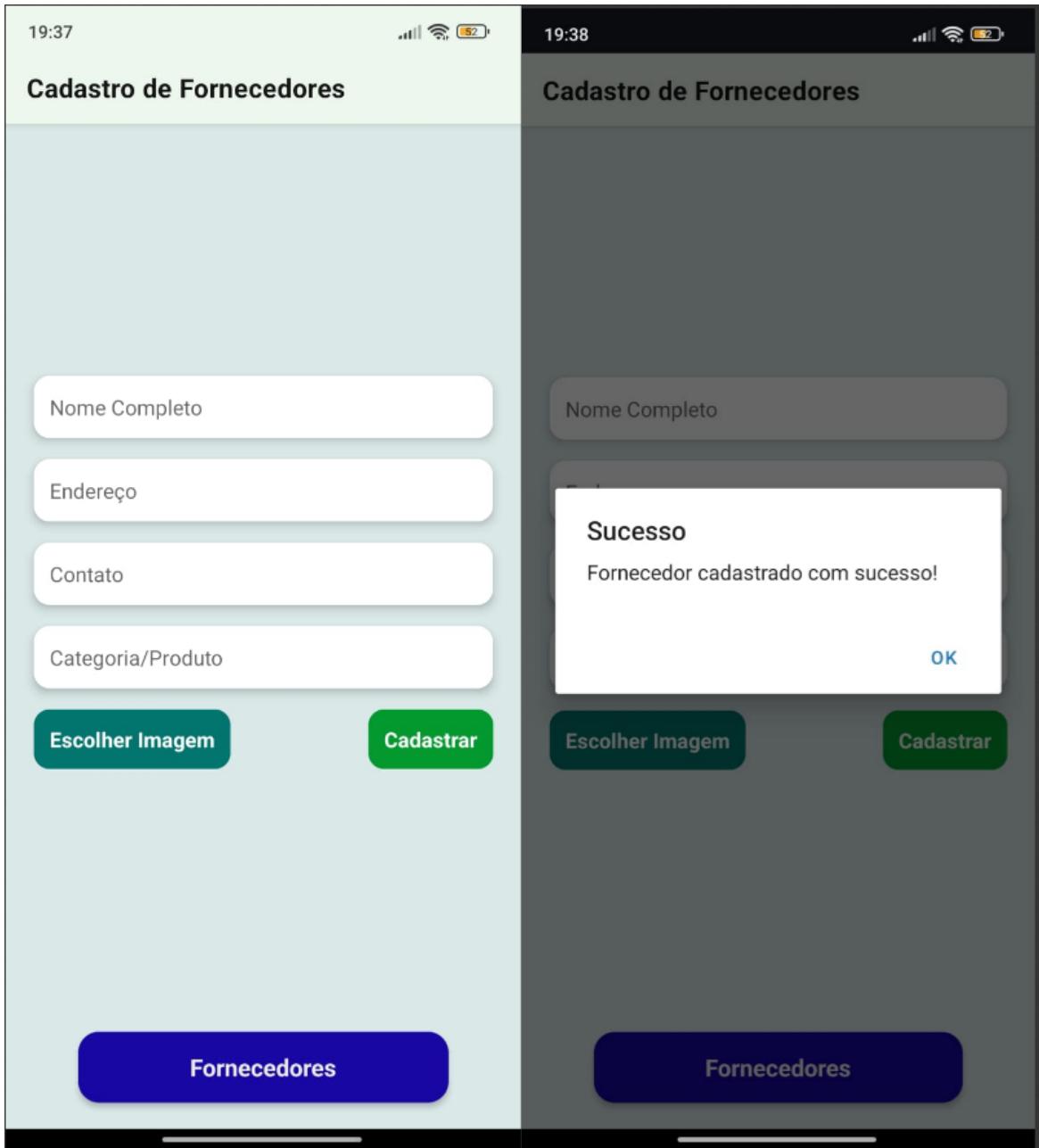
8.4 APP

```
≡ App.jsx ×

1 import React, { useState } from 'react';
2 import { StyleSheet, View, ScrollView, Text, TouchableOpacity } from 'react-native';
3 import { NavigationContainer } from '@react-navigation/native';
4 import { createNativeStackNavigator } from '@react-navigation/native-stack';
5 import CadastroFornecedor from './components/Cadastro';
6 import ListaFornecedores from './components/Fornecedores';
7
8 const Stack = createNativeStackNavigator();
9
10 export default function App() {
11     const [fornecedores, setFornecedores] = useState([]);
12
13     const handleCadastroSubmit = (novoFornecedor) => {
14         setFornecedores([...fornecedores, novoFornecedor]);
15     };
16
17     const handleRemove = (fornecedorId) => {
18         setFornecedores(currentFornecedores => currentFornecedores.filter(f => f.id !== fornecedorId));
19     };
20
21     const handleEdit = (fornecedorId, updatedData) => {
22         setFornecedores(currentFornecedores => currentFornecedores.map(f => {
23             if (f.id === fornecedorId) {
24                 return { ...f, ...updatedData };
25             }
26             return f;
27         }));
28     };
29
30     return (
31         <NavigationContainer>
32             <Stack.Navigator initialRouteName="Cadastro" screenOptions={{
33                 headerStyle: {
34                     backgroundColor: '#edf7ee',
35                 },
36                 headerTintColor: '#141414',
37                 headerTitleStyle: {
38                     fontWeight: 'bold',
39                 }
40             }}>
41                 <Stack.Screen name="Cadastro" options={{ title: 'Cadastro de Fornecedores' }}>
42                     {props => (
```

```
App.jsx x
40    });
41    <Stack.Screen name="Cadastro" options={{ title: 'Cadastro de Fornecedores' }}>
42        {props => [
43            <ScrollView contentContainerStyle={styles.scroll}>
44                <View style={styles.section}>
45                    <CadastroFornecedor onCadastroSubmit={handleCadastroSubmit} {...props} />
46                    <TouchableOpacity style={styles.button} onPress={() => props.navigation.navigate('Fornecedores')}>
47                        <Text style={styles.buttonText}>Fornecedores</Text>
48                    </TouchableOpacity>
49                </View>
50            </ScrollView>
51        )}
52    </Stack.Screen>
53    <Stack.Screen name="Fornecedores" options={{ title: 'Lista de Fornecedores' }}>
54        {() => <ListaFornecedores fornecedores={fornecedores} onRemove={handleRemove} onEdit={handleEdit} />}
55    </Stack.Screen>
56    </Stack.Navigator>
57 </NavigationContainer>
58 );
59 }
60
61 const styles = StyleSheet.create({
62     scroll: {
63         flexGrow: 1,
64         justifyContent: "center",
65     },
66     section: {
67         flex: 1,
68         alignItems: "center",
69         justifyContent: "center",
70         padding: 20,
71         backgroundColor: "#dae8e7"
72     },
73     button: {
74         marginTop: 20,
75         backgroundColor: "#1907a3",
76         borderRadius: 16,
77         width: '80%',
78         paddingVertical: 15,
79         paddingHorizontal: 20,
80         alignItems: 'center',
81         justifyContent: 'center',
82         justifyContent: 'center',
83         elevation: 3,
84         shadowColor: '#000',
85         shadowOffset: { width: 0, height: 2 },
86         shadowOpacity: 0.1,
87         shadowRadius: 1.5,
88     },
89     buttonText: {
90         color: '#ebebeb',
91         fontSize: 18,
92         fontWeight: 'bold'
93     }
94 });
95
```

8.5 RESULTADOS



19:39

Pesquisar por Categoria

Bruno Sampaio Bastos
Endereço: Rio de janeiro
Contato: 24991800999
Categoria: Programação
[Excluir](#)


Sucesso
Fornecedor cadastrado com sucesso!

Bruno Sampaio Bastos
Endereço: Rio de janeiro
Contato: 24099000876
Categoria: Eletromecânica
[Excluir](#)

19:39

Pesquisar por Categoria

Bruno Sampaio Bastos
Endereço: Rio de janeiro
Contato: 24991800999
Categoria: Programação
[Excluir](#)

Confirmar exclusão
Deseja realmente excluir este fornecedor?
[CANCELAR](#) [EXCLUIR](#)