Guia de Componentes Básicos: React Native

Exemplos minimalistas e funcionais prontos para uso em aplicações independentes (Android & iOS).

1. View

O contêiner básico de interface que organiza outros componentes através de Flexbox e aplica estilizações.

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

export default function App() {
  return (
    <View style={styles.container}>
      <View style={styles.box} />
    </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, justifyContent: 'center', alignItems: 'center' },
  box: { width: 100, height: 100, backgroundColor: 'skyblue' },
});
```

2. Text

Componente fundamental para renderização de textos e rótulos na tela.

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

export default function App() {
  return (
    <View style={styles.container}>
      <Text style={styles.text}>Olá, React Native!</Text>
    </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, justifyContent: 'center', alignItems: 'center' },
  text: { fontSize: 20, fontWeight: 'bold' },
});
```

3. Image

Exibe imagens locais ou remotas (requer especificação explícita de largura e altura).

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

export default function App() {
  return (
    <View style={styles.container}>
      <Image
        style={styles.image}
        source={{ uri: 'https://reactnative.dev/img/tiny_logo.png' }}
      />
    </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, justifyContent: 'center', alignItems: 'center' },
  image: { width: 100, height: 100 },
});
```

4. TextInput

Permite a entrada de texto pelo usuário via teclado virtual.

```
import React, { useState } from 'react';
import { View, TextInput, StyleSheet } from 'react-native';

export default function App() {
  const [text, setText] = useState('');

  return (
    <View style={styles.container}>
      <TextInput
        style={styles.input}
        placeholder="Digite algo..."
        value={text}
        onChangeText={setText}
      />
    </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, justifyContent: 'center', padding: 20 },
  input: { height: 40, borderColor: 'gray', borderWidth: 1, paddingHorizontal: 10,
borderRadius: 5 },
});
```

5. Button

Elemento básico de botão nativo para interações de clique simples.

```
import React from 'react';
import { View, Button, Alert, StyleSheet } from 'react-native';

export default function App() {
  return (
    <View style={styles.container}>
      <Button title="Clique Aqui" onPress={() => Alert.alert('Botão Pressionado!')} />
    </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, justifyContent: 'center', alignItems: 'center' },
});
```

6. ScrollView

Contêiner de rolagem genérico que abriga múltiplos componentes verticais ou horizontais.

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

export default function App() {
  return (
    <ScrollView contentContainerStyle={styles.container}>
      <Text style={styles.item}>Item 1</Text>
      <Text style={styles.item}>Item 2</Text>
      <Text style={styles.item}>Item 3</Text>
    </ScrollView>
  );
}

const styles = StyleSheet.create({
  container: { paddingVertical: 50, alignItems: 'center' },
  item: { fontSize: 24, marginVertical: 20 },
});
```

7. FlatList & SectionList

Listas performáticas otimizadas para renderizar apenas os elementos visíveis na tela.

FlatList (Lista Simples)

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

const DATA = [{ id: '1', title: 'Primeiro Item' }, { id: '2', title: 'Segundo Item' }];

export default function App() {
  return (
    <View style={styles.container}>
      <FlatList
        data={DATA}
        keyExtractor={(item) => item.id}
        renderItem={({ item }) => <Text style={styles.item}>{item.title}</Text>}
      />
    </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, paddingTop: 50 },
  item: { padding: 20, fontSize: 18, borderBottomWidth: 1, borderBottomColor: '#ccc' },
});
```

SectionList (Lista Agrupada)

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

const SECTIONS = [{ title: 'Frutas', data: ['Maçã', 'Banana'] }];

export default function App() {
  return (
    <View style={styles.container}>
      <SectionList
        sections={SECTIONS}
        keyExtractor={(item, index) => item + index}
        renderSectionHeader={({ section: { title } }) => <Text style={styles.header}>{title}
</Text>}
        renderItem={({ item }) => <Text style={styles.item}>{item}</Text>}
      />
    </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, paddingTop: 50 },
  header: { fontSize: 20, fontWeight: 'bold', backgroundColor: '#eee', padding: 10 },
  item: { padding: 15, fontSize: 16 },
});
```

8. TouchableOpacity & TouchableHighlight

Componentes para capturar toques com feedback visual configurável.

TouchableOpacity

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

export default function App() {
  return (
    <View style={styles.container}>
      <TouchableOpacity style=
{styles.button} activeOpacity={0.7}>
        <Text style={styles.text}>Toque
Aqui</Text>
      </TouchableOpacity>
    </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, justifyContent:
'center', alignItems: 'center' },
  button: { backgroundColor: '#007AFF',
padding: 15, borderRadius: 8 },
  text: { color: '#fff', fontWeight: 'bold'
},
});
```

TouchableHighlight

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

export default function App() {
  return (
    <View style={styles.container}>
      <TouchableHighlight style=
{styles.button} underlayColor="#0056b3">
        <Text style={styles.text}>Toque
Aqui</Text>
      </TouchableHighlight>
    </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, justifyContent:
'center', alignItems: 'center' },
  button: { backgroundColor: '#007AFF',
padding: 15, borderRadius: 8 },
  text: { color: '#fff', fontWeight: 'bold'
},
});
```

9. Switch

Entrada de dados booleana para alternar entre ligado/desligado (toggle).

```
import React, { useState } from 'react';
import { View, Switch, StyleSheet } from 'react-native';

export default function App() {
  const [isEnabled, setIsEnabled] = useState(false);

  return (
    <View style={styles.container}>
      <Switch value={isEnabled} onValueChange={setIsEnabled} />
    </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, justifyContent: 'center', alignItems: 'center' },
});
```

10. StyleSheet

Abstração de estilo semelhante ao CSS otimizada para React Native.

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

export default function App() {
  return (
    <View style={styles.container}>
      <Text style={styles.text}>Estilizado via StyleSheet</Text>
    </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, backgroundColor: '#f0f0f0', justifyContent: 'center', alignItems:
'center' },
  text: { fontSize: 18, color: 'darkblue', fontWeight: '600' },
});
```