

# **Agenda de Contatos**

Aplicativo com React Native + SQLite

*Inclui boas práticas, modularização e explicações técnicas.*

## Explicações Técnicas

1. Separação de responsabilidades em componentes reutilizáveis.
2. Validação de formulário (email, telefone) e máscara.
3. Atualização reativa com refreshTrigger e debounce em buscas.
4. Componentes com estilo moderno e responsivo.
5. Confirmações antes de excluir contatos.
6. Utilização do expo-sqlite para persistência local.
7. Modularização de helpers e funções utilitárias.
8. Design limpo e acessível com botões claros e feedbacks visuais.

## Trecho do componente ContactList.js

```
import React, { useEffect, useState, useRef } from "react";
import {
  View,
  Text,
  FlatList,
  TextInput,
  TouchableOpacity,
  StyleSheet,
  ActivityIndicator,
  Alert,
} from "react-native";
import { useSQLiteContext } from "expo-sqlite";
import { rs } from "../utils/helpers";

export default function ContactList({ onEdit, refreshTrigger }) {
  const db = useSQLiteContext();
  const [users, setUsers] = useState([]);
  const [search, setSearch] = useState("");
  const [loading, setLoading] = useState(false);
  const listRef = useRef(null);

  useEffect(() => {
    const timeout = setTimeout(() => {
      loadUsers();
    }, 300);
    return () => clearTimeout(timeout);
  }, [search]);

  useEffect(() => {
    loadUsers();
  }, [refreshTrigger]);

  async function loadUsers() {
    if (!db) return;
    setLoading(true);
    try {
      const result = await db.getAllAsync(
        `SELECT * FROM users WHERE firstName LIKE ? OR lastName LIKE ? OR email LIKE ? OR phone LIKE ? ORDER BY
        firstName`,
        [`%${search}%`, `%${search}%`, `%${search}%`, `%${search}%`]
      );
      setUsers(result);
    } catch (error) {
      Alert.alert("Erro", "Erro ao carregar contatos");
    } finally {
      setLoading(false);
    }
  }

  function confirmDelete(user) {
    Alert.alert(
      "Confirmação",
      `Deseja excluir o contato '${user.firstName} ${user.lastName}'?`,
      [
        { text: "Cancelar", style: "cancel" },
        {
          text: "Excluir",
          style: "destructive",
        }
      ]
    );
  }
}
```

```
        onPress: () => deleteUser(user.id),  
      },  
    ]  
  ]
```