Instalar o pacote do SQLite
- comando: npx expo install expo-sqlite

Arquivo App.js

```
import React, { useState, useEffect } from 'react';
import { View, Text, TextInput, Button, FlatList, StyleSheet } from 'react-native';
import * as SQLite from 'expo-sqlite';

const db = SQLite.openDatabase('tarefas.db');

export default function App() {
  const [tarefa, setTarefa] = useState("")
  const [tarefas, setTarefas] = useState([])
  const [idTarefaMod, setIdTarefaMod] = useState(null)

  // Incializar os métodos
  // de criação da tabela
  // e de carregamento das tabelas cadastradas
  useEffect(()=>{
    criarTabela()
    listaTarefas()
  },[])

  // Criação da tabela
  const criarTabela = () => {
    db.transaction((tx) => {
      tx.executeSql(
        'CREATE TABLE IF NOT EXISTS tarefaas(id INTEGER PRIMARY KEY
AUTOINCREMENT, nome TEXT);',
        [],
        (_, result) => { console.log("Tabela criada com sucesso") },
        (_, error) => { console.error("Erro ao criar a tabela", error) }
      )
    })
  }

  // Adicionar uma tarefa
  const adicionarTarefa = () => {
    if (tarefa.trim() === '') return;
    db.transaction((tx) => {
      //Para a tarefa modificada ou editada
      if (idTarefaMod) {
        tx.executeSql(
          'UPDATE tarefaas set nome = ? WHERE id = ?;',
          [tarefa, idTarefaMod],
          (_, result) => {
            if (result.rowsAffected > 0) {
```

```
            console.log("Tarefa atualizada com sucesso")
            setTarefa('')
            setIdTarefaMod(null)
            listaTarefas()
          } else {
            console.error("Tarefa não encontrada", error)
          }
        },
        (_, error) => { "Erro ao atualizar a tarefa", error }
      )
    } else {
      tx.executeSql(
        'INSERT INTO tarefaas (nome) VALUES(?);',
        [tarefa],
        (_, result) => {
          const idTarefa = result.insertId
          if (idTarefa) {
            console.log("Tarefa criada com sucesso")
            setTarefa('')
            listaTarefas()
          } else {
            console.error("Erro ao criar a tarefa", error)
          }
        },
        (_, error) => { console.error("Erro ao criar a tarefa", error) }
      )
    }
  })
}

// Listar as tarefas
const listaTarefas = () => {
  db.transaction((tx) => {
    tx.executeSql(
      'SELECT * FROM tarefaas;',
      [],
      (_, { rows }) => {
        setTarefas(rows._array)
      },
      (_, error) => { console.error("Erro ao listar as tarefas", error) }
    )
  })
}

// Excluir uma tarefa

const excluirTarefa = (id) => {
  db.transaction((tx) => {
```

```
    tx.executeSql(
      'DELETE FROM tarefaas WHERE id = ?;',
      [id],
      (_, result) => {
        if (result.rowsAffected > 0) {
          console.log("Tarefa excluída com sucesso")
          listaTarefas()
        } else {
          console.error("Tarefa não encontrada", error)
        }
      },
      (_, error) => { console.error("Erro ao excluir a tarefa", error) }
    )
  })
}

return (
  <View style={styles.container}>
    <Text style={styles.title}>Tarefas com SQLite</Text>
    <TextInput
      style={styles.input}
      placeholder="Digite o nome da tarefa"
      value={tarefa}
      onChangeText={(text) => setTarefa(text)}
    />
    <Button
      title={idTarefaMod ? 'Salvar Edição' : 'Adicionar Tarefa'}
      onPress={adicionarTarefa}
    />
    <FlatList
      data={tarefas}
      keyExtractor={(item) => item.id.toString()}
      renderItem={({ item }) => (
        <View style={styles.tarefaContainer}>
          <Text>{item.nome}</Text>
          <View style={styles.buttonsContainer}>
            <Button
              title="Editar"
              onPress={() => {
                setTarefa(item.nome);
                setIdTarefaMod(item.id);
              }}
            />
            <Button title="Excluir" onPress={() => excluirTarefa(item.id)} />
          </View>
        </View>
      )}
      style={styles.listaTarefas}
```

```
      />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
  },
  title: {
    fontSize: 20,
    fontWeight: 'bold',
    marginBottom: 20,
  },
  input: {
    borderWidth: 1,
    borderColor: 'gray',
    padding: 10,
    marginBottom: 10,
  },
  listaTarefas: {
    marginTop: 20,
  },
  tarefaContainer: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    paddingHorizontal: 10,
  },
  buttonsContainer: {
    flexDirection: 'row',
  },
});
```