MINHA COMPRA

BRUNO CAROLINO LOPES

JHONATA JACKSON MONTEIRO MOTTA

KAIO HENRIK DE PAULA SILVA

MERIANE DOS REIS DIAS

PEDRO HENRIQUE DA SILVA GÓIS

THIAGO ROBERTO DE SOUZA

Minha Compra

APLICAÇÃO MÓVEL – LISTA DE COMPRAS PARA CONTROLE FINANCEIRO

Eixo 3 – Projeto: Desenvolvimento de uma Aplicação Móvel em um Ambiente de Negócio

Orientador: Mateus Curcino de Lima

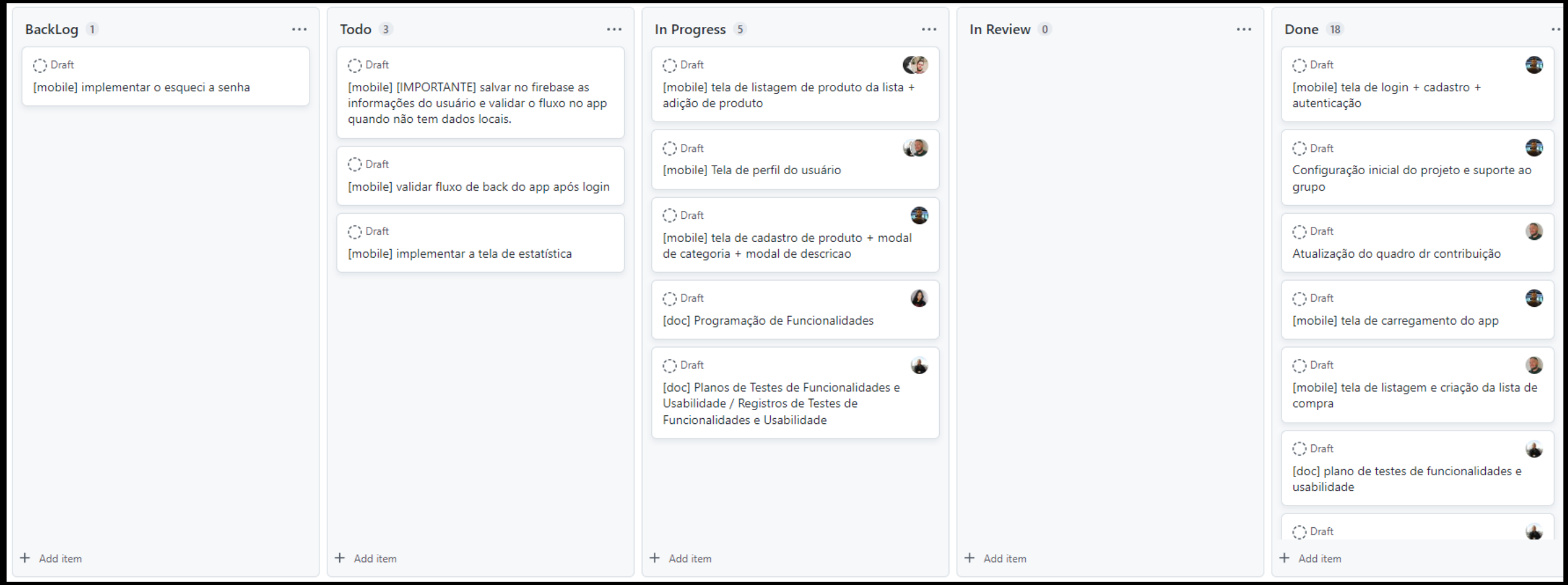
PUC-MINAS – Análise e Desenvolvimento de Sistemas – 3ºP

Dezembro/2022

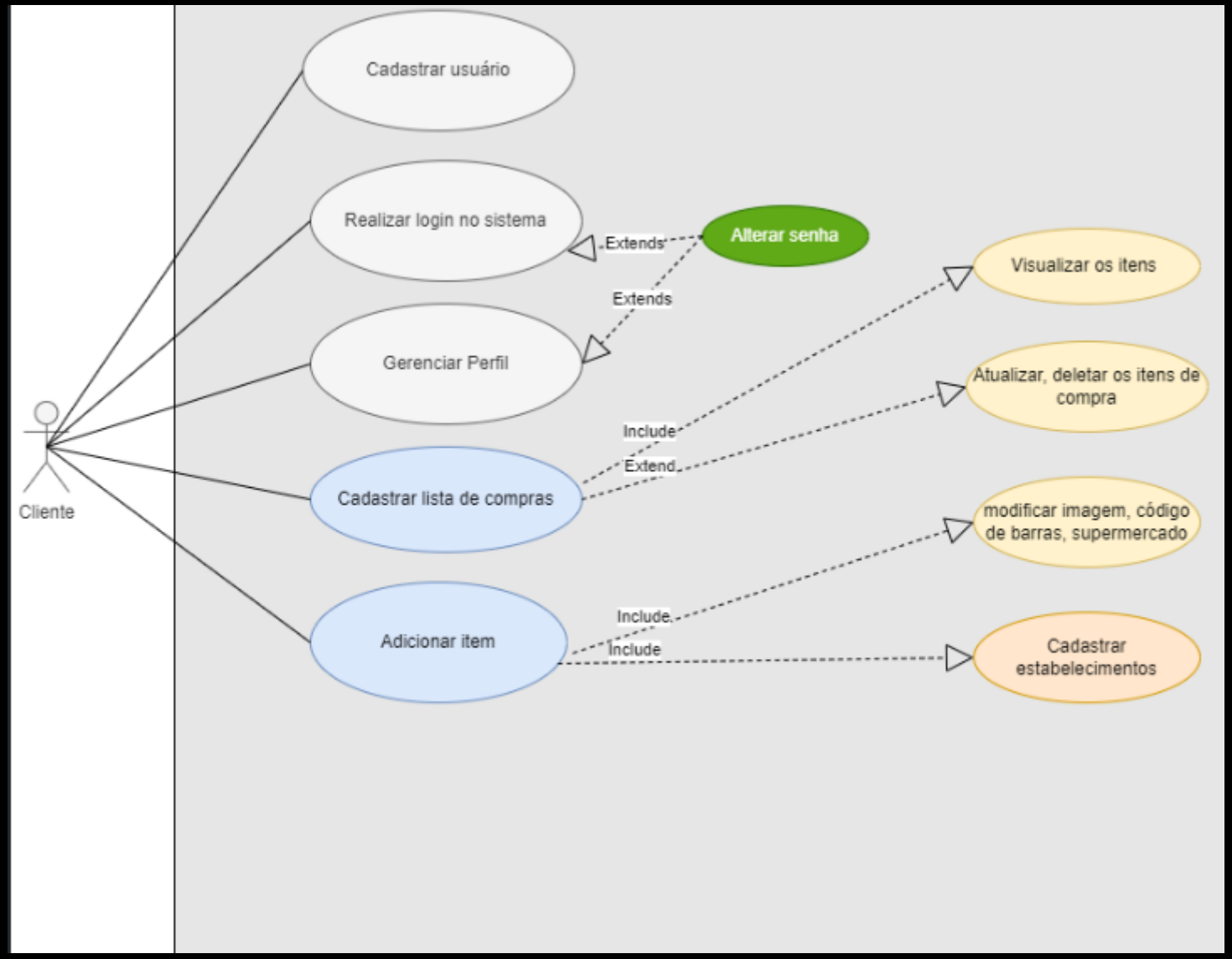
O PROBLEMA

Qual é o problema?	• Falta de planejamento na hora de realizar as compras do mês
Quem tem este problema?	• Todas as pessoas que realizam compras e têm acesso a um smartphone
Por que este problema deve ser resolvido?	• Para que as pessoas possam obter melhor organização pessoal e controle financeiro na hora de planejar e realizar compras.
Como será resolvido?	• Criar um aplicativo de cadastro de listas de compras para controle

PLANEJAMENTO



ESPECIFICAÇÃO

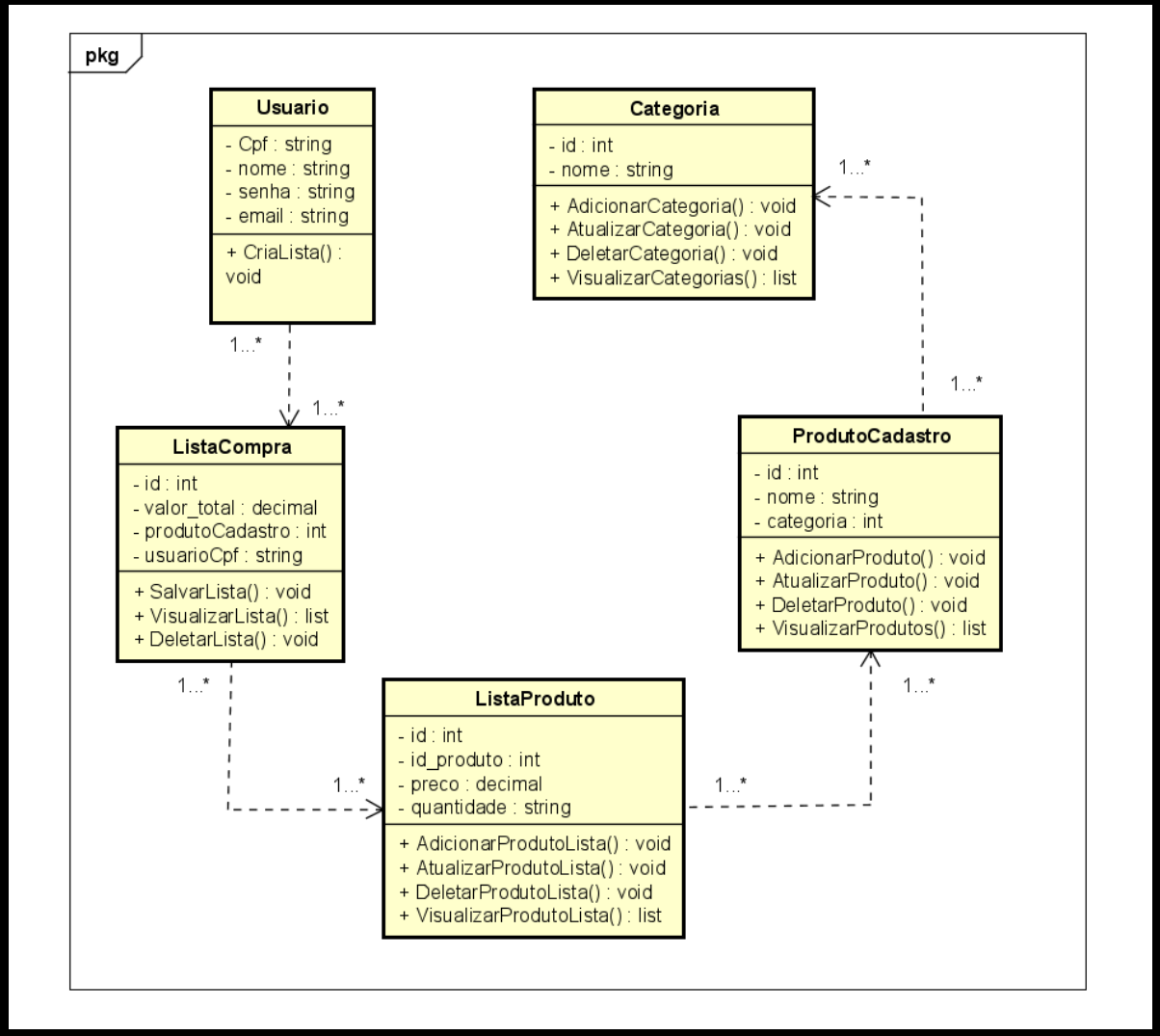


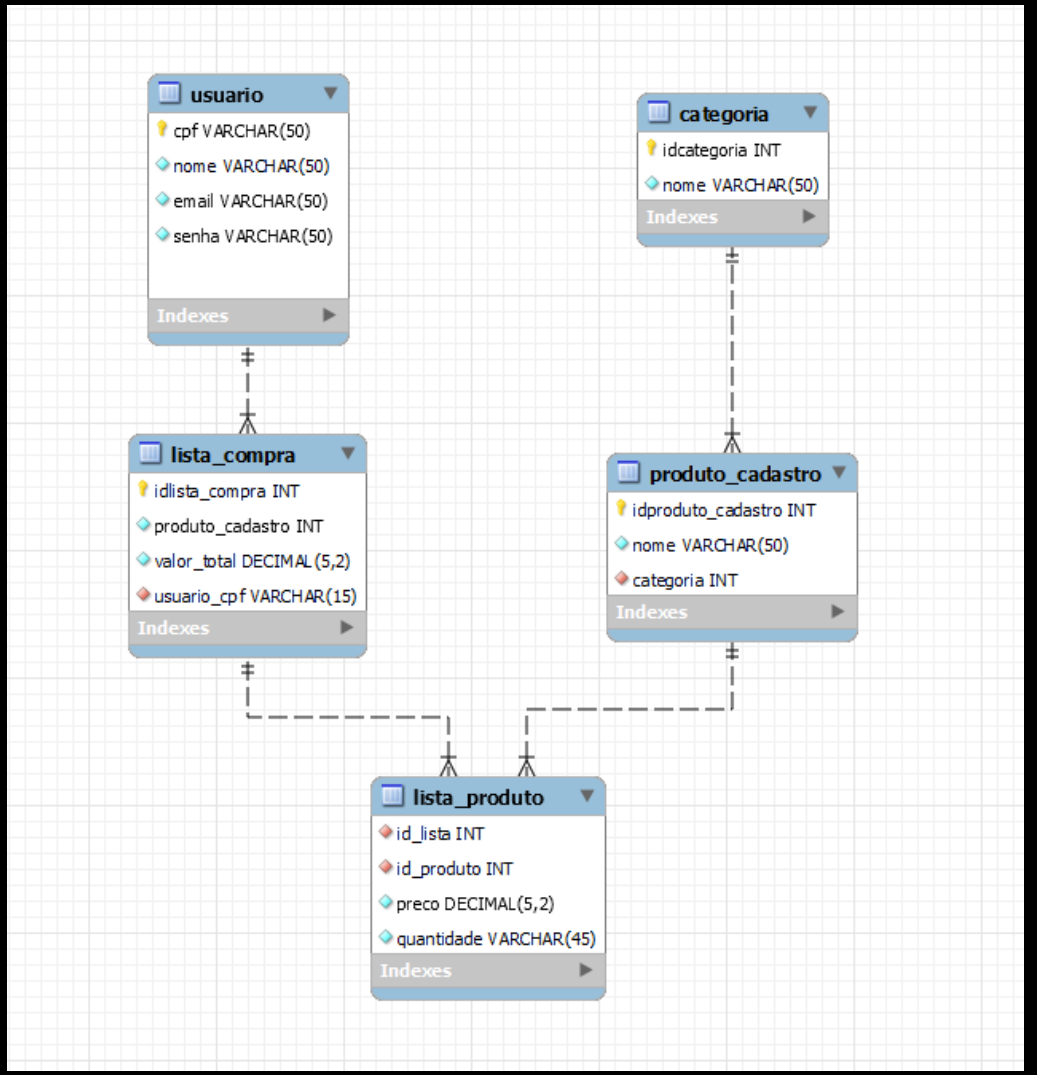
PERSONAS E HISTÓRIAS DE USUÁRIOS

BEATRIZ SILVA		
Informações Pessoais	*Idade*: 34	*Ocupação*: Assistente administrativo
Aplicativos	Zoom	Buscapé
Motivações	Economizar tempo	Economizar Dinheiro
Frustrações	Pouco tempo para a família e hobbies	Produtos caros
História	Sócia de um pequeno escritório que atua em processos de fusões de empresas	

ANTÔNIO AUGUSTO		
Informações Pessoais	*Idade*: 45	*Ocupação*: Vendedor autônomo
Aplicativos	Oferta Esperta	Mercado Livre
Motivações	Maior lucro na revenda de produtos	Economizar tempo
Frustrações	Perder tempo indo em diversos mercados	Baixo lucro em sua loja
História	Dono de uma pequena venda de bairro	

DESIGN


pkg
Usuario
- Cpf : string
- nome : string
- senha : string
- email : string
+ CriaLista() : void
Categoria
- id : int
- nome : string
+ AdicionarCategoria() : void
+ AtualizarCategoria() : void
+ DeletarCategoria() : void
+ VisualizarCategorias() : list
ListaCompra
- id : int
- valor_total : decimal
- produtoCadastro : int
- usuarioCpf : string
+ SalvarLista() : void
+ VisualizarLista() : list
+ DeletarLista() : void
ProdutoCadastro
- id : int
- nome : string
- categoria : int
+ AdicionarProduto() : void
+ AtualizarProduto() : void
+ DeletarProduto() : void
+ VisualizarProdutos() : list
ListaProduto
- id : int
- id_produto : int
- preco : decimal
- quantidade : string
+ AdicionarProdutoLista() : void
+ AtualizarProdutoLista() : void
+ DeletarProdutoLista() : void
+ VisualizarProdutoLista() : list
1...*


usuario
cpf VARCHAR(50)
nome VARCHAR(50)
email VARCHAR(50)
senha VARCHAR(50)
Indexes
categoria
idcategoria INT
nome VARCHAR(50)
Indexes
lista_compra
idlista_compra INT
produto_cadastro INT
valor_total DECIMAL(5,2)
usuario_cpf VARCHAR(15)
Indexes
produto_cadastro
idproduto_cadastro INT
nome VARCHAR(50)
categoria INT
Indexes
lista_produto
id_lista INT
id_produto INT
preco DECIMAL(5,2)
quantidade VARCHAR(45)
Indexes

BANCO DE DADOS

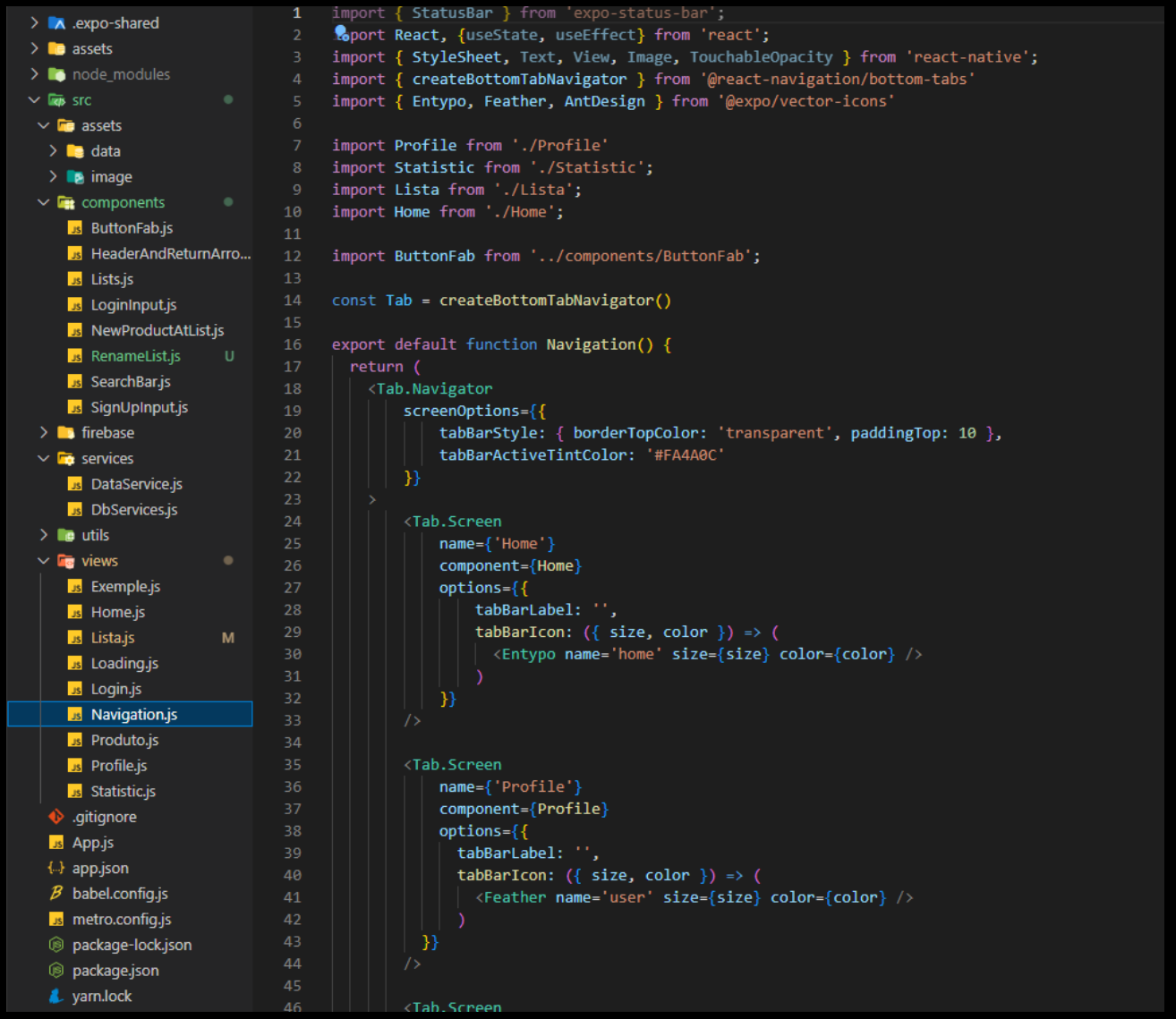
- Banco de dados: SQLite

```
import Database from "./DbServices";
import { defaultData } from "../assets/data/default_data";

const DB_EXEC = Database.getConnection();

export const criaProdutoCadastro = async (param) => {
  let results = await DB_EXEC(
    "insert into produto_cadastro(nome, categoria) values (?,?)",
    [param.nome, param.categoria]
  );
  return results.rowsAffected;
};

export const criaCategoria = async (param) => {
  let results = await DB_EXEC("insert into categoria(nome) values (?)", [
    param.nome,
  ]);
  return results.rowsAffected;
};

export const criaListaDeProduto = async (param) => {
  let results = await DB_EXEC(
    "insert into lista_produto(id_produto, preco, quantidade) values (?,?,?)",
    [param.id_produto, param.preco, param.quantidade]
  );
  return results.rowsAffected;
};

export const criaListaDeCompra = async (param) => {
  let results = await DB_EXEC(
    "insert into lista_compra(id_produto, valor_total, cpf, nome_lista) values (?,?,?,?)",
    [param.id_produto, param.valor_total, param.cpf, param.nome_lista]
  );
  return results.rows._array;
};

export const inserirUsuario = async (param) => {
  let results = await DB_EXEC(
    "insert into usuario(cpf, nome , email, senha) values (?,?,?,?)",
    [param.cpf, param.nome, param.email, param.senha]
  );
  return results.rowsAffected;
};
export const excluiTodosOsUsuarios = async () => {
  let results = await DB_EXEC(" DELETE from usuario");
  return results.rows._array;
};
```

```
import * as SQLite from "expo-sqlite";

export const Database = {
  getConnection: () => {
    const db = SQLite.openDatabase("minha_compra.db");

    db.transaction((tx) => {
      tx.executeSql(
        "create table if not exists usuario (cpf text primary key not null, nome text not null, email text not null, senha text not null);"
      );
      tx.executeSql(
        "create table if not exists produto_cadastro (id integer primary key not null, nome text not null, categoria text not null);"
      );
      tx.executeSql(
        "create table if not exists categoria (id integer primary key not null, nome text not null);"
      );
      tx.executeSql(
        "create table if not exists lista_produto (id integer primary key not null, id_produto integer not null, preco real not null, quantidade integer not null);"
      );
      tx.executeSql(
        "create table if not exists lista_compra (id integer primary key not null, id_produto integer not null, valor_total real not null, cpf integer not null, nome_lista text not null);"
      );
    });

    const ExecuteQuery = (sql, params = []) =>
      new Promise((resolve, reject) => {
        db.transaction((trans) => {
          trans.executeSql(
            sql,
            params,
            (trans, results) => {
              resolve(results);
            },
            (error) => {
              reject(error);
            }
          );
        });
      });

    return ExecuteQuery;
  },
};

export default Database;
```

TECNOLOGIAS

- React Native
- Firebase



TESTE - SOFTWARE

Registro de Testes de Software

Tabela de resumo com os resultados dos testes.

Caso de teste	Descrição	Resultado
CT-01	Cadastramento de usuário	Sucesso
CT-02	Consulta de dados	Sucesso
CT-03	Criação de uma nova lista	Sucesso
CT-04	Acessar lista existente	Sucesso
CT-05	Alterar lista existente	Sucesso
CT-06	Alterar dados do usuário	Sucesso

TESTE - USABILIDADE

Heurísticas	Severidade					T
	0	1	2	3	4	
Visibilidade do Estado do sistema						
Correspondência Sistema - Mundo Real						
Controle e Liberdade do Usuário						
Consistência e Padronização						
Reconhecimento em vez de memorização						
Prevenção de erros						
Flexibilidade e eficiência de uso						
Projeto estético e minimalista						
Ajudar os usuários com os erros						
Ajuda e documentação						
Total de problemas	0	0	0	0	0	0

0 - Sem Importância: não afeta a operação da interface para todos usuários, não sendo encarado necessariamente como um problema de usabilidade.

1- Cosmético: não necessita ser reparado, a menos que haja tempo disponível.

2 - Simples: pode ser reparado, com baixa prioridade de correção.

3 - Grave: deve ser reparado, com alta prioridade de correção.

4 - Catastrófico: deve ser reparado de qualquer forma antes do produto ser disponibilizado.

CONCLUSÕES

- Os requisitos principais propostos para o trabalho foram executados com sucesso.
- Para trabalhos futuros pretendemos implementar a tela de estatística.
- As atividades foram divididas entre os integrantes de acordo com as suas melhores habilidades para alcançarmos o objetivo.
- Apesar de conseguir entregar os requisitos, tivemos dificuldade em gerir o tempo e alguns membros tiveram dificuldades com as tecnologias utilizadas, porém conseguimos trabalhar em conjunto, e os que possuíam maior habilidade conseguiram passar o conhecimento. Demonstrando assim um trabalho em equipe.

OBRIGADO!