



**UnB**

Instituto de  
Ciências Exatas

Departamento de  
Ciência da Computação

## **Técnicas de Programação 2 - 2025/1**

**Professor Jan Mendonça Correa**

**Alan dos Santos (232007830), Arthur Henrique (232000481), Bruno Henrique (221022239), João Gabriel Machado (232003607), João Victor Prata (202028857), Luan Ribeiro (232036958) Lucas Hashimoto (231003504), Rafael Bonach (221008365)**

### **Relatório do projeto - Sistema de Gerência de compras e compartilhamento de preços**

Este documento tem como objetivo descrever sobre o desenvolvimento e funcionamento de um site de consulta de preços de produtos de mercados diferentes. Providencia um ambiente de fácil manejo, útil tanto para impulsionar as vendas dos mercados quanto facilitar o consumo de produtos pelos clientes.

### **Distribuição dos papéis para desenvolvimento:**

**Product Owner:** João Victor Prata

**Scrum Master:** Rafael Bonach

**Full Stack (1 integrante):** Bruno Henrique

**Front-end (3 integrantes):** Arthur Henrique, João Victor Prata, Lucas Hashimoto

**Back-end (4 integrantes):** Alan dos Santos, João Gabriel, Luan Ribeiro, Rafael Bonach

---

## Backlog do produto:

Este projeto considera três tipos diferentes de usuários, sendo necessário haver uma distinção quanto ao funcionamento do site a depender de qual tipo está navegando nele:

**Cliente:** busca e realiza compras dos produtos.

**Gerente:** pertencente a um mercado. Cria, lê, atualiza e deleta dados de produtos de seu mercado à lista de produtos no site.

**Administrador:** mesmas permissões que o gerente, mas para todos os mercados. Pode também criar, ler, atualizar e deletar dados de clientes e gerentes.

## Historietas de usuário:

### ÉPICO 1 - ACESSO E IDENTIFICAÇÃO DE USUÁRIOS:

- O cadastro de mercados é realizado pelo gerente através de uma tela de especificação. Após inseridos e aprovados os dados descritos, um novo produto é registrado no banco de dados para que possa ser buscado pelos clientes.
- O cadastro de clientes e gerentes é realizado através de formulários específicos. O administrador pode acessar, modificar ou excluir qualquer conta através de uma interface de gestão, para manter a integridade do sistema e remover contas maliciosas ou abandonadas.
- O login de clientes é feito por meio de autenticação com e-mail e senha previamente cadastrados. Uma vez autenticado, o cliente é redirecionado à sua área pessoal com listas, preferências e histórico de compras carregados automaticamente.
- O cadastro de novos clientes é realizado por meio de um formulário onde são preenchidos nome, e-mail e senha. Após o envio, os dados são

armazenados no banco de dados e o cliente será direcionado à página inicial.

### **Histórias do Épico 1:**

#### **(CRUD01)**

**COMO** administrador

**QUERO** cadastrar mercados

**PARA** apresentar produtos no site

#### **(CRUD02)**

**COMO** administrador

**QUERO** criar, ler, atualizar e deletar gerentes e clientes

**PARA** testar funcionamento do site, e remover contas maliciosas ou abandonadas

#### **(CRUD03)**

**COMO** gerente

**QUERO** cadastrar meu mercado

**PARA** gerenciar os produtos na minha loja

#### **(CRUD04)**

**COMO** cliente

**QUERO** realizar login

**PARA** acessar dados previamente registrados

#### **(CRUD05)**

**COMO** cliente

**QUERO** realizar cadastro

**PARA** salvar minhas preferências e realizar compras

## ÉPICO 2 - GERENCIAMENTO DE PRODUTOS:

- Quando o administrador identificar produtos inconsistentes, maliciosos ou duplicados ele poderá apagar esses produtos para manter a base de dados limpa.
- O gerente acessa uma tela de cadastro de produto onde informa nome, preço e localização. Ao confirmar os dados, o produto é registrado no sistema e disponibilizado para clientes.
- A visualização de produtos é feita por clientes em páginas de listagem que exibe nome, preço e localização por mercado. O sistema permite aplicar filtros e ordenações por nome, preço ou distância para facilitar a comparação entre mercados.

### Histórias do Épico 2:

#### (CRUDP01)

**COMO** administrador

**QUERO** apagar produtos inconsistentes, maliciosos ou duplicados

**PARA** manter a base de dados limpa

#### (CRUDP02)

**COMO** gerente

**QUERO** cadastrar produtos com nome, preço e localização na loja

**PARA** que meus clientes encontrem facilmente os itens disponíveis

#### (CRUDP03)

**COMO** gerente

**QUERO** ver a lista de produtos disponíveis

**PARA** cadastrar em meu supermercado

#### **(CRUD04)**

**COMO** cliente

**QUERO** ver os produtos organizados por nome, preço e localização

**PARA** comparar ofertas e escolher o melhor custo-benefício

#### **(CRUD05)**

**COMO** cliente

**QUERO** uma tela para ver, atualizar e excluir os meus dados

**PARA** editar as minhas informações no site

#### **(CRUD06)**

**COMO** cliente

**QUERO** criar ou realizar login no site

**PARA** ter minhas informações salvas e realizar edição dos preços

### **ÉPICO 3 - CRIAÇÃO E EDIÇÃO DE LISTA DE COMPRAS:**

- A criação de uma nova lista de compras é iniciada por um botão dedicado. O cliente pode adicionar produtos manualmente ou por código de barras. A lista é salva no banco de dados com nome personalizado e pode ser reutilizada posteriormente.
- A edição de uma lista anterior é realizada ao selecionar uma lista salva. O cliente pode adicionar ou remover itens, alterar quantidades e salvar as modificações, que sobrescrevem a versão anterior no banco de dados.
- A adição de um produto por código de barras é feita com o acionamento da câmera. Ao escanear o código, o sistema busca o produto na base de dados e, se encontrado, adiciona o item à lista. Caso não exista, uma sugestão de cadastro é oferecida.

### Histórias do Épico 3:

#### (CRUDL01)

**COMO** cliente

**QUERO** criar uma nova lista de compras

**PARA** planejar minhas compras com antecedência

#### (CRUDL02)

**COMO** cliente

**QUERO** editar listas anteriores

**PARA** reaproveitá-las e ajustá-las às novas necessidades

#### (CRUDL03)

**COMO** cliente

**QUERO** adicionar produtos à lista

**PARA** agilizar o processo de montagem da lista

#### (CRUDL04)

**COMO** cliente

**QUERO** pesquisar um produto pelo nome

**PARA** adicioná-lo à minha lista de compras

### ÉPICO 4 - LOCALIZAÇÃO E COMPARAÇÃO DE PRODUTOS:

- O cliente poderá informar sua localização via GPS ou manualmente para ver os mercados mais próximos.
- O cliente ordenará os produtos por distância, preço (ou ambos) para encontrar as opções com base na sua situação.
- Será possível ao cliente saber quando o produto não estiver disponível no mercado selecionado, assim ele poderá comprar o mesmo item em outro mercado mais próximo.

## Histórias do Épico 4:

### (LOC01)

**COMO** cliente

**QUERO** informar minha localização (GPS ou manual)

**PARA** ver os mercados mais próximos

### (LOC02)

**COMO** cliente

**QUERO** ordenar os produtos por distância, preço ou ambos

**PARA** encontrar as melhores opções com base na minha situação

### (LOC03)

**COMO** cliente

**QUERO** saber quando um produto não estiver disponível no mercado selecionado

**PARA** que eu possa comprar o mesmo item em outro mercado próximo

## ÉPICO 5 - INCENTIVOS DE VERIFICAÇÃO:

- Produtos com todos os dados preenchidos serão destacados para impulsionar o alcance e as vendas
- Para o cliente, os preços marcados como “verificado” serão destacados para passar uma maior confiança sobre as informações de cada produto.

## Histórias do Épico 5:

### (DES01)

**COMO** gerente

**QUERO** que os produtos com todos os dados preenchidos sejam destacados

**PARA** impulsionar alcance e vendas

**(DES02)**

**COMO** cliente

**QUERO** que os preços marcados como “verificados” apareçam com destaque

**PARA** confiar nas informações ao decidir onde comprar

## **ÉPICO 6 - EXECUÇÃO DA COMPRA:**

- O cliente poderá conferir o valor final da compra conforme os itens forem sendo adicionados à sua lista de compras para ter um melhor controle de seus gastos.
- O cliente poderá verificar se o preço no caixa é o mesmo preço registrado no sistema do mercado, isso evitará cobranças incorretas acerca do produto.

### **Histórias do Épico 6:**

**(VAL01)**

**COMO** cliente

**QUERO** conferir o valor final da compra conforme os itens são adicionados

**PARA** controlar melhor meus gastos

**(VAL02)**

**COMO** cliente

**QUERO** definir o preço do produto no site a partir do valor que estiver na caixa

**PARA** evitar cobranças incorretas

### **Priorização das historietas:**

- Sprint 1:
  - **(CRUD01)**
  - **(CRUD02)**
  - **(CRUD03)**



- Sprint 2
    - (CRUD05)
    - (CRUD06)
    - (CRUDL02)
    - (CRUDL03)
    - (VAL02)
- 

## Estrutura do Banco de Dados:

### Tabela *users*:

**Propósito:** armazena as informações de todos que acessam o sistema.

#### **Campos Importantes:**

- *id*: identificador único para cada usuário.
- *Role*: define o tipo de usuário (cliente, gerente, admin), controlando suas permissões.

### Tabela *produtos*:

**Propósito:** funciona como um catálogo mestre de todos os produtos que podem existir, independentemente de onde são vendidos.

#### **Campos Importantes:**

- *codigo\_de\_barras*: identificador único e principal de um produto, garantindo que produtos como "Coca-Cola" e "coca cola" sejam tratados como o mesmo item.

### Tabela *supermercados*:

**Propósito:** guarda o cadastro de todos os estabelecimentos comerciais.

#### **Campos Importantes:**

- *id*: identificador único para cada mercado.

- *gerente\_id*: link para a tabela *users*, indicando quem é o gerente responsável por aquele mercado.

### **Tabela *listas\_de\_compra*:**

**Propósito:** armazena as listas criadas pelos usuários.

#### **Campos Importantes:**

- *usuario\_id*: link para a tabela *users*, indicando quem é o dono da lista.

### **Tabela *itens\_da\_lista*:**

**Propósito:** é uma tabela de "ligação". Ela conecta os produtos da tabela *produtos* com uma *lista\_de\_compra* específica.

#### **Campos Importantes:**

- *lista\_id* e *produto\_id*: juntos, formam uma chave única para garantir que um mesmo produto não seja adicionado duas vezes na mesma lista.

### **Tabela *registros\_de\_preco*:**

**Propósito:** é a tabela mais importante para a funcionalidade principal. Ela armazena um "evento": quando um usuário ou gerente registra o preço de um produto em um supermercado específico.

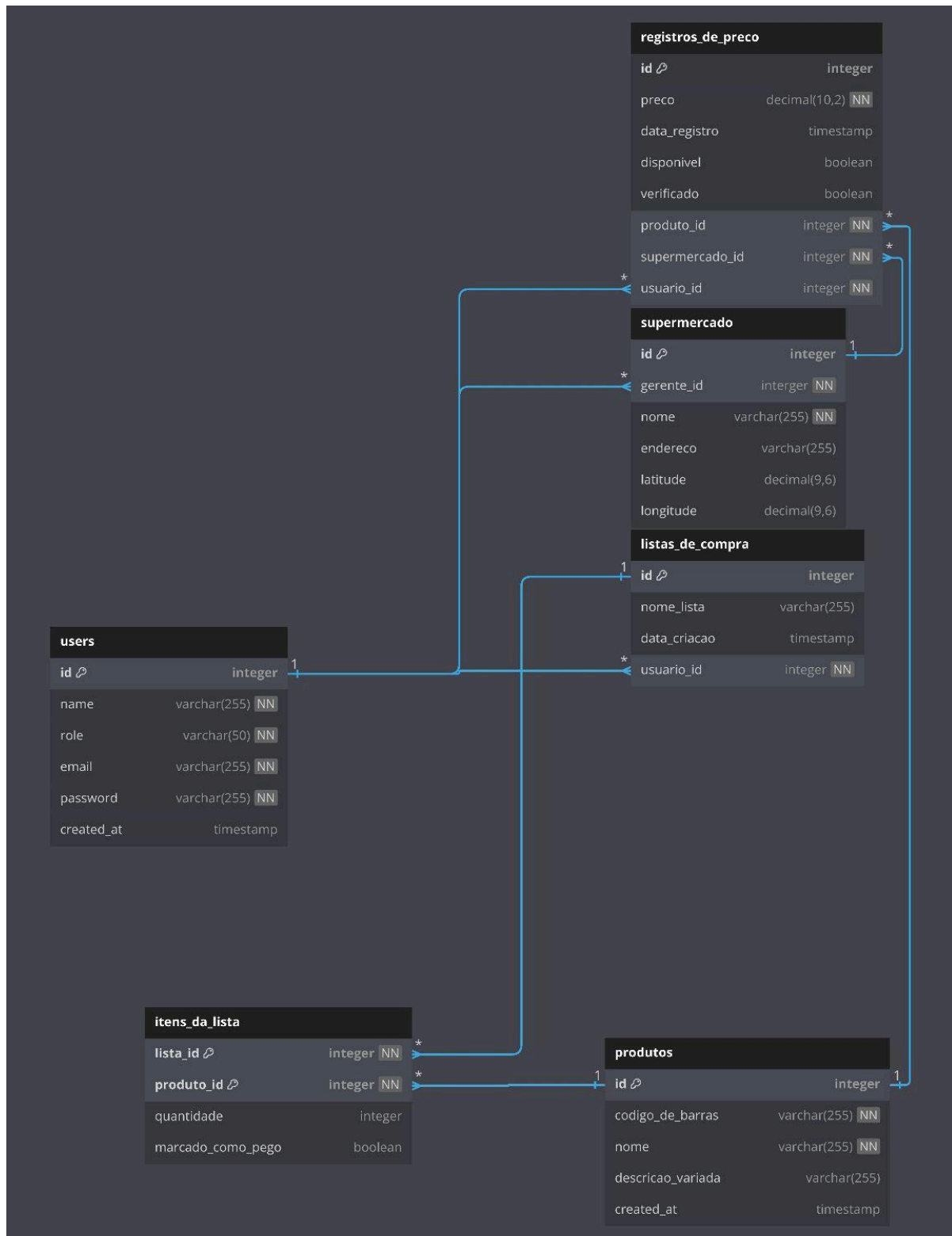
#### **Campos Importantes:**

- *produto\_id*, *supermercado\_id* e *usuario\_id*: conectam as três informações principais.
- *preco*: o valor do produto naquele momento.
- *disponivel*: informa se o produto está em estoque.
- *verificado*: indica se o preço foi confirmado.

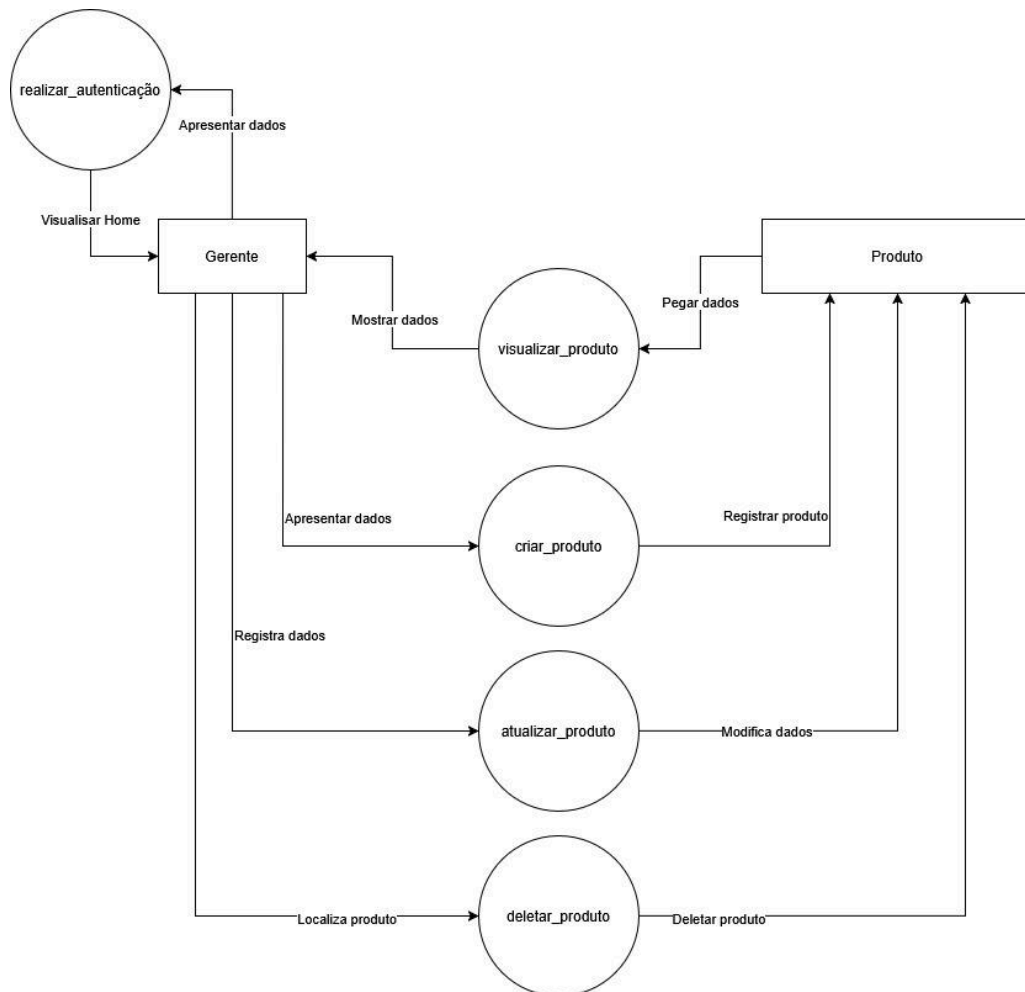
### **Como as Tabelas se Conectam:**

- Um **usuário** pode ter várias **listas de compra**.
  - Uma **lista de compra** pode ter vários **produtos**, e o mesmo **produto** pode estar em várias listas.
-

## Diagrama do Banco de Dados:



## Diagrama de Contexto:



---

## Linguagem de programação utilizada:

Para desenvolvimento do projeto, foi optado pela linguagem Javascript no Front-end e no Back-end. O SGBD utilizado para gerenciar os dados dos produtos, mercados e usuários foi o Postgres, compatível com Javascript usando a biblioteca Prisma (<https://www.prisma.io/>). O Docker também foi utilizado para criar um ambiente isolado e padronizado de desenvolvimento do projeto.

**Padrão de codificação utilizado:** para o Javascript foi utilizado o [Google JavaScript Style](#)

**Programa verificador para o padrão de codificação utilizado:** foi utilizado o ESLint como verificador (<https://eslint.org/docs/latest/>)

---

**Framework de teste:** foi utilizado o para desenvolvimento do Front-end o Jest (<https://jestjs.io/pt-BR/>).

**Bibliotecas e frameworks utilizados:** React para desenvolvimento Frontend (Javascript), Express para desenvolvimento Back End (Node.js)

**Verificador de código:** foi utilizado o ESLint (<https://eslint.org/docs/latest/>)

**Verificador de cobertura:-** foi utilizado o Jest (<https://jestjs.io/pt-BR/>)

**Site usado para controle de versões:** foi utilizado o GitHub tanto para controle de versões, quanto para a metodologia do Kanban.

**Ferramenta de documentação:** foi utilizado o JSDoc (<https://jsdoc.app/>)

**Link do repositório no GitHub:** <https://github.com/joviprata/projeto-tp2>