



Projeto final

Disciplina: Engenharia de Software

Equipe:

Gustavo Almeida Monteiro

Hermeson Bastos Maia

Victor Anthony Pereira Alves

Vitoria Ashiley Lopes Ferreira

Contexto

Cliente: Dona da loja Daisy Store, que é uma loja de venda de roupas com público predominantemente das cidades de Senador Pompeu e Fortaleza

Necessidade

A loja tem como principal plataforma de vendas atualmente o whatsapp, e deseja obter um melhor controle de estoque e um **catálogo online** para seus clientes.



Objetivos e metas:

- Facilitar a interação entre vendedor e cliente.
- Solucionar problemas de gerenciamento de estoque.
- Evidenciar informações sobre os produtos de forma íntegra.
- Desenvolver um sistema web que atenda as necessidades dos usuários.
- Construir um sistema que priorize a experiência do usuário cliente e o controle por parte do usuário administrador.



Elicitação de requisitos

Métodos:

- Entrevista semi-estruturada
- Brainstorming

Principais necessidades coletadas durante a elicitação

- Separar o foco da experiência, uma para o usuário e outra para os ADMs, sendo **Mobile-first para o cliente e desktop para os administradores**.
- Designer mais claro e objetivo, pensando na identidade visual e no **fácil reconhecimento dos produtos** e funcionalidades.
- Deve haver uma área para **controle de estoque**, onde o estoque e suas informações devem ser claras.
- O cliente deve ser **informado sobre a disponibilidade do produto que deseja**.

Exemplo de histórias

US001 - Adicionar produto ao estoque

Como administrador, desejo registrar a entrada de novos produtos no estoque, para manter um controle preciso dos itens disponíveis.

Conversa:

- O vendedor precisa evidenciar as informações dos produtos para o cliente, tornando o processo de compra mais prático, evitando conversas intermediárias, redundâncias e/ou informações precipitadas sobre o estoque.

Critérios de aceitação:

- Dado que o usuário é administrador, quando o usuário adicionar um novo produto ao estoque, então devem ser exibidos os campos para as seguintes informações: descrição, foto, tamanho, quantidade, status de disponibilidade e classificação do produto (filtros).
- Dado que o usuário é administrador, quando o usuário finalizar o cadastro de um produto, um resumo/preview das informações adicionadas deve exibido para que o mesmo finalize a ação ou retorne um passo para ajustar alguma informação.

Exemplo de histórias

US001 - Adicionar produto ao estoque

Como administrador, desejo registrar a entrada de novos produtos no estoque, para manter um controle preciso dos itens disponíveis.

Conversa:

- O vendedor precisa evidenciar as informações dos produtos para o cliente, tornando o processo de compra mais prático, evitando conversas intermediárias, redundâncias e/ou informações precipitadas sobre o estoque.

Critérios de aceitação:

- Dado que o usuário é administrador, quando o usuário adicionar um novo produto ao estoque, então devem ser exibidos os campos para as seguintes informações: descrição, foto, tamanho, quantidade, status de disponibilidade e classificação do produto (filtros).
- Dado que o usuário é administrador, quando o usuário finalizar o cadastro de um produto, um resumo/preview das informações adicionadas deve exibido para que o mesmo finalize a ação ou retorne um passo para ajustar alguma informação.

Exemplo de histórias

US006 - Visualizar detalhes dos produtos

Como usuário cliente, desejo visualizar as informações detalhadas de cada produto incluindo imagens, descrições, preços e disponibilidade, para tomar decisões de compra informadas.

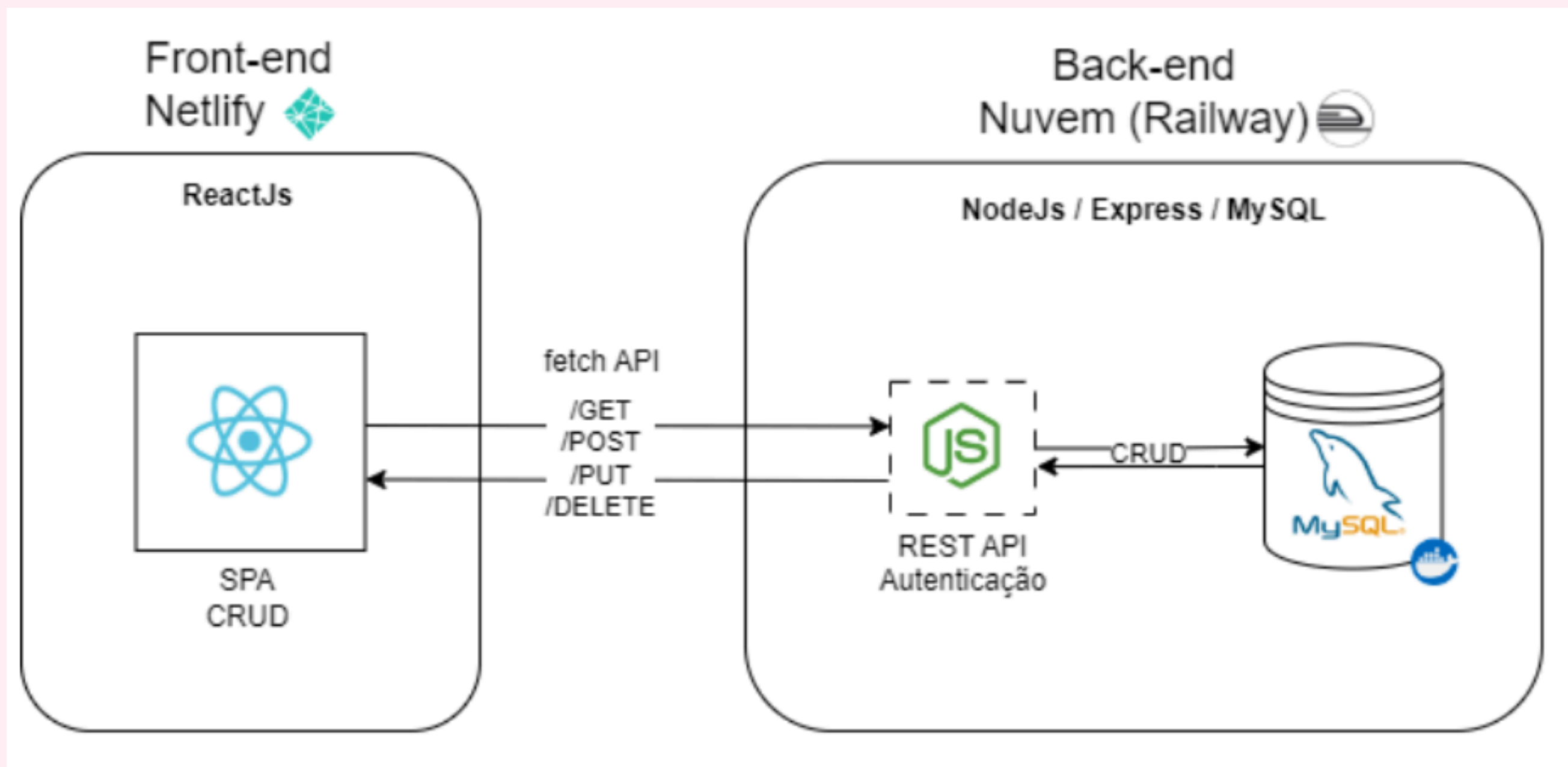
Conversa:

- Além de visualizar a listagem de produtos, o usuário deve também poder visualizar informações mais específicas antes de comprar o produto, como quantidade disponível, cores e outras informações que sejam de interesse do cliente.

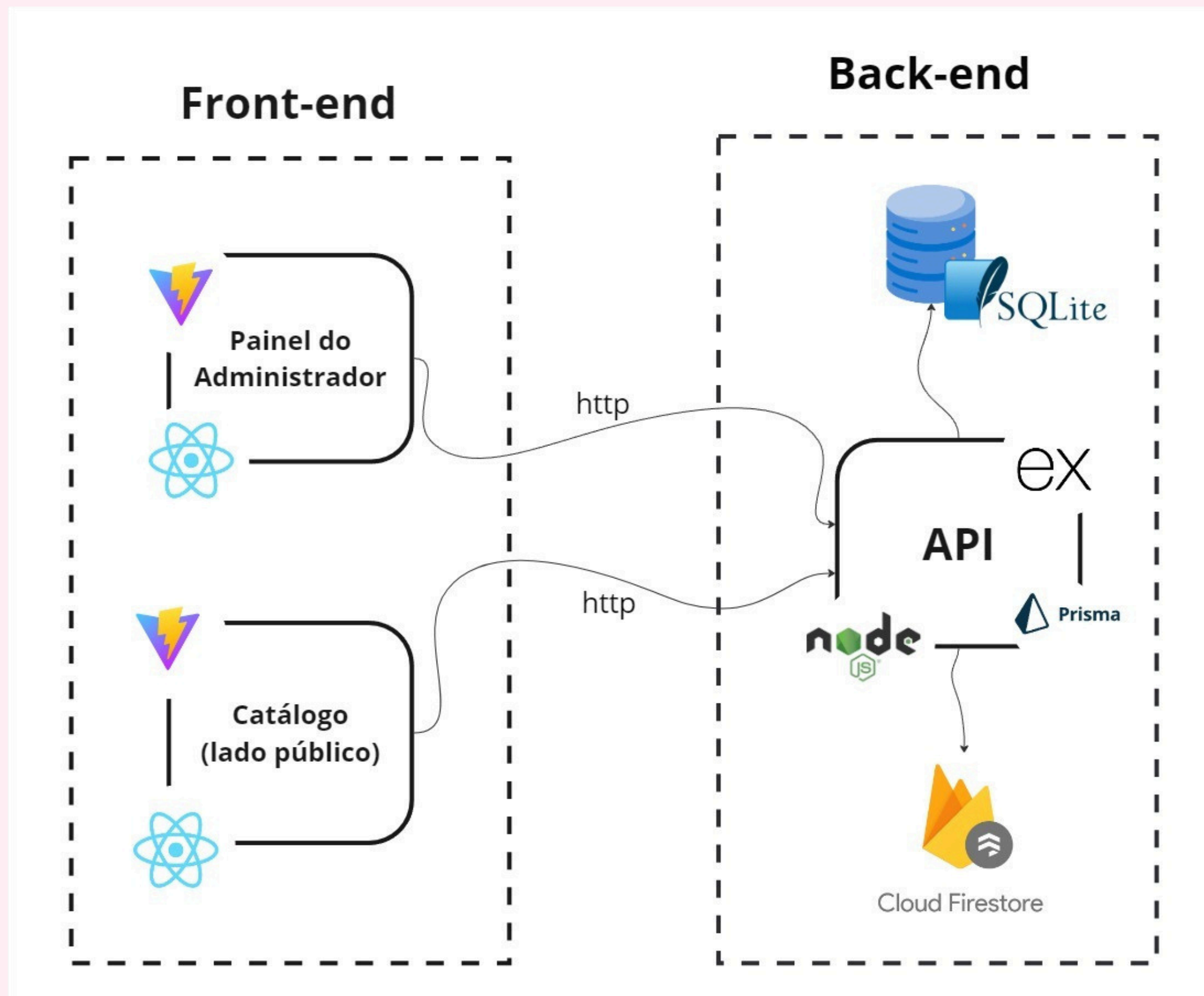
Critérios de aceitação:

- Dado que o usuário é um cliente, quando ele selecionar um produto listado, então o sistema deve exibir uma página completa com o objetivo de informar todas as informações daquele respectivo produto como fotos, tamanho, quantidade disponível, descrição e preço e a opção de adicionar o produto ao pedido.

Arquitetura – Proposta

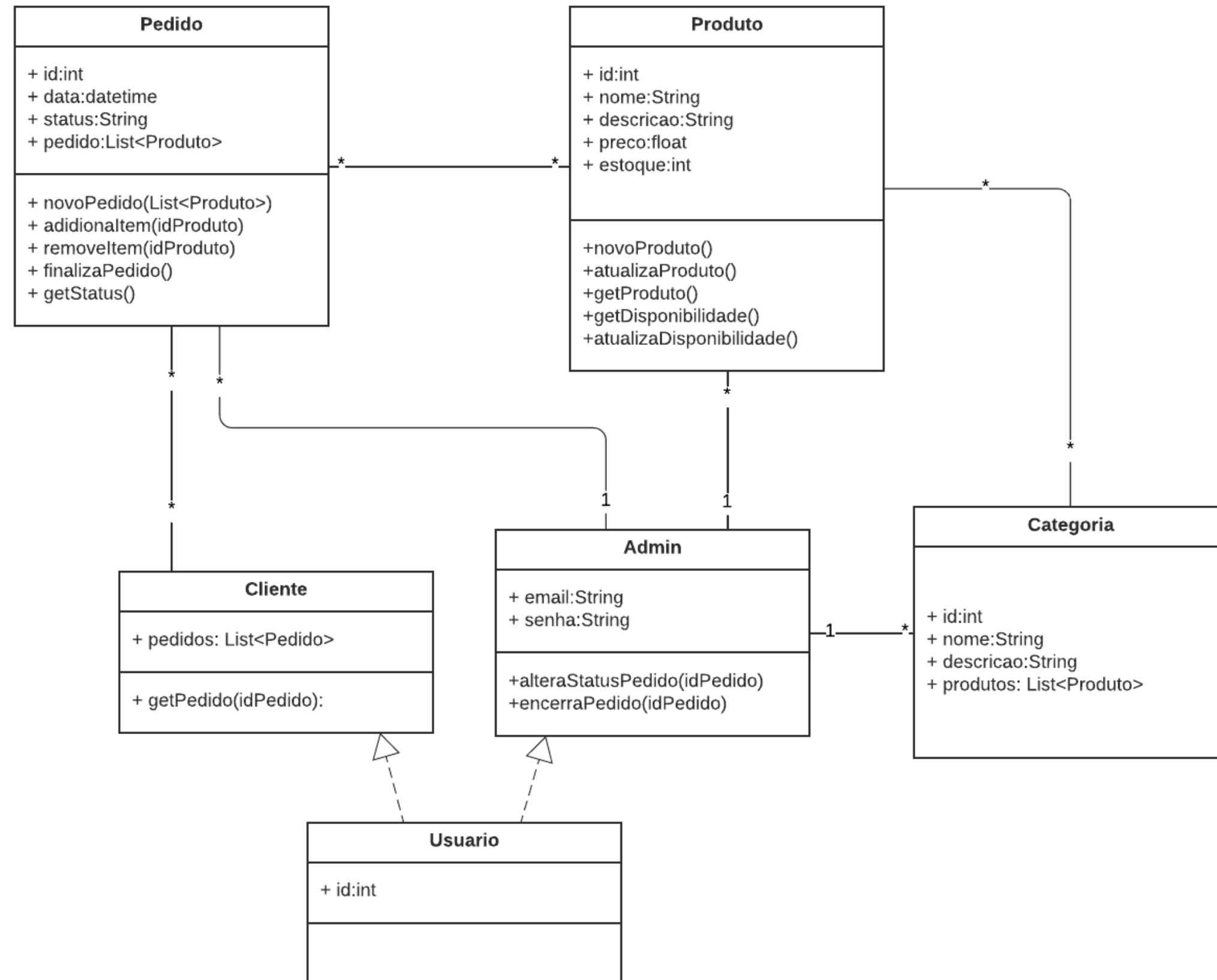


Arquitetura – Atual



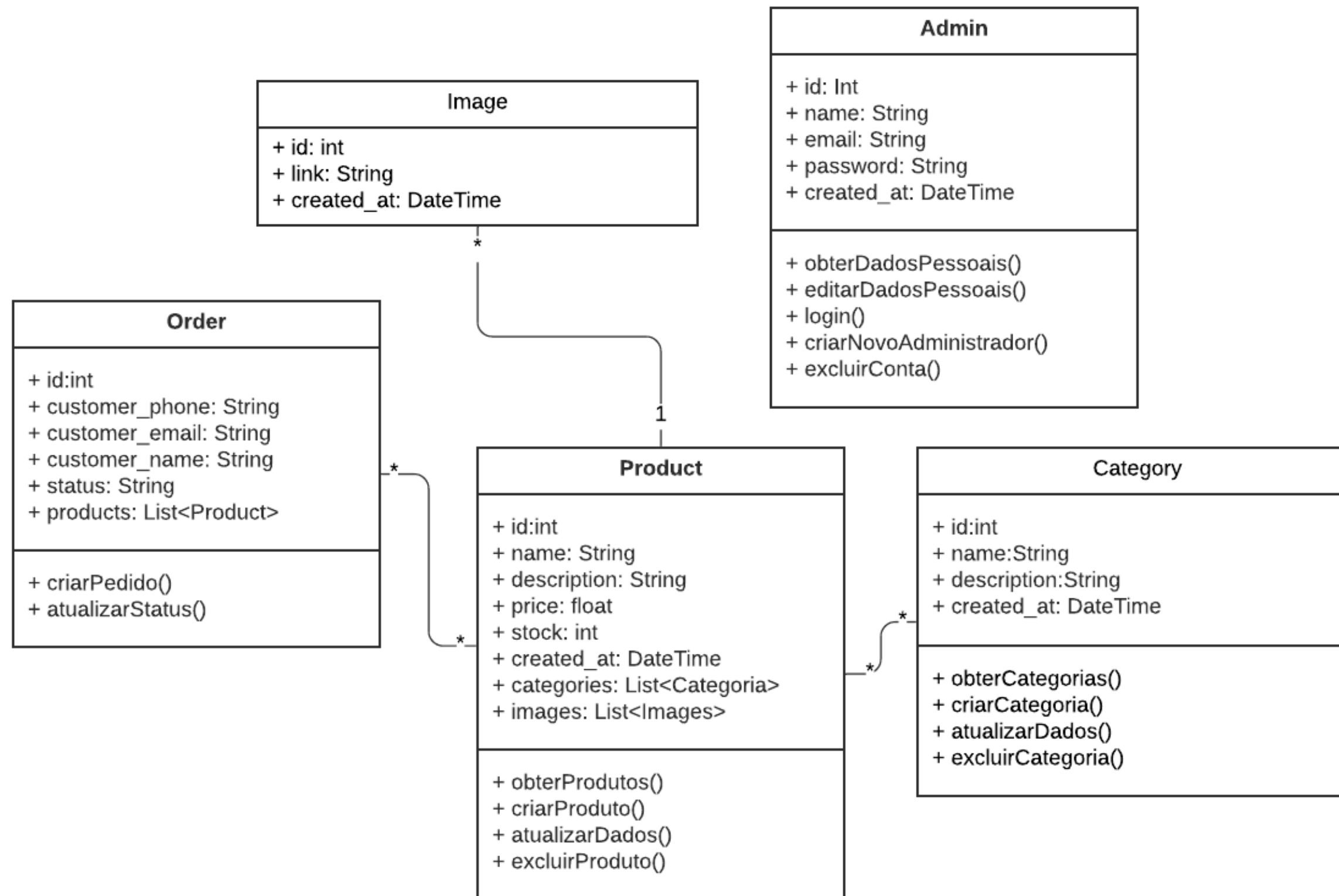
Modelagem

Diagrama de classes – Proposta

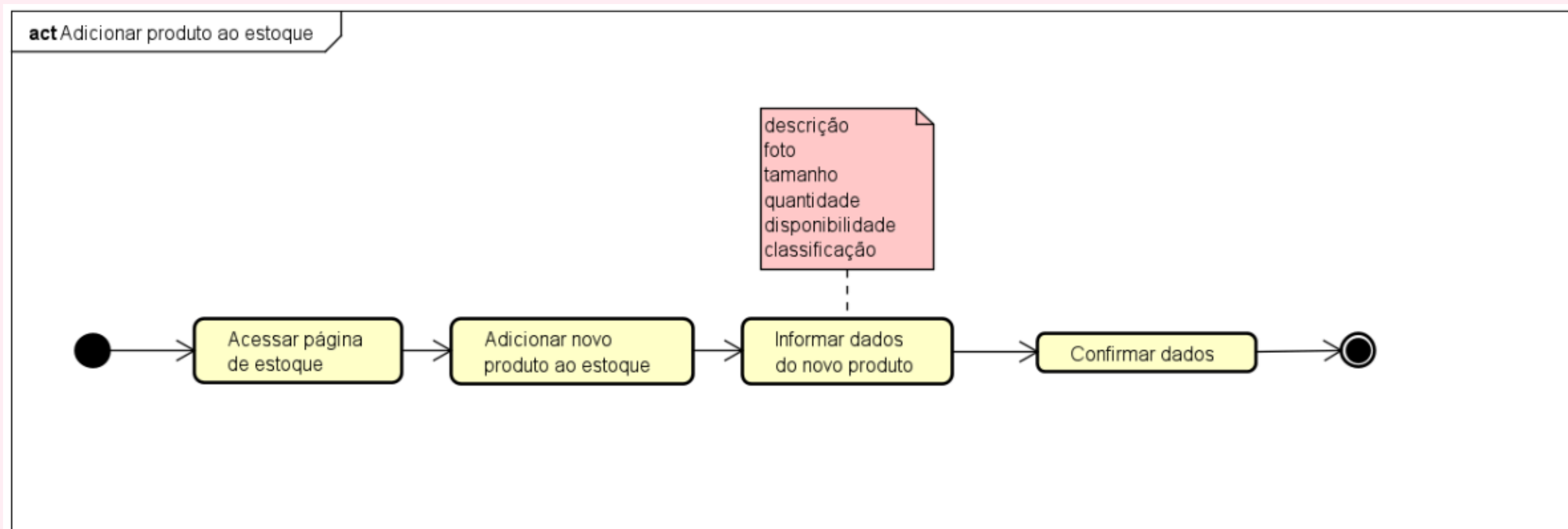


Modelagem

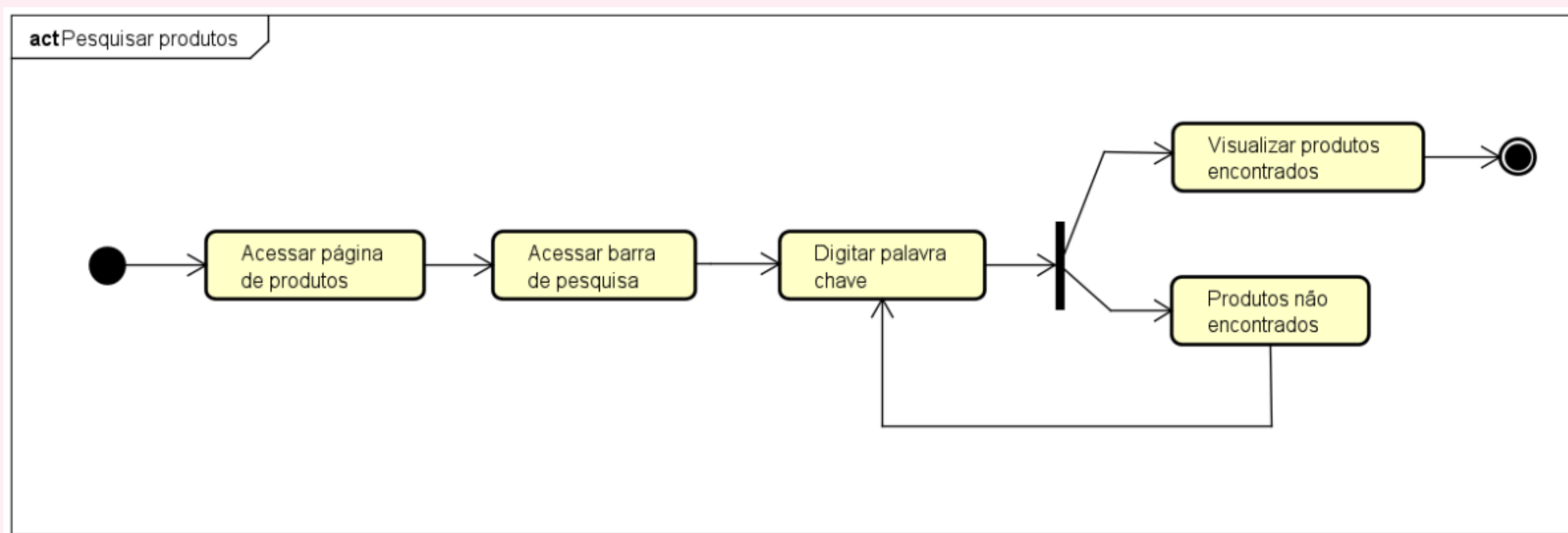
Diagrama de classes - Atual



Modelagem Diagrama de atividades



Modelagem Diagrama de atividades



Testes

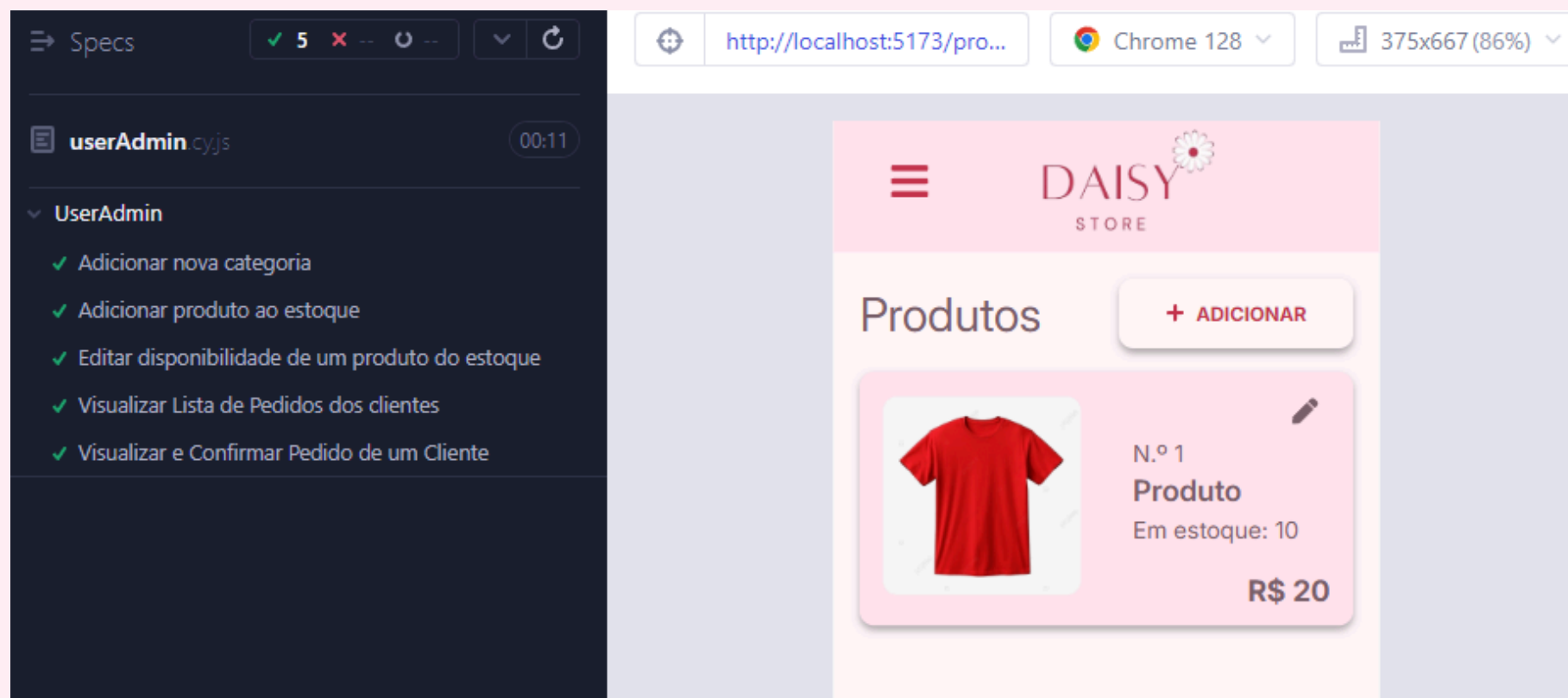


Foram implementados testes unitários com JestJS no backend para validar o comportamento dos Controllers

```
✓ ✓ adminController.test.js
  ✓ ✓ Admin Controller - getAdmins
    ✓ ✓ should return all admins successfully
    ✓ ✓ should handle errors when retrieving admins
✓ ✓ categoryController.test.js
  ✓ ✓ Category Controller
    ✓ ✓ getCategories
      ✓ ✓ should return all categories successfully
      ✓ ✓ should handle errors when retrieving categories
    ✓ ✓ createCategory
      ✓ ✓ should create a category successfully
      ✓ ✓ should handle errors when creating a category
    ✓ ✓ updateCategory
      ✓ ✓ should update a category successfully
      ✓ ✓ should handle not found error when updating a category
      ✓ ✓ should handle errors when updating a category
    ✓ ✓ deleteCategory
      ✓ ✓ should delete a category successfully
      ✓ ✓ should handle errors when deleting a category
> ✓ ✓ orderController.test.js
```



Implementados testes E2E com Cypress para validar as histórias



Produto Final

API: <https://github.com/hermesonbastos/api-daisy-store>

FRONT: <https://github.com/hermesonbastos/daisy-store-crm>