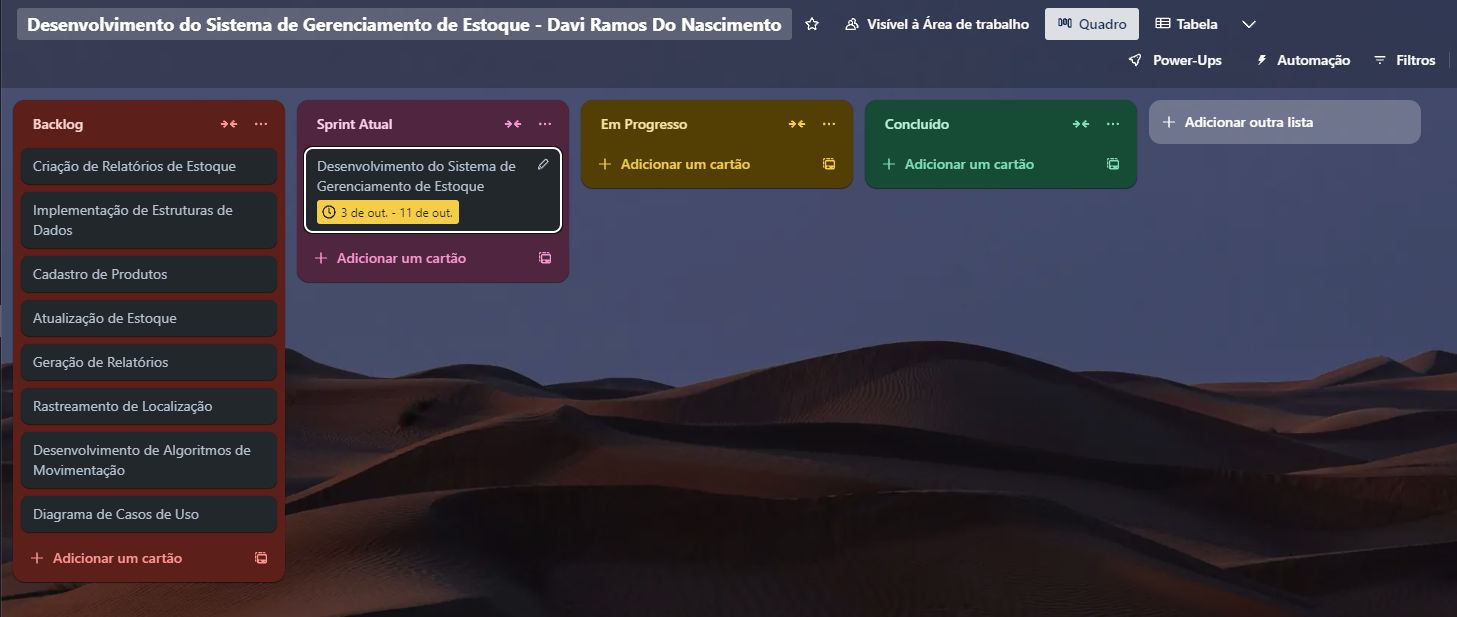
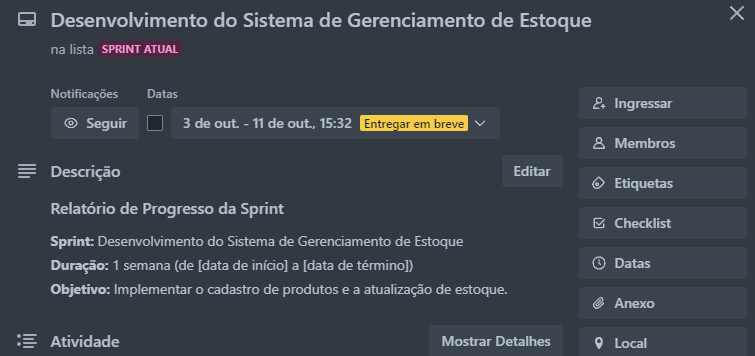
### Passo 1: Quadro Scrum no Trello

1. **Criação do Quadro:**
   * Acesse o [Trello](https://www.trello.com) e faça login ou crie uma conta.
   * Crie um novo quadro nomeado: "Desenvolvimento do Sistema de Gerenciamento de Estoque – Davi Ramos do Nascimento".
   * Configure as listas:
     + **Backlog**: Liste todas as tarefas que precisam ser realizadas.
     + **Sprint Atual**: Tarefas selecionadas para a sprint atual.
     + **Em Progresso**: Tarefas em andamento.
     + **Concluído**: Tarefas finalizadas.
2. **Divisão em Sprints:**
   * Defina a duração das sprints (ex: 1 ou 2 semanas).
   * Selecione tarefas do backlog para a sprint atual e mova para a lista correspondente.
3. **Gestão das Tarefas:**
   * Mova as tarefas entre as listas conforme avança (de "Sprint Atual" para "Em Progresso" e, em seguida, para "Concluído").
   * Utilize comentários, anexos, checklists e datas de entrega para manter tudo organizado.
4. **Relatório ao Final de Cada Sprint:**
   * Revise as tarefas concluídas e escreva um breve relatório sobre o progresso e desafios enfrentados. Inclua o que foi bem-sucedido e o que pode ser melhorado nas próximas sprints.
5. **Print Screens e Link:**
   * Capture print screens do quadro no Trello mostrando a organização e evolução do trabalho.
   * Compartilhe o link do quadro, se necessário.

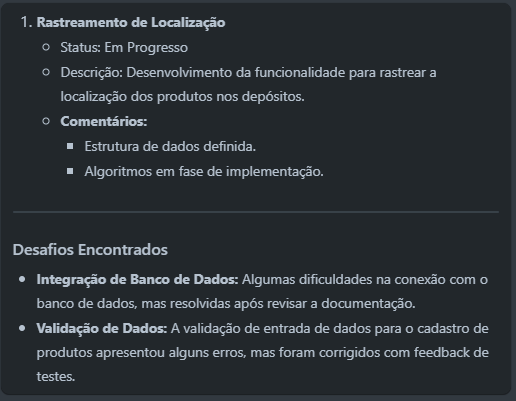


SPRINT ATUAL



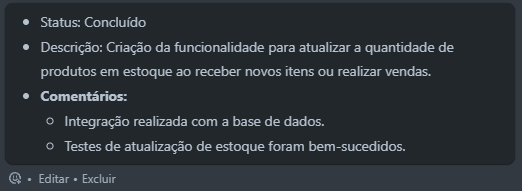
EM PROGRESSO





CONCLUIDO





#### Desafios Encontrados

* **Integração de Banco de Dados:** Algumas dificuldades na conexão com o banco de dados, mas resolvidas após revisar a documentação.
* **Validação de Dados:** A validação de entrada de dados para o cadastro de produtos apresentou alguns erros, mas foram corrigidos com feedback de testes.

#### Avaliação da Sprint

**O que foi bem-sucedido:**

* O cadastro de produtos foi implementado dentro do prazo e funcionou conforme o esperado.
* A atualização de estoque foi bem recebida, com testes confirmando a eficácia da função.

**O que pode ser melhorado:**

* Melhorar a comunicação entre os membros da equipe sobre mudanças no design do sistema.
* Dedicar mais tempo à fase de testes para minimizar erros nas próximas sprints.

### Passo 2: Tabela Verdade

1. **Definição de Variáveis Booleanas:**
   * **P**: Cadastro de Produtos
   * **E**: Atualização de Estoque
   * **L**: Rastreamento de Localização
   * **R**: Relatórios
2. **Tabela Verdade:**

| **P** | **E** | **L** | **R** | **Solução Completa** |
| --- | --- | --- | --- | --- |
| T | T | T | T | T |
| T | T | T | F | F |
| T | T | F | T | F |
| T | T | F | F | F |
| T | F | T | T | F |
| T | F | T | F | F |
| T | F | F | T | F |
| T | F | F | F | F |
| F | T | T | T | F |
| F | T | T | F | F |
| F | T | F | T | F |
| F | T | F | F | F |
| F | F | T | T | F |
| F | F | T | F | F |
| F | F | F | T | F |
| F | F | F | F | F |

1. **Expressão Lógica:**
   * A expressão lógica para a solução completa é:

Soluc¸a˜o Completa=P∧E∧L∧R\text{Solução Completa} = P \land E \land L \land RSoluc¸​a˜o Completa=P∧E∧L∧R

### Passo 3: Estruturas de Dados e Algoritmos

### Definição de Estruturas de Dados:

### class Produto:

### def \_\_init\_\_(self, nome, categoria, quantidade, preco, localizacao):

### self.nome = nome

### self.categoria = categoria

### self.quantidade = quantidade

### self.preco = preco

### self.localizacao = localizacao

### class Estoque:

### def \_\_init\_\_(self):

### self.produtos = []

### Algoritmos de Cadastro e Consulta:

### def cadastrar\_produto(estoque, produto):

### estoque.produtos.append(produto)

### def consultar\_produto(estoque, nome):

### for produto in estoque.produtos:

### if produto.nome == nome:

### return produto

### return None

### Algoritmos de Movimentação:

### def registrar\_entrada(estoque, nome, quantidade):

### produto = consultar\_produto(estoque, nome)

### if produto:

### produto.quantidade += quantidade

### def registrar\_saida(estoque, nome, quantidade):

### produto = consultar\_produto(estoque, nome)

### if produto and produto.quantidade >= quantidade:

### produto.quantidade -= quantidade

### Relatórios e Consultas:

### def relatorio\_estoque(estoque):

### for produto in estoque.produtos:

### print(f'Produto: {produto.nome}, Quantidade: {produto.quantidade}, Localização: {produto.localizacao}')

**Passo 4: Diagrama de Casos de Uso**

* Para criar um Diagrama de Casos de Uso, você pode usar ferramentas como Lucidchart, Draw.io, ou até mesmo papel e caneta. O diagrama deve incluir:
  + **Usuários:**
    - Estoquista
    - Usuário
    - Gerente de Setor
  + **Casos de Uso:**
    - Estoquista: Registrar entrada de produtos
    - Usuário: Emitir relatórios, Solicitar compra de produtos
    - Gerente de Setor: Autorizar compra de produtos
* O diagrama deve mostrar a interação entre os usuários e as funcionalidades do sistema.