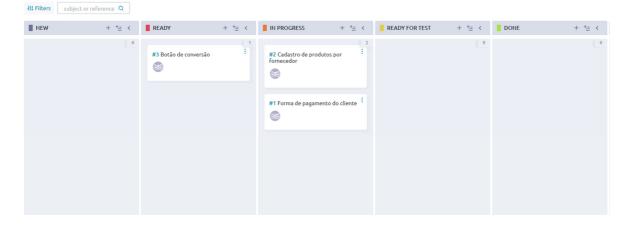
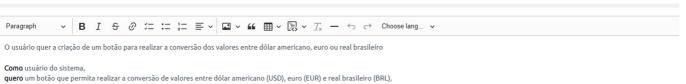


## Kanban









## Épicos Relacionados

Implementação de lógica de conversão com base em taxas atualizadas.

para que eu possa visualizar e comparar os valores em diferentes moedas com facilidade e precisão.

• Seleção de moedas de origem e destino.

#### Interface de Usuário

- Criação de botão visível e funcional na interface.
- o Campos para inserir o valor e selecionar moedas.

### Integração com APIs

o Consulta a uma API de câmbio (ex: ExchangeRate API, [Fixer.io], [Open Exchange Rates]) para obter taxas atualizadas em tempo real.

### Validação e Experiência do Usuário

- Mensagens de erro amigáveis para entradas inválidas.
- Conversão instantânea e clara.

#### Critérios de Aceitação

- Deve haver um campo para inserir o valor a ser convertido.
- O usuário deve poder selecionar a moeda de origem e a de destino entre USD, EUR e BRL.
- Ao clicar no botão de conversão, o sistema deve:
- Realizar a conversão com base na taxa de câmbio atual.
- Exibir o valor convertido de forma destacada.
- Se o valor inserido é numérico e positivo.
- Se ambas as moedas foram selecionadas.

# Casos de Teste – Técnica Step-by-Step – Conversão de Moeda EUR ightarrow BRL

Pré-condições: Sistema com acesso à API de câmbio

- 1. Inserir o valor em EUR: 50
- 2. Selecionar "De: EUR", "Para: BRL"
- 3. Clicar em "Converter"

Resultado esperado: Valor convertido em BRL, com base na cotação atual



Save Cancel

Markdown

# #1 Forma de pagamento do cliente

USER STORY





Desejo um campo para selecionar a forma de pagamento entre cartão e pix

### Como usuário do sistema,

quero um campo para selecionar a forma de pagamento entre cartão e PIX, para que eu possa escolher o método que melhor atenda às minhas necessidades no momento da compra.

#### Épicos Relacionados

## Processo de Pagamento

- Adição de opções de pagamento na finalização da compra.
- Permitir que o usuário escolha entre PIX e cartão.

### Interface de Usuário

- o Campo claro e intuitivo (como radio button, dropdown ou botão de seleção).
- o Atualização dinâmica da interface conforme a forma de pagamento selecionada.

## Fluxos Diferenciados por Pagamento

- Exibir os campos relevantes conforme a forma selecionada:
- o Cartão: número do cartão, validade, CVV, nome do titular.
- PIX: exibição do OR Code ou chave PIX.

- Garantir que uma forma de pagamento seja obrigatoriamente selecionada antes da confirmação do pedido
- o Campos obrigatórios visíveis e validados com clareza.

## Critérios de Aceitação

- Deve existir um campo de seleção com as opções:
- "Cartão"
- "PIX"
- Ao selecionar:
  - Cartão: o sistema deve exibir campos para preenchimento dos dados do cartão.
- PIX: o sistema deve exibir o QR Code ou chave para pagamento.
- Deve ser possível **confirmar o pedido** apenas após selecionar uma das formas de pagamento.
- O sistema deve impedir a continuação sem que uma opção esteja marcada.
- A interface deve ser responsiva e acessível.

# Casos de Teste – Técnica BDD – Seleção de Forma de Pagamento com Cartão

Feature: Selecionar forma de pagamento

Scenario: Usuário escolhe a forma de pagamento por cartão Given que o usuário está na página de checkout When ele seleciona a opção "Cartão" Then os campos para número do cartão, validade, CVV e titular devem ser exibidos And o botão de "Finalizar compra" deve estar habilitado

Save Cancel

Markdown

# #2 Cadastro de produtos por fornecedor

**USER STORY** 





Quero um campo para informar a disponibilidade em estoque dos produtos, com informações de quantidade, marca, cor, tamanho e modelo

Como operador/responsável pelo controle de estoque,

quero um campo ou formulário onde possa informar a disponibilidade dos produtos, incluindo quantidade, marca, cor, tamanho e modelo, para que seja possível gerenciar com precisão o inventário e facilitar o acesso às informações pelos demais setores (ex: vendas, compras, logística).

## Épicos Relacionados

- Gestão de Produtos
- Cadastro e manutenção de informações detalhadas dos produtos.
- · Inclusão de atributos como marca, cor, tamanho e modelo.
- Controle de Estoque
- Registro da quantidade disponível de cada item.
- Atualização automática após entrada e saída de produtos.
- · Relatórios de disponibilidade.
- Integração com Setores
- Disponibilização das informações para os setores de vendas, compras e atendimento.
- Interface do Usuário (UX)
- Tela amigável para consulta e atualização de estoque.
- · Filtros e busca por atributos (ex: tamanho, marca).

## Informações Relevantes (Critérios de Aceitação)

- Deve ser possível cadastrar um produto com os seguintes campos obrigatórios:
- Quantidade disponível
- Сог
- Modelo
- A interface de cadastro deve permitir:
- Edição e exclusão de registros
- Filtros por qualquer um dos atributos
- Visualização rápida do status de estoque (ex: baixo, esgotado, normal)
- Deve haver validação:
- Não permitir cadastrar quantidade negativa
- · Verificação de campos obrigatórios
- As atualizações devem ser refletidas em tempo real (ou próximo disso) nos sistemas que dependem dessas informações.

## Caso de Teste 01 - Cadastro de Produto com Estoque Válido

Pré-condições: Acesso ao módulo de estoque

## Passos:

- 1. Acessar a página de cadastro de produto
- 2. Preencher os campos com dados válidos
- 3. Clicar em "Salvar"

# Dados de entrada:

- Marca: Nike
- · Tamanho: M
- Modelo: Air Max
- · Quantidade: 10

Resultado esperado: Produto salvo com sucesso e exibido na lista

## Caso de Teste 02 - Cadastro de Estoque com Quantidade Inválida

Feature: Validação de quantidade no estoque

Scenario: Usuário tenta cadastrar produto com quantidade negativa Given que o usuário está na tela de novo produto When ele insere "-5" no campo de quantidade And clica em "Salvar" Then uma mensagem de erro deve ser exibida: "Quantidade inválida" And o produto não deve ser salvo

Save Cancel