

Casos de Teste - Formato BDD (Gherkin)

Informações Gerais

Módulo: Carrinho de Compras

User Story: US-001 - Adicionar Produto ao Carrinho

Framework: Cucumber/Gherkin

Autor: Equipe QA

Data: 11/11/2025

TC-BDD-001: Adicionar Produto ao Carrinho com Diferentes Quantidades

```
Feature: Adicionar produtos ao carrinho de compras
  Como um usuário do e-commerce
  Quero adicionar produtos ao meu carrinho
  Para que eu possa revisar e finalizar minha compra posteriormente

  Background:
    Given que estou na página inicial do e-commerce
    And estou logado com o usuário "teste@email.com"
    And meu carrinho está vazio

  @smoke @high-priority
  Scenario: Adicionar um único produto ao carrinho com quantidade padrão
    Given que estou visualizando o produto "Notebook Dell Inspiron 15"
    And o produto tem preço de "R$ 3.299,00"
    And o produto tem estoque disponível de "50" unidades
    When eu clico no botão "Adicionar ao Carrinho"
    Then devo ver a mensagem "Produto adicionado ao carrinho com sucesso!"
    And o contador do carrinho deve exibir "1"
    And o produto deve aparecer no carrinho com quantidade "1"
    And o subtotal do produto deve ser "R$ 3.299,00"
    And o total do carrinho deve ser "R$ 3.299,00"

  @regression
  Scenario: Adicionar produto ao carrinho com quantidade específica
    Given que estou visualizando o produto "Mouse Logitech MX Master 3"
    And o produto tem preço de "R$ 499,00"
    And o produto tem estoque disponível de "200" unidades
    When eu altero a quantidade para "5"
    And eu clico no botão "Adicionar ao Carrinho"
    Then devo ver a mensagem "Produto adicionado ao carrinho com sucesso!"
    And o contador do carrinho deve exibir "5"
    And o produto deve aparecer no carrinho com quantidade "5"
    And o subtotal do produto deve ser "R$ 2.495,00"
    And o total do carrinho deve ser "R$ 2.495,00"

  @regression
  Scenario Outline: Adicionar múltiplos produtos com diferentes quantidades
    Given que estou visualizando o produto "<produto>"
    And o produto tem preço de "<preco>"
    And o produto tem estoque disponível de "<estoque>" unidades
    When eu altero a quantidade para "<quantidade>"
    And eu clico no botão "Adicionar ao Carrinho"
    Then devo ver a mensagem "Produto adicionado ao carrinho com sucesso!"
    And o contador do carrinho deve exibir "<quantidade>"
    And o subtotal do produto deve ser "<subtotal>"
```

```
When eu altero a quantidade para "<quantidade>"
And eu cliço no botão "Adicionar ao Carrinho"
Then devo ver a mensagem "Produto adicionado ao carrinho com sucesso!"
And o contador do carrinho deve exibir "<quantidade>"
And o subtotal do produto deve ser "<subtotal>"
```

Examples:

produto	preco	estoque	quantidade	subtotal	
Teclado Mecânico RGB	R\$ 450,00	100	2	R\$ 900,00	
Webcam Logitech C920	R\$ 550,00	75	1	R\$ 550,00	
Headset Gamer HyperX	R\$ 380,00	150	3	R\$ 1.140,00	
Monitor LG 27" UltraWide	R\$ 1.899,00	30	1	R\$ 1.899,00	

@functional

Scenario: Adicionar produto ao carrinho e verificar persistência

Given que estou visualizando o produto "SSD Samsung 1TB"

And o produto tem preço de "R\$ 789,00"

When eu altero a quantidade para "2"

And eu cliço no botão "Adicionar ao Carrinho"

And eu navego para a página "Categorias"

And eu navego de volta para a página "Carrinho"

Then o produto "SSD Samsung 1TB" deve estar no carrinho

And a quantidade deve ser "2"

And o subtotal deve ser "R\$ 1.578,00"

@functional

Scenario: Adicionar mesmo produto múltiplas vezes incrementa quantidade

Given que estou visualizando o produto "Cadeira Gamer DXRacer"

And o produto tem preço de "R\$ 1.299,00"

And o produto tem estoque disponível de "25" unidades

When eu cliço no botão "Adicionar ao Carrinho"

And devo ver a mensagem "Produto adicionado ao carrinho com sucesso!"

And eu cliço no botão "Adicionar ao Carrinho" novamente

Then o contador do carrinho deve exibir "2"

And o carrinho deve conter apenas "1" linha de produto

And a quantidade do produto deve ser "2"

And o subtotal deve ser "R\$ 2.598,00"

TC-BDD-002: Validações e Restrições ao Adicionar Produtos

Feature: Validações ao adicionar produtos ao carrinho

Como um sistema de e-commerce

Quero validar as entradas do usuário ao adicionar produtos

Para garantir a integridade dos dados e uma boa experiência

Background:

Given que estou na página inicial do e-commerce

And estou logado com o usuário "teste@email.com"

@negative @validation

Scenario: Tentar adicionar produto com quantidade zero

Given que estou visualizando o produto "Teclado Mecânico RGB"

And o produto tem estoque disponível de "100" unidades

When eu altero a quantidade para "0"

And eu cliço fora do campo de quantidade

Then devo ver a mensagem de erro "Quantidade mínima é 1"

And o botão "Adicionar ao Carrinho" deve estar desabilitado

And o produto não deve ser adicionado ao carrinho

@negative @validation

Scenario: Tentar adicionar produto com quantidade negativa

Given que estou visualizando o produto "Mouse Gamer Razer"

And o produto tem estoque disponível de "50" unidades

When eu tento inserir a quantidade "-5"

Then o campo não deve aceitar valores negativos

Or devo ver a mensagem de erro "Quantidade deve ser um número positivo"

And o botão "Adicionar ao Carrinho" deve estar desabilitado

```
Given que estou visualizando o produto "Mouse Gamer Razer"
And o produto tem estoque disponível de "50" unidades
When eu tento inserir a quantidade "-5"
Then o campo não deve aceitar valores negativos
Or devo ver a mensagem de erro "Quantidade deve ser um número positivo"
And o botão "Adicionar ao Carrinho" deve estar desabilitado
```

@negative @validation @boundary

Scenario: Tentar adicionar produto acima do limite máximo permitido

```
Given que estou visualizando o produto "Cabo USB-C"
And o produto tem estoque disponível de "500" unidades
And o limite máximo por pedido é "99" unidades
When eu altero a quantidade para "100"
And eu clico fora do campo de quantidade
Then devo ver a mensagem de erro "Quantidade máxima por pedido é 99 unidades"
And o botão "Adicionar ao Carrinho" deve estar desabilitado
And o produto não deve ser adicionado ao carrinho
```

@negative @validation

Scenario: Tentar adicionar produto sem estoque suficiente

```
Given que estou visualizando o produto "Placa de Vídeo RTX 4090"
And o produto tem estoque disponível de "3" unidades
When eu altero a quantidade para "5"
And eu clico no botão "Adicionar ao Carrinho"
Then devo ver a mensagem de erro "Estoque insuficiente. Apenas 3 unidades disponíveis"
And o produto não deve ser adicionado ao carrinho
And devo permanecer na página do produto
```

@negative @validation

Scenario: Tentar adicionar produto com estoque zerado

```
Given que estou visualizando o produto "Console PlayStation 5"
And o produto tem estoque disponível de "0" unidades
Then o botão "Adicionar ao Carrinho" deve estar desabilitado
And devo ver a mensagem "Produto indisponível no momento"
And o campo de quantidade deve estar desabilitado
```

@negative @security

Scenario: Tentar adicionar produto sem estar autenticado

```
Given que NÃO estou logado no sistema
And que estou visualizando o produto "Smartphone Samsung Galaxy"
When eu clico no botão "Adicionar ao Carrinho"
Then devo ser redirecionado para a página de login
And devo ver a mensagem "Faça login para adicionar produtos ao carrinho"
And o produto não deve ser adicionado ao carrinho
```

@validation @boundary

Scenario Outline: Validar limites de quantidade (valores de fronteira)

```
Given que estou visualizando o produto "Memória RAM 16GB"
And o produto tem estoque disponível de "200" unidades
When eu altero a quantidade para "<quantidade>"
And eu clico no botão "Adicionar ao Carrinho"
Then devo ver "<resultado>"
```

Examples:

quantidade	resultado
1	Produto adicionado ao carrinho com sucesso!
99	Produto adicionado ao carrinho com sucesso!
100	Quantidade máxima por pedido é 99 unidades
0	Quantidade mínima é 1

@functional @validation

Scenario: Adicionar produto e validar atualização do estoque

```
Given que estou visualizando o produto "Processador Intel i9"
And o produto tem estoque disponível de "15" unidades
And o estoque mostrado na página é "15 unidades disponíveis"
When eu altero a quantidade para "10"
And eu clico no botão "Adicionar ao Carrinho"
Then devo ver a mensagem "Produto adicionado ao carrinho com sucesso!"
And ao retornar para a página do produto
Then o estoque disponível deve ser "5 unidades disponíveis"
```

@edge-case

Scenario: Adicionar produto com caracteres especiais na quantidade

```
Given que estou visualizando o produto "Mousepad Gamer"
When eu tento inserir "abc" no campo de quantidade
```

```
@edge-case
Scenario: Adicionar produto com caracteres especiais na quantidade
  Given que estou visualizando o produto "Mousepad Gamer"
  When eu tento inserir "abc" no campo de quantidade
  Then o campo não deve aceitar letras
  And o campo deve permitir apenas números
  When eu tento inserir "1.5" no campo de quantidade
  Then o campo deve aceitar apenas números inteiros
  And devo ver a mensagem "Quantidade deve ser um número inteiro"

@performance
Scenario: Verificar tempo de resposta ao adicionar produto
  Given que estou visualizando o produto "Notebook Gamer"
  And o produto tem estoque disponível
  When eu clico no botão "Adicionar ao Carrinho"
  Then a operação deve ser concluída em menos de "2" segundos
  And devo ver o feedback visual durante o processamento
  And devo ver a mensagem de sucesso após a conclusão
```

Configuração dos Hooks (Cucumber)

```
# hooks.feature

Before:
  """
  - Limpar cookies e cache do navegador
  - Inicializar driver do navegador
  - Configurar resolução da tela (1920x1080)
  - Preparar dados de teste no banco de dados
  - Gerar token de autenticação se necessário
  """

After:
  """
  - Capturar screenshot se o teste falhou
  - Capturar logs do console
  - Limpar dados de teste criados
  - Fechar navegador
  - Gerar relatório de execução
  """

AfterStep:
  """
  - Aguardar 500ms entre passos (para estabilidade)
  - Verificar se há erros JavaScript no console
  """
```

Tags Utilizadas

Tag	Descrição
@smoke	Testes críticos executados em toda build
@regression	Suite completa de regressão
@high-priority	Testes de alta prioridade
@critical-path	Testes que cobrem o fluxo principal de uso

@regression	Suite completa de regressão
@high-priority	Testes de alta prioridade
@negative	Testes de cenários negativos
@validation	Testes de validação de entrada
@boundary	Testes de valor limite
@security	Testes relacionados à segurança
@functional	Testes de funcionalidade
@edge-case	Casos extremos/raros
@performance	Testes de performance

Execução dos Testes

```
# Executar todos os testes
cucumber features/carrinho.feature

# Executar apenas testes smoke
cucumber --tags @smoke

# Executar testes de validação negativos
cucumber --tags "@negative and @validation"

# Executar todos exceto edge cases
cucumber --tags "not @edge-case"

# Gerar relatório HTML
cucumber --format html --out reports/cucumber_report.html
```

Dados de Teste

Usuários

- Email: teste@email.com
- Senha: Teste@123
- Nome: Usuário Teste
- CPF: 123.456.789-00

Produtos

ID	Nome	Preço	Estoque
P001	Notebook Dell Inspiron 15	R\$ 3.299,00	50
P002	Mouse Logitech MX Master 3	R\$ 499,00	200
P003	Teclado Mecânico RGB	R\$ 450,00	100

P001	Notebook Dell Inspiron 15	R\$ 3.299,00	50
P002	Mouse Logitech MX Master 3	R\$ 499,00	200
P003	Teclado Mecânico RGB	R\$ 450,00	100
P004	SSD Samsung 1TB	R\$ 789,00	75

Assertivas Importantes

- ✓ Todas as mensagens devem ser validadas
- ✓ Estados dos botões (habilitado/desabilitado) devem ser verificados
- ✓ Cálculos matemáticos devem ser precisos
- ✓ Tempo de resposta deve ser monitorado
- ✓ Logs e screenshots devem ser capturados em caso de falha