

# Documento de Casos de Teste para a Classe **VendaController**

## Introdução

Este documento descreve os casos de teste para a classe **VendaController**, responsável por gerenciar as operações relacionadas às vendas no sistema. Os testes visam garantir que as funcionalidades implementadas estejam operando conforme o esperado e que os requisitos do sistema sejam atendidos.

---

## Casos de Teste

### Caso de Teste 1

ID: CT-001

**Descrição:** Verificar a exibição do formulário de criação de venda.

**Método:** **GET** `/venda/form`

**Precondições:**

- O usuário está autenticado no sistema.

**Passos:**

- Acessar a URL `/venda/form` via método GET.

**Resultados Esperados:**

- A página do formulário de venda é exibida corretamente.
  - O modelo contém um objeto **Venda** vazio.
- 

### Caso de Teste 2

ID: CT-002

**Descrição:** Listar vendas com um status específico.

**Método:** **GET** `/venda/status/{status}`

**Precondições:**

- Existem vendas cadastradas com o status informado (por exemplo, "ABERTO").

**Passos:**

1. Acessar a URL `/venda/status/ABERTO` via método GET.

**Resultados Esperados:**

- A lista de vendas com status "ABERTO" é exibida.
  - A paginação funciona corretamente.
  - Os atributos do modelo estão preenchidos conforme esperado.
- 

## Caso de Teste 3

ID: CT-003

**Descrição:** Abrir uma nova venda com dados válidos.

**Método:** `POST /venda`

**Precondições:**

- O usuário está autenticado.
- Os dados fornecidos para a venda são válidos.

**Passos:**

1. Enviar uma requisição POST para `/venda` com os dados válidos da venda.

**Resultados Esperados:**

- A venda é criada com sucesso.
  - O usuário é redirecionado para a página da venda criada.
  - Uma mensagem de sucesso "Pedido Salvo" é exibida.
- 

## Caso de Teste 4

ID: CT-004

**Descrição:** Tentar abrir uma nova venda com dados inválidos.

**Método:** `POST /venda`

**Precondições:**

- O usuário está autenticado.
- Os dados fornecidos são inválidos (por exemplo, campos obrigatórios não preenchidos).

**Passos:**

1. Enviar uma requisição POST para `/venda` com dados inválidos.

**Resultados Esperados:**

- A venda não é criada.
  - Os erros de validação são exibidos no formulário.
  - O usuário permanece na página de criação da venda.
- 

## Caso de Teste 5

ID: CT-005

**Descrição:** Buscar detalhes de uma venda existente.

**Método:** `GET /venda/{codigo}`

**Precondições:**

- Existe uma venda com o código informado (por exemplo, código 1).

**Passos:**

1. Acessar a URL `/venda/1` via método GET.

**Resultados Esperados:**

- A página com os detalhes da venda é exibida.
  - Os produtos associados à venda são listados.
- 

## Caso de Teste 6

ID: CT-006

**Descrição:** Adicionar um produto à venda.

**Método:** `POST /venda/addproduto`

**Precondições:**

- A venda existe e está aberta.

- O produto a ser adicionado existe no sistema.

**Passos:**

1. Enviar uma requisição POST para `/venda/addproduto` com os parâmetros:
  - `codigoVen`: código da venda.
  - `codigoPro`: código do produto.
  - `valorBalanca`: valor (se aplicável).

**Resultados Esperados:**

- O produto é adicionado à venda com sucesso.
  - Uma mensagem de sucesso é retornada.
- 

## Caso de Teste 7

ID: CT-007

**Descrição:** Remover um produto da venda.

**Método:** `POST /venda/removeproduto`

**Precondições:**

- A venda existe e contém o produto a ser removido.

**Passos:**

1. Enviar uma requisição POST para `/venda/removeproduto` com os parâmetros:
  - `codigoVen`: código da venda.
  - `posicaoPro`: posição do produto na lista.

**Resultados Esperados:**

- O produto é removido da venda.
  - Uma mensagem de sucesso é retornada.
- 

## Caso de Teste 8

ID: CT-008

**Descrição:** Fechar uma venda com pagamento válido.

**Método:** `POST /venda/fechar`

**Precondições:**

- A venda existe e possui produtos adicionados.
- Os dados de pagamento são válidos.

**Passos:**

1. Enviar uma requisição POST para `/venda/fechar` com os parâmetros necessários:
  - `venda`
  - `pagamentotipo`
  - `valor_produtos`
  - `valor_desconto`
  - `valor_acrescimo`
  - `valores` (lista de valores das parcelas)
  - `titulos` (lista de títulos)

**Resultados Esperados:**

- A venda é fechada com sucesso.
  - Os títulos são gerados corretamente.
  - Uma mensagem de sucesso é retornada.
- 

**Caso de Teste 9**

ID: CT-009

**Descrição:** Obter a lista de títulos disponíveis.

**Método:** `GET /venda/titulos`

**Precondições:**

- Existem títulos cadastrados no sistema.

**Passos:**

1. Enviar uma requisição GET para `/venda/titulos`.

**Resultados Esperados:**

- Uma lista JSON contendo os títulos é retornada.
  - O conteúdo corresponde aos títulos disponíveis no sistema.
-

## Considerações Finais

Este documento abrange os principais casos de teste para as operações da classe `VendaController`. A execução desses testes ajudará a garantir que as funcionalidades de venda estejam funcionando corretamente e que o sistema atenda aos requisitos especificados.

### Recomendações:

- **Automatização:** Sempre que possível, automatize esses testes usando frameworks apropriados para testes de aplicações web.
- **Atualizações:** Mantenha este documento atualizado conforme novas funcionalidades sejam adicionadas ou alteradas.
- **Validação de Dados:** Além dos testes listados, valide os cenários de erro e exceções para garantir a robustez do sistema.