

Status:	Backlog
Projeto:	Dinâmica - Cenário Simulado de E-commerce
Componentes:	Nenhum
Versões afetadas:	Nenhum
Versões corrigidas:	Nenhum
Nível de Segurança:	2f5e4a7de36b207a6f223143c39ffdcdbdc3390d9761ce8b36759b8c93e40913 (Restrictions created for Next Gen projects)
Pai:	Checkout e Pagamentos do E-commerce

Tipo:	História	Prioridade:	Medium
Relator:	Letícia Vidal	Responsável:	Letícia Vidal
Resolução:	Não resolvido(s)	Votos:	0
Categorias:	Funcionalidade		
Estimativa de trabalho restante:	Desconhecido		
Tempo gasto:	Desconhecido		
Estimativa original:	Desconhecido		

Rank:	0ji00027:
Team:	
Start date:	04/dez/25
Sprint:	quadro Sprint 1
Story point estimate:	0

Descrição

## História de Usuário - Ecommerce de Miniaturas produzidas com impressão 3D.

**Como:**

Uuário comprador Lucas, fã de miniaturas de anime feitas por impressão 3D,

**Quero:**

Adicionar produtos ao carrinho e finalizar a compra com pagamento online (cartão, pix ou boleto),

**Para:**

Receber minhas miniaturas de forma rápida, segura e sem erros de cobrança.

### Funcionalidades:

- Visualizar catálogo de miniaturas prontas.
- Adicionar produtos ao carrinho e ajustar quantidades.
- Ver resumo de valores (subtotais, descontos e frete).
- Finalizar compra com endereço de entrega.
- Pagamento via cartão de crédito, PIX e boleto.
- Confirmação automática do pedido após processamento do pagamento.

### Requisitos Funcionais

- O sistema deve permitir incluir, editar quantidade e remover produtos do carrinho.
- O carrinho deve persistir caso o usuário volte ao site (persistência local).
- O sistema deve validar estoque antes de permitir a finalização da compra.
- O sistema deve calcular automaticamente o valor total, incluindo frete, descontos e produto.
- O sistema deve recalcular automaticamente o total sempre que houver mudança no carrinho.
- O sistema deve exibir métodos de pagamento (cartão, PIX, boleto) com fluxos separados.
- Para pagamentos PIX, o QR Code deve expirar em até 30 minutos.
- Após aprovação do pagamento, o sistema deve registrar o status “Pagamento aprovado”.
- O sistema deve registrar todos os dados essenciais do pedido (itens, endereço, valor).
- O sistema deve atualizar o status do pedido para “Pedido confirmado” após registro completo.
- O sistema deve encaminhar o pedido ao módulo de logística e atualizar o status para “Aguardando separação”.
- O sistema deve permitir evolução de status do pedido conforme fluxo operacional: Pagamento aprovado, pedido confirmado, aguardando separação, em separação, em transporte e entregue

- O sistema deve registrar data e hora de cada atualização de status (auditoria).
- O sistema deve exibir confirmação final para o cliente e enviar e-mail com número do pedido.

## Requisitos Não Funcionais:

1. **Segurança:** Todas as transações devem seguir padrões PCI-DSS e criptografia TLS 1.2+.
2. **Disponibilidade:** 99,9% do uptime atual.
3. **Performance:**
4. Página de pagamento deve carregar em até dois segundos.
5. Confirmação de pagamento deve ocorrer em até 5 segundos.
6. **Integridade:** Integração com APIs externas (frete e pagamento) deve ter fallback se falharem.
7. **Usabilidade:** A interface deve atualizar automaticamente os valores do carrinho (subtotal, frete, total) em até 1 segundo após qualquer alteração.

## Critérios de Aceite:

- O usuário consegue adicionar itens ao carrinho e ver valor atualizado instantaneamente.
- Se faltar estoque o sistema informa antes de realizar o pedido
- O cálculo de frete e descontos deve aparecer corretamente no resumo.
- No pagamento via cartão, se a transação for aprovada, o pedido deve ser confirmado. Caso for recusada, uma mensagem de erro descrevendo o problema deve ser exibida na tela.
- No pagamento via PIX, o QR Code deve ser exibido com tempo de expiração de 30 minutos e após o pagamento a confirmação deve ser automática.
- No pagamento via boleto, o boleto gerado deve ter vencimento de 3 dias.
- O usuário recebe confirmação do pedido por e-mail após a conclusão.

## Cenários de erro previstos:

- Cartão recusado.
- Falha na geração do QR Code.
- PIX expirado.
- API de pagamento indisponível.
- Falha no cálculo do frete.
- Item sem estoque.

## APIs externas Envolvidas:

- API de cálculo de frete.
- API de pagamento (cartão/PIX/boleto)

## Testes funcionais:

- Verificar se o usuário consegue visualizar o catálogo de miniaturas prontas.
- Testar se é possível incluir, remover e editar a quantidade de produtos no carrinho.
- Confirmar se o carrinho mantém os itens ao retornar ao site (persistência local).
- Verificar se o sistema calcula automaticamente subtotal, descontos, frete e valor total após qualquer mudança.
- Testar se o sistema impede a finalização da compra quando não há estoque disponível.
- Validar se o resumo de valores é exibido corretamente antes da finalização da compra.

## Testes de pagamento

### Cartão de crédito:

- Confirmar se o pedido é aprovado quando a transação é aceita.
- Verificar se a mensagem de erro aparece corretamente quando o cartão é recusado.

### PIX:

- Verificar se o sistema exibe o QR Code com expiração de 30 minutos.
- Confirmar se, após o pagamento, o pedido é aprovado automaticamente.
- Testar o comportamento quando o QR Code expira.

### Boleto:

- Verificar se o boleto é gerado com vencimento de 3 dias.

## Testes relacionados ao pedido

- Confirmar se, após a aprovação do pagamento, o sistema registra corretamente os dados essenciais do pedido (itens, endereço e valor).
- Verificar se o status muda para “Pagamento aprovado” e depois “Pedido confirmado”.

- Conferir se o sistema encaminha o pedido ao módulo de logística e atualiza para “Aguardando separação”.
- Testar se o fluxo de status segue corretamente a ordem prevista:  
Pagamento aprovado → Pedido confirmado → Aguardando separação → Em separação → Em transporte → Entregue.
- Verificar se cada atualização registra data e hora (auditoria).
- Confirmar se a confirmação final é exibida para o cliente e se o e-mail com o número do pedido é enviado.

## Testes de Cenários de Erro

- Tentar finalizar a compra com cartão recusado.
- Simular falha na geração do QR Code.
- Testar comportamento com PIX expirado.
- Simular API de pagamento indisponível.
- Verificar falha no cálculo de frete.
- Tentar finalizar a compra com item sem estoque.

Gerado em Mon Dec 08 19:51:35 UTC 2025 por Tatinha Souza usando JIRA 1001.0.0-SNAPSHOT#100290-  
rev:01015d7ff7de39974d2b159bd650b7113bf409da.