



Universidade Estadual da Paraíba (UEPB)  
Centro de Ciências e Tecnologia (CCT)  
Curso Bacharelado em Ciências da Computação

Pedro Lucas Vaz Andrade

Otávio Ferreira Clementino

Wagner Tibúrcio da Silva Junior

Relatório de Release 02 - Funcionalidades

Campina Grande - 2025

<b>1. Introdução</b>	<b>3</b>
<b>2. Funcionalidades Implementadas</b>	<b>3</b>
2.1 Cadastro e Gerenciamento de Compradores	3
2.2 Sistema de Autenticação (Login de Comprador)	4
2.3 Carrinho de Compras	5
2.3.1 Estrutura de Dados	5
2.3.2 Funcionalidades do Carrinho	5
2.4 Integração entre Entidades	6
<b>3. Interface com o Usuário (CLI)</b>	<b>6</b>
<b>4. Testes Realizados</b>	<b>7</b>
<b>5. Considerações Finais</b>	<b>7</b>

# 1. Introdução

A Sprint 4 teve como foco a implementação do **sistema de pedidos** no Eng2Marketplace, ampliando significativamente a robustez do processo de finalização de compras. O objetivo principal foi permitir que compradores tenham **rastreabilidade completa** de suas transações, assegurando a persistência dos pedidos e a integridade das operações. Também foram abordados aspectos importantes como **validações de estoque, tratamento de erros críticos e serialização dos dados** em formato JSON.

Além disso, essa etapa também serviu para **corrigir lacunas técnicas** identificadas na Sprint 2, como a ausência de atualização de estoque e de estrutura adequada para o gerenciamento de históricos. A Sprint 4 encerra o ciclo funcional do módulo de vendas, preparando o terreno para futuras integrações com fluxos administrativos e relatórios de desempenho para vendedores.

## 2. Funcionalidades Implementadas

A seguir, estão detalhadas as funcionalidades entregues nesta sprint, segmentadas em dois grandes blocos: funcionalidades centrais do **módulo de pedidos** e suas respectivas **integrações sistêmicas**.

### 2.1 Módulo de Pedidos

Funcionalidade		Descrição	Classes Envolvidas
Criação de Pedidos	de	Ao final da compra, um novo objeto <b>Pedido</b> é instanciado com UUID, data/hora da compra, itens comprados, valor total e status inicial.	<b>Pedido</b> , <b>PedidoController</b> , <b>PedidoRepository</b>
Histórico Compras	de	Permite ao comprador visualizar todas as suas compras passadas por meio de filtro por CPF.	<b>PedidoView</b> , <b>MarketplaceFacade</b>

### 2.2 Integrações Sistêmicas

Funcionalidade		Descrição	Classes Envolvidas
----------------	--	-----------	--------------------

Finalização Compra	de	Converte o conteúdo do carrinho em um pedido, calcula total, armazena o pedido e esvazia o carrinho.	MarketplaceFacade, CompradorController
Validação Estoque	de	No momento da compra, o sistema valida se os produtos têm estoque suficiente. Se sim, decrementar o estoque.	ProdutoController
Serialização Pedidos	de	Os pedidos são salvos no disco em <code>pedidos.json</code> , garantindo persistência mesmo após desligamento do sistema.	PedidoRepository, JsonU

### 3. Interface com o Usuário

A interface de interação com pedidos foi estruturada em **menus simples**, com feedbacks intuitivos para o usuário. As principais interações disponíveis são:

- **Visualizar histórico de pedidos:** via opção no menu principal, com listagem por data, status e valor total.
- **Finalizar compra:** opção acessível após login de comprador, com mensagem clara sobre sucesso ou falha (ex: estoque insuficiente, carrinho vazio).
- **Mensagens de erro e sucesso:** Todos os cenários possuem retornos imediatos ao usuário com mensagens formatadas, seguindo os padrões de UX textual do sistema.

### 4. Testes Realizados

A validação da sprint foi conduzida com foco em **testes manuais de fluxo**, **tratamento de exceções** e **validação de persistência JSON**.

#### 4.1 Cobertura de Testes

Área	Resultado

Testes de PedidoController	85% de cobertura (funções de criar, buscar, listar, atualizar status)
Fluxos críticos	100% dos principais fluxos testados com sucesso
Tempo de carregamento	Histórico de até 100 pedidos carregado em menos de 500ms

## 4.2 Casos de Teste Específicos

Cenário	Resultado
Carrinho vazio	Mensagem de erro e bloqueio da operação
Produto com estoque insuficiente	Impede venda e informa item indisponível
CPF não autenticado	Redirecionamento para tela de login
ID de pedido inválido	Feedback claro com mensagem personalizada
JSON ausente ou corrompido	Criação automática do arquivo ou mensagem

## 5. Considerações Finais

A Sprint 4 representou um marco importante na evolução do Eng2Marketplace, ao consolidar o processo de compras com **registro persistente**, **verificações de estoque**, **tratamento estruturado de erros** e **integração com o carrinho do comprador**. O sistema agora está apto a simular um ciclo realista de compras B2B, com recursos essenciais de rastreabilidade.

A decisão por armazenar os pedidos em um arquivo próprio (`pedidos.json`) aumentou a modularidade da aplicação e facilitou o acesso separado ao histórico, sem interferir diretamente na estrutura dos compradores.