

## **RELEASE 3 - SPRINT 6 (29/05 - 11/06)**

### **View dos requisitos:**

**Requisito 7:** O comprador pode avaliar a compra

### **Tecnologias utilizadas:**

- 1. Linguagem de programação:** Java
- 2. Estruturação:** arquitetura MVC
- 3. Persistência de dados:** JSON

### **Processo de desenvolvimento e atribuições:**

**Gerente:** Raquel Melo

### **Atribuições de Letícia Medeiros:**

#### **Score Comprador - Model:**

- Task 707 - Adicionar score comprador - Model
- Task 702 - Adicionar pontuação da venda - Model

#### **Score Comprador - Controller**

- Task 709 - Adicionar score comprador - Controller
- Task 708 - Adicionar score vendas - Controller

#### **Documentação:**

- Task 771 - Documentar funcionalidade Score de comprador
- Task 714 - Diagrama de caso de uso REQ06

### **Atribuições de Raquel Melo:**

#### **Score Comprador - View:**

- Task 706 - Exibir comprador - View
- Task 705 - Exibir pontuação da venda - View

#### **Score Compradoa - View:**

- Task 6.0 - Implementar nova exibição de lojas na View

### **Testes:**

- Task 704 - Testes venda - View
- Task 717 - Teste comprador - View

### **Documentação:**

- Task 713 - Diagrama de sequência - REQ06
- Task 715 - Documentar o processo de desenvolvimento

#### Atribuições de Vinícius Araújo:

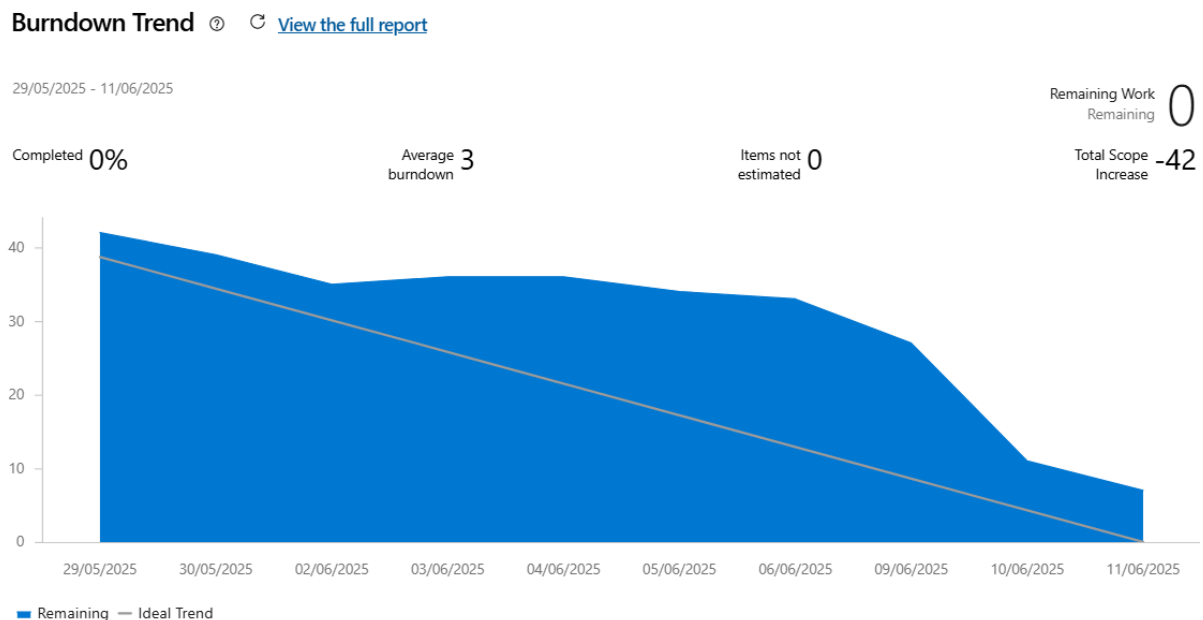
##### Testes:

- Task 703 - Testes venda - Controller
- Task 716 - Testes comprador - Controller

##### Documentação:

- Task 710 - Documentar funcionalidades do Score de vendas
- Task 712 - Diagrama de classes - REQ06

#### Gráfico de Burndown:



#### Processo de assegurar vantagens ao comprador:

##### 1. Score do comprador

No requisito implementado, o score é uma funcionalidade implementada com o objetivo de atribuir e gerenciar uma pontuação associada a cada comprador do sistema. Essa métrica pode ser utilizada para representar diferentes aspectos relacionados ao comportamento ou histórico do usuário, como fidelidade, frequência de compras ou até mesmo reputação dentro da plataforma.

O score está definido como um atributo na classe Comprador, que representa os dados básicos de um cliente no sistema. Esse valor é inicializado com zero no momento em que um novo comprador é instanciado. Além disso, a classe disponibiliza métodos getters e setters para manipulação do atributo: `getScore()` retorna o valor atual do score e `setScore(int score)` atualiza o valor do score com o valor informado.

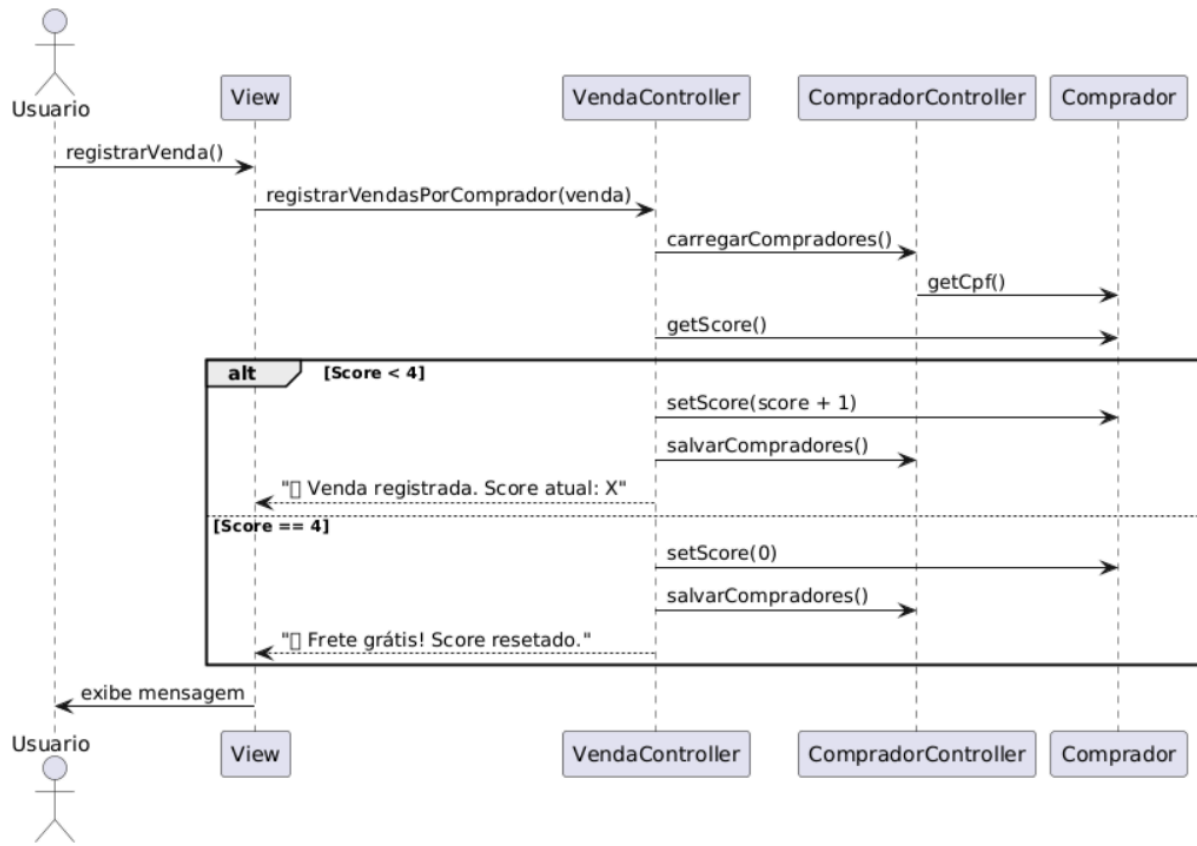
## **2. Atualização do score**

A atualização do score ocorre por meio do método estático `adicionarScore(String cpf, int pontos)`, presente na classe `CompradorController`. Este método carrega a lista completa de compradores utilizando o método `carregarCompradores()`, localiza o comprador pelo CPF fornecido como parâmetro, atualiza o score desse comprador somando o valor dos pontos passados, por meio dos métodos `getScore()` e `setScore()` e salva novamente a lista de compradores no arquivo JSON através do método `salvarCompradores(List<Comprador> compradores)`.

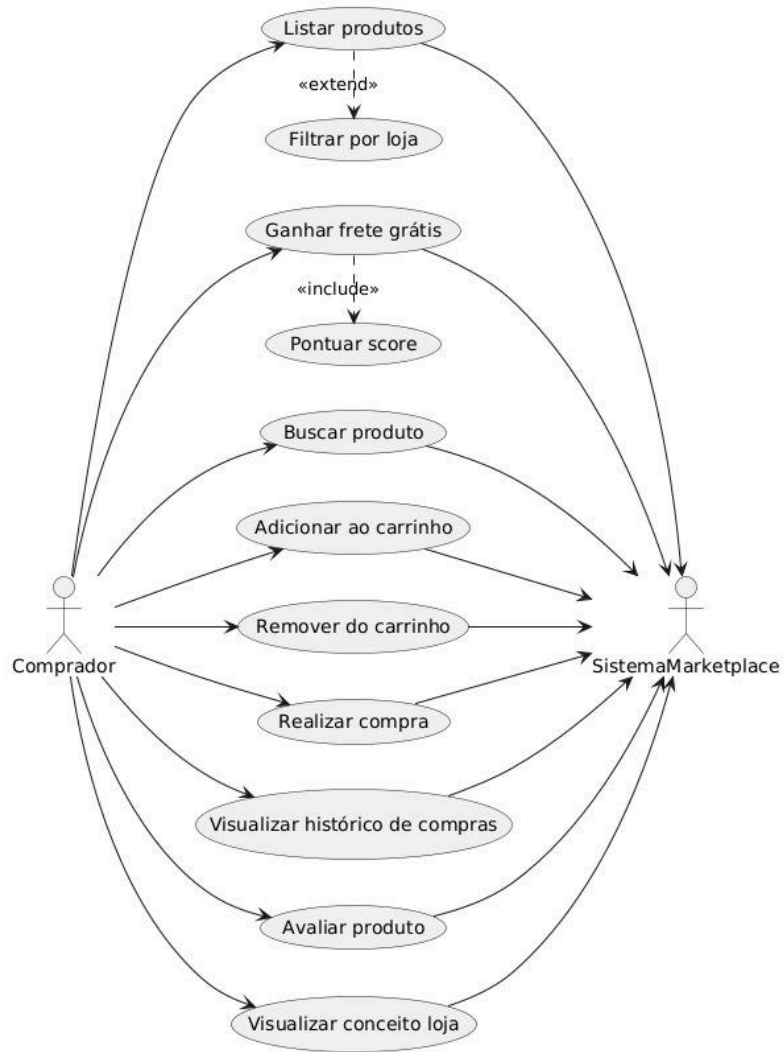
## **3. Vantagem: frete grátis**

Foi estabelecida uma regra de negócio que quando o cliente acumula 5 pontos no score, ele ganha frete grátis em sua próxima compra. Essa verificação pode ser realizada sempre que o score for atualizado ou durante a finalização de uma compra, chamando o método `getCompradorPorCpf(String cpf)` para obter o estado atual do comprador e verificar se o seu score é maior ou igual a 5. Após o uso do benefício, o método `setScore(int score)` pode ser utilizado para reduzir ou zerar os pontos utilizados.

## Diagrama de sequência:



## Diagrama de caso de uso:



# Diagrama de classes:

