

# Relatório da Sprint Scrum

Isabela Yabe

Lavinia Dias

Rodrigo Kalil

17 de outubro

## 1 Introdução

Este documento serve como um relatório da sprint realizada de **03** a **17** de outubro de 2024.

## 2 Backlog da Sprint

### 2.1 Itens de backlog selecionados para a Sprint

- **RF33 - Reportar Problemas:** Como usuario quero poder reportar problemas com as vending machines ou com a rede social e o gestor pode visualizá-las para que sejam resolvidas.
- **RF14 - Perfil da Vending Machine:** Como usuário, quero acessar o perfil de cada vending machine para ver sua localização, produtos disponíveis e histórico de avaliações, me ajudando a escolher a máquina mais adequada.
- **NR A - Visualizar Perfil de Produto:** Como usuário, quero poder visualizar informações de um produto para decidir sobre sua compra.

### 2.2 Itens de backlog preparados para acionamento

- **RF7 - Comentar sobre Produto:** Como usuário, quero poder escrever um comentário sobre um produto para compartilhar minha experiência com outros usuários.
- **RF10 - Visualizar Estoque:** Como gestor, quero visualizar o estoque atual de cada produto em cada vending machine para tomar decisões informadas.
- **RF20 - Relatórios Básicos:** Como administrador ou membro da equipe de manutenção, quero gerar relatórios básicos sobre vendas, avaliações e estoque para monitorar o desempenho das vending machines.

## 3 Resultados da Sprint

- **RF33: Concluída** - Foi feita uma visualização. Foi feita uma lógica de registro das mensagens. O banco de dados é acessado para escrita.
- **RF14: Não Concluída** - Foi feita uma visualização que acessa um mock em json. Foi feito o arquivo de testes e o esboço da classe para conexão ao banco de dados, mas a conexão não foi completamente implementada no contexto desse requisito. Não há uma visualização que acesse o banco de dados.
- **NR A: Não Concluída** - Aqui foi criada uma classe DatabaseManager a fim de ser um intermediador entre o nosso banco de dados MySql e o python. Dessa classe surgirão as classes filhas que serão nossas tabelas de produtos, VM, reclamações, comentários... Também foi criada a classe DatabaseManagerCenter, para controlar globalmente o banco de dados. É nessa instância onde são instanciadas todas demais tabelas e toda manipulação é feita através dessa classe.

Para o front-end foi criado de maneira análoga as mesmas classes como Client, como por exemplo o DatabaseManagerClient, para integrarmos através do flask o back e o front. A finalização do front-end ainda não foi finalizada, falta apenas corrigirmos erro de URL.

- RF7: **Concluída** - Foi feita uma visualização. Foi feita uma lógica similar à da RF33 para registro. O banco de dados é acessado para escrita.
- RF10: **Concluída** - Foi feita a visualização. Foi feita uma lógica de consulta ao banco de dados.

## 4 Retrospectiva

### O que funcionou bem

- O time se comunicou frequentemente, e seguiu o acordo em relação ao uso de testes e à rede de apoio.
- O time avançou na modelagem e implementação da base de dados, o que deve facilitar o trabalho futuro, pois muitos requisitos dependem dessa interação com os dados.
- Após estudos conseguimos entender bem o problema e estruturar o projeto de uma forma coesa.

### O que pode ser melhorado

- No meio da sprint, houve dúvidas quanto à modelagem inicial, que levou o time a buscar o scrum master. Chegou-se a um consenso, mas a modelagem inadequada de início causou atraso.

## 5 Conclusão

Nessa sprint, o time solidificou sua percepção sobre o problema e trabalhou na modelagem de classes e da base de dados. Enquanto o desenvolvimento de alguns itens seguiu de forma autônoma e obteve uma entrega, o de outros, nos quais o time tentou desenvolver o banco de dados e seu acesso a partir de uma mesma base acoplada de código, não foi concluído por dificuldade do time em relação a essa base.

Alguns itens foram, pois, postergados, mas o melhor direcionamento da estruturação da base de dados e do acesso a ela deve tornar o restante da implementação deles mais rápida.