

Shopping Store

Aplicação de e-commerce



Integrantes da Equipe

- **Líder Técnico**

- João Vitor Nascimento Ramos

- **Integrantes**

- Ian Rodrigues Alexandrino
- Leonardo Ribeiro Barbosa Santos
- Murilo Carlos Novais
- Renata C. de Oliveira Moreno
- Danilo da Conceição Santos
- Alexandre de Souza Amaral

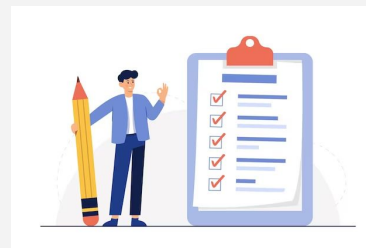


Problema

- O problema consiste na atualização de um sistema de e-commerce, estabelecido há cinco anos, que atualmente lida com a comercialização de relógios, laptops e smartphones.
- Essa solicitação surge em consonância com as novas diretrizes do modelo de negócios da empresa e visa manter sua competitividade no mercado.
- A modernização proposta implica uma transição estratégica, direcionando o sistema para a venda de produtos tecnológicos de alto valor agregado, como os servidores das marcas Dell e Positivo.
- Diante desse contexto, torna-se crucial reestruturar e atualizar o sistema de e-commerce para atender aos padrões tecnológicos contemporâneos.
- Essas melhorias incluíram a atualização da versão do framework Java, a otimização do backend e a remodelagem do banco de dados.

Objetivos

- Atualizar, adaptar e otimizar um e-commerce desenvolvido há cinco anos
- Estimular o trabalho em equipe, promover a comunicação eficaz e desafiar os participantes a resolver problemas complexos
- Demonstrar profundo entendimento dos conceitos avançados de Java (abordados na residência)
- Adotar boas práticas de desenvolvimento e mostrar capacidade para atender a novas demandas tecnológicas
- Demonstrar criatividade e proatividade na busca por soluções inovadoras



Identificação do pontos negativos

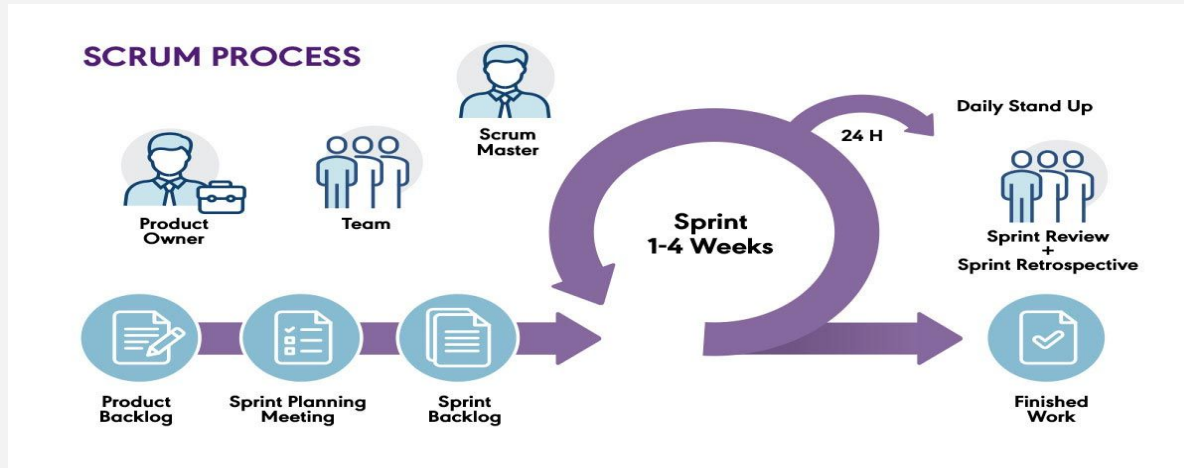
- Versão do Spring Boot
- Versão do Java
- Uso de banco de dados Mongo DB
- Versões desatualizadas de dependências
- Sistema não funciona

Planejamento de melhorias

- Atualização da versão do Spring Boot para 3.2.5
- Atualização do Java para versão 17
- Atualização das dependências.
- Re-modelagem do sistema para um banco de dados relacional PostgreSQL.
- Inserção de novas dependências ao projeto
- Aplicação de padrões de projeto nas verificações de dados em transições de cadastro, update e delete.
- Expansão do uso de spring security para funcionalidades de segurança do sistema

Metodologia de gerenciamento

- Foi optado pela metodologia Scrum para o desenvolvimento do projeto, principalmente devido à sua abordagem ágil e iterativa.

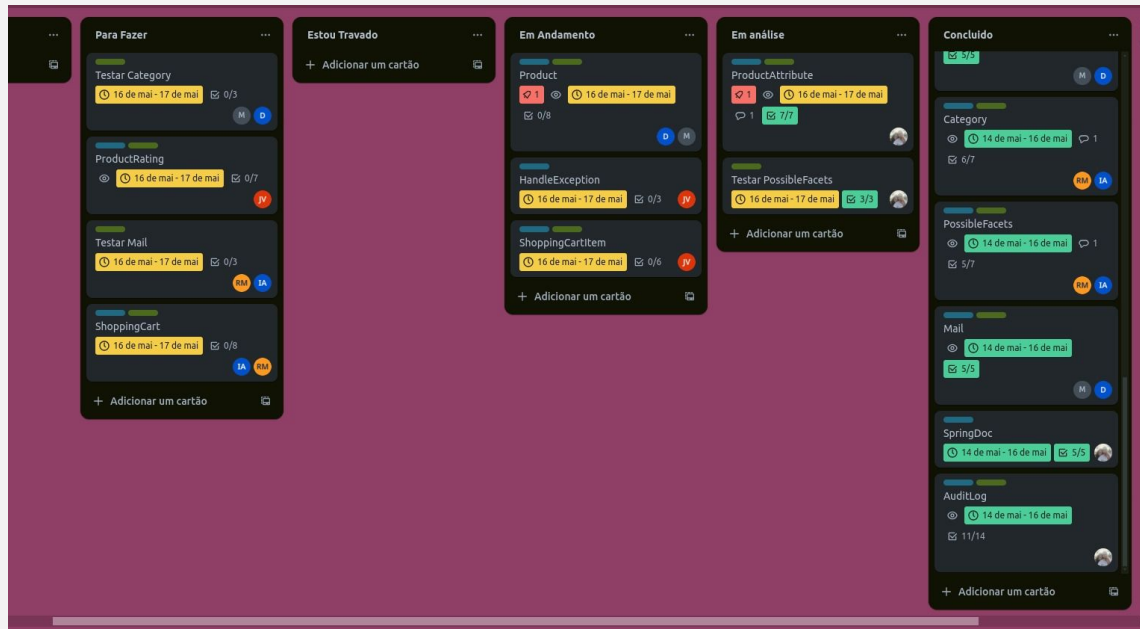


Ferramenta de gerenciamento

- A ferramenta escolhida para o gerenciamento do projeto foi o Trello, devido à sua simplicidade, flexibilidade e capacidade de promover a colaboração da equipe.
- O Trello oferece uma interface intuitiva que facilita a criação, organização e acompanhamento de tarefas, tornando-o ideal para uma equipe que enfrenta prazos apertados e está se familiarizando com novas práticas de trabalho.



Ferramenta de gerenciamento



Estratégia para Versionamento

- Foi definido o uso do Git como sistema de controle de versão, com a criação de um repositório no GitHub.
- A estratégia adotada envolve a criação de branches individuais para cada tarefa ou funcionalidade, que serão posteriormente 'mergeadas' na branch "desenvolvimento".
- Após revisão e testes, as alterações serão então 'mergeadas' na branch "main", que representa a versão estável do sistema.



git



GitHub

Gestão de Riscos

- **Interoperabilidade entre grupo**

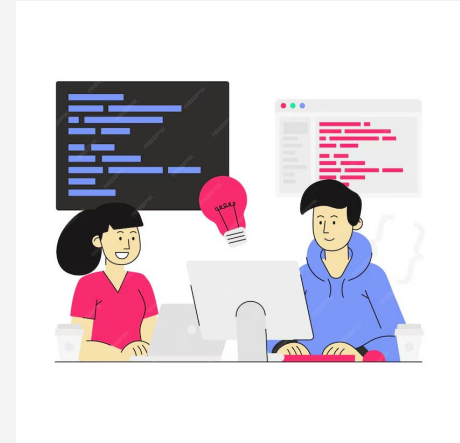
- Como forma de garantir que qualquer membro da equipe consiga dar continuidade ao trabalho de outro, sem que haja desperdício de tempo com análise de código, foi estabelecido um Framework Interop Group e adotado padrões de escrita de código.

- **Mitigar atrasos e evitar bugs**

- Adotamos a estratégia de programação em pares dividindo nossa força de trabalho em sub-equipes de 2 pessoas como forma de agilizar a execução das atividades e contingenciar a entrega caso um membro da equipe fique indisponível ou simplesmente não consiga executar a tarefa.
- Somado a isso, são realizadas reuniões diárias onde cada um dos participantes do grupo relata como foi o desenvolvimento de suas atividades e se houve alguma dificuldade ou impedimento na execução da tarefa.

Sub-equipes

Sub-equipe	Integrantes	
1	Leonardo Ribeiro Barbosa Santo	Alexandre de Souza Amaral
2	Danilo da Conceição Santos	Murilo Carlos Novais
3	Ian Rodrigues Alexandrino	Renata C. de Oliveira Moreno
4	João Vitor Nascimento Ramos	

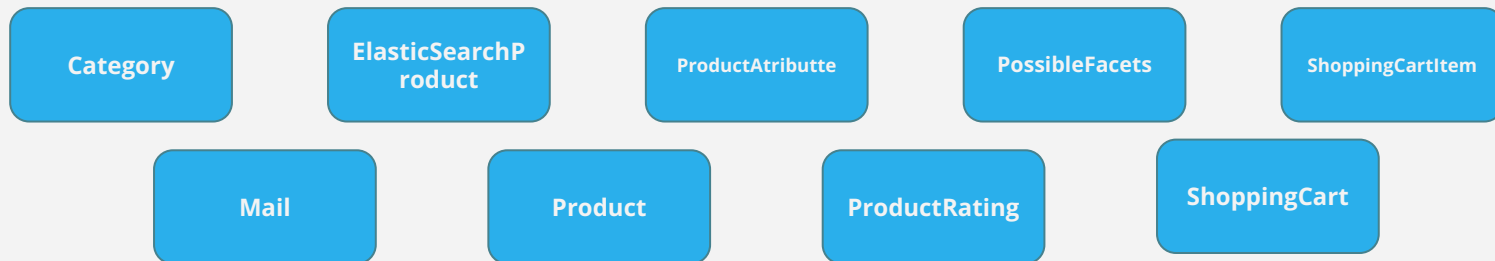


Desenvolvimento do Projeto

- **Planejamento Inicial**

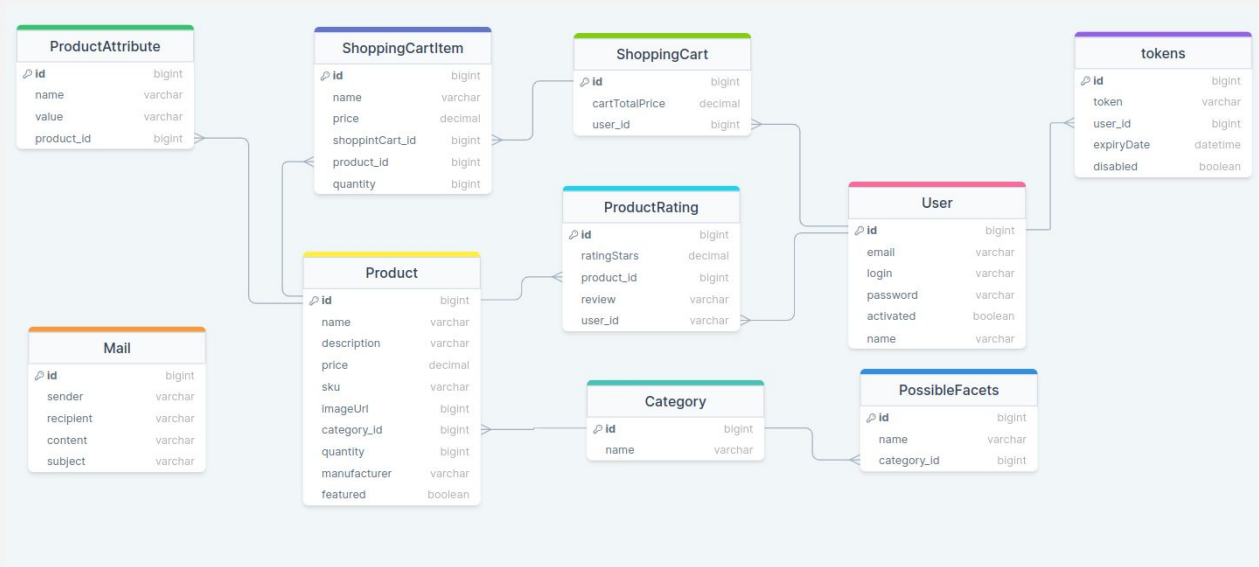
- Foi feita uma análise das entidades e da regra de negócio para entender o funcionamento do sistema legado.

- **Banco de Dados Legado**



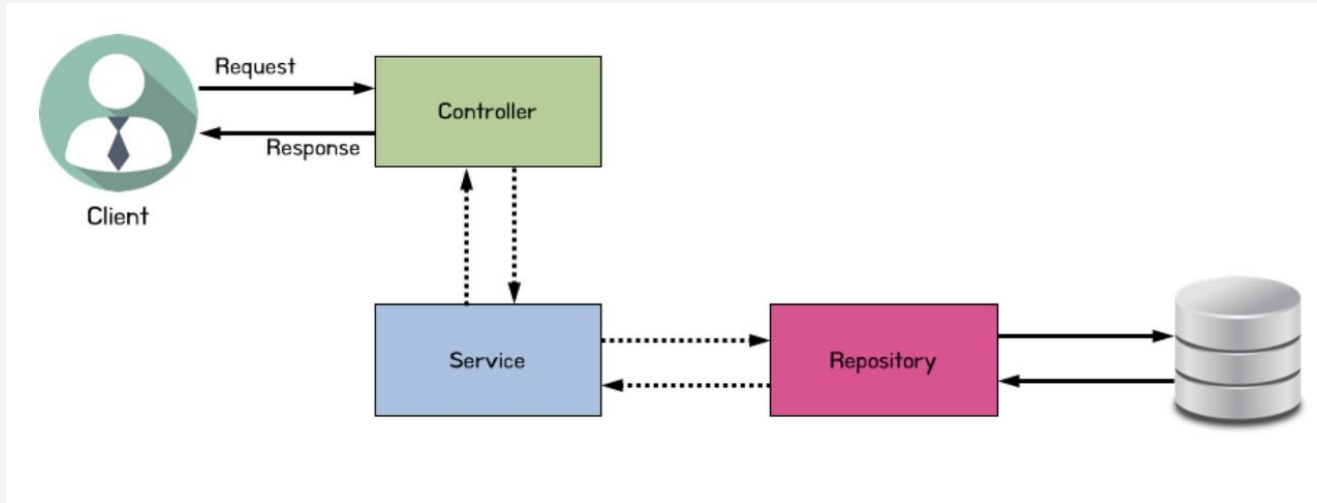
Desenvolvimento do Projeto

- Modelagem para banco de dados relacional PostgreSQL



Desenvolvimento do Projeto



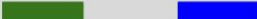
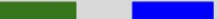


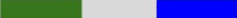



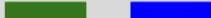
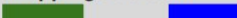



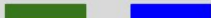




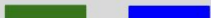
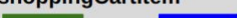








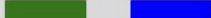







- Modelo de camadas



Desenvolvimento do Projeto

- **Product Backlog**

- O backlog do produto é a lista priorizada de trabalho para a equipe de desenvolvimento, derivado do roteiro do produto e seus requisitos.
- Uma lista de pendências ágil bem organizada não só torna o planejamento da liberação e da iteração (sprint) mais fácil, como também transmite tudo em que sua equipe pretende trabalhar e faz com que o tempo da equipe seja fixo.

Configurar Spring Doc 	Implemetar package AuditLog 	Implementar Entity e Repository Mail 	Implementar Entity e Repository Category 	Inserir dependencias 
Implementar Package Security 	Implementar Entity e Repository PossibleFacets 	Criar Entity e Repository ProductAttribute 	Criar Entity e Repository Product 	Criar Entity e Repository ShoppingCart 
Criar Entity e Repository ProductRating 	Criar Entity e Repository ShoppingCartItem 	Implementar HandleException 	Criar e testar Service ShoppingCart 	Criar e testar Service ProductAttribute 
Criar e testar Service ShoppingCartItem 	Criar e testar Service ProductRating 	Criar e testar Service Mail 	Criar e testar Service Category 	Criar e testar Service Product 
Criar e testar Service PossibleFacets 	Criar e testar Controller shoppingCartItem 	Criar e testar Controller ShoppingCart 	Criar e testar Controller Category 	Criar e testar Controller ProductAttribute 
Criar e testar Controller Product 	Criar e testar Controller PossibleFacets 	Criar e testar Controller ProductRating 	Implementar Envio de Email Para registro do usuario 	Implementar Envio de Email Para recuperação de senha do usuario 
Implementar Persistencia dos emails enviados 	Comentarios Spring Doc nos Controllers 	Implementar ValidationMessages 	Popular banco de Dados 	Criar e testar Entity Payment 
Criar e testar Service Payment 	Criar e testar Controller Payment 	Configurar Permissão nos controllers 		

Desenvolvimento do Projeto

- **Planejamento das sprints**

- O projeto foi apresentado à equipe no dia 13 de maio, marcando o início de uma série de reuniões e planejamentos, incluindo a execução de 7 sprints, com o objetivo de concluir a solução até o dia 24 de maio.

Sprint	Data de Inicio	Data de Fim
1	14 de maio, 12:00	16 de maio, 11:00
2	16 de maio, 15:00	17 de maio, 11:00
3	17 de maio, 15:00	20 de maio, 11:00
4	20 de maio, 15:00	21 de maio, 11:00
5	21 de maio, 15:00	22 de maio, 11:00
6	22 de maio, 15:00	23 de maio, 11:00
7	23 de maio, 15:00	24 de maio, 11:00

Configurar Spring Doc ██████████	Implemetar package AuditLog ██████████	Implementar Entity e Repository Mail ██████████	Implementar Entity e Repository Category ██████████	Inserir dependencias ██████████
Implementar Package Security ██████████	Implementar Entity e Repository PossibleFacets ██████████	Criar Entity e Repository ProductAtributte ██████████	Criar Entity e Repository Product ██████████	Criar Entity e Repository ShoppingCart ██████████
Criar Entity e Repository ProductRating ██████████	Criar Entity e Repository ShoppingCartItem ██████████	Implementar HandleException ██████████	Criar e testar Service ShoppingCart ██████████	Criar e testar Service ProductAtributte ██████████
Criar e testar Service ShoppingCartItem ██████████	Criar e testar Service ProductRating ██████████	Criar e testar Service Mail ██████████	Criar e testar Service Category ██████████	Criar e testar Service Product ██████████
Criar e testar Service PossibleFacets ██████████	Criar e testar Controller shoppingCartItem ██████████	Criar e testar Controller ShoppingCart ██████████	Criar e testar Controller Category ██████████	Criar e testar Controller ProductAtributte ██████████
Criar e testar Controller Product ██████████	Criar e testar Controller PossibleFacets ██████████	Criar e testar Controller ProductRating ██████████	Implementar Envio de Email Para registro do usuario ██████████	Implementar Envio de Email Para recuperação de senha do usuario ██████████
Implementar Persistencia dos emails enviados ██████████	Comentarios Spring Doc nos Controllers ██████████	Implementar ValidationMessages ██████████	Popular banco de Dados ██████████	Criar e testar Entity Payment ██████████
Criar e testar Service Payment ██████████	Criar e testar Controller Payment ██████████	Configurar Permissão nos controllers ██████████		

Sprint	Cor
1	
2	
3	
4	
5	
6	
7	
8	

Desenvolvimento de Projeto

- **Sprint 1**

- Foi apresentado aos colaboradores as regras para a interoperabilidade do grupo.

1	2	3	4
Implementar e testar package AuditLog	Inserir dependências	Implementar e testar Entity e Repository Category	Implementar e testar o package Security
Configurar Spring Doc	Implementar e testar Entity e Repository Mail	Implementar e testar Entity e Repository PossibleFacets	

- Todos os erros identificados nessa sprint foram corrigidos. Os testes não implementados foram realocados como tarefas para próxima sprint.

Desenvolvimento do Projeto

- **Sprint 2**

- Foi apresentado aos colaboradores o plugin JaCoCo responsável por ilustrar a cobertura de testes. A cobertura atual de testes estava em 40%

1	2	3	4
Testar Entity e Repository PossibleFacets	Implementar e testar Entity e Repository Product	Implementar e testar Entity e Repository ShoppingCart	Implementar e testar Entity e Repository ProductRating
Implementar e testar Entity e Repository ProductAttribute	Testar Entity e Repository Category	Testar Entity e Repository Mail	Implementar e testar Entity e Repository ShoppingCartItem
			Implementar e testar HandleException

- Devido aos padrões adotados, foi possível que equipes diferentes pudessem facilmente testar entidades que não haviam escrito.
- A cobertura de testes passou de 40% para 65%.

Desenvolvimento do Projeto

- **Sprint 3**

- Após a conclusão do desenvolvimento das camadas de entities e repositories na sprint 2, deu-se início à implementação da camada de services.

1	2	3	4
Implementar e testar Service Category	Implementar e testar Service Product	Implementar e testar Service ProductAttribute	Implementar e testar Service ProductCart
Implementar e testar Service PossibleFacets	Implementar e testar Service ProductRating	Implementar e testar Service Mail	Implementar e testar Service ProductCartItem
			Buscar aumentar cobertura de testes

- Os colaboradores apresentaram uma certa dificuldade na injeção inversa de dependência. No entanto conseguiram concluir o trabalho esquecendo de apenas algumas validações.
- A cobertura de testes subiu de 65% para 71%.

Desenvolvimento do Projeto

- **Sprint 4**

- Nesta sprint foi destinado o prosseguimento na criação dos métodos dos services, introduzindo os métodos de update e disable.

1	2	3	4
Implementar e testar Service Category	Implementar e testar Service Product	Implementar e testar Service ProductAttribute	Implementar e testar Service ProductCart
Implementar e testar Service PossibleFacets	Implementar e testar Service ProductRating	Implementar e testar Service Mail	Implementar e testar Service ProductCartItem
			Buscar aumentar cobertura de testes

- Apesar do aumento na quantidade de linhas de código, a cobertura de testes permaneceu em 71%.
- Não foram mais detectados erros significativos na estrutura do código.

Desenvolvimento do Projeto

- **Sprint 5**

- Com os services parcialmente prontos, chegou o momento de iniciar o desenvolvimento dos controllers. Inicialmente, não foram solicitados métodos de teste para esses controllers.

1	2	3	4
Implementar Controller Category	Implementar Controller Product	Implementar Service ProductAttribute	Pesquisar como preencher o banco de dados
	Implementar Controller ProductRating	Implementar Controller PossibleFacets	Implementar o serviço de e-mail no controlador responsável por disparar e-mails.

- A cobertura de testes caiu para 62%.
- Banco de dados não foi preenchido

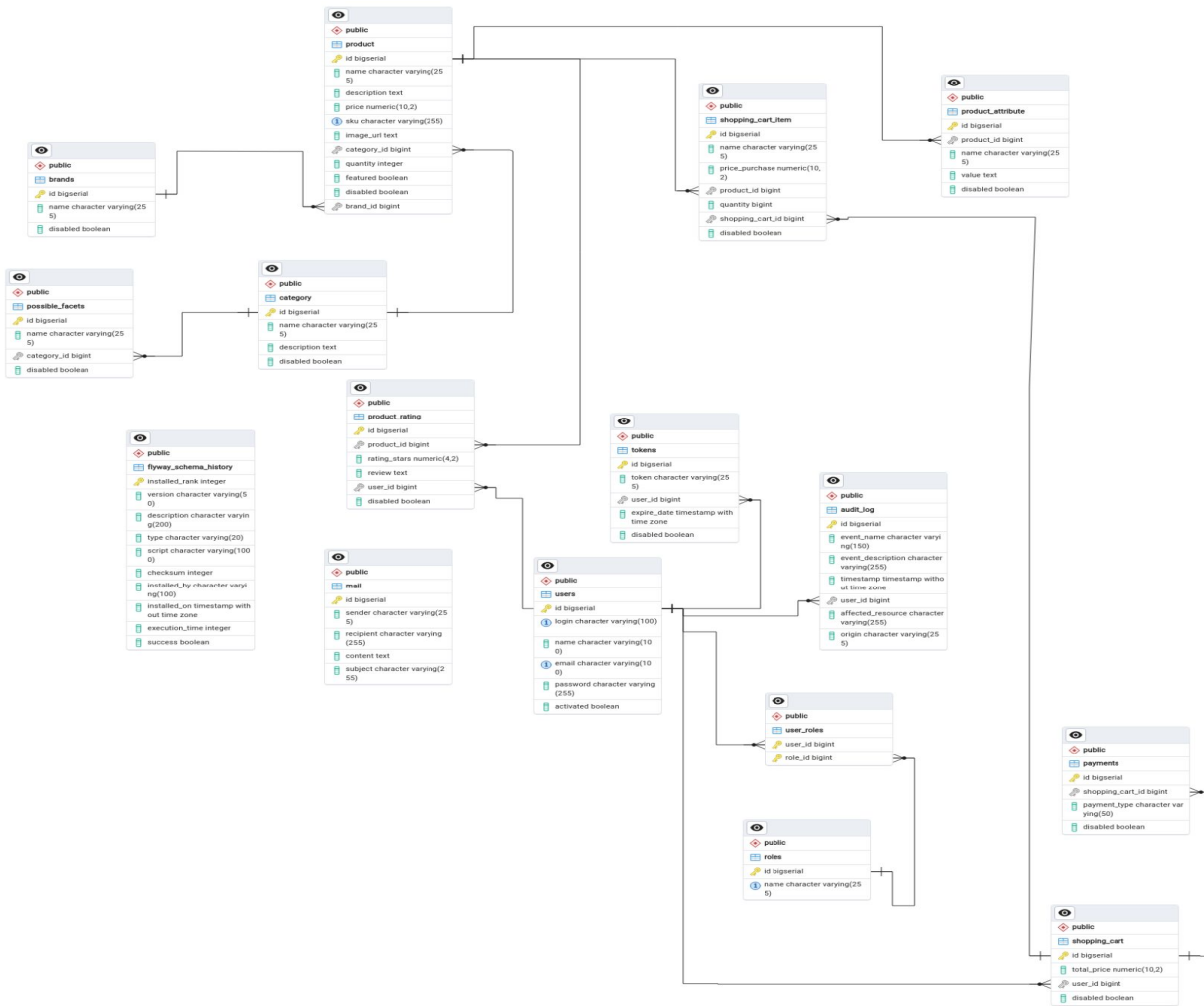
Desenvolvimento do Projeto

- **Sprint 6**

- O intuito principal dessa sprint foi aumentar a cobertura de testes e terminar a implementação dos controllers.

1	2	3	4
Aumentar Cobertura de Testes	Implementar Controller ProductRating	Implementar novos metodos de busca em todos os services	Implementar Controller Para o Shopping-Cart
	Implementar Controller ProductAttribute	Implementar ValidationMessages.properties	Implementar Controller Para o ShoppingCartItem.
		Implementar comentarios Spring Doc nos controllers	

- Apesar de aumentar a quantidade de testes realizados, a cobertura de código caiu de 65% para 58%.



Desenvolvimento do Projeto

- **Sprint 7**

- Essa sprint consistiu basicamente em aumentar a cobertura de testes e popular o banco de dados.

1	2	3	4
Aumentar cobertura de testes nos validations	Aumentar cobertura de testes nos <i>controllers</i>	Aumentar cobertura de testes nos <i>services</i>	Aplicar seeders
Aumentar cobertura de testes nas entitys			Aplicar controle de permissão aos controllers.

- Nessa sprint conseguimos elevar a cobertura de 58% para 66%. Conseguimos popular o banco de dados.

Desenvolvimento do Projeto

- **Sprint 8**

- Essa sprint não estava no planejamento e ocorreu virtude do adiamento do prazo. Dessa forma, decidimos concentrar nossos esforços na melhoria dos testes.

1	2	3	4
Aumentar cobertura de testes nos validations	Aumentar cobertura de testes nos <i>controllers</i>	Aumentar cobertura de testes nos <i>services</i> remanescentes	Finalizar relatorio
Aumentar cobertura de testes no Serviceshop-pingCartItem	Aumentar cobertura de testes no Service ProductRating		

- Nessa sprint conseguimos elevar a cobertura de 66% para 84%.

Funcionalidades Desenvolvidas

- **Segurança**

- Registro de usuário
- Ativação do usuário
- Login
- Recuperação de senha



Funcionalidades Desenvolvidas

- **Brand**

- Registrar Brand
- Buscar Brand por ID
- Buscar todas as Brands
- Atualizar Brand
- Desabilitar Brand



Funcionalidades Implementadas

- **Category**
 - Registrar Category
 - Buscar Category por ID
 - Buscar todas as categories
 - Buscar categorias por nome
 - Atualizar Category
 - Desabilitar Category

Funcionalidades Implementadas

- **PossibleFacets**

- Registrar Atributo de categoria
- Buscar Todas as PossibleFacets
- Buscar PossibleFacets por ID
- Buscar possibleFacets por categoria
- Atualizar PossibleFacets
- Desabilitar PossibleFacets

Funcionalidades Implementadas

- **Product**
 - Registrar Produto
 - Listar Produtos
 - Buscar Produto por ID
 - Buscar Produtos por Categoria
 - Atualizar Produto
 - Desabilitar Produto

Funcionalidades Implementadas

- **ProductAttribute**

- Registrar ProductAttribute
- Listar ProductAttributes
- Listar ProductAttributes Desabilitados
- Buscar ProductAttributes por ID
- Buscar ProductAttributes por Produto
- Buscar ProductAttributes Desabilitados por Produto
- Atualizar ProductAttribute
- Desabilitar ProductAttribute

Funcionalidades Implementadas

- **ProductRating**

- Registrar Avaliação de Produto
- Listar Avaliações de Produto
- Buscar Avaliação de Produto por ID
- Buscar Avaliações por Produto
- Buscar Avaliações por Usuário
- Atualizar Avaliação de Produto
- Desabilitar Avaliação de Produto



Funcionalidades Implementadas

- **Purchase**
 - Realizar Compra
 - Cancelar Compra



Funcionalidades Implementadas

- **Payments**
 - Registrar Pagamento
 - Listar Todos os Pagamentos
 - Listar Pagamentos por Usuário
 - Desabilitar Pagamento



Dúvidas?

