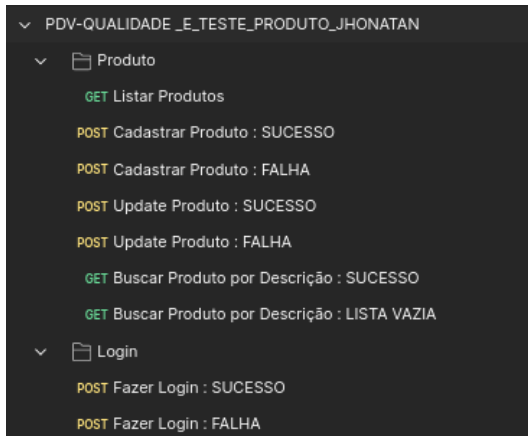


## Entrega 2 - JHONATAN

### 1. Testes Unitários e de Integração

- Melhorar e aumentar os casos de testes unitários, isolando suas dependências (**FEITO**)
- Implementar casos de teste de integração (**FEITO**)

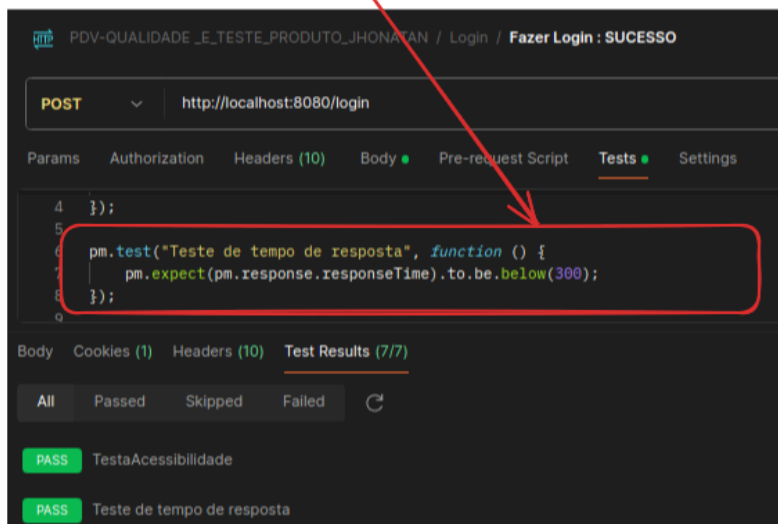


### 2. Testes de Sistema e Requisitos Não Funcionais

- Implementar testes de sistema considerando requisitos funcionais (**FEITO**)
  - **Selenium:**
    - Login (**FEITO**)
    - Buscar Produto (**FEITO**)
    - Cadastrar produto (**FEITO**)
- Pelo menos um requisito não funcional (atributos de qualidade - Exemplo: desempenho, segurança) (**FEITO**)
  - Realizado teste de desempenho no login

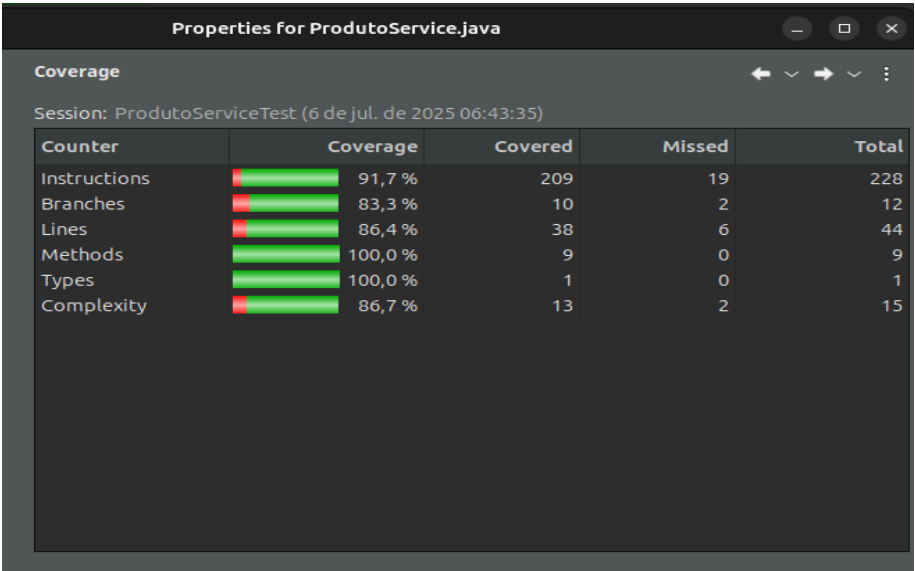
#### Teste de desempenho

#### Testando tempo de resposta da requisição de login

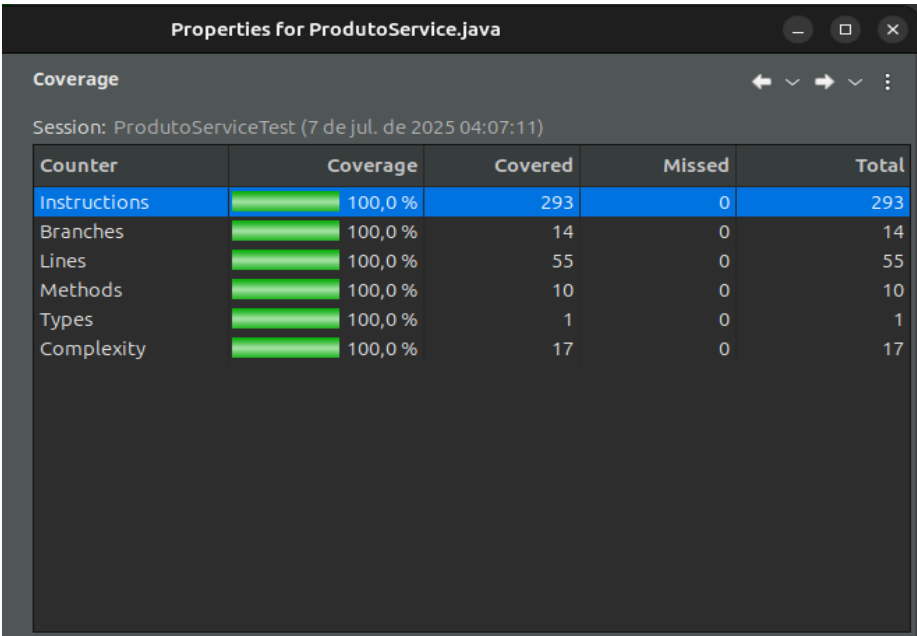


3. Testes Estruturais (Cobertura de Código)(FEITO)

- Estrutural (ao menos 80% de cobertura no critério todas-arestas)
  - **Classe de ProdutoService:**
    - Coverage classe de teste - ENTREGA 1

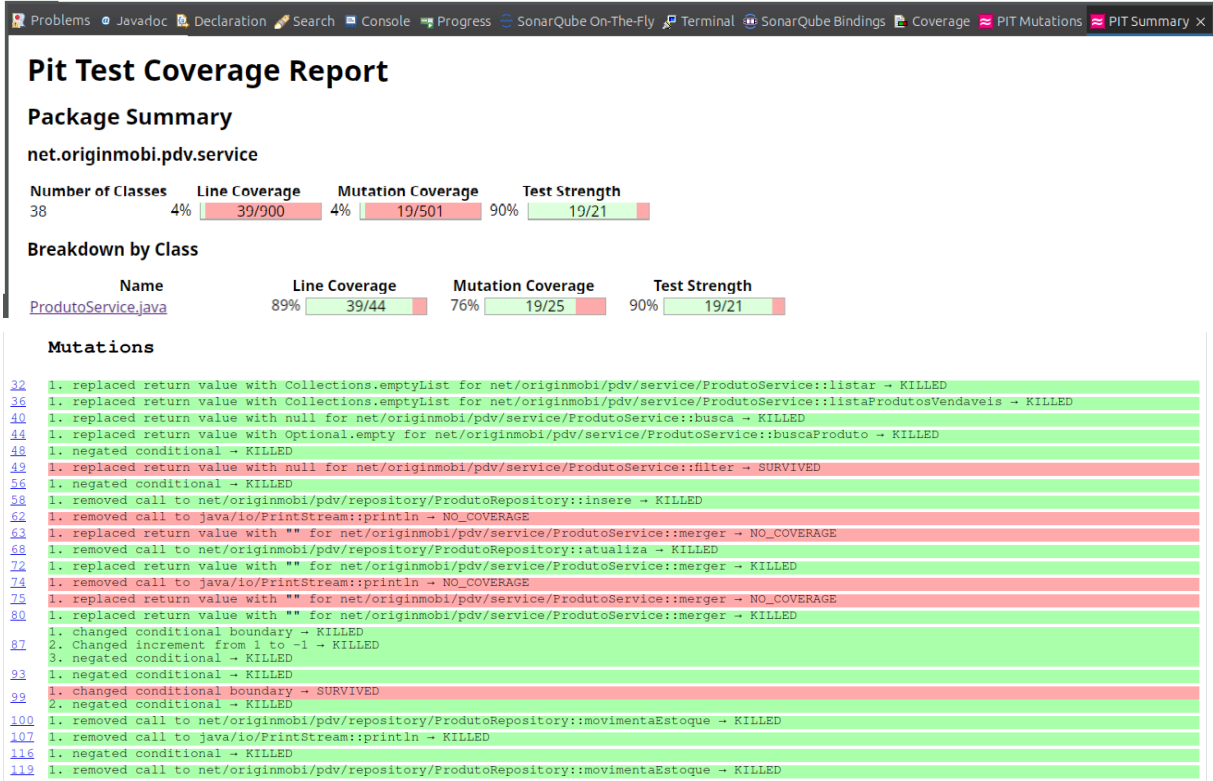
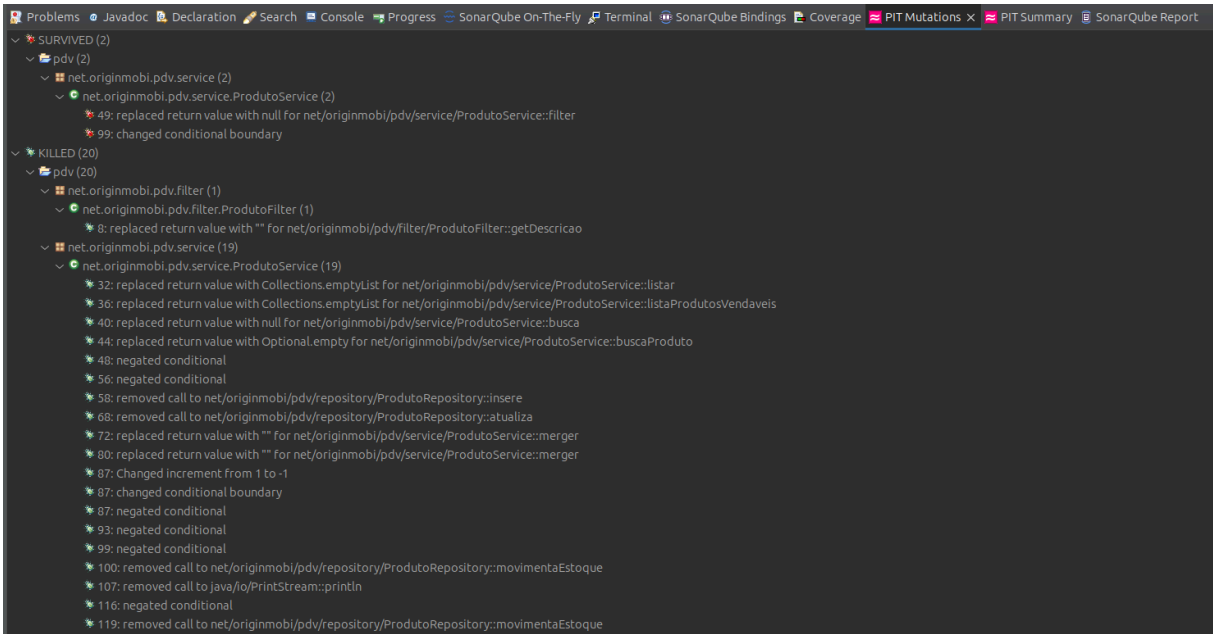


- Coverage classe de teste - ENTREGA 2

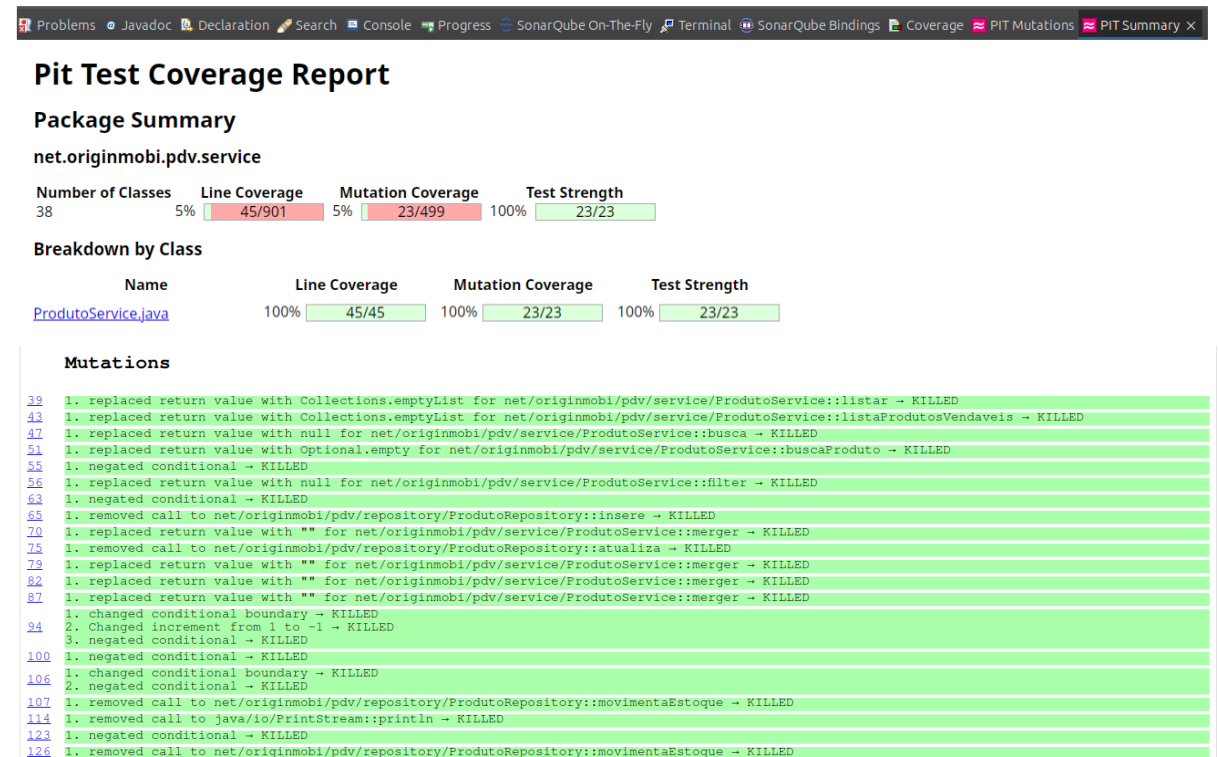
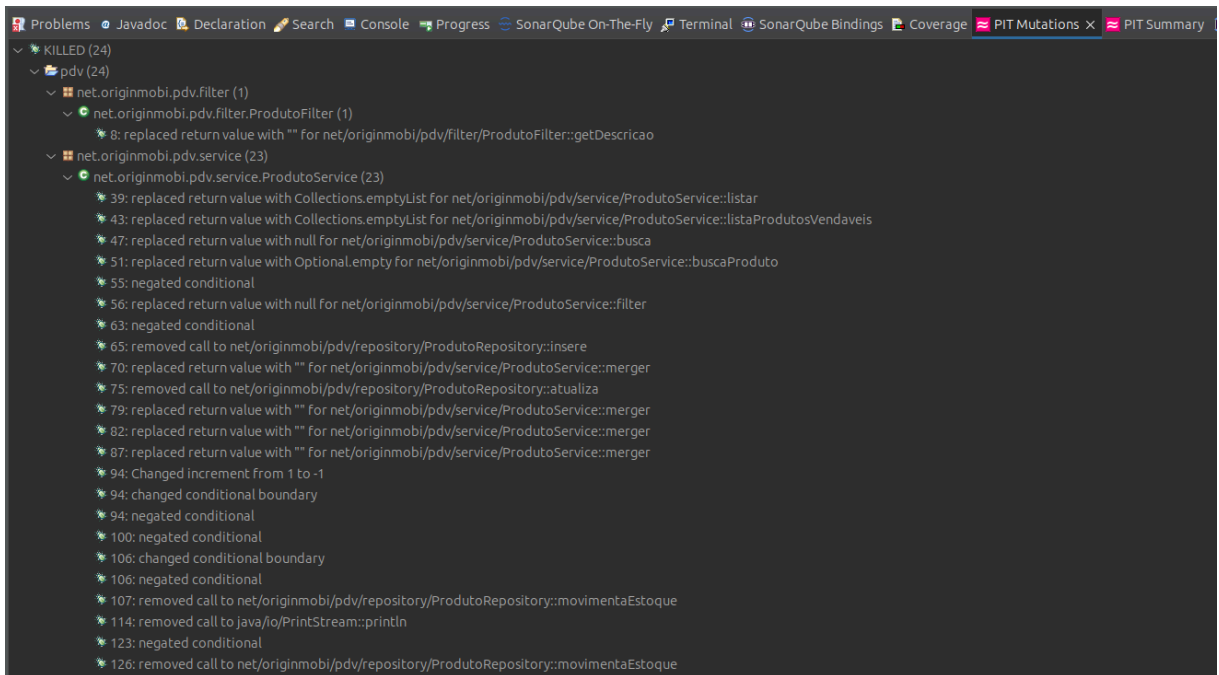


4. Testes Baseados em Defeitos (Muta  o)(FEITO)

- Baseada em Defeitos (80% de escore de muta  o nas mesmas classes do teste estrutural)
  - Classe de ProdutoService:
    - PIT Mutation Test classe de teste - ENTREGA 1

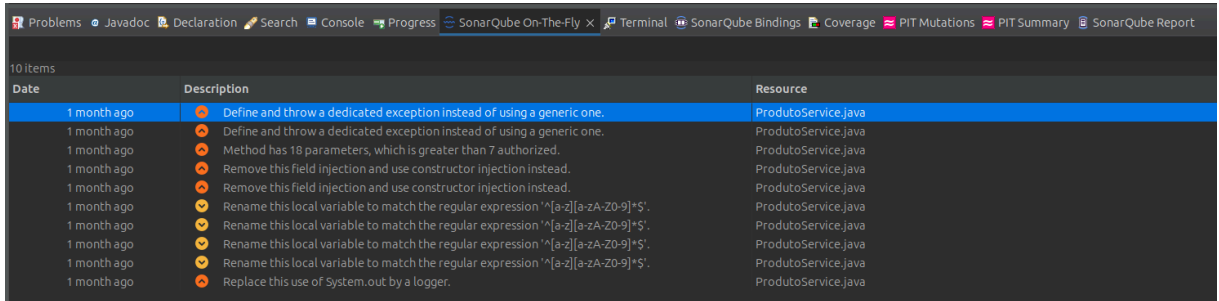


## ■ PIT Mutation Test classe de teste - ENTREGA 2



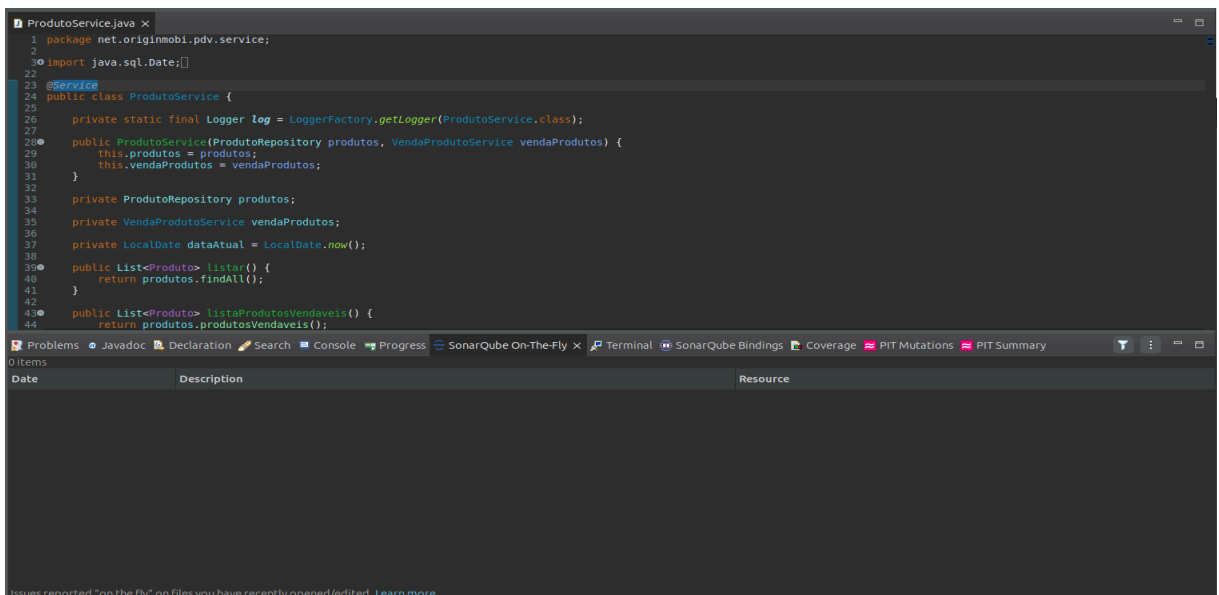
## 5. Relatório de Inspeção do Código-Fonte (SonarQube)(FEITO)

- Relatório de inspeção do código-fonte (usando SonarQube).
  - Análise utilizando SonarQube da classe ProdutoService - ENTREGA 1



Date	Description	Resource
1 month ago	Define and throw a dedicated exception instead of using a generic one.	ProdutoService.java
1 month ago	Define and throw a dedicated exception instead of using a generic one.	ProdutoService.java
1 month ago	Method has 18 parameters, which is greater than 7 authorized.	ProdutoService.java
1 month ago	Remove this field injection and use constructor injection instead.	ProdutoService.java
1 month ago	Remove this field injection and use constructor injection instead.	ProdutoService.java
1 month ago	Rename this local variable to match the regular expression '[a-z][a-zA-Z0-9]*\$'.	ProdutoService.java
1 month ago	Rename this local variable to match the regular expression '[a-z][a-zA-Z0-9]*\$'.	ProdutoService.java
1 month ago	Rename this local variable to match the regular expression '[a-z][a-zA-Z0-9]*\$'.	ProdutoService.java
1 month ago	Rename this local variable to match the regular expression '[a-z][a-zA-Z0-9]*\$'.	ProdutoService.java
1 month ago	Replace this use of System.out by a logger.	ProdutoService.java

- Análise utilizando SonarQube da classe ProdutoService - ENTREGA 2



Date	Description	Resource
------	-------------	----------

```
1 package net.originmobi.pdv.service;
2
3 import java.sql.Date;
4
5 @Service
6 public class ProdutoService {
7
8     private static final Logger log = LoggerFactory.getLogger(ProdutoService.class);
9
10    public ProdutoService(ProdutoRepository produtos, VendaProdutoService vendaProdutos) {
11        this.produtos = produtos;
12        this.vendaProdutos = vendaProdutos;
13    }
14
15    private ProdutoRepository produtos;
16    private VendaProdutoService vendaProdutos;
17    private LocalDate dataAtual = LocalDate.now();
18
19    public List<Produto> listar() {
20        return produtos.findAll();
21    }
22
23    public List<Produto> listarProdutosVendaveis() {
24        return produtos.produtosVendaveis();
25    }
26}
```

## 6. Melhorias Implementadas(FEITO)

- Criação de exception personalizado
- Criação de DTO ProdutoMergerDTO para o método merger com objetivo de qualidade de código e reduzir complexidade
- Ajuste no nome das variáveis
- Limpeza nos '@Autowired' desnecessários
- Adição de dependência Lombok para código mais limpo
- Ajuste nos testes para se adequar às melhorias
- Ajuste no ProdutoController para se adequar ao método merger usando o novo DTO ProdutoMergerDTO