Documento Final - Trabalho Prático 2

Alunos: Amanda Ingrid, Bruno Silveira, Gabriel Gomide e Mateus Samartini

O software selecionado para o Trabalho Prático 2 foi uma aplicação web/mobile sobre gestão de condomínios, desenvolvida neste semestre, na disciplina Trabalho Interdisciplinar: Aplicações Distribuídas. O sistema em si foi desenvolvido em Flutter no front-end, com Java Spring no back-end.

Com ênfase na última tecnologia citada acima, os testes realizados foram baseados na estratégia de testes de caixa branca, em que foram desenvolvidos casos de testes unitários para cada método público implementado no sistema, contemplando no total 14 classes e 46 métodos testados diretamente. De modo a garantir que os métodos estão funcionando de acordo com a forma esperada tanto em seus fluxos principais, quanto em seus fluxos alternativos. Assim, assegurando a obtenção de maior qualidade no produto desenvolvido.

Em que, para o desenvolvimento desses testes foram utilizadas as seguintes tecnologias: o JUnit, que consiste em um framework open source para o desenvolvimento de testes unitários em Java e o Mockito, que é um framework open source para Java que permite simular o comportamento de outras classes, isso é, este auxilia a testar a lógica de um método que depende de métodos de outras classes.

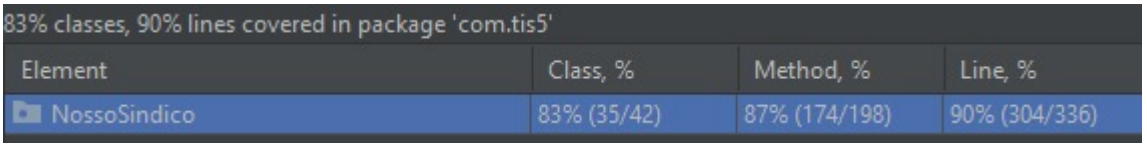
No que tange aos testes realizados, foram implementados 82 casos de testes referentes aos métodos das camadas de serviço e controle do backend da aplicação, em que as demais camadas, de domínio e de repositório, não seriam testadas diretamente, uma vez que essas representam apenas abstrações das entidades e interfaces de acesso aos dados, não possuindo regras de negócio. Entretanto, ao longo do processo de testes, essas classes foram utilizadas por meio de mocks no caso dos repositórios, visto que a estratégia era testar a lógica dos métodos não a integração entre os componentes, e pelo uso de construtores, getters e setters no caso das classes de domínio.

Dessa forma, visando obter a maior cobertura de teste possível, foi elaborado ao menos um caso para cada método, de modo que aqueles métodos que possuíssem fluxos alternativos relevantes tiveram mais casos elaborados, a

princípio, um novo caso para cada fluxo identificado. Com isso, obtendo uma cobertura satisfatória de métodos e linhas de código.

Para realizar o cálculo da cobertura de testes foram utilizadas as próprias ferramentas disponibilizadas pela IDEA Intellij, que apresenta uma opção de rodar toda a suíte de testes com cobertura, que já disponibiliza além dos resultados de todos os testes outras informações relevantes sobre a cobertura atingida, como o percentual de classes, métodos e linhas alcançadas no total, e também esses mesmos dados especificados para cada pacote e classe. Assim, permitindo precisar com segurança a cobertura de casos de teste alcançada no todo e em cada classe especificamente. Ademais, ainda vale ressaltar que ainda é possível visualizar visualmente a cobertura das linhas dentro dos métodos de cada classe, o que ajuda a indicar possíveis fluxos não testados. Para isso, é disposto no canto esquerdo da tela tracejados verdes indicando que determinada linha foi testada, e tracejados vermelhos caso contrário.

A seguir, seguem os resultados obtidos pela cobertura de testes:
Foram desenvolvidos 82 casos de teste, os quais obtiveram cobertura de 83% das classes, 87% dos métodos e 90% das linhas.



Ademais, vale ressaltar os resultados específicos de cada classe, em que estão sendo dispostas a cobertura de linhas e de métodos de cada uma:


java 83% classes, 90% lines covered
com.tis5.NossoSindico 83% classes, 90% lines covered
Controller 100% classes, 98% lines covered
ApartamentoController 100% methods, 100% lines covered
AvisoController 100% methods, 100% lines covered
CondominioController 100% methods, 95% lines covered
DespessaController 100% methods, 100% lines covered
EspacoController 100% methods, 100% lines covered
ReservaController 100% methods, 100% lines covered
UsuarioController 100% methods, 100% lines covered
domain 77% classes, 85% lines covered
AcessoCondominioResource 62% methods, 62% lines covered
Apartamento 100% methods, 100% lines covered
Aviso 100% methods, 100% lines covered
AvisoResource 62% methods, 62% lines covered
CadCondAptoResource 78% methods, 78% lines covered
Condominio 100% methods, 100% lines covered
Despessa 100% methods, 100% lines covered
DespessaResource 100% methods, 100% lines covered
Espaco 70% methods, 70% lines covered
LoginUsuario 57% methods, 57% lines covered
Reserva 70% methods, 70% lines covered
ReservaResource 88% methods, 88% lines covered
SindicoResource 80% methods, 80% lines covered
Usuario 100% methods, 100% lines covered
UsuarioComApto 90% methods, 90% lines covered
repository
Service 100% classes, 92% lines covered
ApartamentoService 100% methods, 100% lines covered
AvisoService 100% methods, 100% lines covered
CondominioService 88% methods, 96% lines covered
DespessaService 100% methods, 83% lines covered
EspacoService 100% methods, 100% lines covered
ReservaService 100% methods, 78% lines covered
UsuarioService 87% methods, 88% lines covered
NossoSindicoApplication 0% methods, 0% lines covered