

Aluno: Lucas de Souza Ferreira

Matéria: Projeto de Bloco: Engenharia Disciplinada de Software

TP4



TP4: Integração, Refatoração e CI/CD - Relatório de Evidências

Aluno: Lucas de Souza Ferreira **Disciplina:** Projeto de Bloco **Data:** 08/12/2025

1. Integração dos Sistemas (Requisito 2)

Descrição: Evidências da integração entre o Sistema de Eventos e o Sistema de Usuários.

Evidência 1: Código de Integração no Service

Objetivo: Demonstrar o código da classe `EventoService.java` focado no método `criarEvento`, destacando a linha onde `usuarioService.buscarPorId(usuarioId)` é chamado.

```
public Evento criarEvento(Evento evento, Long usuarioId) {  
    // 1. Validação de regra de negócio (nome duplicado)  
    validarNomeDuplicado(evento.getNome(), null);  
  
    // 2. Busca e associa o usuário (Integração)  
    Usuario criador = usuarioService.buscarPorId(usuarioId);  
    evento.setUsuario(criador);  
  
    // 3. Salva o evento  
    return eventoRepository.save(evento);  
}
```

Evidência 2: Teste com Mock de Integração

Objetivo: Demonstrar o teste `EventoServiceTest.java` mostrando a anotação `@Mock` `private UsuarioService` e o uso do método `verify(usuarioService...)`.

```

@ExtendWith(MockitoExtension.class)
class EventoServiceTest {

    @Mock
    private EventoRepository eventoRepository;

    @Mock
    private UsuarioService usuarioService;

    @InjectMocks
    private EventoService eventoService;

    @Test
    void deveLancarExcecao_QuandoCriarEventoComNomeDuplicado() {
        // 1. Cenário (Given)
        Evento eventoNovo = new Evento("Evento Repetido", "Descricao");
        Long usuarioId = 1L;

        // Simula a validação de nome duplicado
        when(eventoRepository.findByNome("Evento Repetido")).thenReturn(Optional.of(new Evento("Evento Repetido", "Descricao")));

        // 2. Ação e Verificação (When & Then)
        // Verificamos se a exceção de validação é lançada
        assertThrows(ValidacaoException.class, () -> {
            eventoService.criarEvento(eventoNovo, usuarioId); // Assinatura atualizada
        });

        // 3. Verificação Extra
        // Garantir que NUNCA tentamos buscar o usuário ou salvar
        verify(usuarioService, never()).buscarPorId(anyLong());
        verify(eventoRepository, never()).save(any());
    }
}

```

2. Refatoração e Clean Code (Requisito 1)

Descrição: Demonstração da aplicação de princípios SOLID e modularização.

Evidência 3: Injeção de Dependência via Construtor

Objetivo: Demonstrar o construtor da classe `EventoService.java`, evidenciando que **NÃO** utilizamos `@Autowired` nos atributos, mas sim a injeção via construtor.

```

@Service
public class EventoService {
    private final EventoRepository eventoRepository;
    private final UsuarioService usuarioService;

    public EventoService(EventoRepository eventoRepository, UsuarioService usuarioService) {
        this.eventoRepository = eventoRepository;
        this.usuarioService = usuarioService;
    }

    public List<Evento> listarTodos() {
        return eventoRepository.findAll();
    }

    public Evento buscarPorId(Long id) {
        return eventoRepository.findById(id)
            .orElseThrow(() -> new RecursoNaoEncontradoException("Evento não encontrado"));
    }
}

```

Evidência 4: Fragmentos Thymeleaf (Reuso)

Objetivo: Demonstrar a pasta `resources/templates/fragments` ou código HTML utilizando `th:replace`.

```

</div>
</div>
</div>

<footer th:replace="~{fragments/_footer :: footer-scripts}"></footer>

body>
html>

```

3. Testes Automatizados e Cobertura (Requisito 4)

Descrição: Execução da suíte de testes e validação da métrica de cobertura.

Evidência 5: Log de Execução dos Testes (Sucesso)

Objetivo: Terminal após rodar `mvn test` (ou aba "Run" da IDE) com a mensagem de sucesso.

```
willump@lucas-popos:/mnt/Linux_Outros/Desenvolvimento/INFNET/PB/TP4/TP4/com-cliente-projeto$ mvn test -Dtest=
\!CadastroEventoE2ETest
mpleted initialization in 2 ms
2025-12-08T23:15:56.867-03:00 INFO 23693 --- [          main] .c.p.c.c.EventoControllerIntegrationTest : St
arted EventoControllerIntegrationTest in 2.402 seconds (process running for 54.778)
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 5.039 s -- in com.cliente.projeto.crud
pb.controller.EventoControllerIntegrationTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 10, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:05 min
[INFO] Finished at: 2025-12-08T23:16:01-03:00
[INFO] -----
```

Evidência 6: Tentativa de Teste E2E (Firefox)

Objetivo: Log do terminal mostrando a execução do `CadastroEventoE2ETest` e a inicialização do driver (`org.openqa.selenium.firefox.FirefoxDriver`).

```
willump@lucas-popos:/mnt/Linux_Outros/Desenvolvimento/INFNET/PB/TP4/TP4/com-cliente-projeto$ export JAVA_HOME=
=/mnt/Linux_Outros/Desenvolvimento/INFNET/PB/TP1/.jdk/jdk-17.0.9+9
export PATH=$JAVA_HOME/bin:$PATH
java -version
mvn test

Driver info: org.openqa.selenium.firefox.FirefoxDriver
Capabilities {acceptInsecureCerts: true, browserName: firefox, browserVersion: 145.0.1, moz:accessibilityCheck
s: false, moz:buildID: 20251118022223, moz:geckodriverVersion: 0.36.0, moz:headless: true, moz:platformVersion:
6.17.4-76061704-generic, moz:processID: 22065, moz:profile: /tmp/rust_mozprofilexZ7FQz, moz:shutdownTimeout: 60000,
moz:webdriverClick: true, moz:windowless: false, pageLoadStrategy: normal, platformName: linux, proxy: Proxy(),
setWindowRect: true, strictFileInteractability: false, timeouts: {implicit: 0, pageLoad: 300000, script: 30000},
unhandledPromptBehavior: dismiss and notify, userAgent: Mozilla/5.0 (X11; Linux x86...)
Session ID: 02fe6c8f-7990-4790-8562-121370899b5b
[INFO]
[ERROR] Tests run: 11, Failures: 0, Errors: 1, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD FAILURE
[INFO] -----
```

Evidência 7: Relatório de Cobertura (JaCoCo)

Objetivo: Tabela de cobertura do arquivo `target/site/jacoco/index.html`.

[Sessions](#)com-cliente-projeto

com-cliente-projeto

Element	Missed Instructions	Cov. Missed	Branches Cov. Missed	Cxty Missed	Lines Missed	Methods Missed	Classes Missed
com.cliente.projeto.crudpb.controller	29671	19% 62	25% 13	18 70	88 10	14	0 2
com.cliente.projeto.crudpb.service	81122	60% 39	75% 9	22 16	45 7	16	0 2
com.cliente.projeto.crudpb.exception	4558	56%	n/a 3	7 11	23 3	7	1 3
com.cliente.projeto.crudpb.model	2152	71%	n/a 6	18 9	31 6	18	0 2
com.cliente.projeto.crudpb.dto	1738	69%	n/a 2	5 2	5 2	5	1 3
com.cliente.projeto.crudpb	5	37%	n/a 1	2 2	3 1	2	0 1
Total	465 of 809	42% 9 of 20	55% 34	72 110	195 29	62	2 13
Created with JaCoCo 0.8.12.202403310830							

4. Automação com CI/CD (Requisitos 3 e 5)

Descrição: Definição da esteira de Integração Contínua.

Evidência 8: Arquivo de Workflow do GitHub Actions

Objetivo: Conteúdo do arquivo `.github/workflows/ci.yml`.

```
name: Java CI com Maven (Build, Teste, Cobertura e E2E)

on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]
  workflow_dispatch:

jobs:
  build-test-coverage:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout do repositório
        uses: actions/checkout@v4

      - name: Set up JDK 17
        uses: actions/setup-java@v4
        with:
          java-version: '17'
          distribution: 'temurin'
          cache: 'maven'

      # Etapa para instalar o Google Chrome
      - name: Instalar Google Chrome
        run: |
          sudo apt-get update
          sudo apt-get install -y google-chrome-stable

      - name: Build, Teste e Cobertura (com Display Virtual)
        run: |
          # Inicia o display virtual (XVFB)
          Xvfb :99 -screen 0 1920x1080x24 &

          # Executa os testes apontando para o display
          DISPLAY=:99 mvn clean verify

      - name: Upload do Relatório de Cobertura (JaCoCo)
        uses: actions/upload-artifact@v4
        with:
          name: jacoco-html-report
          path: target/site/jacoco/
```

The screenshot shows a GitHub Actions workflow summary for a file named `ci.yml`. The workflow is triggered by a push event. The summary includes the following information:

- Triggered via push 1 minute ago**
- Status:** Success
- Total duration:** 50s
- Artifacts:** 1

The workflow is triggered by a push event from user `Willummp` on the `main` branch, with commit `be6fd14`.

The workflow file `ci.yml` is located at `on: push`. The workflow consists of a single job named `build-test-coverage`, which has a duration of 47s and a status of Success.

Resumo dos Arquivos Utilizados

Os seguintes arquivos foram base para as evidências acima:

- `src/main/java/com/cliente/projeto/crudpb/service/EventoService.java`
- `src/test/java/com/cliente/projeto/crudpb/service/EventoServiceTest.java`
- `.github/workflows/ci.yml`