

Aula 5 – Técnicas de Desenvolvimento Ágil

INPRO – Desenvolvimento Web 3º Ano

Aula 5 · Técnicas de Desenvolvimento Ágil (50 min)

Data: 16/09/2025 · Professor: Sanddro Lib

1. POR QUE ESTA AULA IMPORTA?

Você já pensou em subir código em produção todo dia sem medo?

Dominar TDD, Pair Programming e CI/CD põe você no grupo de devs que entregam rápido e com qualidade. Isso faz diferença no currículo — e no contracheque.

2. OBJETIVOS DA AULA

- Experimentar o ciclo Red → Green → Refactor (TDD).
- Colaborar em Pair Programming, trocando papéis a cada 2 min.
- Visualizar um pipeline CI/CD completo rodando na nuvem.
- Identificar code smells e aplicar refatoração guiada por testes.

3. AGENDA RELÂMPAGO (50 MIN)

Minuto | Atividade | Resultado Esperado

00-05 | ■ Bug Ninja: encontre o erro | Entender valor de testes.

05-12 | ■ Talk F1 Pit-Stop | “Processo > Pressa” gravado na mente. 12-22 | ■ Kata FizzBuzz TDD (Live-Coding) | Cobertura > 90 %.

22-30 | ■ Pair Programming Relay | Todos os testes verdes.

30-38 | ■ Demo CI/CD no GitHub Actions | Pipeline rodando ao vivo.

38-44 | ■ Refatoração & Code Review | Checklist de PR criado pela turma. 44-48 | ■ Feature Flag Demo | Entender deploy seguro.

48-50 | ■ Retro “Continue/Pare/Comece” | 2 compromissos para a próxima aula. 4.

DESAFIOS EM AULA

4.1 Bug Ninja

```
if (user.isAdmin = true) { GrantAccess(); } // use "==" para corrigir
```

→ Primeiro(a) que corrigir ganha um adesivo especial.

4.2 Kata TDD

- Escreva o teste que falha (Red).
- Faça passar com solução mínima (Green).
- Limpe o código (Refactor).
- Linguagem-alvo: C# (ou Java/Kotlin se preferir).

4.3 Pair Programming Relay

- Repositório com bug no cálculo de juros.

- Driver/Navigator trocam a cada 2 min.
- Critério de sucesso: testes + legibilidade.

5. FERRAMENTAS ONLINE RECOMENDADAS

Necessidade | Ferramenta | Acesso

IDE no navegador | Replit ou Gitpod | replit.com · gitpod.io

Execução rápida de C# | dotnetfiddle | dotnetfiddle.net

Pair remoto | VS Code Live Share | extensão gratuita

CI/CD | GitHub Actions | integrado ao repositório

Análise de código | SonarCloud | sonarcloud.io

Feature Flags | Unleash (hosted) | getunleash.io

(todas têm planos gratuitos suficientes para o curso)

6. MISSÃO PÓS-AULA · “Refator-Plus PR”

Entrega (via GitHub) | Requisitos | Deadline

Pull Request | ≥ 1 teste falha \rightarrow Green \rightarrow Refactor · Cobertura $\geq 80\%$ | 20/09 – 23h59

Peer-Review | 1 comentário construtivo recebido | 20/09 Evidência Pipeline | print verde anexado | 20/09

Peso: 5 pts (2 participação + 2 qualidade PR + 1 review).

7. CRITÉRIOS DE AVALIAÇÃO

- Clareza do código (nomes, responsabilidades).
- Efetividade dos testes (happy + edge cases).
- Uso adequado do Git (commits pequenos, mensagens imperativas).
- Qualidade do feedback dado ao colega.

8. DICA DE OURO

“Testes não são rede de segurança — são trampolim. Quanto mais alto queremos saltar, mais confiamos

nele. Preparem seus laptops, plug-ins e café. Vamos codar como quem faz pit-stop em 2 segundos! ■■