### **Plano de Projeto: Implementação e Avaliação de Testes TDD/BDD em Projetos Reais**

Este plano está dividido em fases para organizar o trabalho do semestre. O sucesso do projeto dependerá de seguir a metodologia de forma rigorosa, especialmente na coleta e análise de dados.

#### **Fase 1: Planejamento e Definição**

O objetivo desta fase é solidificar o que será feito e como. É a base para todo o resto do trabalho.

1. **Definir o Escopo e as Questões de Pesquisa:**  
   O documento do projeto exige que a metodologia contenha um objetivo e questões de pesquisa. Para um estudo comparativo, precisamos definir o que exatamente queremos comparar.
   * **Tema escolhido:** Implementação e Avaliação de Testes TDD/BDD em Projetos Reais
   * **Objetivo Sugerido:** Avaliar e comparar os impactos da adoção das práticas de TDD e BDD no processo de desenvolvimento de software, focando em métricas de qualidade, produtividade e percepção do desenvolvedor.
   * **Questões de Pesquisa:**
     + **QP1:** Qual abordagem (TDD ou BDD) resulta em maior cobertura de testes estrutural no projeto selecionado?
     + **QP2:** Como a adoção de TDD e BDD impacta o tempo de desenvolvimento de novas funcionalidades?
     + **QP3:** Qual a percepção da equipe de desenvolvimento em relação à clareza dos requisitos e à facilidade de manutenção do código ao utilizar TDD vs. BDD?
2. **Selecionar o Projeto Real (Amostra):**  
   Vamos precisar de um "projeto real" para aplicar as abordagens.
   * **Opção A (Projeto Novo):** Desenvolver um projeto pequeno do zero. A equipe pode desenvolver a mesma funcionalidade duas vezes, uma com TDD e outra com BDD, para uma comparação direta.
   * **Opção B (Projeto Existente):** Escolher um projeto open-source de pequeno a médio porte e adicionar novas funcionalidades a ele usando ambas as abordagens em ramos (branches) diferentes.
   * **Critérios de Seleção:** O projeto deve ter requisitos claros e ser complexo o suficiente para que a diferença entre as abordagens seja perceptível, mas simples o suficiente para ser concluído no prazo da disciplina.
3. **Definir as Métricas de Avaliação:**  
   A metodologia precisa de métricas claras.
   * **Métricas Quantitativas:**
     + **Cobertura de Código:** Medir a cobertura de linha, ramo e função gerada pelos testes de cada abordagem.
     + **Densidade de Defeitos:** Número de bugs encontrados após a entrega da funcionalidade.
     + **Tempo de Desenvolvimento:** Horas gastas para completar uma funcionalidade (desde o entendimento até a aprovação dos testes).
   * **Métricas Qualitativas:**
     + **Percepção da Equipe:** Aplicar um questionário simples (ex: usando Escala Likert) para a equipe avaliar a clareza dos testes, a facilidade de refatoração e a confiança no código gerado por cada abordagem.

#### **Fase 2: Execução e Coleta de Dados**

Esta é a fase prática, onde iremos implementar e coletar os dados definidos.

1. **Implementação:**
   * Configurar o ambiente e o projeto escolhido.
   * Implementar as funcionalidades planejadas, aplicando TDD e BDD em ramos separados para garantir o isolamento das abordagens.
   * Documentar o tempo gasto em cada tarefa.
2. **Coleta de Dados:**
   * Após cada implementação, rodem as ferramentas de cobertura de código e registrem os resultados.
   * Realizem testes exploratórios para encontrar bugs e registrem-nos.
   * Apliquem os questionários de percepção ao final da implementação de cada grande funcionalidade.
3. **Reuniões Semanais de Progresso:**  
   Conforme exigido, teremos reuniões semanais com a professora. Para cada reunião iremos preparar:
   * Um resumo do que foi feito na semana.
   * Os dados preliminares coletados.
   * Quaisquer dificuldades ou bloqueios encontrados.
   * O planejamento para a próxima semana.

#### **Fase 3: Análise, Escrita e Finalização**

1. **Análise dos Resultados:**
   * Compilar todos os dados quantitativos em tabelas e gráficos.
   * Analisar as respostas qualitativas em busca de padrões.
   * Cruzar os dados para responder às questões de pesquisa definidas na Fase 1.
2. **Escrita do Artigo Científico/Relatório:**
   * A equipe optou pelo formato Relatório
   * Usaremos o modelo abaixo como guia.
3. **Preparação do Repositório:**
   * Organizarem todo o material (código, scripts, dados brutos, questionários) no repositório remoto.

### **Escolha do Repositório Remoto**

O documento exige a disponibilização de um repositório remoto. Sugestão de escolha:

**GitHub**.

* **Por que o GitHub?**
  + **Gratuito:** É o padrão da indústria para projetos open-source e oferece repositórios públicos gratuitos.
  + **Colaboração:** Facilita o trabalho em equipe com ferramentas de pull request, issues e projects.
  + **Integração:** Integra-se facilmente com ferramentas de CI/CD (como GitHub Actions) para automatizar testes, o que pode ser um diferencial no projeto.
  + **Visibilidade:** Dá visibilidade ao trabalho para a comunidade.
* **Estrutura Sugerida para o Repositório:**  
  /  
  ├── README.md # Descrição do projeto, como executar, etc.  
  ├── project/ # Código fonte do projeto que vocês testaram/desenvolveram  
  │ ├── branch-TDD/  
  │ └── branch-BDD/  
  ├── scripts/ # Scripts usados para coleta ou análise de dados  
  ├── data/ # Dados brutos coletados (em .csv, .json, etc.)  
  ├── report/ # Versão final do artigo ou relatório em PDF  
  └── LICENCE # Uma licença de código aberto (ex: MIT)

#### **Modelo** 1**: Relatório Técnico**

*(****Tom:*** *Mais detalhado e direto. Foco em registrar o processo completo para uma audiência interna, como a professora e a própria equipe.)*

**Relatório Técnico do Projeto da Disciplina de Testes de Software 2025.1**

Título do Projeto: Implementação e Avaliação de Testes TDD/BDD em Projetos Reais

Equipe: Guilherme Caio, Júlio Bem, Maria Luísa Mendes, Mateus Nepomuceno e Vitor Malta

Professora: Paola Accioly

Data: 12/08/2025

**1. Introdução**

Descreva o objetivo do projeto conforme proposto na disciplina. Apresente o tema escolhido (TDD vs. BDD) e explique por que a equipe optou por um estudo empírico comparativo.

**2. Motivação**

(Similar ao artigo, mas pode ser mais detalhado). Explique o problema prático que motivou o estudo. Inclua um exemplo detalhado do fluxo de desenvolvimento de uma pequena funcionalidade usando ambas as abordagens para ilustrar as diferenças. 11

**3. Metodologia Detalhada**

Seja extremamente detalhista aqui.

* **3.1. Planejamento:** Descreva as decisões tomadas na Fase 1, incluindo as QPs e as métricas.
* **3.2. Ambiente de Desenvolvimento:** Liste todas as ferramentas, linguagens e frameworks utilizados (ex: Python 3.9, Pytest, Behave, VS Code).
* **3.3. Execução do Estudo:** Narre o processo. "Na semana 1, implementamos a funcionalidade de login usando TDD. O processo levou 8 horas e os testes podem ser encontrados no arquivo tests/test\_login.py. Na semana 2, a mesma funcionalidade foi implementada com BDD..."
* **3.4. Coleta de Dados:** Detalhe como cada métrica foi coletada, incluindo os comandos usados no terminal se aplicável.

**4. Resultados e Análise**

Apresente os dados brutos em apêndices e resuma-os aqui em tabelas e gráficos. Para cada resultado, faça uma análise imediata. "A cobertura de código para a versão TDD foi de 95%, enquanto a BDD foi de 91%. Acreditamos que isso ocorreu porque TDD incentiva testes de unidade mais granulares..."

**5. Discussão**

Discuta as lições aprendidas pela equipe. Quais foram os maiores desafios? A equipe mudou sua percepção sobre TDD/BDD ao longo do projeto? Quais seriam as recomendações para outras equipes que fossem usar essas práticas?

**6. Trabalhos Relacionados**

(Similar ao artigo). Liste e resuma brevemente outros estudos relevantes que vocês leram para embasar o projeto.

**7. Conclusão e Trabalhos Futuros**

Resuma os achados do projeto de forma clara. Conclua se, para o contexto estudado, uma abordagem se mostrou superior ou se elas são adequadas para cenários diferentes. Liste as sugestões para trabalhos futuros.

**Apêndices:**

* **A.** Questionários aplicados.
* **B.** Tabelas com dados brutos coletados.
* **C.** Link para o repositório do GitHub com todos os artefatos. 12