UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE ENGENHARIA DE SOFTWARE

MANUTENÇÃO E INTEGRAÇÃO DE SOFTWARE

TP2 - MANUTENÇÃO ADAPTATIVA

Instruções

O trabalho deve ser realizado em **grupos** (os mesmos do trabalho anterior).

Toda a documentação e o código devem estar organizados em um **repositório no GitHub**.

Cada grupo deve apresentar:

- O problema adaptativo enfrentado.
- A adaptação implementada.
- As evidências de funcionamento antes e depois.

A entrega deve ser feita com o **link do repositório GitHub** enviado por um integrante na aba do trabalho do Google Classroom.

Caso o sistema não tenha suporte a alguma das estratégias propostas, a equipe deve **adaptar a ideia ao seu contexto** (ex.: se não consome API externa, simular consumo de um endpoint de teste).

Descrição do Trabalho

Aplicar na prática o conceito de Manutenção Adaptativa de Software, promovendo ajustes no sistema já desenvolvido pela equipe, de modo a responder a mudanças de ambiente, dependências, APIs externas e requisitos externos (legais/regulatórios). O trabalho deve evidenciar a adaptação realizada, com documentação completa no GitHub.

Simulação de Mudança de Dependência

- Escolher uma dependência (framework, biblioteca ou SDK) usada no projeto.
- Atualizar para uma versão mais recente e resolver problemas de compatibilidade.
- Caso não haja dependência, incluir uma biblioteca auxiliar (ex.: formatação de datas, autenticação JWT, etc.).
- **Evidência**: print do erro ou incompatibilidade inicial + código funcionando após adaptação.

Cenário de Mudança de Regulamentação

Adaptar uma funcionalidade do sistema em resposta a uma nova **lei ou política de uso**.

- Exemplos:
  - LGPD → adição de botão “Excluir Conta”.
  - Inclusão de termos de uso aceitos pelo usuário.
  - Regras de acessibilidade (ex.: contraste mínimo, labels de formulário).

Evidência: prints da interface antes e depois + trecho de código/documentação legal simulada.

Migração de API Externa

Para sistemas que consomem API: migrar de um endpoint antigo para outro (ex.: de uma versão v1 para v2).

Para sistemas que não consomem API: simular integração com uma API pública simples (ex.: https://jsonplaceholder.typicode.com ou ViaCEP).

- **Uso do Postman**: obrigatório para testar requisições antes de integrar no sistema.
- **Evidência**: requisições no Postman (prints/exportação JSON) + execução no sistema

Etapa 5 – Estrutura Recomendada

```
├ 📂 manutenção-adaptativa
| ├ 📜 README.md                  # Apresentação geral do trabalho
| ├ 📜 plano-estrategia.md        # Descrição do Plano de Estratégia Adaptativa
| ├ 📜 evidencia1.md              # Evidência da estratégia 1
| ├ 📜 evidencia2.md              # Evidência da estratégia 2
| ├ 📜 evidencia3.md
├ 📂 src                          # Código do sistema atualizado
├ 📂 evidencias
| ├ 🖼️ prints/                    # Screenshots de testes, antes/depois
| └ 🎥 videos/                    # (opcional) pequenos vídeos demonstrando adaptação
├ 📜 CHANGELOG.md                 # Histórico de mudanças feitas no sistema
└ 📜 RELATORIO.md                 # Relatório final (síntese das adaptações)
```

Critérios de Avaliação

O trabalho será avaliado com **nota total de 10 pontos**, distribuídos da seguinte forma:

Critério	Pontos	Descrição
Documentação no GitHub	2,0	Organização da árvore de pastas, clareza no README e relatórios em Markdown.
Evidência das execuções das manutenções	2,0	Prints, vídeos, commits e comparação antes/depois.
Execução das 3 estratégias adaptativas	3,0	Cada uma implementada com clareza e coerência no sistema do grupo.
Plano de Estratégia Adaptativa	2,0	Documento estruturado, cobrindo todas as seções propostas, com boas práticas e visão de longo prazo.
Clareza e apresentação final (RELATORIO.md + apresentação)	1,0	Síntese bem escrita, linguagem clara, reflexão crítica.