Trabalho Prático – Git, GitHub e GitFlow (Sem Pull Request)

# Objetivo

Praticar colaboração em repositório GitHub usando o fluxo GitFlow (branches main, develop, feature/\*, release/\*, hotfix/\*) de forma simplificada: sem abrir Pull Requests, apenas criando branches, fazendo merges e resolvendo conflitos manualmente.

# Papéis

Aluno A (Owner): cria o repositório e organiza o fluxo.

Aluno B e Aluno C (Colaboradores): recebem convite, implementam tarefas e coordenam merges.

# Fase 0 — Preparação do Repositório

1. Aluno A cria o repositório no GitHub e adiciona B e C como colaboradores.  
2. Cria a branch develop a partir de main.  
3. Faz o commit inicial com os arquivos da aplicação (HTML, CSS e JS).  
4. Equipe decide convenções de nomes de branch (feature/nome, release/x.y.z, hotfix/ajuste).

# Fase 1 — Definição de Tarefas

- Criar um arquivo docs/tarefas.md com a lista:  
 • incrementar de 2 em 2  
 • renomear função setCount  
 • tema escuro ajustável  
 • corrigir título no modo claro (hotfix posterior)  
  
Distribuição:  
- Aluno B: incremento + rename.  
- Aluno C: tema escuro ajustável.  
- Aluno A: coordenação, merges e hotfix.

# Fase 2 — Features Paralelas (Provocar Conflitos)

Aluno B (feature/incremento-rename):  
- Renomear setCount para updateCount.  
- Alterar incremento para 2 em 2.  
- Alterar cor primária no styles.css para verde.  
- Alterar título no index.html para incluir 'Equipe B'.  
  
Aluno C (feature/tema-ajustavel):  
- Alterar incremento também para 2 em 2 (mesma linha do B).  
- Alterar cor primária para vermelho.  
- Ajustar lógica do toggleTheme para modificar mais elementos.  
- Alterar título no index.html para incluir 'Modo Escuro'.

# Fase 3 — Integração e Conflitos

1. Aluno A mescla primeiro uma das features em develop.  
2. Depois mescla a segunda feature em develop.  
3. Resolver conflitos em styles.css, app.js e index.html.  
4. Decidir em conjunto qual versão manter.  
5. Registrar as decisões no arquivo docs/decisoes.md.

# Fase 4 — Release

1. Criar branch release/1.0.0 a partir de develop.  
2. Atualizar index.html com a versão 1.0.0.  
3. Criar docs/release-notes.md listando funcionalidades.  
4. Mesclar release/1.0.0 em main.  
5. Criar tag v1.0.0.

# Fase 5 — Hotfix

1. Simular bug no título ao voltar para o modo claro.  
2. Aluno B cria hotfix/titulo-claro a partir de main.  
3. Corrige a lógica em app.js.  
4. Faz merge direto em main.  
5. Sincroniza main com develop.

# Fase 6 — Finalização

- Atualizar docs/decisoes.md com registro dos conflitos.  
- Atualizar README.md com nomes dos integrantes.  
- Garantir que main tem versão estável v1.0.0.  
- Subir todos os commits e tags para o GitHub.

# Critérios de Avaliação

- Estrutura de branches (main, develop, feature, release, hotfix): 2,0  
- Conflitos criados e resolvidos: 3,0  
- Documentação (tarefas.md, decisoes.md, release-notes.md): 2,0  
- Release com versão e tag: 1,5  
- Hotfix integrado corretamente: 1,5