



## Desafio Avançado em Git: Fluxo de Trabalho Profissional

### Grupos de 4 Alunos

#### 1. Contexto e Objetivo

No desenvolvimento de software, dominar o Git é essencial para manter um fluxo de trabalho colaborativo, organizado e transparente. Este desafio tem como objetivo testar e aprimorar suas habilidades no uso do Git, simulando um ambiente real de integração e colaboração em projetos de software. Os grupos que cumprirem os critérios estabelecidos poderão ser dispensados da prova final.

---

#### 2. Descrição do Desafio

##### Projeto Base

- **Aplicação:** Desenvolvam uma aplicação simples (por exemplo, um gerador de senhas, uma calculadora). O foco deve ser na organização e no uso adequado das ferramentas do Git, e não na complexidade do código.
- **Estrutura:** O projeto deve conter, no mínimo:
  - Um arquivo principal (de sua preferência) com a implementação da funcionalidade escolhida.
  - Um arquivo README.md contendo:
    - Descrição do projeto.
    - Instruções para instalação e execução.
    - Requisitos e dependências.

##### Configuração e Versionamento

- **Repositório Git:**
  - Inicializem um repositório Git local para o projeto.



Professor: Ítalo Nunes Pereira

Turma: 5º Período

Disciplina: DEVOPS

- Criem um repositório remoto (GitHub,) e conectem-no ao repositório local.
- **Organização das Branches:**
  - A branch principal deverá ser main.
  - Criem uma branch de desenvolvimento chamada **develop** para centralizar as alterações e integrações.
  - Implementem novas funcionalidades em branches separadas (por exemplo, feature/gerador-senhas ou feature/calculadora).

## Desenvolvimento e Integração

- **Commits e Mensagens:**
  - Realizem commits frequentes com mensagens claras e que sigam as convenções semânticas (ex.: feat: adiciona função de validação, fix: corrige bug na exibição).
- **Pull Requests (PR):**
  - Ao concluir uma funcionalidade, abram um pull request para mesclar as alterações da branch de funcionalidade na branch **develop** e, posteriormente, integrem as mudanças à branch main.
  - Simulem um processo de revisão de código, utilizando comentários e resolvendo eventuais conflitos de merge.
- **Versionamento:**
  - Após a integração, criem uma tag (por exemplo, v1.0) para marcar a versão final do projeto.

## Documentação do Fluxo de Trabalho

- Criem um arquivo chamado **GITFLOW.md** que descreva detalhadamente:
    - A estratégia de criação e gerenciamento das branches.
    - Os critérios para os commits e a abordagem para as revisões.
    - O procedimento de merge e uso de tags.
-



### 3. Critérios de Avaliação

O desafio será avaliado com base nos seguintes critérios:

- **Organização do Projeto:**
    - Estrutura clara e funcional do código.
    - Documentação completa no README.md com instruções detalhadas.
  - **Uso do Git:**
    - Histórico de commits bem estruturado e descritivo.
    - Uso adequado das branches (main, develop e branches de funcionalidades).
    - Execução correta dos pull requests, com revisão e resolução de conflitos.
    - Marcação de versões com tags.
  - **Fluxo de Trabalho Documentado:**
    - O arquivo GITFLOW.md deve refletir uma estratégia coerente de versionamento e integração.
- 

### 4. Entrega e Apresentação

- **Repositório Remoto:**
  - O projeto completo deve ser enviado para um repositório remoto.
  - O link do repositório deverá ser compartilhado com o professor dentro do prazo estipulado.
- **Apresentação:**
  - Cada grupo deverá estar preparado para apresentar seu fluxo de trabalho, explicar as decisões tomadas e demonstrar como resolveram eventuais conflitos.
- **Prazo:**



**Professor: Ítalo Nunes Pereira**

**Turma: 5º Período**

**Disciplina: DEVOPS**

- O prazo para entrega será de 30 minutos.