



Disciplina: Gerência de Configuração e Evolução de Software
Prof Johnatan Oliveira
Curso: Engenharia de Software

Atividade prática: Trabalho colaborativo com GitHub

Objetivo: Praticar o uso do GitHub para o desenvolvimento colaborativo de software em equipe.

Requisitos:

- Cada aluno deve ter uma conta no GitHub.
- Os alunos devem estar divididos em equipes de pelo menos 3 pessoas.
- Cada equipe deve ter um projeto de software para trabalhar.

Passo 1: Configuração do projeto

1. Cada equipe deve criar um repositório para o projeto no GitHub.
2. Cada equipe deve convidar todos os membros da equipe para colaborar no repositório.
3. Cada membro da equipe deve clonar o repositório em sua máquina local.

Passo 2: Trabalhando com branches

1. Cada membro da equipe deve criar uma nova branch para trabalhar em um recurso específico do projeto.
2. Cada membro deve trabalhar em sua própria branch e enviar as alterações para o repositório no GitHub.
3. Quando a funcionalidade estiver pronta, o membro deve criar uma solicitação de pull (pull request) para incorporar as alterações na branch principal do projeto.
4. Os outros membros da equipe devem revisar o código, deixar comentários e sugerir alterações se necessário.

Passo 3: Gerenciando conflitos

1. Em alguns casos, dois membros da equipe podem trabalhar no mesmo arquivo ou linha de código e gerar conflitos.
2. Nesse caso, o GitHub permite que os membros da equipe comparem as diferenças entre as versões e resolvam os conflitos.
3. Os membros da equipe devem discutir a melhor maneira de resolver o conflito e enviar as alterações.

Passo 4: Melhorando a qualidade do código

1. Cada membro da equipe deve estar atento à qualidade do código e seguir as melhores práticas de codificação.
2. Os membros da equipe podem usar as revisões de código para ajudar a melhorar a qualidade do código.
3. A equipe pode configurar integrações com ferramentas de análise estática de código para identificar problemas de qualidade.

Passo 5: Gerenciando tarefas

1. A equipe pode usar os problemas (issues) do GitHub para gerenciar as tarefas do projeto.
2. Os membros da equipe podem criar problemas para novas funcionalidades, correções de bugs ou outras tarefas.
3. A equipe pode usar as etiquetas (labels) para organizar e priorizar os problemas.
4. Quando uma tarefa estiver concluída, a equipe pode fechar o problema.