

Thiago Vinicius Maciel Moura

Savio Pereira Soares

TURMA 113

Artigo: Git e GitHub

Thiago Vinicius Maciel Moura

Savio Pereira Soares

TURMA 113

Artigo: Git e GitHub

Trabalho apresentado no curso

Técnico em Desenvolvimento

de sistemas integrados ao ensino

médio do Instituto Federal de São

Paulo. Professor(a): Claudete de Oliveira Alves

Sumário

Introdução.....	4
Conceitos de Git	5
Conceitos de GitHub	7
Diferenças entre Git e GitHub.....	8
Exemplos e Casos de uso	9
Conclusão.....	10
Referências	11

Introdução

O *Git* e *GitHub* são duas ferramentas diferentes que colaboram de maneira integrada, e são utilizadas por *devs* (desenvolvedores) para tornar o desenvolvimento de software mais eficiente.

O GitHub é uma espécie de rede social para devs, aonde é possível armazenar e compartilhar projetos de softwares com outras pessoas. Já o Git é um sistema de controle de versão de arquivos, ou seja, ele guarda as alterações do projeto que ele está monitorando.

Essas duas ferramentas são muito úteis para equipes de desenvolvimento conseguirem controlar o versionamento do código, rastrear mudanças no mesmo, e também colaborar eficientemente com toda a equipe. Nesse artigo, vamos comentar e nos aprofundar um pouco mais sobre cada uma dessas funções, suas capacidades, e diferenças entre elas.

Conceitos de Git

Git é uma ferramenta open-source de controle de versão, o que significa que ele rastreia quaisquer alterações nos arquivos que está monitorando, e cria um histórico completo com cada modificação feita nos arquivos. Existem várias outras ferramentas de controle de versão no mercado, mas o Git se diferencia por conta de suas funcionalidades adicionais, como o *branch* e os *merges*, que vão ser explicados mais adiante.

É importante ter um sistema de controle de versões em um projeto para que possamos garantir que não existem conflitos entre os códigos de todos os programadores no projeto, além de que, se qualquer coisa dar errado no projeto como um todo, a única coisa que os desenvolvedores precisam fazer é reverter a uma versão antiga do projeto e continuar a partir dali, em vez de ter que reiniciar o projeto inteiro desde o começo.

O Git funciona a base de comandos no terminal, que são para controlar o repositório (pasta do projeto). Dentre os comandos, estão:

- **git init [nome do projeto]:** Utilizado para criar um repositório em um diretório qualquer.
- **git add: nome_do_arquivo:** Usado para preparar alterações em arquivos, os preparando para o próximo commit.
- **git commit -m “descrição do commit”:** Cria uma mensagem de commit, que vai para o histórico de mudanças do projeto.
- **git status:** Exibe informações sobre as modificações feitas, e o status de preparação dos seus arquivos.
- **git log:** Permite visualizar uma lista cronológica do histórico de commits
- **git diff arquivo1.txt:** Compara as alterações entre o diretório de trabalho e o commit mais recente. Também pode comparar dois commits, usando o comando *git diff commit1 commit2*.
- **git rm:** Remove arquivos do seu diretório de trabalho, e prepara a remoção para o próximo commit.
- **git mv:** Usado para renomear e mover arquivos no diretório de trabalho, para renomear, *git mv arquivo_a_ser_renameado.txt novo_nome.txt*, e para mover o arquivo, *git mv arquivo.txt pasta_destinataria/*.
- **git config:** Configura diversos aspectos do Git, incluindo informações de preferências do usuário.

1.1 Repositório

Repositório é basicamente a pasta do projeto, onde fica armazenado os seus arquivos e o histórico de versões.

1.2 Commits

As *commits*, são como pontos na história do projeto. São formadas por pequenas alterações feitas pelos desenvolvedores, e acompanham um descritivo que resume as alterações no *commit* feito. Por exemplo, se alguém alterou o arquivo CSS de uma página específica de um projeto, essa modificação precisa ser descrita pelo desenvolvedor que a fez, com uma descrição breve como “Centralizei a div imagens na tela”.

Diferente de outras ferramentas de controle de versão, em vez de apenas registrar as diferenças nos códigos de cada commit, o Git registra o conteúdo de cada arquivo em cada commit, o que significa que cada commit tem sua versão do arquivo original do diretório do trabalho, o que agiliza e muito as operações do Git.

1.3 Branchs

As *branchs* são ramificações que divergem da linha principal do projeto, funcionando sem alterar a mesma, e em muitas ferramentas de controle de versão, esse é um processo difícil de se fazer, já que exige que você crie uma nova cópia do diretório do código-fonte, o que acaba demorando em projetos maiores. Git se destaca dos outros nesse quesito, já que ele consegue criar branches de maneira leve e quase que instantanea.

Para isso, o Git ao criar um branch, ele na verdade cria uma nova referência (ponteiro) para um commit específico dentro do projeto, em vez de copiar toda a pasta do projeto para uma nova pasta separada. Isso facilita e muito todo o processo, além de deixar o repositório do projeto mais leve no PC do usuário.

1.4 Merge

Merge é o processo de “fundir” dois branches juntos. Isso faz com que qualquer mudança que você tenha feito no branch alvo, também se faça no branch atual, fazendo com que os dois se tornem um só. É útil para correção de problemas, atualizações, e implementações que você queira adicionar no projeto, sem ter que reescrevê-las no branch principal.

1.5 Clone

O *git clone* é um comando usado para criar uma cópia de um repositório remoto, para um diretório local, permitindo que você tenha uma cópia do projeto para conseguir trabalho em seu desenvolvimento.

Conceitos de GitHub

Fundado em 2008, o GitHub é uma plataforma de hospedagem de código-fonte online que fornece controle de versão e recursos colaborativos para desenvolvedores. Com o Git como base, o GitHub permite que os desenvolvedores armazenem, compartilhem e colaborem em projetos de software de forma eficiente. Ao longo dos anos, tornou-se uma ferramenta essencial no mundo da tecnologia, facilitando o gerenciamento de diferentes versões de código-fonte e promovendo a colaboração entre equipes.

O GitHub registra todas as alterações feitas em um projeto, permitindo que os desenvolvedores monitorem as mudanças ao longo do tempo. A tecnologia Git, que serve de base, é extremamente eficaz nessa tarefa, garantindo um controle preciso sobre as diferentes versões do projeto.

O GitHub tem diversas ferramentas e recursos para auxiliar os desenvolvedores, como os issues que servem para relatar bugs, solicitar recursos ou discutir problemas em um projeto, isso ajuda a equipe a priorizar e resolver problemas de forma eficiente, também tem as pull requests que permitem que os desenvolvedores enviem suas alterações para revisão e aprovação antes de serem incorporadas ao projeto principal, garantindo que as alterações sejam testadas e avaliadas antes de serem implementadas, outro recurso é o GitHub Pages que deixa que os desenvolvedores hospedem sites estáticos diretamente no GitHub. Isso é útil para criar documentação, sites de projeto ou até mesmo blogs.

Diferenças entre Git e GitHub

O Git é um software de controle de versão local que permite aos desenvolvedores salvar snapshots de seus projetos ao longo do tempo, sendo ideal para uso individual. Já o GitHub é uma plataforma baseada na web que incorpora os recursos de controle de versão do Git, permitindo que os desenvolvedores trabalhem colaborativamente em projetos.

O Git armazena cada versão salva como uma "snapshot" do projeto, em vez de uma lista de alterações. Isso permite que os desenvolvedores consultem versões anteriores e criem novas versões quando o projeto é modificado. Além disso, o Git permite que os desenvolvedores "empurrem" e "puxem" mudanças de e para outros computadores, tornando-o um Sistema de Controle de Versão Distribuído. Porém como é um software local instalado na máquina individual, ele não consegue ler as edições que outros desenvolvedores podem estar fazendo em tempo real. Isso significa que se você e um colega de equipe estiverem trabalhando em um projeto simultaneamente, não poderão ver o trabalho um do outro em tempo real.

O GitHub é uma plataforma de colaboração que utiliza o Git para permitir que vários desenvolvedores trabalhem em um único projeto em tempo real, com repositórios de código armazenados na nuvem. Além disso, os repositórios GitHub estão disponíveis publicamente, o que possibilita que desenvolvedores de todo o mundo interajam e contribuam com o código uns dos outros, prática conhecida como "codificação social", tornando o GitHub uma espécie de rede social para profissionais da web.

Exemplos e Casos de uso

O Git e o GitHub são ferramentas poderosas que podem ser usadas tanto por indivíduos quanto por equipes de desenvolvimento para gerenciar e colaborar em projetos de software. Indivíduos podem usar o Git para manter seu código organizado e atualizado, criando branches e fazendo commits, enquanto equipes de desenvolvimento podem usar o Git para colaborar em projetos complexos, criando branches e fazendo pull requests. Já o GitHub permite que indivíduos hospedem e compartilhem seu código, enquanto equipes de desenvolvimento podem usá-lo para trabalhar em projetos complexos, utilizando recursos como issues, pull requests e code review para garantir a qualidade e integridade do código.

Exemplos comuns, Git é usado para gerenciar mudanças em projetos e a colaboração em projetos de código aberto. Enquanto o GitHub é usado para hospedar e permitir contribuições de desenvolvedores de todo o mundo, além da hospedagem de portfólios de projetos, como o “README” onde o GitHub é usado por desenvolvedores para demonstrar suas habilidades e experiências, mostrando seus projetos e código de forma pública.

Conclusão

Na produção desse artigo, aprendemos que tanto o Git quando o GitHub são ferramentas muito importantes que facilitam e muito a vida de vários desenvolvedores. A maneira como eles tornam fácil organizar projetos, além de oferecerem a possibilidade de ter diferentes versões do mesmo projeto é muito interessante e muito prática. As duas ferramentas são essenciais para qualquer pessoa que trabalhe na área.

Concluimos que, se você trabalha na área de desenvolvimento de sistemas, o Git e o GitHub são muito importantes de se ter conhecimento sobre, especialmente nos tempos atuais, aonde várias empresas e startups utilizam das duas ferramentas. É importante saber, pelo menos, o básico de cada uma das duas ferramentas.

Referências

[Conceitos de Git](#)

[Conceitos de GitHub](#)

[Diferença entre Git e GitHub](#)

[Exemplos e Casos de uso](#)