sesc escola cuiabá

Amanda latorraca melo

git e github

cuiabá

2024



amanda latorraca melo

git e github

curiosidades

trabalho apresentado pelo curso de

jogos digitais

orientador: wanderson

**Cuiabá**

2024

**RESUMO**

Git é um sistema de controle de versão de arquivos. Através deles podemos desenvolver projetos na qual diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas.

O Git clone é um comando para baixar o código-fonte existente de um repositório remoto (como o Github, por exemplo).

Existem algumas maneiras de baixar o código-fonte, mas eu prefiro o clone com o modo https:

**Para que serve:** Git é um software de controle de versões desenhado por Linus Torvalds, o criador do Linux. O propósito do Git é levar um registro de mudanças e coordenar o trabalho de várias pessoas em um repositório compartilhado.

**Como usar:** O GIT é consideravelmente simples de usar. Para começar, você pode criar um repositório ou conferir um já existente. Após a instalação, um simples git-init irá deixar tudo pronto. Da mesma maneira, o comando git clone pode criar uma cópia de um repositório local para um usuário

**COMANDOS GIT**

### ***git add***

Move as alterações do diretório de trabalho para a área de staging. Assim, você tem a oportunidade de preparar um instantâneo antes de realizar o commit ao histórico oficial.

### ***Git Branch***

Esse comando é a ferramenta de administração de ramificações de uso geral. Permite criar ambientes de desenvolvimento isolados em um único repositório.

**O que faz o Git Checkout**

Além de verificar commits antigos e revisões de arquivos antigos, o git checkout também é a forma de navegar pelas ramificações existentes. Combinado com os comandos básicos do Git, é a maneira de trabalhar em determinada linha de desenvolvimento.

### ***Git Clean***

Remove os arquivos não monitorados do diretório de trabalho. Esse é o correspondente lógico do git reset, que (em geral) só opera em arquivos monitorados.

### ***git clone***

Cria cópia de um Repositório do Git existente. A clonagem é o modo mais comum dos desenvolvedores obterem uma cópia de trabalho de um repositório central.

***git init***

Inicializa novo repositório do Git. Se você quiser colocar um projeto sob controle de revisão, este é o primeiro comando que você precisa aprender.

### ***git log***

Permite explorar as revisões anteriores de determinado projeto. Ele oferece várias opções de formatação para exibir o commit de instantâneos.

### ***Git merge***

Uma maneira poderosa de integrar alterações de ramificações divergentes. Após bifurcar o histórico do projeto com o git branch, o git merge permite unificar o histórico de novo.

### ***git pull***

É a versão automatizada do git fetch. Faz o download de ramificação de um repositório remoto e faz a mesclagem imediata na ramificação atual. É o equivalente Git do svn update.

### ***Git push***

É o oposto de buscar (com algumas ressalvas). Permite que você mova uma ramificação local para outro repositório, o que é um modo conveniente de publicar as contribuições. É semelhante ao svn commit, mas envia uma série de commits em vez de um único conjunto de alterações

### ***Git commit***

Faz um commit do instantâneo preparado no histórico do projeto. Combinado com git add, esse processo define o fluxo de trabalho básico para todos os usuários do Git.

### ***git commit --amend***

Informação da marcação -- amend ao git commit permite corrigir o commit mais recente. Isso é muito útil quando você esquece de preparar um arquivo ou para omitir informações importantes da mensagem do commit.

### ***git config***

Uma maneira conveniente de definir opções de configuração para a instalação do Git. Em geral, sua utilização só é necessária de modo imediato após a instalação do Git em máquina de desenvolvimento nova.

### ***git fetch***

A busca faz o download a partir da ramificação de outro repositório, junto com todos os commits e arquivos associados. Mas, não tenta integrar nada em o repositório local. Assim, você tem a oportunidade de inspecionar as alterações antes de fazer o merge no projeto.

### ***git rebase***

Permite mover ramificações, o que ajuda a evitar commits de merge desnecessários. A história linear resultante é, em geral, muito mais fácil de entender e explorar.

Github Resumo

O GitHub é considerado é uma ferramenta essencial para engenheiros de software, com uma popularidade sem igual. Atualmente, ele acomoda mais de 25 milhões usuarios isso significa que há um número considerável de profissionais que estão procurando o GitHub para melhorar o fluxo de trabalho e a colaboração.

Em suma, o GitHub é um serviço baseado em nuvem que hospeda um sistema de controle de versão (VCS) chamado Git. Ele permite que os desenvolvedores colaborem e façam mudanças em projetos compartilhados enquanto mantêm um registro detalhado do seu progresso.

Para melhor entender o que é o GitHub e como ele funciona, precisamos dar uma olhada mais a fundo.

**Para que serve:** GitHub hospeda repositórios do Git e fornece aos desenvolvedores ferramentas para enviar um código melhor por meio das funcionalidades de linha de comando.

**Como usar:** O GitHub Pages publica qualquer arquivo estático do qual você faz push no repositório. É possível criar seus próprios arquivos estáticos ou usar um gerador de site estático para que ele crie o site para você. Também pode personalizar seu próprio processo de criação localmente ou em outro servidor.