



Git

Material das Aulas

Apresentação e introdução ao tema

O que vamos aprender?

O que é o git?

Branch e Commit

Versionamento de códigos

Criando outras linhas cronológicas

Repositório

Instalação e comandos básicos

Material

Download do GIT

Git Bash

Iniciando projetos

git init

git status

git add . ou nomeDoArquivo

git commit -nomeDaBranch "comentário"

git config

informações do commit

Github e Repositório remoto

O que é

Github

Pesquisando Testes Técnicas

Criando seu repositório

Linkando repositório local com o remoto

Branch e Merge

Git checkout -b nome da branch

git merge nome da Branch

git fetch

Fork, Pull Request e Github Pages

Fork

Git Clone

Pull Request

Github Pages

Material das Aulas

- Manual de Sobrevivência do Git

Apresentação e introdução ao tema

O que vamos aprender?

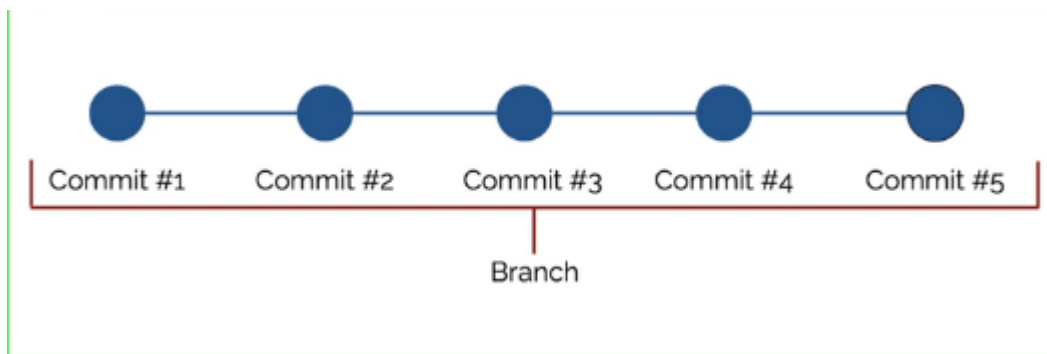
- Commit, branch e repositório

O que é o git?

- Software de versionamento usado para controlar e registrar as alterações que acontecem durante um projeto

Branch e Commit

- Temos uma linha cronológica e cada ponto representa uma nova alteração, ou seja, uma nova versão.
- Essa linha lógica chamamos de **branch** e cada ponto na linha cronológica chamamos de **commit**
- Usamos o commit para representar cada ponto de alteração no projeto



Versionamento de códigos

- Se precisamos voltar em algum commit, ou desfazer alguma alteração, isso é possível através do Git

Criando outras linhas cronológicas

Main

- Linha cronológica principal que vai normalmente para produção

- Versão estável do projeto.

Branches

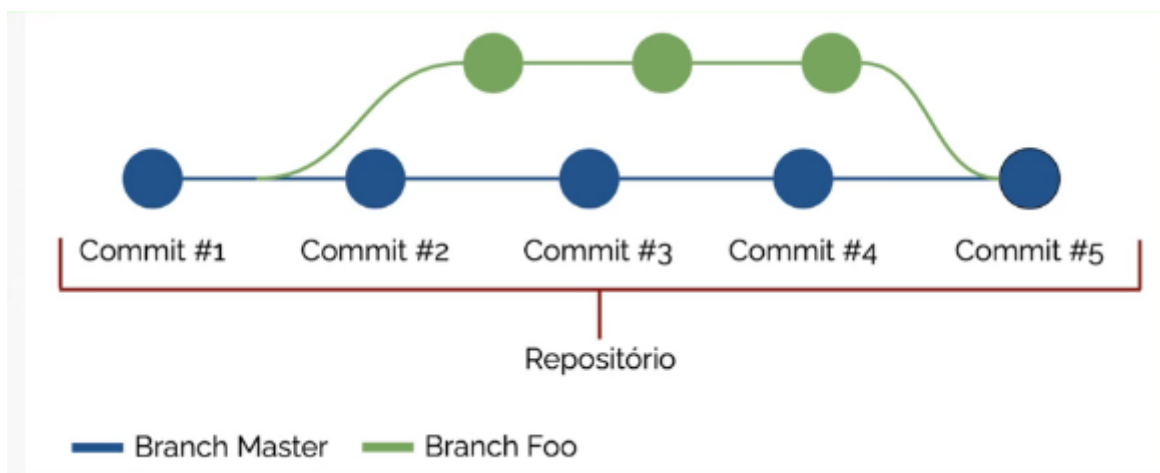
- Novas linhas cronológicas que podemos criar
- Normalmente cada integrante que trabalha na equipe tem uma branch.

Merge

- É a acoplação ou união de uma linha cronológica secundária, na linha cronológica principal
- Incorporação de **branch** na **main**

Repositório

- Conjunto de linhas cronológicas com Branches, Main etc. que formam nosso projeto.



Instalação e comandos básicos

Material

- Git e Terminal

Download do GIT

- Acesse o site e a documentação
- Baixe de acordo com seu sistema operacional

Git Bash

- Terminal que vamos usar para utilizar o Git
- Usamos comando de terminais e comandos gits.

Iniciando projetos

```
pwd // Vemos em que pasta estamos

cd // Mudamos o diretório

mkdir projetos-coronos // Criamos essa pasta

ls // Vemos o arquivo dentro de onde estamos

cd projeto-cronos

touch index.txt // Para criar um arquivo
```

git init

- Permite iniciarmos um repositório GIT **local na nossa máquina**
- A partir do momento que iniciamos o repositório local, criamos uma Branch chamada **master**

git status

- Mostra os commits que fizemos, as alterações e se há algum arquivo que não foi adicionado na nossa branch, além de qual branch estamos

git add . ou nomeDoArquivo

- Adiciona um arquivo ao Commit.
- O atalho ponto, adicionamos todas as alterações do nosso projeto no commit.

git commit -nomeDaBranch “comentário”

- Forma que temos para descrever a alteração que fizemos.
- Adicionamos um comentário dentro do nosso commit e da branch

git config

- Precisamos configurar nosso perfil do GIT para fazermos nosso primeiro commit
- Diz quem tá gerenciando os commits.

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

informações do commit

- Todas as informações do nosso commit são registradas na linha do tempo, para podermos desfazer, altera-la ou remove-la.

```
master (root-commit) 1c4a270] adicionando arquivo index.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.txt
```

Github e Repositório remoto

O que é

- Como ter um repositório remoto para compartilhar nosso código e trabalhar com outros profissionais através disso.
- Temos opções como: Github, Bitbucket (para empresas), GitLab (criação de novos ambientes personalizados e opções de segurança)

Github

- Github.com
- Crie sua conta, pensando em processos seletivos com um usuário condizente.
- Funciona como portfólio, dá pra ver seus commits durante o ano.
- Você consegue fazer o gerenciamento dos seus repositórios e projetos.
- Também consegue ver cada commit dentro de projetos.

Pesquisando Testes Técnicas

- Conseguimos ver os testes técnicos de empresas e repositórios de outros usuários

Criando seu repositório

- Define um nome, descrição, definir a visibilidade e adicionar um arquivo de descrição e porta inicial do projeto
- Git ignore dizendo quais arquivos não vamos subir para os repositórios

- License dizendo sobre a licença que vamos usar.

Linkando repositório local com o remoto

```
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin url do repositório github
git push -u origin main
```

Normalmente eu uso esses comandos

```
git init

git remote add origin url

git add .

git commit -m "Meu primeiro Commit"

git push origin master
```

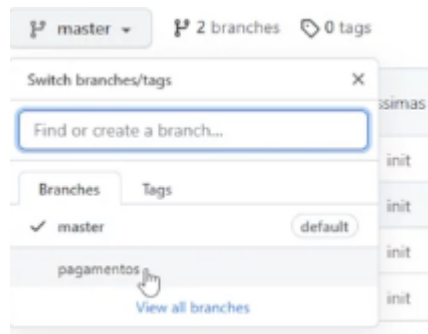
- Recebemos uma mensagem indicando todas as informações e arquivos do nosso repositório remoto! 🎉🎉

Branch e Merge

- Criar novas branches para fazer merge entre essas branches e atualizar nosso repositório local e remoto.

Git checkout -b nome da branch

- Mudamos para linha/branch que nomeamos.
- Fazemos todo o processo de `git add, commit e push` para o repositório
- Uma nova branch foi criada no nosso repositório
- Temos a possibilidade de ver no Github nossa branch



git merge nome da Branch

- Primeiro temos que estar na nossa branch para juntar, com `git checkout nome da branch`
- Fazemos a união, junção ou Merge da Branch com a master
- Depois disso, fazemos o `git push origin nome da branch`
- Na master deixamos tudo que já foi testado e está estável
- Na outras branches deixamos o que estamos ainda desenvolvendo.

git fetch

- Vemos se a junção da branch com a master (linha do tempo principal) foi feita com sucesso.

Fork, Pull Request e Github Pages

Fork

- Permite pegarmos todo o projeto de um repositório de outro usuário e criar uma copia para o nosso todo no Github
- Ele traz todos os commits feitos anteriormente

Git Clone



- Podemos fazer também o clone do repositório de outra pessoa com o comando `git clone url`

Pull Request

- Para subir as atualizações feitas na branch master, que é a versão estável, precisamos fazer uma requisição para subir para essa branch estável.

- Podemos fazer pull request para outros repositório também com o intuito de contribuir.
- Depois de termos enviado o nosso commit, vamos em **Contribute** e **Open Pull Request** dentro do Github, devemos colocar a mensagem que queremos enviar dentro do Pull-Request, além do nome dela.
- Todas as pessoas conseguem ver seu Pull-Request e todos os outros PRs.

alterando readme.md #1719

 **Open** fronthendy wants to merge 1 commit into jcbombardelli:master from fronthendy:master 



-u significa upstream, a primeira alteração que fazemos

Github Pages

- Serve para publicar o nosso projeto do Github
- Conseguimos selecionar qual Branch queremos publicar.
- O github já nos da um link com o deploy da nossa aplicação
- Podemos acessar qualquer página da nossa aplicação, colocando no endereço o nome do arquivo `.html`
- Por padrão, a primeira página lida é o `index.html`