



Apresentação

Prof. Thiago Xavier

Disciplina: Arquitetura de software

Objetivo



Apresentação do Git

Já imaginou que você pode voltar no tempo?

Poder criar uma linha do tempo



Gerência de configuração de software

Durante o desenvolvimento do software queremos saber:

- O que mudou e quando mudou?
- Por que mudou?
- Quem fez a mudança?
- Podemos reproduzir esta mudança?



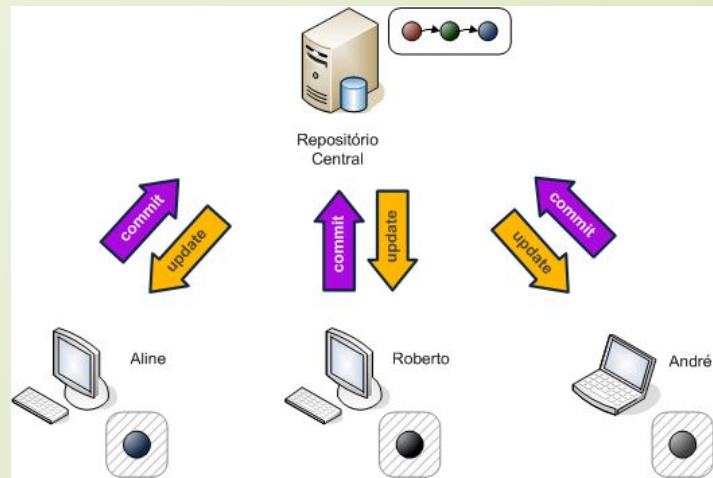
Tópicos – gerência de configuração de software

- Identificação;
- Documentação;
- Controle;
- Auditoria.



O que é o git-scm?

- Sistema de controle de versão distribuído;
- SCM – Source control management (Controle de gerenciamento de versão);
- Criado por Linus Torvalds (2005);
- Auxiliar no desenvolvimento do Linux.



O que é o controle de versão?

Você precisa editar um código que está no seu Google drive;

- Você faz o download do arquivo;
- Faz as alterações necessárias;
- Salva o arquivo no drive novamente.



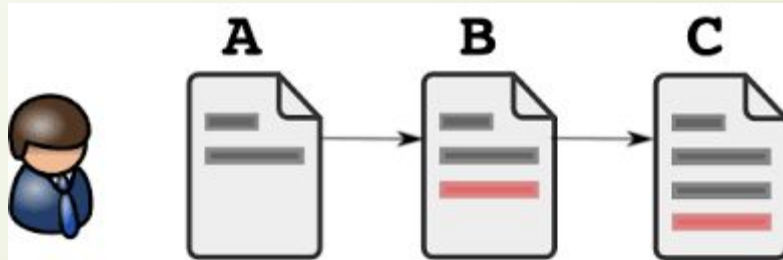
Mas qual o problema?

- Seu colega também quer editar o código;
- Vocês baixam o mesmo arquivo junto;
- Você edita e salva no Dropbox;
- Seu colega edita e salva no Dropbox e acaba sobrescrevendo seu código.



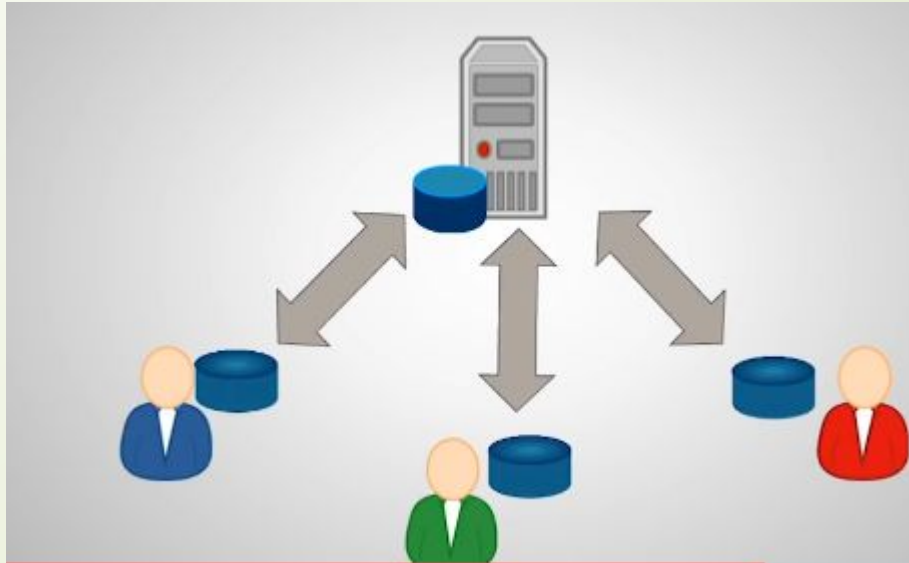
Controle de versão

Controle de versão faz o 'merge' (junção) entre as alterações;



Git é distribuído

- Todos possuem uma cópia;
- Controle de versão sem conexão com o servidor;



GitHub


Servidor de repositórios Git;
Surgiu em 2008;
+76 milhões de usuário (Atualizado em 2023).



Vamos praticar!

Para começar, criem uma conta no Github:

www.github.com



Sign in to GitHub

Username or email address

Password [Forgot password?](#)

Sign in

New to GitHub? [Create an account.](#)

Instalar o git

<https://git-scm.com/downloads>:



- Acessar o Gitbash



Criando o primeiro repositório local

1. Primeiro vamos acessar uma pasta local, para isso utilizado o comando cd + “caminho da pasta”

```
AB1239450@A541732 MINGW64 ~  
$ cd /C/projetos/PrimeiroProjetoGit
```

2. Agora estamos no caminho que será o repositório local.
3. Após acessar a pasta, executar o comando:
`git init`
 - Será criado o diretório configurado como um repositório git;
 - Acesse a pasta que será criado o projeto.

Validar repositório

1. Para validar, podemos utilizar o comando `ls -al`. O comando “al” é para mostrar os arquivos ocultos.

`ls -al`

```
AB1239450@A541732 MINGW64 /c/projetos/PrimeiroProjetoGit (master)
$ ls -al
total 8
drwxr-xr-x 1 AB1239450 1049089 0 ago 18 14:24 ./
drwxr-xr-x 1 AB1239450 1049089 0 ago 18 14:21 ../
drwxr-xr-x 1 AB1239450 1049089 0 ago 18 14:24 .git/
```

Configurar

- Configure seu nome e e-mail:

```
git config --global user.name "username"
```

```
git config --global user.email "email"
```


Criar arquivos

- Antes de começarmos os comandos, é importante criar um arquivo de exemplo;
- Podemos criar um arquivo chamado index.html;

Comando:

```
touch index.html
```

Comando git status

- Verificar estado dos arquivos/diretórios

Comando:

`git status`

```
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
  
index.html
```

Git Add

- Recebe uma lista de arquivos como parâmetros;

`git add .` (adicionar todos os arquivos)

ou

`git add "nome do arquivo"`

```
AB1239450@A541732 MINGW64 /c/projetos/primeiroprojeto (master)
$ git add .
```

Git commit

- Registrar **commit** com todos os arquivos adicionados "git add";
- Se o parâmetro de mensagem não for passado, abrirá um editor de texto para escrever a mensagem;
- Utilizamos o commit para descrever o versionamento adicionado.

```
AB1239450@A541732 MINGW64 /c/projetos/primeiroprojeto (master)  
$ git commit -m "Alteracao"
```

Git Log

- Avaliar os pontos da história;
- Possibilita revisar a linha do tempo.

comando: `git log`

Git show

- Observar o que foi feito nos pontos;
- git show e o id apresentado pelo git log;

comando: git show 95f91abaa1b32d248fe4f6bba0f167cb186af188

```
commit 95f91abaa1b32d248fe4f6bba0f167cb186af188
Author: thiagoxavier <xavier7132@gmail.com>
Date:   Wed Feb 19 14:13:42 2020 -0300

    projeto inicial

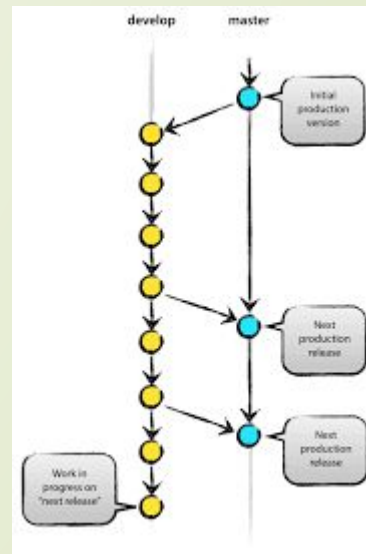
thiago.xavier@MTZDTI-N17998 MINGW64 /c/faculdade/automacao/aulaUm (master)
$ git show 95f91abaa1b32d248fe4f6bba0f167cb186af188
commit 95f91abaa1b32d248fe4f6bba0f167cb186af188
Author: thiagoxavier <xavier7132@gmail.com>
Date:   Wed Feb 19 14:13:42 2020 -0300

    projeto inicial
```

Criar universos paralelos - branch

- Criar linhas do tempo alternativa;
- Permite fazer alterações sem afetar a linha principal (master);
- São ramificações

Comando: `git branch "nome da branch"`



Acessar uma branch

- Para acessar a branch que se deseja trabalhar;

comando: `git checkout "nome da branch"`

para confirmar, podemos utilizar o `git status` para confirmar a branch;

Visualizar todas as branches

- Para visualizar todos, devemos estar na master;

git checkout master;

git branch (para listar todas as branches)

```
chiago.xavier@MTZDTI-N17998 MINGW64 /c/faculdade/automacao/aulaUm (master)
$ git branch
feature/incluiBody
* master
```

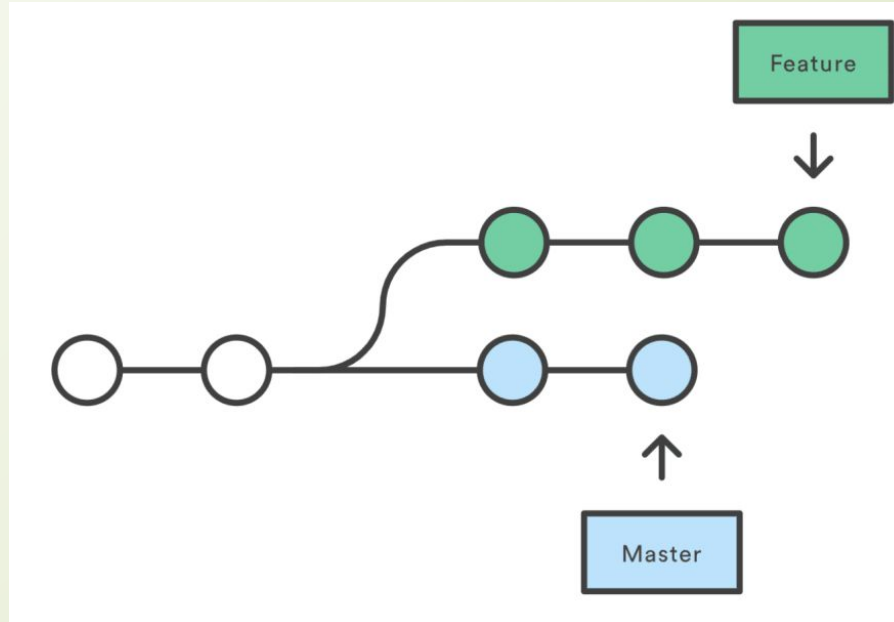
Unindo linhas do tempo

- Dê um checkout na master;
git checkout master
- Faça uma junção das linhas do tempo
(merge);
git merge e o “nome da linha do tempo”



Excluire a branch

- `git branch -D "nome da branch";`



Criar um repositório remoto no github

- Crie um repositório no Github;
- Configure o repositório remoto;
- Crie um arquivo;
- Faça o upload das alterações.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner



professorthiagoxavier ▾

Repository name *

Great repository names are short and memorable. Need inspiration? How about [legendary-robot?](#)

Description (optional)

Sincronizar o repositório local com o repositório criado

1. Ao criar um projeto, é gerado um endereço. Exemplo:
[git@github.com:professorthiagoxavier/primeiroProjeto.git](https://github.com/professorthiagoxavier/primeiroProjeto.git)
2. Para sincronizarmos com a pasta local, basta utilizar o comando
git remote

git remote add origin

[git@github.com:professorthiagoxavier/primeiroProjeto.git](https://github.com/professorthiagoxavier/primeiroProjeto.git)

```
AB1239450@A541732 MINGW64 /c/projetos/primeiroprojeto (master)
$ git remote add origin git@github.com:professorthiagoxavier/primeiroProjeto.git
```

Comando push

- push: sincronizar com o repositório remoto;
- Agora basta sincronizar utilizando o comando push

git push -u origin master

```
AB1239450@A541732 MINGW64 /c/projetos/primeiroprojeto git (master)  
$ git push -u origin master
```

Revisão de comandos

Comando	Descrição
git status	Mostrar o status dos commits
git log	retornar os últimos commits feitos
git add	adicionar arquivo(s)
git push	enviará todos os commits que ainda não estiverem sido enviados para o repositório .

Estado do arquivos

- Não monitorado (untracked). Quando criamos o arquivo ele passa a ser não monitorado;
- Modificado (modified);
- Preparado (staged). Quando realiza git add, a gente coloca o arquivo para este estado;
- Consolidado (committed) .

Exercícios

1. Remova uma linha de um arquivo.
2. Adicione uma nova linha no mesmo.
3. Adicione um novo arquivo com pelo menos duas linhas.
4. Veja as diferenças do commit pelo GitHub