Desenvolvimento Web

Aula 02 - Git (Parte 1)

Turma: ADS2AN-PLA

Prof. Igor Moreira Félix

Universidade São Judas Tadeu
São Paulo - SP

Introdução

- Organizar o trabalho desenvolvido

Introdução

- Organizar o trabalho desenvolvido
- Software da calculadora, com dois desenvolvedores

Introdução

- Organizar o trabalho desenvolvido
- Software da calculadora, com dois desenvolvedores
- Como trabalhar ao mesmo tempo no mesmo projeto?

Introdução

- Organizar o trabalho desenvolvido
- Software da calculadora, com dois desenvolvedores
- Como trabalhar ao mesmo tempo no mesmo projeto?
- Onde guardar e compartilhar o código?

Introdução

- Organizar o trabalho desenvolvido
- Software da calculadora, com dois desenvolvedores
- Como trabalhar ao mesmo tempo no mesmo projeto?
- Onde guardar e compartilhar o código?
  - Por e-mail?

Introdução

- Organizar o trabalho desenvolvido
- Software da calculadora, com dois desenvolvedores
- Como trabalhar ao mesmo tempo no mesmo projeto?
- Onde guardar e compartilhar o código?
  - Por e-mail?
  - Pendrive?

Introdução

- Organizar o trabalho desenvolvido
- Software da calculadora, com dois desenvolvedores
- Como trabalhar ao mesmo tempo no mesmo projeto?
- Onde guardar e compartilhar o código?
  - Por e-mail?
  - Pendrive?
- Como juntar as partes?

Introdução

- Organizar o trabalho desenvolvido
- Software da calculadora, com dois desenvolvedores
- Como trabalhar ao mesmo tempo no mesmo projeto?
- Onde guardar e compartilhar o código?
  - Por e-mail?
  - Pendrive?
- Como juntar as partes?
- Como fazer backups?

sãojudas universidade

Introdução

- Organizar o trabalho desenvolvido
- Software da calculadora, com dois desenvolvedores
- Como trabalhar ao mesmo tempo no mesmo projeto?
- Onde guardar e compartilhar o código?
  - Por e-mail?
  - Pendrive?
- Como juntar as partes?
- Como fazer backups?
  - Pastas? versao1, versao2, versao3...

Introdução

sãojudas universidade

git

- Ferramenta de controle de versões distribuído

sãojudas universidade

git

- Ferramenta de controle de versões distribuído
- Mantém o histórico de todas as alterações

sãojudas universidade

git

- Ferramenta de controle de versões distribuído
- Mantém o histórico de todas as alterações
- Equipe trabalha ao mesmo tempo

sãojudas universidade

git

- Ferramenta de controle de versões distribuído
- Mantém o histórico de todas as alterações
- Equipe trabalha ao mesmo tempo
- Em caso de erro, basta voltar à versão anterior

sãojudas universidade

git

- Ferramenta de controle de versões distribuído
- Mantém o histórico de todas as alterações
- Equipe trabalha ao mesmo tempo
- Em caso de erro, basta voltar à versão anterior
- Cada diretório de trabalho é um **repositório**

sãojudas universidade

git

- Ferramenta de controle de versões distribuído
- Mantém o histórico de todas as alterações
- Equipe trabalha ao mesmo tempo
- Em caso de erro, basta voltar à versão anterior
- Cada diretório de trabalho é um **repositório**
- Software livre e de código aberto!

sãojudas universidade

git

- Ferramenta de controle de versões distribuído
- Mantém o histórico de todas as alterações
- Equipe trabalha ao mesmo tempo
- Em caso de erro, basta voltar à versão anterior
- Cada diretório de trabalho é um **repositório**
- Software livre e de código aberto!
- Precisa ser instalado em sua máquina de trabalho!

sãojudas universidade

git - Principais comandos

- Inicializando um repositório Git em uma pasta existente

```
# git init
```

sãojudas universidade

git - Principais comandos

- Inicializando um repositório Git em uma pasta existente

```
# git init
```

A partir deste instante, os arquivos criados no diretório poderão ter sua versão controlada pelo Git.

git - Estados de um arquivo do ponto de vista do git

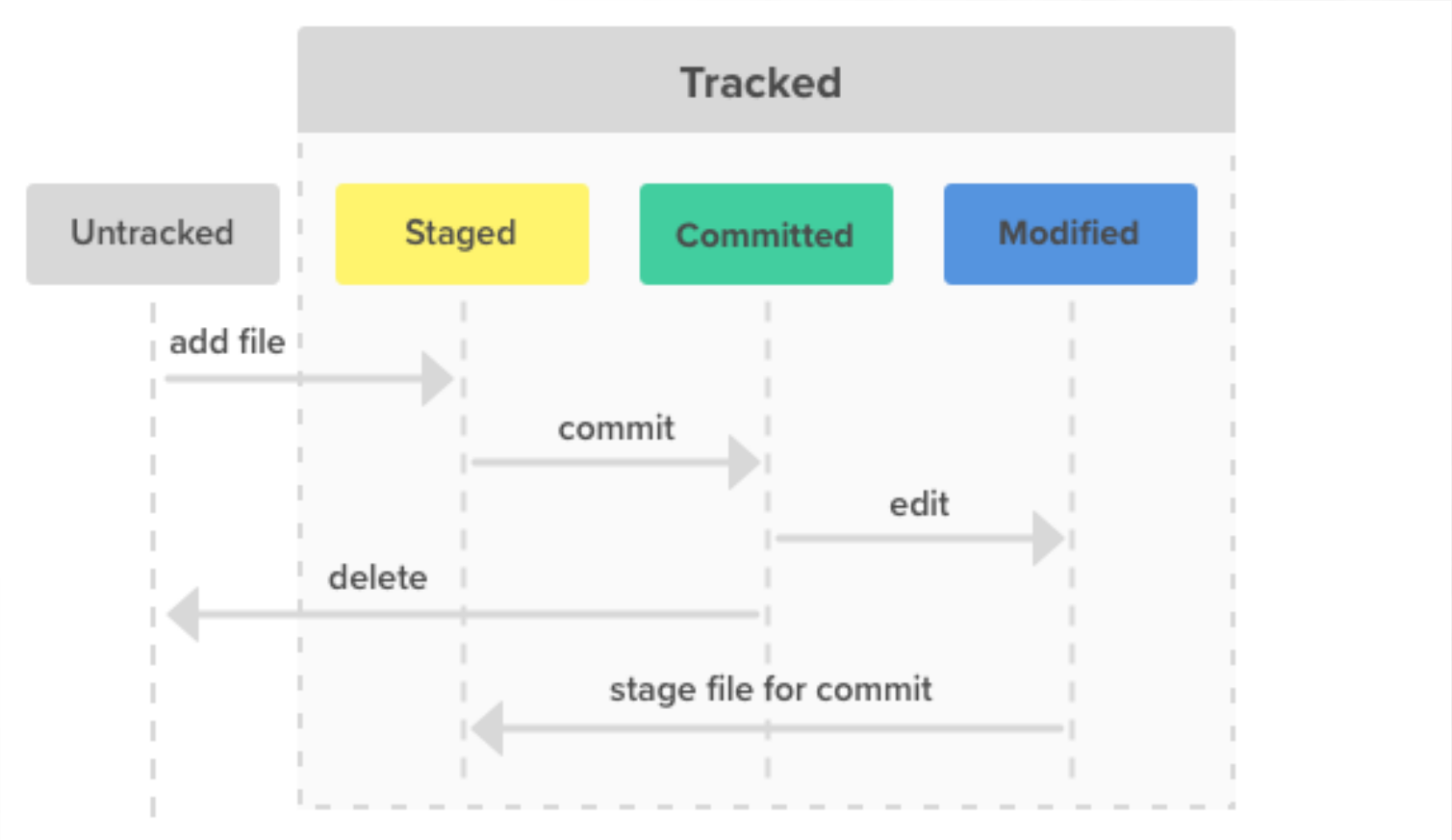

Figura: Estados e transições possíveis

git - Principais comandos

- Informando ao Git que desejamos controlar a versão de um arquivo

```
# git add arquivo.java
```

sãojudas universidade

git - Principais comandos

- Informando ao Git que desejamos controlar a versão de um arquivo

```
# git add arquivo.java
```

Agora, o arquivo *arquivo.java* está no estado Staged. Ou seja, será incluído no próximo commit.

sãojudas universidade

git - Principais comandos

- Para fazer um commit (ou seja, tornar as alterações permanentes), usamos o comando:

```
# git commit -m “Meu primeiro commit”
```

sãojudas universidade

git - Principais comandos

- Para fazer um commit (ou seja, tornar as alterações permanentes), usamos o comando:

```
# git commit -m “Meu primeiro commit”
```

O parâmetro -m do comando significa “message” e o valor a seguir fica associado ao commit.

git - Principais comandos

- Para informar o usuário que está realizando os commits:

```
# git config --global user.email
'seuemail@email.com'

# git config --global user.name
'Seu nome'
```

git - Principais comandos

- Para verificar o estado atual de seu repositório, usamos o comando:

```
# git status
```

```
# git status -s
```

sãojudas universidade

git - Principais comandos

- Para conferir o histórico de commits:

```
# git log
```

sãojudas universidade

GitHub

- Repositório remoto de repositórios git
- Sincronização entre o repositório local e remoto
- Há diversos servidores Git disponíveis como o GitHub e o GitLab
- Criar uma conta no GitHub
  - https://github.com/
- Criar um novo repositório

git - Principais comandos

- Adicionando um remote (adicionar ao repositório local um remote que representa o repositório que acabamos de criar no Github):

```
# git remote add origin
https://github.com/meuusuario/meu-repositorio.git
```

git - Principais comandos

- Adicionando um remote (adicionar ao repositório local um remote que representa o repositório que acabamos de criar no Github):

```
# git remote add origin
https://github.com/meuusuario/meu-repositorio.git
```

Nesse instante, temos um remote em nosso repositório cujo nome é origin.

git - Principais comandos

- Fazendo upload ao repositório remoto:

```
# git push origin master
```

sãojudas universidade

git - Principais comandos

- Fazendo upload ao repositório remoto:

```
# git push origin master
```

master é o nome da branch em que estamos trabalhando atualmente. Esse assunto será tratado posteriormente.

sãojudas universidade

git - Principais comandos

- Clonando um repositório:

```
# git clone
https://github.com/meuusuario/meu-repositorio.git
```

git - Principais comandos

- Clonando um repositório:

```
# git clone
https://github.com/meuusuario/meu-repositorio.git
```

Nesse caso, o diretório local já está sendo monitorado pelo git, portanto não precisamos usar o git init.