GIT

ProWay

O que é Git

Git é um sistema de controle de versão distribuído livre e de código aberto, projetado para lidar com tudo, de pequenos a grandes projetos, com velocidade e eficiência.

ProWay

Qual a vantagem

Imagine poder descobrir alterações feitas em um projeto, quando foram feitas e quem fez? Imagine você poder usar o CTRL+Z em todo seu projeto desde o início.

ProWay

Características do GIT

GIT permite aos desenvolvedores ter uma infinidade de ramos de código completamente independente. Criação, exclusão e fusão desses ramos é simples e não leva tempo

ProWay

Características do GIT

No GIT, todas as operações são atômicas. Isso significa que uma ação pode ter sucesso ou falhar (sem fazer nenhuma alteração). Isso é importante porque em alguns sistemas de controle de versão (como o CVS) onde as operações não são atômicas, se uma operação de repositório é suspensa, ela pode deixar o repositório em um estado instável.

ProWay

Características do GIT

No GIT, todas as operações são atômicas. Isso significa que uma ação pode ter sucesso ou falhar (sem fazer nenhuma alteração). Isso é importante porque em alguns sistemas de controle de versão (como o CVS) onde as operações não são atômicas, se uma operação de repositório é suspensa, ela pode deixar o repositório em um estado instável.

ProWay

Instalação

www.git-scm.com

ProWay

Configuração

```
$ git config — global color.status auto
$ git config — global color.branch auto
$ git config — global color.diff auto
$ git config — global color.ui always
$ git config — global core.editor vim
$ git config — global user.name "meunome"
$ git config — global user.email "eu@example.com"
```

ProWay

Hospedagem

LOCAL: Para usar o git localmente não é necessário nenhuma instalação extra, apenas usar git init e pronto servidor local já está funcionando.

GITHUB: O GitHub é um Serviço de Web Hosting Compartilhado para projetos que usam o controle de versionamento Git, ele possui funcionalidades de uma rede social como feeds, followers, wiki e um gráfico que mostra como os desenvolvedores trabalham as versões de seus repositórios. *https://github.com/*

Bitbucket Serviço semelhante ao Github. *https://bitbucket.org/*

Gitlab: Semelhante ao Github. *https://about.gitlab.com/*

ProWay

Como iniciar um repositório

Existem duas maneiras de iniciar um repositório: pelo local ou pela hospedagem.

ProWay

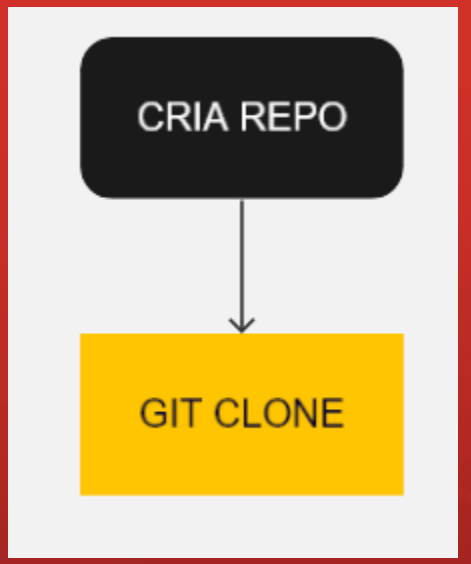
O que muda entre elas é a forma como se vai conectar os dois ambientes. Sim são dois ambientes.

1. Repositório local: no seu pc
2. *Repositório online: hospedagem*

Ainda pode exisitir um terceiro: que seria o servidor onde será dado o deploy das alterações para produção

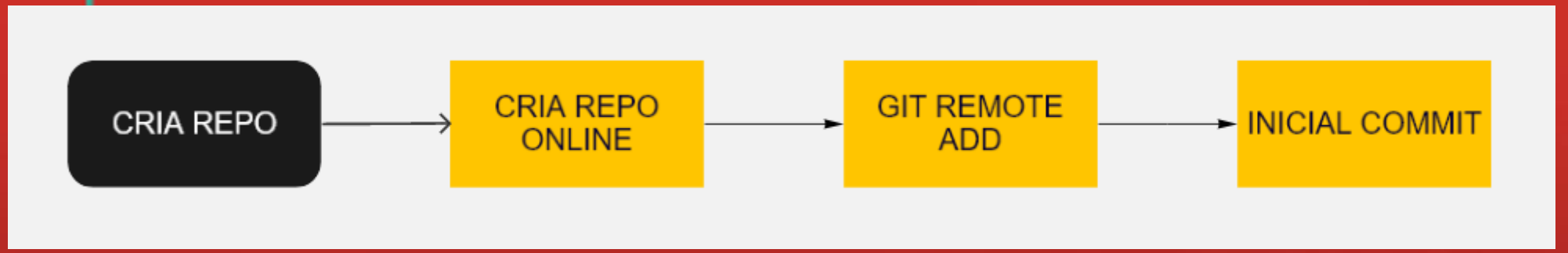
ProWay

Criando repositório a partir do online



ProWay

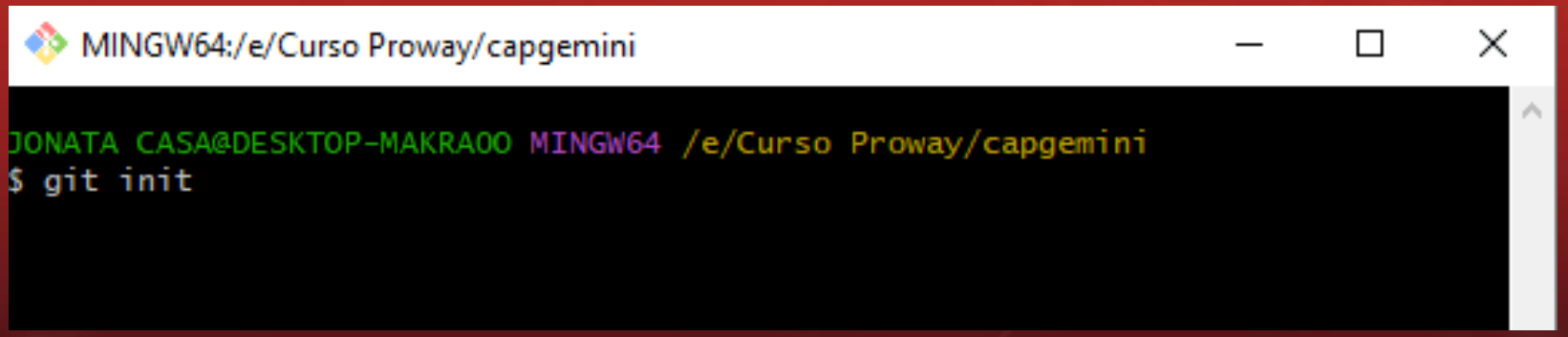
Criando repositório a partir do local



ProWay

Criando um repositório local

Para iniciar um repositório basta criar uma pasta onde você deseja e com o git bash aplicar o comando: git init



ProWay

Adicionando um repositório remoto ao local

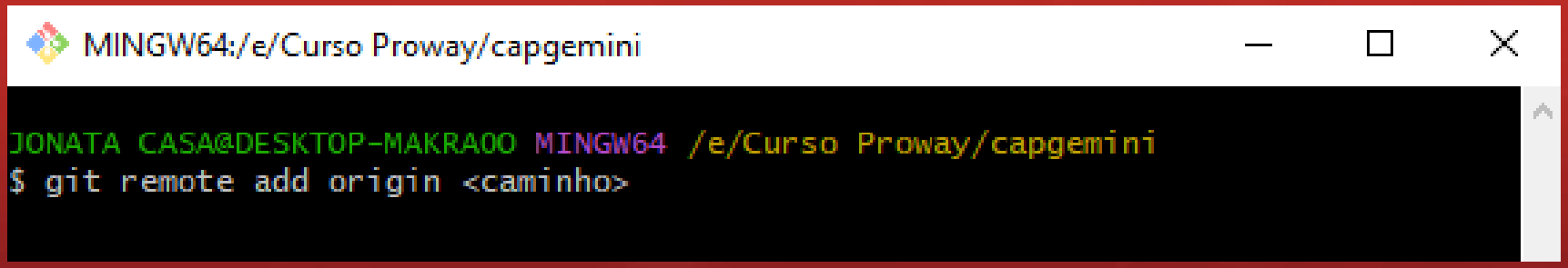
No caso de já termos um repositório local criado e efetuarmos a criação do repositório online posteriormente, precisamos fazer o processo de vinculação do repositório online com o nosso local

ProWay

Adicionando um repositório remoto ao local

Para fazermos isso vamos utilizar o

comando git remote add



ProWay

Adicionando um repositório remoto ao local

Agora temos que adicionar tudo, commitar e pushar.

Para isso vamos utilizar 3 comandos na ordem que vamos listar

ProWay

Adicionando um repositório remoto ao local

git add

```
MINGW64:/e/Curso Proway/capgemini

JONATA CASA@DESKTOP-MAKRAOO MINGW64 /e/Curso Proway/capgemini
$ git add .
```

git commit

```
MINGW64:/e/Curso Proway/capgemini

JONATA CASA@DESKTOP-MAKRAOO MINGW64 /e/Curso Proway/capgemini
$ git commit -m "descrição do commit"
```

git push

```
MINGW64:/e/Curso Proway/capgemini

JONATA CASA@DESKTOP-MAKRAOO MINGW64 /e/Curso Proway/capgemini
$ git push origin branch
```

ProWay

Criando um repositório online

Este é o procedimento mais simples e fácil, basta criar o repositório online e depois clonar.

Posso ainda clonar repositórios públicos de outros desenvolvedores.

ProWay

Clonando um repositório

Para clonar um repositório basta utilizar o comando git clone seguido pelo caminho do repositório.

O clone pode ser feito por HTTPS ou SSH

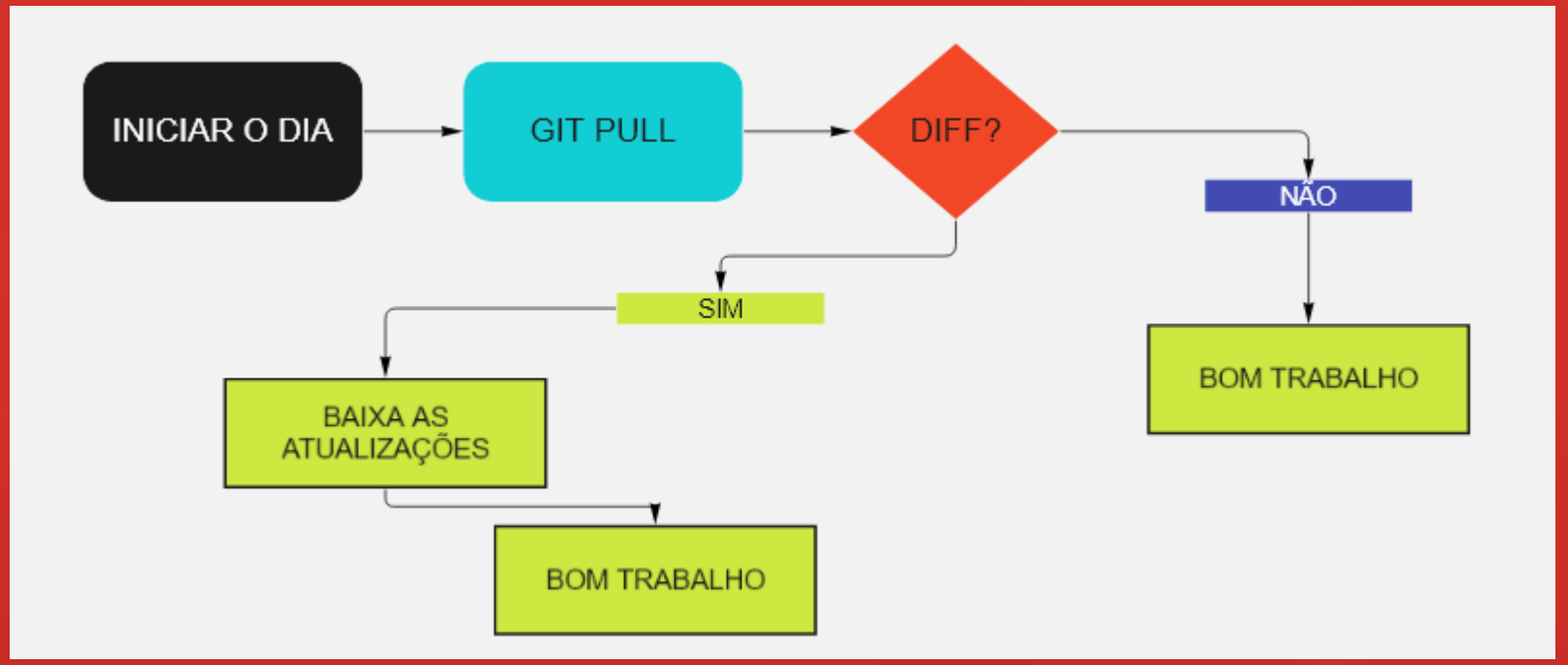
ProWay

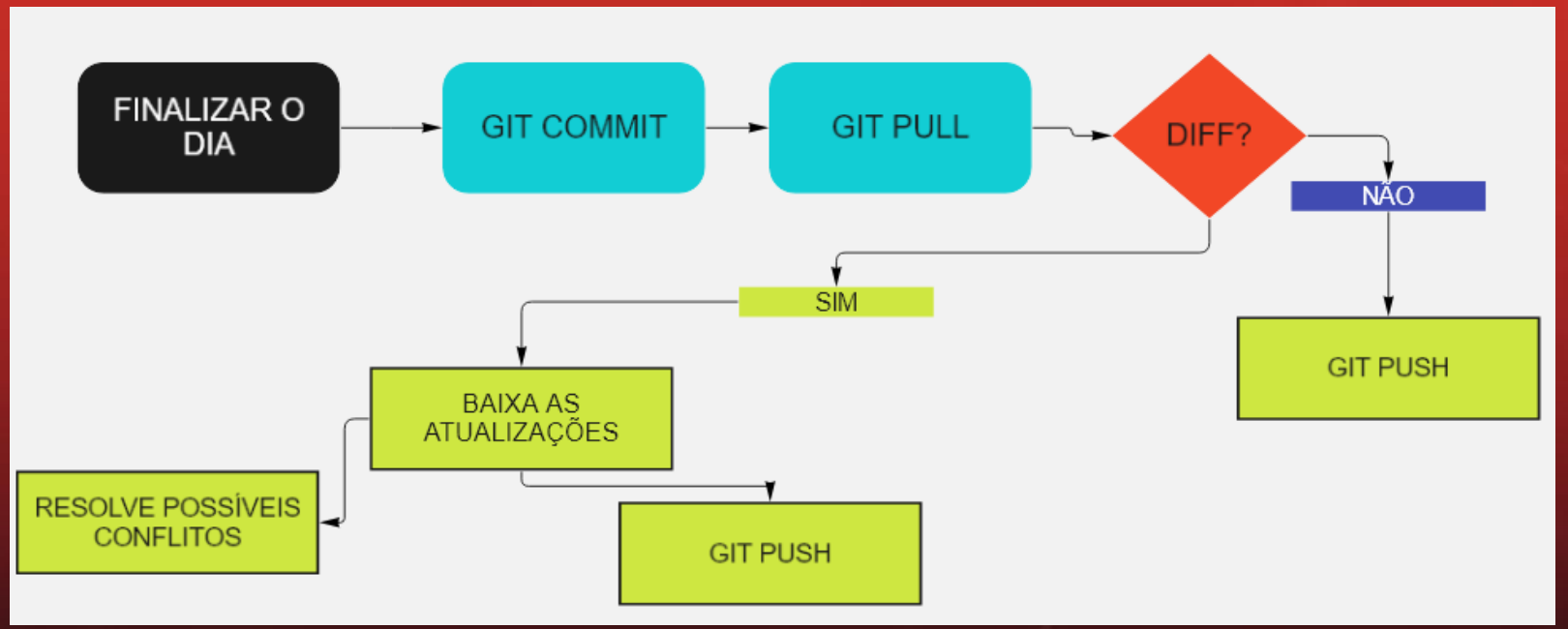
Dia a dia no git

Devemos criar uma rotina diária para utilização do git, para manter assim nossos códigos sempre atualizados tanto em nossa máquina quanto no repositório.

ProWay

ProWay


INICIAR O DIA
GIT PULL
DIFF?
NÃO
SIM
BOM TRABALHO
BAIXA AS ATUALIZAÇÕES
BOM TRABALHO


FINALIZAR O DIA
GIT COMMIT
GIT PULL
DIFF?
NÃO
SIM
GIT PUSH
BAIXA AS ATUALIZAÇÕES
RESOLVE POSSÍVEIS CONFLITOS
GIT PUSH

AO LONGO DO DIA CODAMOS, MODIFICAMOS LINHAS DE CÓDIGO E O NOSSO REPOSITÓRIO VAI CRIANDO STAGES DE QUALQUER ARQUIVO QUE TENHAMOS ALTERADO

ProWay

PARA VERIFICAR O STAGE DO REPOSITÓRIO UTILIZAMOS O COMANDO GIT STATUS

ProWay

ARQUIVOS NÃO ADICIONADOS AO STAGE DE COMMIT FICAM NA COR VERMELHO

ProWay

ARQUIVOS ADICIONADOS AO STAGE PARA COMMIT FICAM NA COR VERDE

ProWay

Commitando alterações

GIT ADD . -> APÓS CONCLUIR NOSSOS TRABALHOS PRECISAMOS EFETUAR UM COMMIT PARA RESGUARDAR NOSSAS ALTERAÇÕES, UTILIZAMOS O COMANDO GIT ADD OU GIT ADD <ARQUIVO>

ProWay

Commitando alterações

GIT COMMIT –M “DESCRIÇÃO” . ->

APÓS ADICIONAR OS ARQUIVOS AO STAGE EFETUAMOS O COMMIT DE FATO.

ProWay

Commitando alterações

GIT PUSH -> APÓS O COMMIT REALIZADO PRECISAMOS EMPURRAR ELE PARA A HOSPEDAGEM GIT QUE UTILIZAMOS, PARA ISSO UTILIZAMOS GIT PUSH OU GIT PUSH ORIGIN <BRANCH>

ProWay

Commitando alterações

SEMPRE QUE FORMOS EFETUAR UM PUSH EM UMA BRANCH DIFERENTE DA MASTER OU MAIN DEVEMOS INFORMAR O ORIGIN <BRANCH>

ProWay

Exercícios individuais

1. Crie um repositório online e faça clone
2. Crie um repositório local e faça todo o procedimento para colocar este repositório para o servidor
3. Crie arquivos no repositório, faça commits, altere arquivos, faça novos commits,

ProWay

Desfazendo commits (resolvendo cacas)

Programas são feitos por seres humanos.

Logo, estão sujeitos a erros, que podem acabar passando desapercebidos e cairem em produção.

ProWay

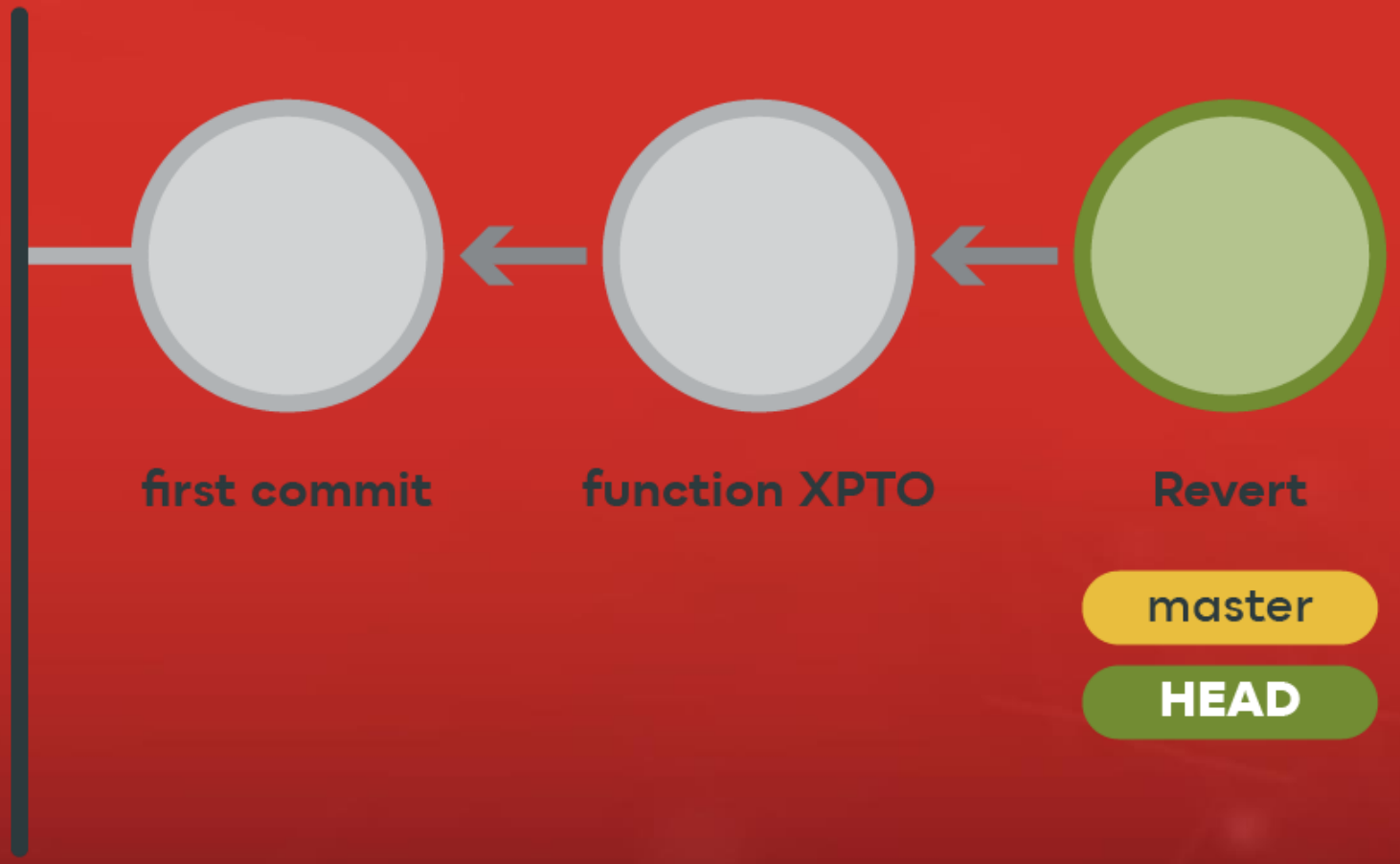
Para isso temos duas formas de "desfazer commits" e resolver o problema que são:

REVERT e RESET

ProWay

REVERT: cria um novo commit desfazendo tudo que foi realizado no commit selecionado a ser desfeito. Como pode ser visto na imagem abaixo, o segundo commit incluiu uma funcionalidade XPTO, mas posteriormente foi verificado que essa funcionalidade poderia ser descartada, então um commit de revert foi gerado.

ProWay



ProWay

RESET: faz que a funcionalidade XPTO conste como se nunca tivesse existido no histórico. Considere a imagem abaixo como resultado de um reset ao invés de revert do exemplo anterior

ProWay

first commit

function XPTO

master

HEAD

ProWay

Branchs - Ramificações

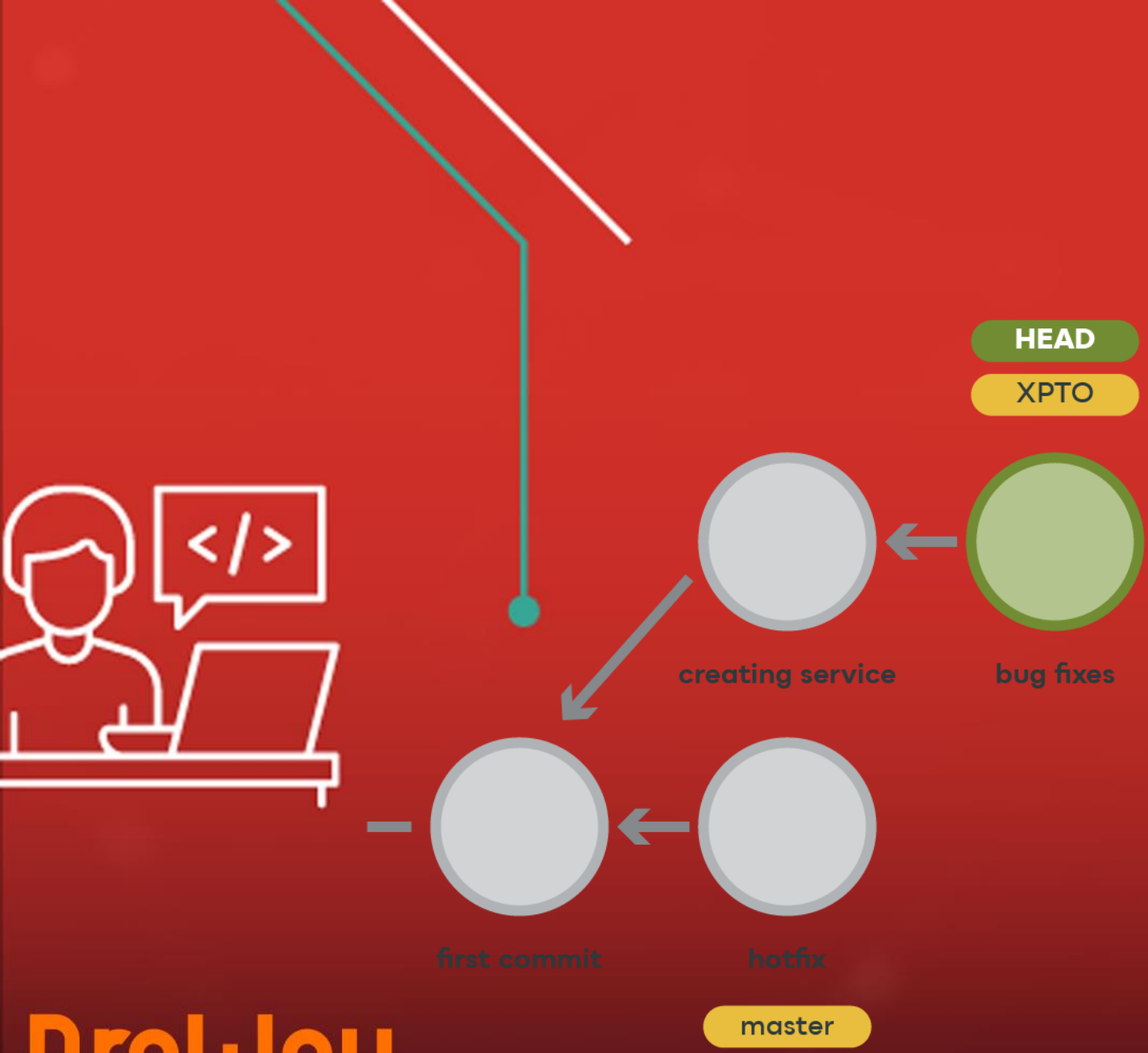
Em teoria, uma branch é uma ramificação de um projeto. Na prática, no Git, uma branch é uma referência para um commit. A branch principal de um projeto chama-se “master”. Porém, há iniciativas para alterar este nome a longo prazo, devido ao termo “master” ser considerado racista.

ProWay

Benefícios de usar branchs

- Rápido e fácil de ser criada: 1 arquivo que contém uma referência;
- Permitem a experimentação: Os membros da equipe podem isolar seu trabalho para que ele não afete os outros até que o trabalho esteja pronto;
- Suporte a várias versões: As ramificações permitem suporte a várias versões do projeto simultaneamente;
- Permite processos de revisão de arquivo: As ramificações permitem incluir processos de revisão de arquivos;

ProWay



Na imagem, uma ramificação do projeto é criado a partir da versão "first commit". Nesse instante, cada ramo terá seus próprios commits enquanto não houver algum mecanismo de fusão entre os ramos. Os principais mecanismos de fusão são: merge ou rebase.

ProWay