Olabi

O PODER DO FUTURO

Formação introdutória à tecnologia
para jovens mulheres negras

Introdução à Programação, Git e Github

preta lab

Introdução à Programação, Git e Github

Cronograma

• Linhas de comando
• Versionamento, Git e GitHub
• Exercícios

Linhas de comando

Algumas interfaces interpretam linhas de comando através de textos, que manipulam arquivos em nossos computadores:
• Command Power / CMD - interpretador Windows simples e funcional;
• PowerShell - criado pela Microsoft, é bem mais robusto, com capacidade maior de programação;
• Bash - criado como *software* livre, é um Unix Shell com linguagem de comando, como o anterior.

Comandos básicos

- pwd: encontra o caminho para o diretório atual da pasta na qual você está;
- ls: lista todos arquivos da pasta na qual você está;
- cd nome-da-pasta: entra em uma pasta dentro da pasta na qual você está;
- cd ~: volta para a pasta raiz na qual você estava;
- cd .. : volta uma pasta;
- mkdir nome-da-pasta : cria uma pasta;
- rm nome-do-arquivo : deleta um arquivo;
- rm -r nome-da-pasta: deleta um repositório;
- whoami : identifica usuário logado.

Exercício 1

Abra o *bash*;
Identifique o usuário;
Confirme a pasta na qual você está;
Crie uma pasta;
Entre na pasta;
Crie um arquivo e insira uma frase;
Tire uma *print* e mostre pra gente!*

Conteúdo metodológico: Amanda Silva

***Exercício 1 – Resposta**

Abra o bash
Identifique o usuário (whoami)
Confirma a pasta na qual está (pwd)
Crie uma pasta (mkdir nome-da-pasta)
Entre na pasta (cd nome-da-pasta)
Crie um arquivo e insira uma frase (pelo terminal: echo frase -> nome.txt linda -> nome.txt; excluindo terminal: rm -f nome.txt)
Tire uma print *e mostre pra gente!*

Exercício 2

Abra o terminal *bash*;
Confirme a pasta na qual está;
Entre na pasta criada anteriormente;
Apague o arquivo criado;
Volte uma pasta;
Apague a pasta criada.*

Exercício 2 – Resposta*

Abra o bash
Confirme pasta na qual estiver (pwd);
Entre na pasta criada antes (cd nome-da-pasta);
Apague o arquivo criado (rm nome-do-arquivo);
Volte uma pasta (cd);
Apague a pasta criada (rm -r nome-da-pasta).

Versionamento de código

Controle de versão

Software que gerencia mudanças de um arquivo de qualquer tipo (.doc, HTML, XML...), é muito usado nas empresas. Com ele, é possível rastrear um arquivo desde o início. Pode ser usado em qualquer tipo de projeto - dos mais simples até os mais complexos - e é utilizado pela maioria das empresas.

Git

O Git é um *software* livre inicialmente projetado e desenvolvido por Linus Torvalds para o desenvolvimento do Kernel Linux. É rápido e eficiente, especialmente em projetos grandes. Git é um dos sistemas de controle de versões existentes mais usados, principalmente no desenvolvimento de *softwares*.

Ferramentas de versionamento

GitHub

GitHub é uma plataforma de hospedagem de código-fonte com controle de versão que utiliza Git. Ela permite que qualquer pessoa cadastrada contribua em projetos privados e/ou de código-fonte aberto (*open source*), de qualquer lugar do mundo.

Git – Conceitos básicos

• **repositório**: pasta/local no qual o projeto é armazenado;
• **clone**: clona (copia) uma versão do repositório remoto do projeto para o local;
• ***branches*** (galhos): útil no desenvolvimento coletivo, permite que cada usuário tenha seu "bracinho" (versão) dentro do projeto de maneira independente;
• ***pull***: puxa as últimas alterações do repositório remoto para o local;
• ***commit***: controla a versão de um arquivo registrando uma mensagem que identifica as últimas alterações;
• ***push***: puxa as últimas alterações do repositório remoto para o local;
• ***merge***: unifica *branches* diferentes;
• ***fork***: cópia de um projeto para a sua conta do GitHub;
• ***pull request***: solicitação de *merge* do seu *branch* em um projeto de outra pessoa;
• ***rebase***: segue o raciocínio do *merge*, mas apaga parte dos *commits* no histórico. Recomendado para uso entre *branches* de desenvolvedores, não diretamente no principal;

Ex.:
git init git *add* git *commit* git *push*

repositório (sem git)	área	repositório local
repositório (com git)	temporária (stagging)	repositório remoto

Remoto Local

Comandos iniciais

Git — Comandos básicos

• git init: inicia o git no repositório local;
• git *add*: adiciona um arquivo modificado ao *stagging* (área temporária);
• git *status*: mostra o status dos arquivos modificados;
• git *commit* -m “mensagem”: cria um *commit;*
• git *pull*: puxa as atualizações mais recente (remoto -> local);
• git *push*: envia as atualizações mais recentes (local -> remoto);
• git *remote add origin* caminho: adiciona seu repositório local ao remoto;
• git *checkout:* nome-arquivo: descarta as alterações locais do arquivo informado.

Exercício 3

Comece com um Git no terminal;
Crie uma pasta;
Navegue até a pasta e inicie o git;
Crie um arquivo e verifique seu status;
Adicione o arquivo ao *stage* do Git;
Faça um *commit;*
Faça um *push*.

Exercício 4

Crie um repositório localmente e inicie o git;
Adicione um arquivo *markdown* chamado README, com seu nome e prato favorito, e faça um *commit;*
Adicione uma curiosidade sobre você e faça outro *commit;*
Publique o repositório no seu GitHub.

Aprofundando no Git

Configurações iniciais

• git config: global user.name “mandypry”;
• git config: global user.email “amanda.adgti@gmail.com”;
• git config: list;
• git: config global:
=
unset user.name “seu usuário”;
git config global: unset user.email “nome@email.com”.

Exercício 5 (algoritmos)

Faça um *fork* do repositório;
Clone o repositório para a sua máquina;
Crie um novo *branch* com seu nome (exemplo: amanda-silva);
Faça *commits* com a resolução dos exercícios;
Atualize seu repositório remoto.

DESAFIO EXTRA

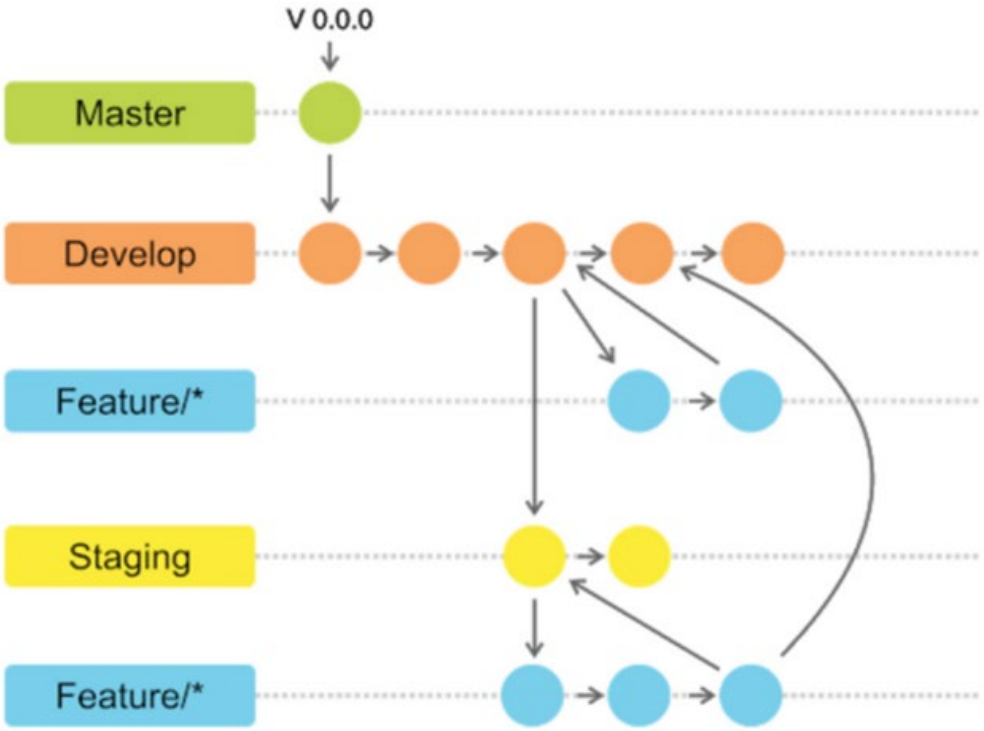
Abra um *pull request* para o repositório original.

Git – conceitos básicos

- git init;
- git *add* ou git *add* nome-do-arquivo;
- git *commit* -m "mensagem do *commit*" git status;
- git *remote add origin* url-do-repo ou ssh-do-repo;
- git remote -v;
- git *push origin* nome-do-*branch*;
- git *clone* url-do-repo;

Git – Sobre branches

- git *branch*;
- git *checkout* -b nome-da-branch;
- git *checkout* nome-da-branch;
- git *branch* -d nome-da-branch;
- git *push origin – delete* nome-do-*branch;*
- git *merge* nome-do-*branch*.



Git – sobre forks

• git clone url-do-seu-*fork*;
• git *remote add upstream* url-do-repo-original;
• git *fetch* (ou *pull*) *upstream*;
• git *rebase* (ou *merge*) *upstream/main*

Exercício 6

No repositório/ pasta local:
- Crie um branch com seu nome;
- Crie um arquivo com seu nome e escreva algo dentro;
- Adicione esse arquivo ao git;
- Faça um commit;
- Faça um git push para o seu repositório;
- No seu repositório no GitHub faça um pull request entre forks, ou seja, envie seu branch para o do repositório original.

Exercício 7

- Crie um repositório no GitHub, sem o Readme.md;
- Crie uma pasta no seu computador;
- Abra o git *bash*;
- Adicione seu repositório remoto como de origem;
- Dê o comando git init;
- Crie um branch com seu nome;
- Dentro da pasta, crie um arquivo e escreva algo;
- Adicione o arquivo ao git;
- Faça um *commit*;
- Envie as alterações para o seu repositório remoto

No último passo, colocar o nome do seu *branch* local após o *origin*!

Obrigada pela presença, maravilhosas!

amanda-silva-dev

FICHA TÉCNICA

Equipe Olabi: Gabriela Agustini, Silvana Bahia, Aldren Flores, Davi Arloy, Joyce Santos, Amanda Oliveira, Roberta Hélcias, Clara Queiroz e Rodrigo Schmitt

Gestão do Projeto: Aldren Flores

Assistente do Projeto: Joyce Santos

Facilitadora: Vera Félix

Comunicação: Raiz Digital

Social media: Raiz Digital e Amanda Oliveira

Coordenadora Técnica: Amanda Silva

Percurso metodológico: Amanda Silva

Conteúdo metodológico das apostilas: Amanda Silva

Revisão gramatical: Luciana Moletta

Design: Bruna Martins

Professoras: Amanda Silva, Letícia Furtado, Lisandra Souza, Mônica Santana, Renata Nunes e Simara Conceição

Monitoras: Ana Beatriz dos Santos, Angela Karolina Lopo e Renata Silva

Apoio: Disney

Introdução à Programação, Git e Github

O PODER DO FUTURO

Formação introdutória à tecnologia para jovens mulheres negras

preta lab

Olabi