INTRODUÇÃO AO GIT/GITHUB

GIT foi criado em 2005, criado por Linus Torvalds(Criador por trás do Linux)

Linux foi feito de forma colaborativa, e precisava ter um sistema que criasse versões que fossem complexas suficientes para que suportasse pessoas do mundo todo utilizando ao mesmo tempo.

SOFTWARE não é feito se não for colaborativo.

GITHUB, Empresa da Microsoft

GITHUB E GIT APARECEM GERALMENTE JUNTOS POR SEREM FORTES NO MERCADO.

São completamente diferente uma da outra.

Embora sejam muito utilizadas não estão sozinha

Há outra opções como BITKEEPER CVS para o GIT

E para GITHUB há o Gitlab etc.

Benefícios de estar aprendendo amba as tecnologias

* Controle de versão
* Armazenamento em nuvem
* Trabalho em equipe
* Melhorar seu código
* Reconhecimento

NAVEGAÇÃO BÁSICA NO TERMINAL & INSTALAÇÃO.

GUI X CLI

O que vamos aprender?

* Mudar de pastas
* Listar as pastas
* Criar pastas/arquivos
* Deletar pastas/arquivos

WINDOWS

- cd – PARA NAVEGAR PELAS PASTAS

- dir – PARA LISTAR

- mkdir – Para criar pasta

- del(deletar arquivo)/rmdir (deletar pasta) /S /Q

- ls retorna oque tem na pasta em si

cd .. retrocede para a navegação o anterior

cls – limpar o terminal do Windows

echo – devolve uma palavra digitada

* Redirecionado de fluxo / utilizar junto ao echo.

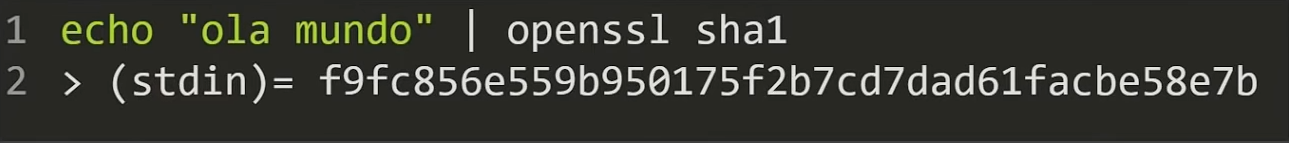
ENTENDENDO COMO O GIT FUNCIONA POR BAIXO DOS PANOS.

* SHA1

A sigla SHA significa Secure Hash Algorithm (Algoritmo de Hash Seguro), é um conjunto de funções hash criptográficas projetadas pela NSA(Agência de Segurança Nacional dos EUA).

A encriptação gera conjunto de caracteres identificador de 40 dígitos.

É uma forma curta de representar um arquivo.



* Objetos fundamentais(Blobs)

BLOBS armazena metadados do GIT (TIPO DO OBJETO/ TAMANHO/UMA \0 + CONTEUDO)

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Tree(arvores) armazena BLOBS, Nome do arquivo, será responsável por montar toda a estrutura dos arquivos de onde está localizados, pode ser apontado tanto pra um blob como para outra Tree, a tree tem um sha1 com os metadados das Tree, se mudar um arquivo por consequência estará mudando toda a estrutura

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Diagrama, Aplicativo

Descrição gerada automaticamente

Commit, objeto que irá juntar tudo, oque dará sentido para oque está fazendo, ele aponta para uma arvore, aponta para um parente(ultimo commit), aponta para um autor, aponta para uma mensagem.

Eles tb possuem um SHA1, significa que se alterar uma blob, irá gerar um sha1 da blob, a mesma tem uma tree apontando para ela, irá altera os metadados da tree e o commit aponta para uma arvore que também pode apontar para outras arvores, e caso vc mude algo vc pode refletir na arvore que irá refletir no seu commit.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

* Sistema distribuído
* Segurança

CHAVES SSH E TOKENS

Chave SSH

Forma de estabelecer uma conexão segura e encriptada entre duas máquinas

Primeiros comando com GIT

* Iniciar o GIT
* git init
* Iniciar o versionamento
* git add
* Criar um commit
* git commit

Criando um repositório

flag ls -a (arquivos ocultos)

git add \*

git commit -m “commit inicial”

Adicionando um arquivo dentro da pasta

Texto

Descrição gerada automaticamenteInterface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamenteTexto

Descrição gerada automaticamente

Markdown = estrutura simples, porem pode fazer coisas complexas.

Ciclo de vida dos arquivos

GIT INIT

Diagrama

Descrição gerada automaticamente com confiança média

Untracked = git não reconhece a existência.

Unmodified = arquivo ainda não alterado

Modified = arquivo alterado

Staged = Conceito chave

Quando é movido para Staged, está aguardando para virar um commit.

Commit retorna os arquivos para unmodified, para começar o ciclo novamente.

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamenteInterface gráfica do usuário, Texto

Descrição gerada automaticamente

Linha do tempo

Descrição gerada automaticamente

Linha do tempo

Descrição gerada automaticamente

Git status – ajuda a monitorar os status dos arquivos, caso ele esteja untracked/modified/staged etc, e será utilizado com frequencia

mv = comando para mover = mv nome do arquivo . /pasta

git commit -a para comitar

git add = mover para a área de stage

git add nomeArquivo = movendo os arquivos para staged

git commit -m “msg” = envelopou os arquivos e criou o arquivo no git chamado commit

Criando repositório no GITHUB

Caso queira alterar verificar no

Git config --list

Git config –global –unset user.email

git config –global user.email “”

Copiar https no github e dentro do GITBASH empurrar para a origem remota

Git remote add origin “(link do repositorio)”

Git remote -v (lista de repositórios remotos cadastrados)

Git status (para verificar se não tem nenhuma pendencia)

Git push origin master

RESOLVENDO CONFLITOS

Git pull origin master – para puxar o conteúdo e corrigir o conflito de Merge

Para clonar um código do github utilizar

Git clone (url)