Aula 1

Introdução ao Git e GitHUB

O que é ?

Aula:

Git é um sistema de versionamento de código distribuído.

Obs1: O software não se cria sozinho ele é criado de forma colaborativa.

Exemplo reportagem e controle de direção.

Outra Fonte:

De longe, o sistema de controle de versão moderno mais usado no mundo hoje é o Git. O Git é um projeto de código aberto maduro e com manutenção ativa desenvolvido em 2005 por Linus Torvalds, o famoso criador do kernel do sistema operacional Linux. Um número impressionante de projetos de software depende do Git para controle de versão, incluindo projetos comerciais e de código-fonte aberto. Os desenvolvedores que trabalharam com o Git estão bem representados no pool de talentos de desenvolvimento de software disponíveis e funcionam bem em uma ampla variedade de sistemas operacionais e IDEs (Ambientes de Desenvolvimento Integrado).

Tendo uma arquitetura distribuída, o Git é um exemplo de DVCS (portanto, Sistema de Controle de Versão Distribuído). Em vez de ter apenas um único local para o histórico completo da versão do software, como é comum em sistemas de controle de versão outrora populares como CVS ou Subversion (também conhecido como SVN), no Git, a cópia de trabalho de todo desenvolvedor do código também é um repositório que pode conter o histórico completo de todas as alterações.

Além de ser distribuído, o Git foi projetado com desempenho, segurança e flexibilidade em mente.

Aula:

Benefícios:

. Controle de versão

. Armazenamento em nuvem

. Trabalho em equipe

. Melhorar seu código

. Reconhecimento

Entendendo como o Git funciona

Tópicos:

SHA1

Objetos Fundamentais

Sistema Distribuido

Segurança

SHA1

O Secure Hash Algorithm é o conjunto de funções hash criptográficas projetadas pela NASA que gera uma encriptação na forma de um conjunto de caracteres identificador de 40 dígitos.

Objetos fundamentais:

Blobs: “Bolha” ou bloco de encriptação, contém metadados de arquivos.

Trees: “Arvores”, armazenam as *blobs* e também contém metadados dos quais nome do arquivo, estrutura onde estão os arquivos, SHA1 do metadado da Tree.

Commits: “Junta toda a informação e organiza para fazer sentido”. Também contendo SHA1 dos quais também estão inseridos a *tree*, parente, autor, messagem “inicio..” e  *timestamp.*

Sistema distribuído seguro:

Todos os *manteiners* possuem versões confiáveis do repositório original.

Formas de autenticação:

Chave SSH:

A chave SSH é fornecida através do sistema após cadastramento através de um arquivo que identifica e autentica a máquina que foi cadastrada.

Tokens:

Os tokens são arquivos que auxiliam na autenticação de usuários.

Security frase:

São as informações de login e senha cadastradas.

Comando básicos de navegação no terminal:

Windows:

cd / cd .. – navega de uma pasta para outra, cd entra e cd .. sai retorna.

Dir – relaciona todos os diretórios contidos na pasta

Mkdir – Cria novo diretório

Del / rmdir – Deleta

Linux:

Cd – navega de uma pasta para outra

Ls – relaciona todos os diretórios

Mkdir – cria novo diretório

-rm / -rf – deleta

Comandos Git

Iniciar Git – inicia versionamento e cria um commit

Git init – iniciar um repositório

Git add – Cria arquivos

Git commit – m “”– Cria commit e deixa mensagem.

Git add \* - Repositório local para Stage Area

Git add . – Repositório local para Stage Area

Comandos remote

Git remote -v

Git push origin main – empurra arquivos para a nuvem

Git pull – puxa arquivos na nuvem.