GitHub – Git-SCM (2005)

Sistema de Controle de Versão Distribuído

SCM – Source Control Management

- What changed?

- Why it was changed?

- Who made the changes?

- The changes can be reproduced?

Tópicos

Identificação

Documentação (Interna / Desenvolvimento)

Controle (Teste/Verificação)

Auditoria (Alinhamento com projeto)

Artefatos:

- Código Fonte

- Documentação do Software

- Manual de Usuário

**Controle de Versão:**

Exemplo: - Múltipla edição/sobrescrição de arquivos por diferentes agentes

Controle de versão possibilita o merge (junção)

Ferramentas de controle de versão:

- Subversion (SVN) (Mais antigo)

- Mercurial (Hg)

- CVS – Concurrent Versioning System (legado)

- Bazaar (pouco utilizado)

- Git mais rápido e eficiente

- Git tem execução distribuída (cada agente tem espelho do código)

- Possibilidade de execução sem conexão com servidor

# O que é o GitHub

Git – ferramenta de controle de versão

GitHub – um servidor de repositórios Git

Outros servidores:

GitHub – maioria dos projetos armazenados

Assinatura mensal para repositórios privados

GitLab – Tem repositórios privados sem necessidade de pagamento

Bitbucket – Suporta Git e Mercurial (repositório privado – limite 5 pessoas) (Não tem chave SSH)

SourceForge – Git/Mercurial/SVN

**Instalação:**

sudo apt-get install git

Escolher uma Interface Gráfica

Recomendação GitEye

<https://www.collab.net/products/giteye>

# Básico em Git

## Chave SSH

<https://help.github.com/en/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

Ajuda

git help <command>

Documentação: <https://git-scm.com/>

# Comandos Git

## Git Add

Adicionar os aqruivos novos e modificados para o próximo commit:

git add <lista de arquivos>

Adicionar todos os arquivos na pasta para o próximo commit

git add .

Registrar o commit com todos os arquivos que usou git add

git commit [-m “Mensagem”]

Se o parâmetro Mensagem não for passado, abrirá um editor de texto para escrever a mensagem (Configurar o editor de texto)

git config –global core.editor gedit

git config --global core.editor "/Applications/Sublime\ Text.app/Contents/SharedSupport/bin/subl -n -w"

<https://swcarpentry.github.io/git-novice/02-setup/>

## Git Workflow

Maior parte dos trabalhos consiste nestas tarefas

* Editar
* Commitar
* Soncronizar com o repositório remoto

Revisão de comandos

Status do repositórios – arquivos não adicionados/adicionaos ao commit

git status

Últimos commits realizados

git log

Adicionar arquivos para o commit

git add

Criar o commit

git commit

Enviar todos commits que ainda não foram enviados

git push

# Estado dos arquivos

Não monitorado

untracked

Modificado

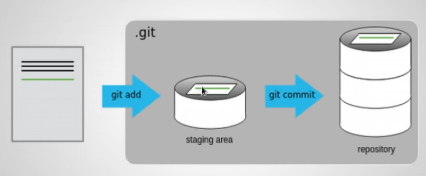
modified

Preparado (após git add)

staged

Consolidado (após o git commit)

commited



# Diff

Exibir diferenças entre commits e branchs

git diff

Diferenças em diretório específico

git diff [path]

Exibir alterações no último commit

git diff HEAD~1

# Aula 3

## Git Clone

Baixa o repositório remoto (cria o repositório com os dados remotos)

git clone

Outra forma de criar um repositório local

## Git Pull

Baixa as alterações do repositório remoto (Mantém o repositório sincronizado com os últimos commits de uma branch)

git pull

## Navegação no histórico

Permite verificar como um arquivo ou todo repositórios estava em um determinado commit

git checkout <commit> <file>

Retornar do modo detached head – para o estado atual (último commit)

git checkout máster

## Desfazendo alterações

Desfazer todas as alterações que não estejam no Stage (antes de git add) desde o último commit

git checkout -- <path\_or\_filename>

Para todos os arquivos:

git checkout -- .

Desfazer alterações desde o último commit incluindo o Stage (após git add)

Git checkout HEAD -- <path\_or\_filename>

## Git Revert

Reverte um commit – criando um commit “ao contrário” do commit realizado anteriormente

git revert <commit>

## Git reset

Reseta o repositório para um determinado commit

git reset <commit>

Resetar e remove todas as alterações (CUIDADO AO USAR! NÃO USAR SE JÁ ESTIVER PUBLICADO (enviado para GitHub – melhor usar revert))

git reset –hard <commit>

Útil para desfazer últimos commits antes do push

Reverter último commit:

git reset HEAD~1

Reverter últimos 2 commits (inclusive as alterações nos arquivos)

git reset HEAD~2 --hard