

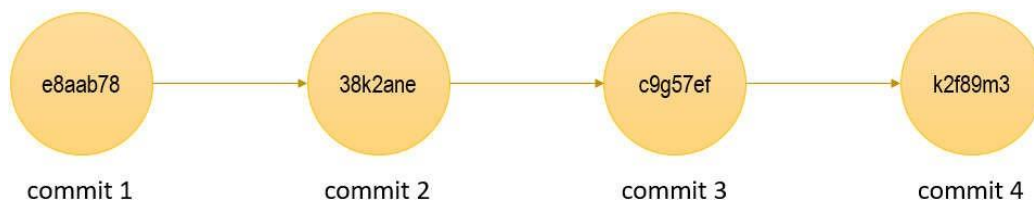
Nivelamento sobre Git e Github

Objetivo geral:

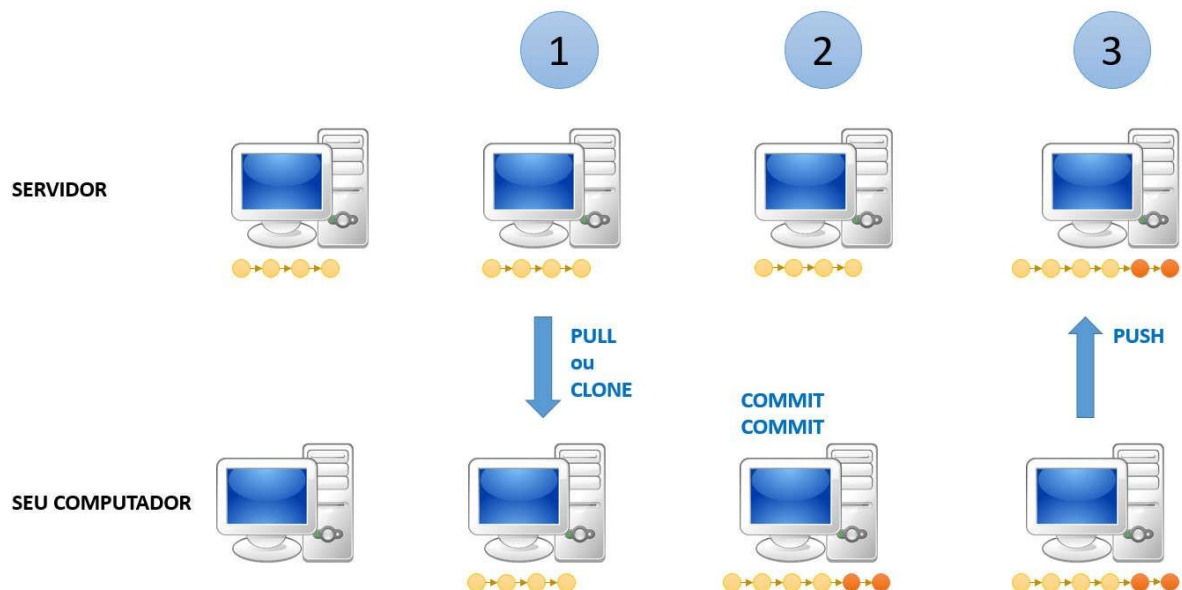
- Oferecer um breve nivelamento sobre aspectos teóricos e procedimentos básicos de Git e Github necessários para acompanhar os projetos do curso
- Vamos abordar: usuário único / *branch* único

O que é Git e Github?

GIT - é um **sistema de versionamento**: você controla as modificações de um projeto por meio de versões chamadas "commits".



Um projeto controlado pelo Git é chamado de **repositório de versionamento**. Tipicamente uma cópia "oficial" do repositório fica salvo em um **servidor** (repositório remoto). Cada pessoa que trabalha no projeto pode fazer uma cópia do repositório para seu computador (repositório local). A pessoa então faz suas alterações no projeto (novos commits) e depois salva as alterações no servidor.



Operações principais:

| | |
|---------------|---|
| CLONE | Copia o repositório remoto para seu computador |
| PULL | Atualiza seu repositório local em relação ao repositório remoto |
| COMMIT | Salva uma nova versão (tipicamente no seu repositório local) |
| PUSH | Envia o repositório local para o repositório remoto |

Como utilizar o Git no meu computador?

- É preciso ter um sistema Git instalado em seu computador
 - **Linux**: já possui
 - **Mac**: já possui / instalação semiautomática
 - **Windows**: <https://git-scm.com>
- O Git pode ser usado por ou por **terminal de comando**, ou por um aplicativo gráfico.

GITHUB - é um serviço de hospedagem de repositórios Git remotos.

- Possui uma interface gráfica web: github.com
- É uma plataforma social (usuários, página de perfil, seguidores, colaboração, etc.). **Dica: currículo!**
- Maior serviço do mundo de hospedagem de projetos de código aberto
- Modelo de cobrança: gratuito para projetos de código aberto, pago para projetos privados
- Alternativas: BitBucket, GitLab, etc.

Instalação do Git e GitBash no Windows

Download: <https://git-scm.com>

Opções recomendadas:

- Use Vim as Git's default editor
- Use Git from the Windows Command Prompt
- Use the OpenSSL library
- Checkout Windows-style, commit Unix-style line endings
- Use MinTTY
- **DESMARCAR**: Enable Git Credential Manager

Demo - criar um novo projeto

ATENÇÃO USUÁRIOS WINDOWS: configurar o sistema para mostrar extensões de arquivos

Painel de controle -> Opções de pasta -> Modo de exibição

- Desmarcar "Ocultar as extensões de tipos de arquivos conhecidos"

Passos:

1. Crie um novo projeto no seu ambiente de desenvolvimento
2. Crie um novo repositório no Github
 - a. **ATENÇÃO:** se seu ambiente de desenvolvimento não gera o arquivo `.gitignore` automaticamente, escolha o seu tipo desejado de `.gitignore` na tela de criação de repositório do Github
3. Abra um terminal na pasta do seu projeto
4. **IMPORTANTE:** certifique-se de estar identificado no Git do seu computador (nome e email)
`git config --list`
`git config --global user.name "Nelio Alves"`
`git config --global user.email "acnelio@gmail.com"`
5. Faça os comandos a seguir

| | |
|---|--|
| <code>git init</code> | Inicia um novo repositório local na pasta do seu projeto |
| <code>git remote add origin https://github.com/acnelio/meuprojeto.git</code> <i>ATENÇÃO: troque pelo caminho do seu repositório do Github</i> | Associa seu repositório local ao repositório remoto, com o apelido de "origin" |
| <code>git pull origin master / git pull origin main</code> <i>ATENÇÃO: este comando só é necessário se você criou o <code>.gitignore</code> pelo Github</i> | Atualiza seu repositório local em relação ao repositório remoto |
| <code>Git checkout -b main</code> (Esse passo é no caso do branch ser main) | Checkout para mudar o branch, e <code>-b</code> para criar caso ele não exista |
| <code>git status</code> | Verifica arquivos |
| <code>git add .</code> | Adiciona todos arquivos ao stage |
| <code>git commit -m "Projeto criado"</code> | Salva uma nova versão do projeto |
| <code>git push -u origin master</code> | Envia o repositório local para o repositório remoto |
| Nota: nas próximas vezes basta fazer: <code>git push</code> | |

Arquivo .gitignore

ATENÇÃO USUÁRIOS WINDOWS: configurar o sistema para mostrar extensões de arquivos

Painel de controle -> Opções de pasta -> Modo de exibição

- Desmarcar "Ocultar as extensões de tipos de arquivos conhecidos"

Checklist:

- Pra quê serve o arquivo `.gitignore`?
- Como saber o que colocar no arquivo `.gitignore`?

Demo - Trabalhando com um projeto existente

Passos:

1. **IMPORTANTE:** certifique-se de estar identificado no Git do seu computador (nome e email)

USUÁRIOS WINDOWS: verifique o "Cofre" (gerenciador de credenciais)

```
git config --list
git config --global user.name "Nelio Alves"
git config --global user.email "acenelio@gmail.com"
```

2. Se o projeto já não estiver no seu computador, *Clonar* o repositório do Github

```
git clone https://github.com/acenelio/meuprojeto.git
```

3. Altere o que você precisar no projeto
4. *Commitar* as mudanças feitas no projeto

4.1. Verificar arquivos:

```
git status
```



4.2. Realizar o commit:

```
git add .
git commit -m "Mensagem explicativa"
```

4.3. Listar histórico de commits:

```
git log --oneline
```

5. Salvar projeto atualizado no seu Github (*push*)

```
git push
```

Importante entender - a sequência de commits deve ser respeitada

POR EXEMPLO:

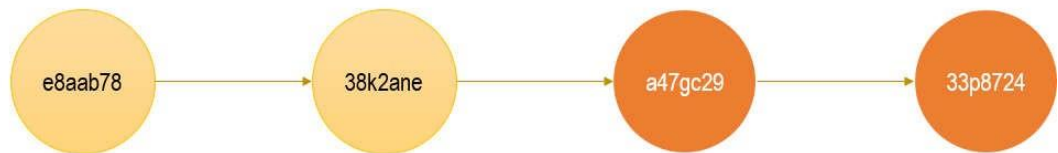
Suponha que o repositório do seu computador estava em uma versão atrasada em relação ao Github, daí você esquece de atualizá-lo (`git pull origin master`) e faz algumas alterações.

O Git não vai aceitar o *push* neste caso.

**NO
GITHUB:**



**NO SEU
COMPUTADOR:**



Kit de primeiros socorros

| PROBLEMA | COMANDO |
|---|--|
| Quero desfazer tudo que eu fiz desde o último commit | <code>git clean -df</code> <code>git checkout -- .</code> |
| Preciso remover o último commit, porém mantendo os arquivos do jeito que estão. | <code>git reset --soft HEAD~1</code> |
| Preciso remover o último commit, inclusive as alterações nos arquivos. | <code>git reset --hard HEAD~1</code> |
| Quero alterar temporariamente os arquivos do projeto de modo a ficarem no estado do commit informado. ATENÇÃO: não podem haver modificações não commitadas no projeto. NOTA: para voltar ao último commit faça: <code>git checkout master</code> | <code>git checkout <código do commit></code> <i>EXEMPLO: <code>git checkout e8a52f3</code></i> |
| Preciso apagar o último commit no Github | <code>git push -f origin HEAD^:master</code> |
| Quero mudar o meu repositório remoto "origin" | <code>git remote set-url origin https://github.com/acenelio/novoprojeto.git</code> |
| Entrei no VIM por engano. Como sair? | Tecle ESC, depois digite :q! e tecla ENTER |