[Uma breve citação para começar]

"Isso aqui vai ser uma palestra de GitHub intermediário."

— Um sábio contemporâneo (2025)

Paráfrase livre de uma lenda que se foi...

for_code[ ]

[Capacitação GitHub]

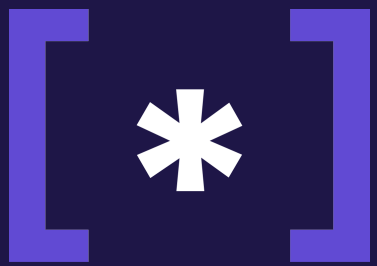
Ronivaldo D. Andrade - Vice Diretor de Projetos – 8 de outubro de 2025

[O que é GitHub?]

- Plataforma de hospedagem de código-fonte
- Utiliza sistema de controle de versão Git
- Permite colaboração em projetos
- Oferece recursos como:
  - GitHub Pages
  - GitHub Actions
  - Pull Requests
  - Code Review

Mão na massa!

No Capítulo 2 do Guia, a partir do Item 2.2, temos um guia resumido de como criar sua primeira conta no GitHub.



[GitHub Student Developer Pack]

Benefícios para Estudantes

- **Conta GitHub Pro gratuita**
- **3.000 minutos** do GitHub Actions (vs 2.000)
- **2GB** de armazenamento (vs 500MB)
- **GitHub Copilot** incluso
- **180 horas** de Codespaces (vs 120)
- **Confira o Guia, Capítulo 2, Item 2.2.1 para maiores informações.**

Como obter

1. Adicionar e-mail acadêmico
2. Acessar education.github.com/pack
3. Enviar comprovante de matrícula
4. Aguardar a análise do documento e a aprovação

[Instalação e Configuração]

Ferramentas Necessárias:

- Winget - Gerenciador de pacotes
- Git - Controle de versão
- GitHub CLI - Interface de linha de comando

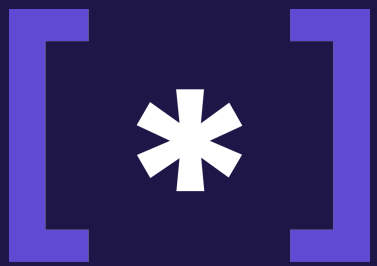
Comandos de Instalação:

- `winget install Git.Git`
- `winget install GitHub.cli`
- `gh auth login`

Configuração Básica do Git

```
git config --global user.name "Seu Nome"
git config --global user.email "seu@email.com"
```

Confira o Guia, Capítulo 3 e 4 para maiores informações.



[Comandos Git Essenciais]

Comandos Básicos:

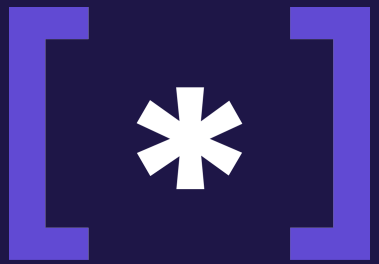
- `git init` - Inicializa repositório
- `git clone` - Clona repositório remoto
- `git add` - Adiciona arquivos ao stage
- `git commit` - Salva alterações
- `git push` - Envia para o remoto
- `git pull` - Atualiza do remoto

Fluxo Básico:

1. `git add .`
2. `git commit -m "msg"`
3. `git push`

Branching:

- `git branch`
- `git checkout -b`
- `git merge`



[Git LFS - Arquivos Grandes]

O que é Git LFS?

Extensão do Git para versionar arquivos grandes (imagens, vídeos, modelos)

- **Problema:** Git tradicional não lida bem com arquivos >100MB
- **Solução:** Armazena apenas ponteiros no repositório
- **Conteúdo real** fica em servidor separado

Como usar

```
git lfs install
git lfs track "*.psd"
git lfs track "*.zip"
git add .gitattributes

git add arquivo-grande.zip
```

[Commits Profissionais]

Boas Práticas

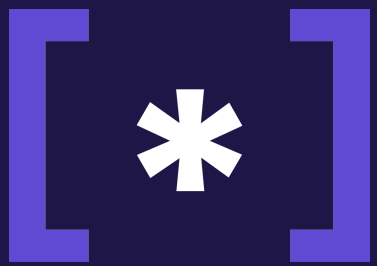
- **Commits atômicos** - Uma mudança lógica por commit
- **Mensagens claras** - Assunto + descrição (se necessário)
- **Modo imperativo** - "Adiciona", "Corrige", "Remove"
- **Referência a issues** - "Closes #42", "Fixes #123"

Exemplo de Commit

git commit -m "<tipo>(escopo): <descrição>"
git commit -m "feat(login): adiciona autenticação via OAuth2"

Implementa sistema de login usando OAuth2 do Google.
Inclui validação de tokens e tratamento de erros.

Closes #15



[Merge vs Rebase]

Merge

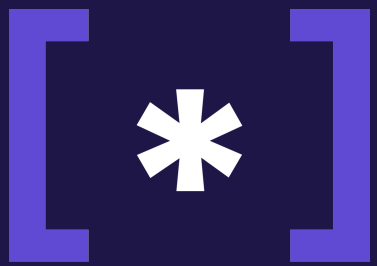
- Preserva histórico completo
- Cria commit de merge
- Mais seguro para histórico compartilhado
- `git merge feature-branch`

Rebase

- Histórico linear e limpo
- Reescreve histórico
- Ideal para branches locais
- `git rebase main`

Importante!

- **Nunca** faça rebase em histórico compartilhado
- Use `--force-with-lease` em vez de `--force`
- Combine com a equipe qual estratégia usar



[Pull Requests]

O que são Pull Requests?

Solicitação formal para integrar mudanças de uma branch em outra, com revisão de código

Fluxo Básico:

1. Criar branch para feature/bugfix
2. Desenvolver e commitar localmente
3. Fazer push da branch
4. Criar PR no GitHub
5. Revisão e aprovação
6. Merge no branch principal

Via GitHub CLI

```
gh pr create --title “Minha feature” --body “Descrição”
```

[GitHub Pages & Actions]

GitHub Pages

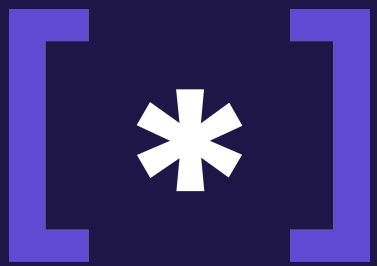
- Hospedagem gratuita de sites estáticos
- Suporte a domínios customizados
- Integração com Jekyll, Hugo, etc.
- Atualização automática

GitHub Actions

- Automação de CI/CD
- Testes automatizados
- Deploy automático
- Workflows customizáveis

Exemplo de Integração

Push na main → Tests → Build → Deploy no GitHub Pages



[Assinatura GPG de Commits]

Por que assinar commits?

- **Autenticidade** - Confirma que você fez o commit
- **Integridade** - Garante que não foi alterado
- **Verificação** - Mostra "Verified"no GitHub

Configuração:

1. Gerar chave GPG: `gpg --full-generate-key`
2. Configurar Git: `git config --global user.signingkey ID`
3. Adicionar chave pública no GitHub
4. Commitar com `git commit -S -m "msg"`

[Exercícios Práticos]

O que vamos praticar?

- Criar repositório via GitHub CLI
- Clonar e fazer mudanças
- Trabalhar com branches
- Fazer Pull Requests
- Configurar GitHub Pages
- Criar workflow básico no Actions

Mão na massa!

Agora acessem o capitulo 9 do Guia que preparei, pois lá tem o passo a passo para alguns exercícios.

for_code[ ]

Obrigado!

Capacitação em GitHub

Vice Diretor de Projetos: Ronivaldo D. Andrade

8 de outubro de 2025