APOSTILA DE GIT PARA INICIANTES EM EQUIPE (WINDOWS E LINUX)

Guia prático para aprender Git colaborando em um projeto compartilhado

---

ÍNDICE

---

1. Introdução ao Git e Controle de Versão

O **Git** é uma ferramenta essencial para desenvolvedores, permitindo que vários programadores colaborem no mesmo projeto de forma organizada e segura.

Vantagens:

- Acompanhar cada alteração feita no projeto
- Reverter facilmente erros ou falhas
- Trabalhar em equipe sem sobrescrever o trabalho dos colegas

---

2. Instalação do Git

Windows

1. Baixe o instalador: Git for Windows
2. Execute o instalador e aceite as opções padrão.
3. Após a instalação, você poderá utilizar o Git pelo **Prompt de Comando (CMD)** ou pelo **Git Bash** (recomendado).

Linux (Debian/Ubuntu)

bash
CopiarEditar
sudo apt update
sudo apt install git

---

3. Configuração Inicial do Git

Após a instalação, configure seu nome e e-mail para associar suas alterações ao seu perfil:

bash
CopiarEditar
git config --global user.name "Seu Nome"
git config --global user.email "seuemail@exemplo.com"

Verifique se as configurações foram salvas corretamente:

bash
CopiarEditar
git config --list

---

4. Configurando Autenticação Automática

Opção 1: Credential Manager (Recomendado para Windows)

O **Credential Manager** armazena suas credenciais de forma segura.

No Windows, execute:

bash
CopiarEditar
git config --global credential.helper manager

Agora, ao executar git push ou git pull pela primeira vez, suas credenciais serão salvas automaticamente.

Opção 2: Configurar SSH (Recomendado para segurança e projetos maiores)

1. **Gerar chave SSH:**

bash
CopiarEditar
ssh-keygen -t ed25519 -C "seuemail@exemplo.com"

2. **Adicionar a chave ao agente SSH:**

bash

CopiarEditar
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519

3. Copiar a chave pública:

bash
CopiarEditar
cat ~/.ssh/id_ed25519.pub

4. Adicionar essa chave ao seu repositório remoto (ex.: GitHub → *Settings* → *SSH and GPG keys*)
5. Testar conexão:

bash
CopiarEditar
ssh -T git@github.com

---

5. Comandos Essenciais do Git

Iniciar um repositório:

bash
CopiarEditar
git init

Clonar um repositório existente:

bash
CopiarEditar
git clone <URL_DO_REPOSITORIO>

Verificar o status do repositório:

bash
CopiarEditar
git status

Adicionar arquivos ao controle de versão:

bash
CopiarEditar
git add <nome_do_arquivo>
git add . # Adiciona todos os arquivos

Criar um commit com uma mensagem clara:

bash
CopiarEditar
git commit -m "Descrição objetiva da alteração"

Enviar as alterações para o repositório remoto:

bash
CopiarEditar

git push origin <nome_da_branch>

Baixar as alterações mais recentes do repositório remoto:

bash
CopiarEditar
git pull origin <nome_da_branch>

---

6. Fluxo de Trabalho Colaborativo (Workflow)

Fluxo padrão para projetos em equipe:

1. Clonar o repositório:

   bash
   CopiarEditar
   git clone <URL_DO_REPOSITORIO>

2. Criar uma branch para sua tarefa:

   bash
   CopiarEditar
   git checkout -b feature/nome-da-tarefa

3. Adicionar suas alterações:

   bash
   CopiarEditar
   git add .
   git commit -m "Implementação da tela de cadastro"

4. Enviar sua branch para o repositório remoto:

   bash
   CopiarEditar
   git push origin feature/nome-da-tarefa

5. Criar um Pull Request (PR) no GitHub/GitLab/Bitbucket
6. Após aprovação, fazer merge com a branch principal:

   bash
   CopiarEditar
   git checkout main
   git merge feature/nome-da-tarefa

---

7. Branches e Controle de Versões

Criar uma nova branch:

bash
CopiarEditar
git checkout -b feature/nova-funcionalidade

Mudar de branch:

bash
CopiarEditar
git checkout main

Listar branches existentes:

bash
CopiarEditar
git branch

Excluir uma branch após merge:

bash
CopiarEditar
git branch -d feature/nova-funcionalidade

---

8. Revertendo Alterações e Histórico

Verificar histórico de commits:

bash
CopiarEditar
git log

Reverter um commit específico:

bash
CopiarEditar
git revert <hash_do_commit>

Resetar o repositório para um estado anterior:

bash
CopiarEditar
git reset --hard <hash_do_commit>

---

9. Exercício Prático: Projeto Colaborativo para 7 Alunos

Desafio:

Cada aluno será responsável por desenvolver uma parte do site de uma loja virtual.

Passo 1: Clonar o repositório

bash
CopiarEditar
git clone <URL_DO_REPOSITORIO>

Passo 2: Criar uma branch individual

bash
CopiarEditar
git checkout -b feature/nome-aluno-pagina-produtos

Passo 3: Tarefas individuais

- **Aluno 1:** Criar a página inicial
- **Aluno 2:** Criar a página de cadastro de clientes
- **Aluno 3:** Implementar a página de login
- **Aluno 4:** Criar a página de produtos
- **Aluno 5:** Desenvolver a página de checkout
- **Aluno 6:** Criar o formulário de contato
- **Aluno 7:** Implementar o rodapé e cabeçalho

Passo 4: Subir as alterações

bash
CopiarEditar
git add .
git commit -m "Adicionada página de produtos"
git push origin feature/nome-aluno-pagina-produtos

Passo 5: Criar Pull Request

Cada aluno deve abrir um **Pull Request** e solicitar revisão.

Passo 6: Simular erro e revertê-lo

1. Após um commit incorreto:

bash
CopiarEditar
git revert <hash_do_commit>

2. Corrija o erro e adicione um novo commit corrigindo.
3. Faça push das correções.

Com essa apostila e prática, vocês estarão preparados para trabalhar com Git de forma profissional e eficiente.