

E) Empregabilidade Santander Coders 2024.1 | #1245

Isaias Soares

Id: 1245022



## Comandos Git

### Introdução à Importância dos Comandos Git para a Empregabilidade

No competitivo mercado de trabalho de desenvolvimento de software, a habilidade de gerenciar e colaborar em projetos de código é mais do que essencial; é um diferencial decisivo. O **Git**, um dos sistemas de controle de versão mais populares, está no centro dessa dinâmica. Ele permite que desenvolvedores acompanhem e gerenciem as mudanças no código ao longo do tempo, colaborando efetivamente com outros programadores. Dominar os principais comandos Git e ser capaz de manuseá-los através da linha de comando não é apenas uma habilidade técnica valiosa, mas também uma ferramenta crucial para quem está buscando o primeiro emprego na área de tecnologia.

Vamos destacar alguns pontos que conectam diretamente o domínio do Git com a empregabilidade:

- Exigência do Mercado:** Muitas empresas de tecnologia, especialmente aquelas que trabalham com desenvolvimento ágil, exigem proficiência em Git como parte de suas qualificações básicas para desenvolvedores.
- Habilidade Demonstrável:** Em entrevistas de emprego para posições de desenvolvimento, frequentemente são realizados testes práticos que incluem o uso do Git. Mostrar competência nesses comandos pode ser um fator decisivo.
- Colaboração e Trabalho em Equipe:** O Git é uma ferramenta essencial para colaboração em projetos de código, uma habilidade altamente valorizada no ambiente de trabalho. Saber como gerenciar versões de código e contribuir em projetos compartilhados é fundamental para qualquer desenvolvedor.
- Ambiente Profissional Realista:** A maioria dos ambientes de desenvolvimento profissional utiliza o Git. Conhecer os comandos da linha de comando prepara você para o fluxo de trabalho e as práticas que encontrará no dia a dia de uma empresa de tecnologia.

**5. Projetos Pessoais e Portfólio:** Saber usar o Git permite que você gerencie melhor seus projetos pessoais e contribua para projetos de código aberto. Um portfólio bem gerenciado no GitHub, por exemplo, é uma vitrine poderosa para suas habilidades técnicas aos potenciais empregadores.

Ao longo deste curso, você não apenas aprenderá os fundamentos do Git, mas também desenvolverá uma habilidade essencial que aumentará significativamente suas chances de sucesso no mercado de trabalho. Os comandos `git clone`, `git add`, `git commit`, `git status`, `git push` e `git pull` são mais do que meras ferramentas de gestão de código; eles são os alicerces sobre os quais você pode construir uma carreira próspera e dinâmica em tecnologia.

## Principais Comandos Git

---

### 1. git clone

- **Descrição:** Este comando é utilizado para clonar um repositório existente em um servidor remoto para a sua máquina local.
- **Exemplo de Uso:** `git clone https://github.com/exemplo/repositorio.git`
- **Explicação Detalhada:** Ao executar `git clone`, você deve fornecer a URL do repositório que deseja clonar. Este comando cria uma nova pasta no seu computador (com o nome do repositório, a menos que você especifique outro nome) e copia todos os arquivos e o histórico de versões para esta pasta.

### 2. git add

- **Descrição:** O comando `git add` é usado para adicionar alterações nos arquivos do seu diretório de trabalho à área de preparação (staging area).
- **Exemplo de Uso:** `git add arquivo.txt` ou `git add .` para adicionar todas as alterações.
- **Explicação Detalhada:** Quando você faz alterações em arquivos no seu repositório, essas mudanças são inicialmente "não rastreadas" pelo Git. Utilizando o `git add`, você sinaliza ao Git quais alterações deseja incluir no próximo commit. O comando `git add .` adiciona todas as alterações atuais ao staging.

### 3. git commit

- **Descrição:** Utilizado para registrar (commitar) as alterações preparadas na área de staging no repositório.
- **Exemplo de Uso:** `git commit -m "Adicionada nova funcionalidade"`

- **Explicação Detalhada:** O `git commit` captura o estado atual das alterações que você adicionou com o `git add`, criando um novo ponto na história do seu repositório. A opção `-m` permite que você adicione uma mensagem descritiva junto com o commit, o que é uma prática recomendada para manter um histórico claro das mudanças.

#### 4. git status

- **Descrição:** Exibe o estado atual do seu repositório, mostrando arquivos que foram modificados, adicionados ou que estão prontos para serem commitados.
- **Exemplo de Uso:** `git status`
- **Explicação Detalhada:** Este comando é útil para obter uma visão geral rápida do que está acontecendo no seu repositório. Ele mostra quais arquivos foram alterados, quais estão no staging e quais estão pendentes para serem adicionados ao staging.

#### 5. git push

- **Descrição:** Envia as alterações commitadas do seu repositório local para o repositório remoto.
- **Exemplo de Uso:** `git push origin main`
- **Explicação Detalhada:** Após commitar suas mudanças localmente, você pode usar `git push` para enviá-las ao repositório remoto. Este comando requer que você especifique o nome do repositório remoto (geralmente `origin`) e o branch para o qual deseja enviar as alterações (como `main` ou `master`).

#### 6. git pull

- **Descrição:** Atualiza seu repositório local com as mudanças presentes no repositório remoto.
- **Exemplo de Uso:** `git pull origin main`
- **Explicação Detalhada:** Este comando é usado quando outras pessoas fizeram alterações no repositório remoto e você quer atualizar seu repositório local com essas mudanças. Ele basicamente faz um `git fetch` seguido de um `git merge`, trazendo as alterações do repositório remoto e mesclando-as com seu trabalho local.

## Habilitando o Editor Web do GitHub

Além de dominar os comandos Git, é igualmente importante familiarizar-se com as ferramentas e recursos disponíveis no GitHub. Uma funcionalidade particularmente útil é o editor web, que permite editar arquivos diretamente no navegador. Para habilitar e

utilizar o editor web do GitHub, siga estes passos:

1. **Acesse o Repositório:** Vá até o repositório onde deseja fazer a edição.
2. **Escolha o Arquivo:** Navegue até o arquivo que você deseja editar.
3. **Edite Online:** Clique no ícone de lápis (editar) no canto superior direito do arquivo para abrir o editor web.
4. **Faça as Alterações:** Edite o arquivo conforme necessário diretamente no navegador.
5. **Commit:** Depois de fazer as alterações, role para baixo até o final da página, insira uma mensagem de commit e confirme suas alterações.

Essa funcionalidade é especialmente útil para correções rápidas, atualizações de documentos ou alterações menores que não exigem um ambiente de desenvolvimento local completo.

Próximo Tópico