**🧠 O que é o GitHub?**

O **GitHub** é uma plataforma de hospedagem de código-fonte que usa o sistema de controle de versão **Git**. Ele permite que desenvolvedores colaborem em projetos de software de forma organizada, segura e eficiente.

Além de armazenar código, o GitHub facilita o controle de versões, colaboração em equipe, rastreamento de erros (issues), revisão de código (pull requests) e integração com ferramentas de automação (CI/CD, por exemplo).

**🔧 Funcionalidades principais do GitHub**

**1. 📁 Repositórios (Repositories)**

Um **repositório** é onde um projeto é armazenado. Ele pode conter arquivos de código, documentação, imagens, scripts, etc.

Existem dois tipos de repositórios:

* **Públicos:** qualquer pessoa pode visualizar e clonar.
* **Privados:** somente pessoas autorizadas têm acesso.

Repositórios também permitem controle de versões, o que facilita acompanhar o histórico de mudanças do projeto.

**2. ✅ Commit**

Um **commit** é uma ação que salva as alterações feitas no código para o repositório local. Ele registra:

* O que foi alterado.
* Quem alterou.
* Quando foi alterado.
* Uma mensagem descritiva da mudança.

Exemplo de uso:

git commit -m "Corrige bug na tela de login"

Cada commit cria um ponto no histórico do projeto que pode ser recuperado posteriormente.

**3. 🔄 Fetch**

O comando **fetch** busca as últimas atualizações de um repositório remoto (por exemplo, no GitHub), mas **não as aplica** ao seu repositório local imediatamente.

Exemplo:

git fetch origin

É útil para ver o que mudou no projeto remoto antes de integrar com seu código local.

**4. ⬇️ Pull**

O comando **pull** faz duas coisas:

* Primeiro, executa um git fetch.
* Depois, faz um **merge** (junção) das mudanças do repositório remoto no seu repositório local.

Exemplo:

git pull origin main

Use quando quiser atualizar seu projeto local com as últimas versões do projeto no GitHub.

**5. ⬆️ Push**

O comando **push** envia seus commits locais para o repositório remoto (GitHub). Ou seja, ele **atualiza o projeto online** com as alterações feitas localmente.

Exemplo:

git push origin main

É a ação final para publicar seu trabalho no GitHub.

**6. 🌍 Público (Public)**

Ao tornar um repositório **público**, você permite que qualquer pessoa no GitHub:

* Veja seu código.
* Baixe (clone) o projeto.
* Siga as atualizações.

Isso é comum em projetos open source, onde a colaboração aberta é incentivada.

**✅ Conclusão**

O **GitHub** é uma ferramenta essencial para programadores e equipes de desenvolvimento. Ele facilita a colaboração, organização e versionamento de projetos. Entender bem suas funcionalidades — como **commit**, **push**, **pull**, entre outras — é fundamental para quem deseja trabalhar com desenvolvimento de software de forma profissional