

Git e GitHub

A by Ana Thereza Motta



O que é Git?

Git é um sistema de controle de versão criado por Linus Torvalds. O Git permite que os desenvolvedores acompanhem as mudanças feitas no código-fonte, registrem quem efetuou as alterações e restaurem versões anteriores do código.

1 Controle de Versão

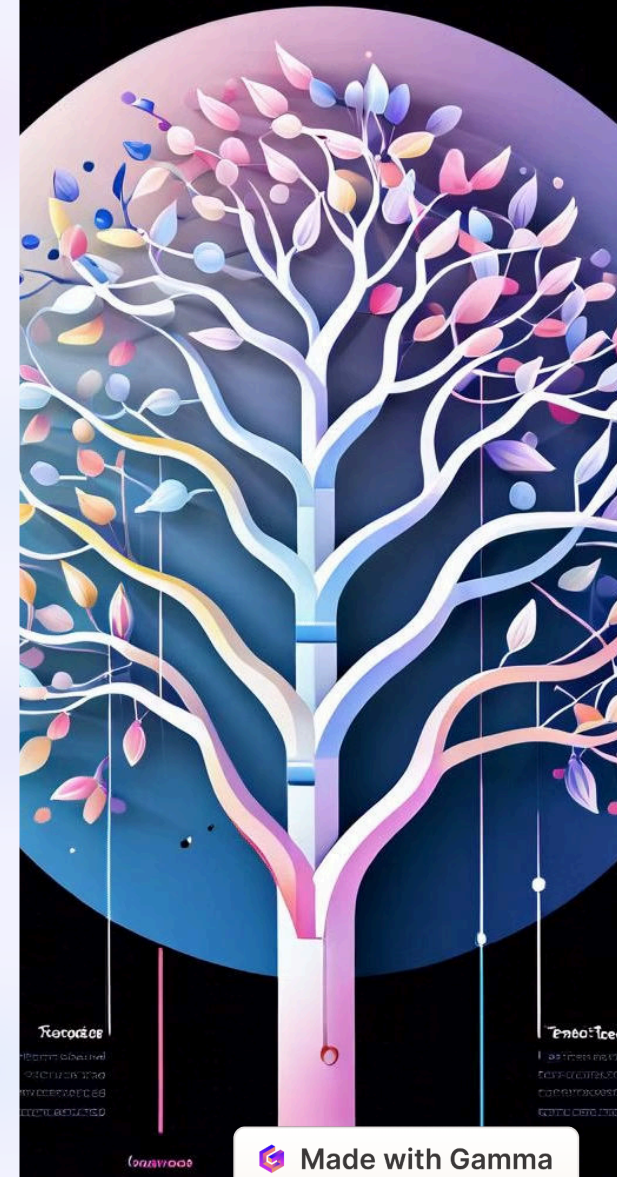
O Git permite acompanhar as mudanças no código-fonte, registrando quem efetuou as alterações.

2 Ramificações de Recursos

O Git oferece ramificações de recursos, permitindo que cada desenvolvedor trabalhe em mudanças específicas sem afetar a ramificação principal do projeto.

3 Mesclagem de Mudanças

Quando as mudanças estão prontas, elas podem ser mescladas de volta à ramificação principal.



GitHub: O que é?

GitHub é um projeto de gestão baseado em nuvem e uma plataforma de organização que incorpora os recursos de controle de versão do Git. Isso significa que todos os usuários do GitHub podem acompanhar e gerenciar as mudanças feitas para o código-fonte em tempo real, enquanto têm acesso a todos os outros recursos do Git disponíveis no mesmo lugar.

Controle de Versão

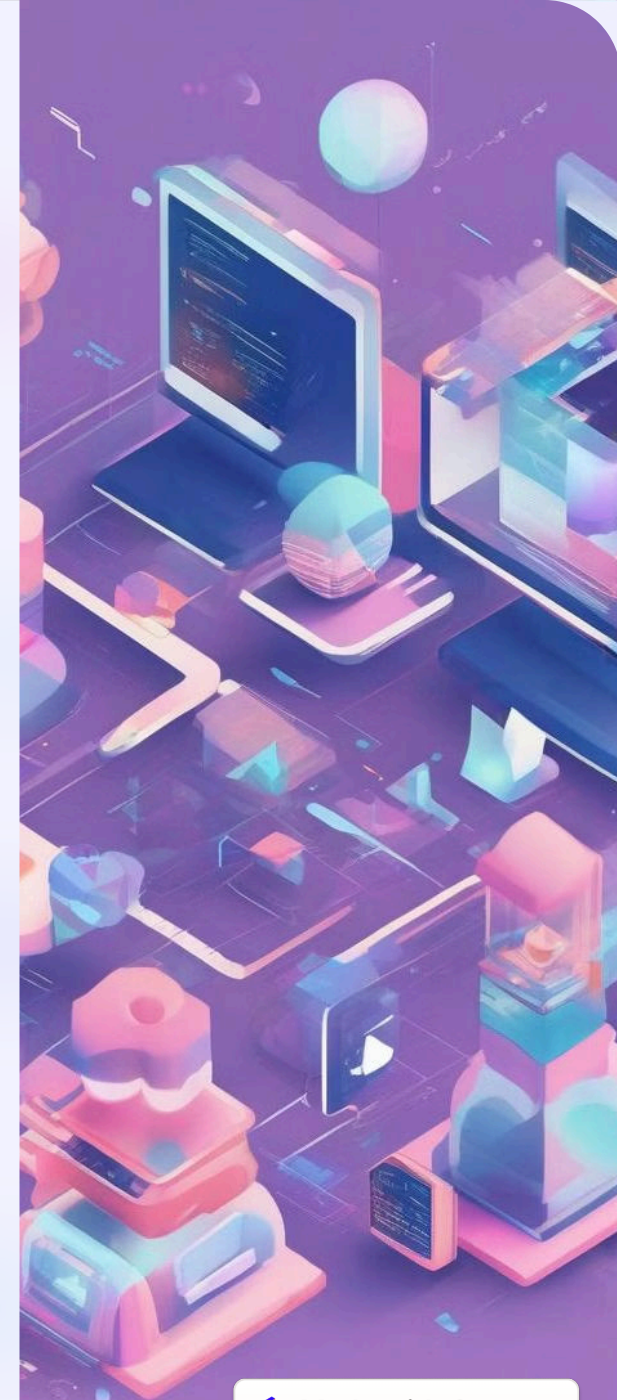
O GitHub permite acompanhar as mudanças no código-fonte em tempo real. Além de oferecer um repositório online para armazenar o seu código Git, permitindo que você acesse e gerencie seus projetos de qualquer lugar.

Colaboração

O GitHub facilita a colaboração entre desenvolvedores, permitindo que eles trabalhem juntos em projetos.

Gerenciamento de Projetos

O GitHub oferece ferramentas para gerenciar projetos, como issues, pull requests e wikis.



Importância do Git e GitHub

Atualmente, praticamente todas as empresas que atuam na área de TI utilizam o Git e GitHub para realizar o controle de versão do código. Isso permite que o desenvolvedor possa gerenciar o código-fonte e seu histórico de mudanças.

Uso Padrão

A utilização do Git se tornou um **requisito básico** para a maioria das vagas de desenvolvimento de software. Empresas buscam profissionais que dominem essa ferramenta para garantir um processo de desenvolvimento eficiente e organizado.

Comunidade

O GitHub possui uma comunidade global ativa de desenvolvedores, onde você pode encontrar ajuda, colaborar em projetos, aprender novas tecnologias e se manter atualizado com as últimas tendências do mercado.

Portifólio

O GitHub é muito utilizado como um portfólio do desenvolvedor para exibir seus projetos e conhecimentos sobre algumas ferramentas e linguagens de programação.

Configurações Iniciais do Git

Vamos realizar a configuração do seu usuário no Git. Abra o power shell, gitbash ou cmd e digite os comandos abaixo.

- 1** Nome de Usuário
`git config --global user.name "Seu nome"`
- 2** Email
`git config --global user.email "seuemail@gmail.com"`
- 3** Editor
`git config --global core.editor notepad`
- 4** Editor VS Code
`git config --global core.editor code`

Criando uma Conta no GitHub

Acesse o site do [GitHub](#) e crie uma conta. Você precisará fornecer um nome de usuário, endereço de email e senha.



Nome de Usuário

Escolha um nome de usuário único que represente você.



Email

Forneça um endereço de email válido para receber notificações e atualizações.



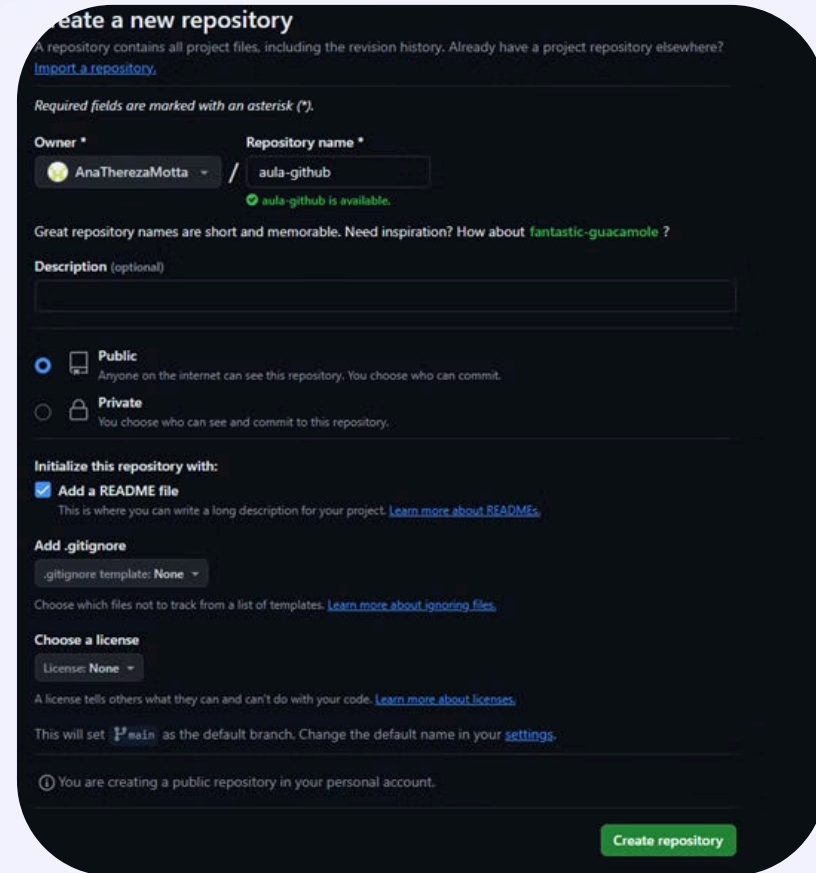
Senha

Crie uma senha forte que seja difícil de adivinhar.

Criar um Repositório no GitHub

Siga estes passos para criar um novo repositório no GitHub:

1. Acesse a [página de criação de repositório](#).
2. Dê o nome do seu repositório de "aula-github".
3. Selecione a opção "Public".
4. Selecione a opção de criar um arquivo "README".
5. Clique em "Criar repositório" para finalizar.



The screenshot shows the 'Create a new repository' page on GitHub. At the top, it says 'Create a new repository' and provides a brief explanation of what a repository is. Below this, it asks if the user already has a project repository elsewhere and provides a link to 'Import a repository'. The form then lists required fields, marked with an asterisk (*). The 'Owner' field is set to 'AnaTherezaMotta' and the 'Repository name' field is set to 'aula-github', with a green checkmark indicating 'aula-github is available'. Below the name fields, there is a suggestion for repository names: 'Great repository names are short and memorable. Need inspiration? How about fantastic-guacamole?'. The 'Description' field is optional and currently empty. The 'Visibility' section has two options: 'Public' (selected) and 'Private'. The 'Initialize this repository with' section has a checked box for 'Add a README file'. Below this, there are options for 'Add .gitignore' (set to 'None') and 'Choose a license' (set to 'None'). At the bottom, there is a note: 'You are creating a public repository in your personal account.' and a green 'Create repository' button.

Seu novo repositório está pronto para uso! Agora você pode começar a adicionar arquivos e colaborar com outras pessoas.

Clonando um Repositório Remoto

Copie o link do seu repositório remoto e clone-o no GitBash. Mude para o novo diretório local da sua máquina que contém o clone do repositório remoto.

1

Copiar Link

Copie o link do seu repositório remoto.

2

Clonar Repositório

```
git clone https://github.com/AnaTherezaMotta/aula-github.git
```

3

Mudar Diretório

```
cd aula-git-github
```

4

Verificar Estado

```
git status
```

```
on branch main  
Your branch is up to date with 'origin/main'.
```


Adicionar um arquivo no repositório local e enviar para o repositório remoto

1

Na sua pasta aula-github crie um arquivo com o nome arquivo.js no VSCode ou Notepad

2

Adicione o conteúdo abaixo no arquivo.js:

```
function media(num1,num2){  
var media = (num1 + num2) / 2;  
if(media >= 7){  
console.log("Aprovado!!!");  
}else{  
console.log("Reprovado!")  
}  
}
```

3

Verificar estado do repositório:

\$git status

```
branch main  
your branch is up to date with 'origin/main'.  
  
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
    arquivo.js  
  
nothing added to commit but untracked files present (use "git add" to track)
```

Adicionar um arquivo no repositório local e enviar para o repositório remoto

Adicionar o arquivo

Execute o comando `git add`. Adiciona todas as alterações no diretório atual e seus subdiretórios à área de preparação.

1

Verificar status

Use `git status` para verificar o estado do seu repositório.

```
branch main
your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   arquivo.js
```

Fazer o commit

Execute `git commit -m "primeiro commit do projeto"` para criar um novo commit com suas alterações.

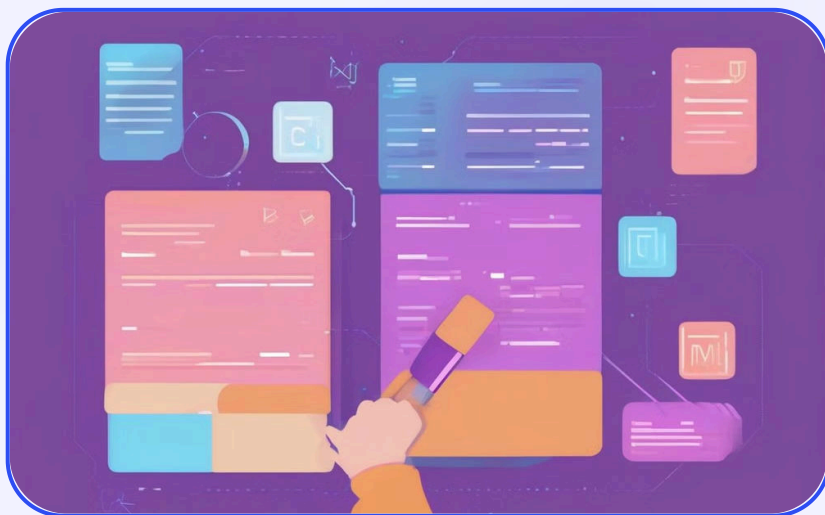
3

4

Enviar para o remoto

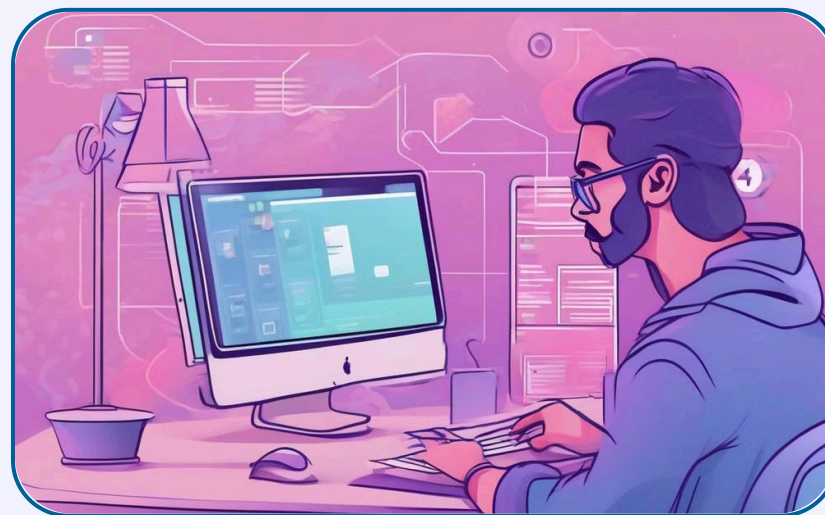
Finalmente, use `git push` para enviar suas alterações do repositório local para o repositório remoto.

Adicionando arquivos ao repositório remoto



Verifique o estado

Execute o comando `git status` para ver o estado atual do seu repositório local.



Atualize o repositório remoto

Acesse a página do seu repositório no GitHub e atualize-a para ver as mudanças.

Obrigada por sua
atenção!

