

Primeiros passos no GitHub

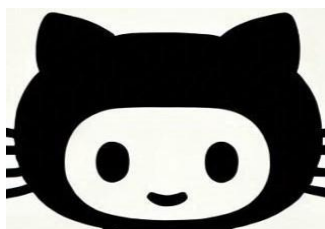
O guia definitivo para Iniciantes



SILVIA GUIMARÃES

Sumário

1. Introdução ao GitHub
2. Configurando o Ambiente
3. Navegando pelo GitHub
4. Criando Seu Primeiro Repositório
5. Realizando Commits e Pushes
6. Trabalhando com Branches
7. Trabalhando com Forks
8. Colaborando com Pull Requests
9. Resolvendo Conflitos
10. Templates no GitHub
11. Trabalhando com Issues
12. Utilizando GitHub Actions
13. Markdown Básico
14. GitHub Pages
15. GitHub CLI
16. GitHub Desktop (Interface Gráfica)
17. Readme
18. Dicas
19. Agradecimento e Nota da Autora



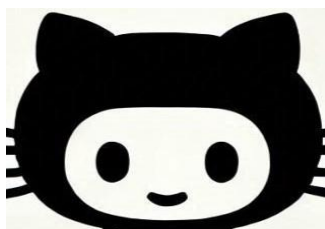
Prefácio

Bem-vindo ao fascinante universo do GitHub, onde a colaboração e o controle de versão se encontram em perfeita harmonia. Este guia foi meticulosamente preparado para você, que está prestes a dar os primeiros passos nessa jornada de descoberta e domínio de uma das ferramentas mais poderosas do mundo da programação.

Imagine o GitHub como um grande castelo, majestoso e cheio de segredos. Cada repositório é uma sala repleta de suas ideias, projetos e realizações. Este castelo digital oferece as chaves para abrir portas para um novo mundo de versionamento de códigos e colaboração online. Aqui, você aprenderá a manusear essas chaves com maestria, descobrindo caminhos e atalhos que tornarão sua jornada ainda mais interessante.

Neste guia, você será conduzido por uma trilha de aprendizado que começa com os fundamentos e avança para técnicas mais sofisticadas. Nosso objetivo é que, ao final desta jornada, você esteja não apenas preparado, mas também confiante para explorar e contribuir com projetos de qualquer tamanho e complexidade no GitHub.

Prepare-se para embarcar nesta aventura com mente aberta e espírito curioso. O castelo do GitHub está à sua espera, pronto para revelar seus segredos e lhe transformar no mestre do seu próprio código. Vamos começar!



01

INTRODUÇÃO AO GITHUB



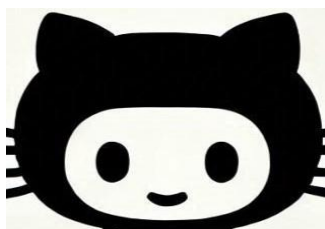
Capítulo 1: Introdução ao GitHub

O que é o GitHub?

Imagine um grande castelo onde você pode guardar e compartilhar seus projetos de programação. Cada sala desse castelo é um repositório, um espaço organizado para armazenar e proteger suas ideias e créditos de autoria. No GitHub, você também encontra ferramentas para trabalhar em equipe, registrar histórico de mudanças e aprender com outros desenvolvedores.

Por que usar o GitHub?

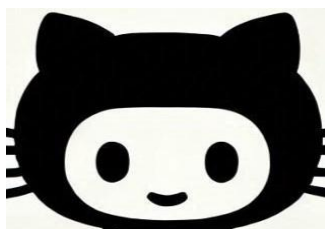
Imagine que você está construindo um quebra-cabeças com seus amigos. O GitHub é o quadro onde vocês montam as peças. Ele guarda todas as etapas do progresso, evita que alguém perca peças e garante que todos possam colaborar ao mesmo tempo.



Capítulo 1: Introdução ao GitHub

Conceitos Básicos

- **Commits:** Como salvar cada progresso do seu quebra-cabeças. Não importa o tamanho do passo, cada um é importante.
- **Branches:** Novos caminhos para explorar ideias sem estragar o que já está pronto. É como testar cores diferentes para o fundo do quebra-cabeças.
- **Pull Requests:** Quando você quer que seus amigos revisem e aceitem suas sugestões para melhorar o projeto.
- **Forks:** Cópias do projeto para você criar algo novo baseado na ideia original. Um verdadeiro "e se...?" do mundo do desenvolvimento.



02

CONFIGURANDO O AMBIENTE



Capítulo 2: Configurando o Ambiente

Instalação do Git

Para usar o GitHub, você precisa do Git, que é como um tradutor entre seu computador e o castelo do GitHub.

- **Baixe o Git:** Acesse git-scm.com e escolha a versão para seu sistema operacional.
- **Instale:** Siga os passos do instalador, aceitando as opções recomendadas.
- **Teste a instalação:** Abra o terminal e digite `git --version`.

Criando uma Conta no GitHub

Pense nisso como criar seu avatar em um jogo online.

- **Acesse github.com** e clique em *Sign Up*.
- **Escolha um nome de usuário** criativo, uma senha segura e um email.
- **Confirme seu email** e pronto! Você está pronto para explorar.

Configurando o Git

Para que o Git saiba quem você é, configure seu nome e email:

```
sh
```

```
git config --global user.name "Seu Nome"
```

```
git config --global user.email "seuemail@exemplo.com"
```



Capítulo 2: Configurando o Ambiente

Outras Formas de Usar o GitHub

- **GitHub Desktop:** Interface gráfica para quem prefere cliques em vez de comandos.
- **Extensão para VS Code:** Integre diretamente no editor e facilite o trabalho.
- **App Mobile:** Gerencie repositórios onde estiver.
- **SSH:** Um jeito seguro e sem senhas para se conectar ao GitHub.



03

NAVEGANDO PELO GITHUB



Capítulo 3: Navegando pelo GitHub

Painel de Controle

Pense no painel inicial do GitHub como um mapa do tesouro. Você vê:

- Seus repositórios.
- Notificações de atividades.
- Sugestões de projetos populares.

Explorando Repositórios

Use a barra de busca no topo da página para descobrir projetos interessantes. É como ter uma lupa mágica que mostra oportunidades e inspirações.

Personalizando Seu Perfil

Seu perfil é sua identidade. Adicione uma foto, conte sobre suas habilidades e compartilhe links para outros projetos. Um perfil completo atrai colaborações e mostra seu profissionalismo.



04

**CRIANDO SEU
PRIMEIRO
REPOSITÓRIO**



Capítulo 4: Criando Seu Primeiro Repositório

O que é um Repositório?

Pense em um repositório como uma prateleira onde você organiza os livros do seu projeto. Tudo fica arrumado, fácil de acessar e compartilhar.

Passo a Passo

- Clique no botão verde New.
- Dê um nome e descreva o projeto.
- Escolha se ele será Public ou Private.
- Adicione um README para explicar do que se trata o projeto.

Clonando um Repositório

É como copiar um livro para ler e escrever suas próprias notas:

```
sh
git clone URL_DO_REPOSITORIO
```



05

REALIZANDO COMMITS E PUSHES



Capítulo 5: Realizando Commits e Pushes

O que é um Commit?

É como registrar cada passo que você deu no projeto. Inclua sempre uma mensagem clara para saber o que mudou.

Como Fazer um Commit

- **Adicione os arquivos:**

```
sh
```

```
git add .
```

- **Salve o progresso:**

```
sh
```

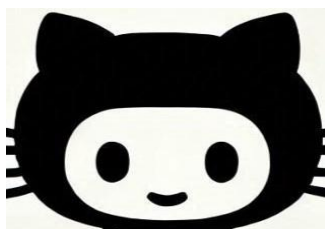
```
git commit -m "Descreva a mudança"
```

Push: Compartilhando Suas Mudanças

Para enviar as atualizações para o GitHub:

```
sh
```

```
git push origin main
```



06

TRABALHANDO COM BRANCHES



Capítulo 6: Trabalhando com Branches

O que é um Branch?

Um branch é como criar uma rota alternativa para testar novas ideias sem afetar o caminho principal. Imagine que você está escrevendo um livro e quer experimentar um final diferente sem mudar o original.

Criando e Alternando entre Branches

- Para criar um branch:

```
sh
```

```
git branch nome-do-branch
```

- Para alternar para o novo branch:

```
sh
```

```
git checkout nome-do-branch
```

Mesclando Branches

Quando a ideia estiver pronta para ser integrada ao projeto principal:

- Volte para o branch principal:

```
sh
```

```
git checkout main
```

- Mescle o branch com as alterações:

```
sh
```

```
git merge nome-do-branch
```



07

TRABALHANDO COM FORKS



Capítulo 7: Trabalhando com Forks

O que é um Fork?

Um fork é como copiar um projeto para sua biblioteca pessoal. Isso permite que você experimente e faça alterações sem impactar o original. Ideal para colaborar em projetos públicos.

Criando um Fork

- Vá até o repositório original no GitHub.
- Clique no botão Fork no canto superior direito.
- O repositório será copiado para o seu perfil.

Atualizando Seu Fork

Se o repositório original for atualizado, sincronize seu fork:

```
sh
git remote add upstream
URL_DO_REPOSITORIO_ORIGINAL
git fetch upstream
git merge upstream/main
```



08

COLABORANDO COM PULL REQUESTS



Capítulo 8: Colaborando com Pull Requests

O que é um Pull Request?

É como pedir ao dono do projeto para incluir suas alterações no trabalho principal. Um processo colaborativo para revisar e integrar ideias.

Criando um Pull Request

- No GitHub, clique em Pull Requests e depois em New Pull Request.
- Compare sua branch ou fork com a branch principal do repositório.
- Adicione uma descrição clara das mudanças e clique em Create Pull Request.



09

RESOLVENDO CONFLITOS



Capítulo 9: Resolvendo Conflitos

O que são Conflitos?

Conflitos ocorrem quando duas pessoas fazem alterações diferentes na mesma linha de um arquivo. O Git pede sua ajuda para decidir qual versão manter.

Resolvendo Conflitos

- Abra o arquivo com conflito e procure por marcas como: Plaintext

```
<<<<<< HEAD
```

```
Versão 1
```

```
=====
```

```
Versão 2
```

```
>>>>>>
```

- Escolha a versão correta e remova as marcas.
- Salve o arquivo e faça um novo commit:

```
sh
```

```
git add arquivo_resolvido
```

```
git commit -m "Conflito resolvido"
```



10

TEMPLATES NO GITHUB



Capítulo 10: Templates no GitHub

O que são Templates?

Templates são repositórios pré-configurados para iniciar novos projetos rapidamente, já com estrutura e configurações básicas.

Criando um Template

- Vá até as configurações do repositório.
- Marque a opção **Make this repository a template**.
- Salve as alterações.

Usando um Template

- Acesse o repositório template.
- Clique em **Use this template**.
- Preencha as informações do novo repositório e clique em **Create repository**.



11

TRABALHANDO COM ISSUES



Capítulo 11: Trabalhando com Issues

O que são Issues?

Issues são ferramentas para acompanhar bugs, tarefas e melhorias em um projeto. É como uma lista de afazeres colaborativa.

Criando uma Issue

- No repositório, clique em Issues.
- Clique em New Issue.
- Descreva o problema ou tarefa e clique em Submit new issue.

Gerenciando Issues

- **Atribuindo Issues:** Você pode designar issues para membros da equipe, garantindo que cada tarefa tenha um responsável.
- **Etiquetas:** Use labels para categorizar e priorizar issues. Isso ajuda a organização e facilita a visualização.
- **Milestones:** Agrupe issues em milestones para monitorar o progresso de grandes funcionalidades ou fases do projeto.



12

UTILIZANDO GITHUB ACTIONS



Capítulo 12: Utilizando GitHub Actions

O que são GitHub Actions?

GitHub Actions são ferramentas de automação que permitem criar e executar pipelines de CI/CD (Integração Contínua/Entrega Contínua) diretamente no GitHub.

Criando uma Action

- **No repositório, clique em Actions.**
- **Escolha um template ou crie um workflow personalizado.**
- **Defina os eventos que irão acionar o workflow** (por exemplo, push, pull request).
- **Adicione jobs e steps** para especificar as ações a serem realizadas.

Exemplos de Uso

- **Build e Testes:** Automatize a construção e testes do seu código.
- **Deploy:** Configure a automação do deploy para ambientes de staging e produção.
- **Notificações:** Envie alertas para canais de comunicação como Slack ou email quando um workflow for concluído.



13

MARKDOWN BÁSICO

Capítulo 13: Markdown Básico

O Markdown é uma linguagem de marcação leve e simples que você pode usar para adicionar formatação aos arquivos do GitHub. Pense no Markdown como um guia que transforma seu texto puro em documentos elegantes e formatados.

Elementos Básicos do Markdown

- **Títulos:** Para criar títulos, use o símbolo #. Quanto mais #, menor será o título.

Markdown

Título Principal

Subtítulo

Sub-subtítulo

- **Ênfase:** Use asteriscos ou underscores para itálico e negrito.

Markdown

itálico ou _itálico_

****negrito**** ou __negrito__



Capítulo 13: Markdown Básico

- **Listas:** Crie listas ordenadas e não ordenadas com números ou asteriscos.

Markdown

1. Item 1

2. Item 2

- Item A

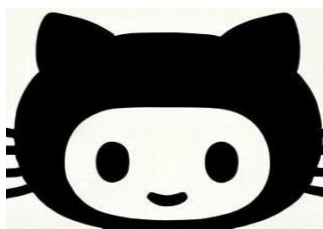
- Item B

- **Links e Imagens:** Insira links e imagens com colchetes e parênteses.

markdown

[Texto do Link](URL)

![Alt Text](URL da Imagem)



14

GITHUB PAGES



Capítulo 14: GitHub Pages

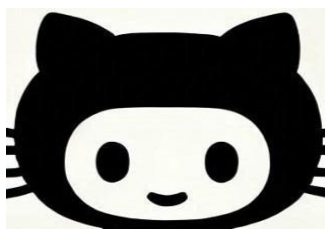
O GitHub Pages permite que você transforme seus repositórios em websites estáticos com facilidade. É como transformar seu castelo de códigos em uma exposição pública, onde todos podem ver seu trabalho.

Configurando o GitHub Pages

- **Escolha o Repositório:** Vá para o repositório que deseja publicar.
- **Acesse as Configurações:** Clique em **Settings**.
- **Habilite o GitHub Pages:** Encontre a seção de GitHub Pages e selecione a branch e a pasta que deseja publicar.
- **Escolha um Tema:** Utilize o Jekyll para escolher um tema atraente para seu site.

Customizando Seu Site

Depois de configurar, você pode editar o conteúdo diretamente no repositório, utilizando HTML, CSS e Markdown para personalizar seu site conforme desejar.



15

GITHUB CLI



Capítulo 15: GitHub CLI

A GitHub CLI é uma ferramenta que permite interagir com o GitHub diretamente pelo terminal, tornando suas operações mais ágeis e eficientes. É como ter uma varinha mágica que realiza tarefas complexas com poucos comandos.

Instalando a GitHub CLI

- **Baixe e Instale:** Visite GitHub CLI e siga as instruções para seu sistema operacional.
- **Autenticação:** Conecte a CLI ao seu GitHub com o comando:

```
sh  
gh auth login
```

Comandos Básicos da GitHub CLI

- **Criar um Repositório:** Crie um novo repositório rapidamente.

```
sh  
gh repo create meu-repo
```



Capítulo 15: GitHub CLI

- **Clonar Repositórios:** Clone qualquer repositório para seu computador.

```
sh
```

```
gh repo clone nome/repo
```

- **Gerenciar Issues e Pull Requests:** Crie, visualize e gerencie issues e pull requests sem sair do terminal.

```
sh
```

```
gh issue create
```

```
gh pr create
```

Com a GitHub CLI, você tem o poder do GitHub ao seu alcance, sem sair do terminal.



16

GITHUB DESKTOP (INTERFACE GRÁFICA)



Capítulo 16: GitHub Desktop (Interface Gráfica)

Para acessar o GitHub Desktop (interface gráfica), siga estes passos:

Instalação

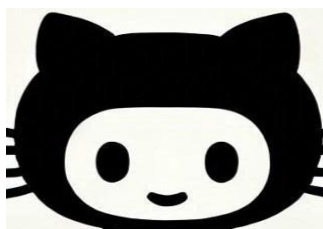
1. Acesse o site oficial: <https://desktop.github.com/>.
2. Clique em "Download" (Windows) ou "Download for macOS" (Mac).
3. Execute o instalador e siga as instruções.

Acesso

1. Após instalar, procure "GitHub Desktop" no menu Iniciar (Windows) ou Spotlight (Mac).
2. Clique para abrir.

Configuração Inicial

1. Conecte-se à sua conta GitHub.
2. Autorize o acesso.
3. Configure seu nome de usuário e e-mail.



Capítulo 16: GitHub Desktop (Interface Gráfica)

Interface Gráfica

A interface inclui:

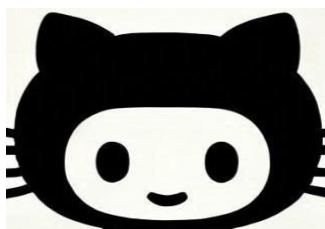
1. ***Repositories***: Gerencie repositórios locais e remotos.
2. ***Changes***: Visualize alterações nos arquivos.
3. ***Commits***: Registre mudanças.
4. ***Branches***: Gerencie ramificações.
5. ***Pull/Push***: Atualize repositórios.

Principais Ações

1. ***Criar repositório***: Clique em "Create a repository".
2. ***Clonar repositório***: Clique em "Clone a repository".
3. ***Fazer commit***: Selecione arquivos alterados e adicione mensagem.
4. ***Publicar***: Clique em "Publish" ou "Push".

Recursos Adicionais

1. Documentação oficial: <https://docs.github.com/en/desktop>
2. Tutoriais: <https://github.com/desktop/desktop/blob/main/docs/tutorials.md>



17

README



Capítulo 17: README

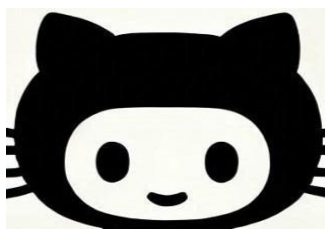
No GitHub, o arquivo README é essencial para apresentar seu projeto. Aqui está um passo a passo para criar um README:

Criar o arquivo README

1. Acesse o repositório do seu projeto no GitHub.
2. Clique em "Add file" (Adicionar arquivo).
3. Selecione "Create new file" (Criar novo arquivo).
4. Nomeie o arquivo como "README.md" (extensão Markdown).

Estrutura básica do README

1. Título do projeto
2. Descrição breve
3. Instalação/Configuração
4. Uso
5. Funcionalidades
6. Contribuição
7. Licença
8. Contato/Links



Capítulo 17: README

Formatação com Markdown

1. Títulos: # (h1), ## (h2), ### (h3)
2. Negrito: ***texto***
3. Itálico: *_texto_*
4. Listas: - item1, - item2
5. Links: url
6. Imagens: !url

Exemplo de README

...

Projeto Exemplo

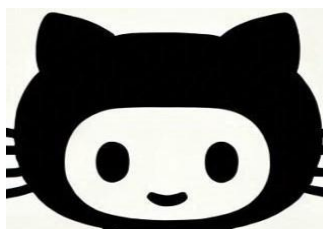
Descrição breve do projeto.

Instalação

- Clone o repositório
- Instale dependências

Uso

- Execute o comando `npm start`
- Acesse `http://localhost:3000`



Capítulo 17: README

Contribuição

- Faça um fork
- Crie uma branch
- Envie um pull request

Licença

MIT

Contato

[Seu nome](<https://seusite.com>)
...

Dicas adicionais

1. Mantenha o README atualizado.
2. Use imagens e ícones para ilustrar.
3. Inclua badges para indicar status (build, cobertura, etc.).
4. Use links para documentação adicional.
5. Seja conciso e claro.

Recursos adicionais:

GitHub Help: <https://help.github.com/en/articles/about-readmes>

Markdown Guide: <https://www.markdownguide.org/>

Exemplos de

README: <https://github.com/matiassingers/awesome-readme>



18

DICAS



Capítulo 18: Dicas

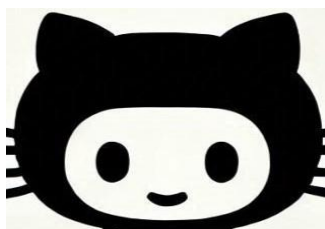
Dicas Finais

- **Pratique Regularmente:** O GitHub é uma habilidade que melhora com o uso. Comece com projetos simples.
- **Leia Documentações:** A documentação oficial do GitHub e do Git são recursos incríveis para aprofundar seu conhecimento.
- **Participe de Comunidades:** Contribuir para projetos open source ou participar de fóruns ajuda a aprender com outros desenvolvedores.

Explorando Mais

- **GitHub Projects e Issues:** Organize tarefas e colabore com equipes.
- **Markdown:** Aprenda essa linguagem simples para criar documentos profissionais no GitHub.

Agora você tem um guia completo e detalhado para começar a utilizar o GitHub. Este e-book foi criado para ser um recurso valioso, ajudando iniciantes a navegar pelo GitHub de maneira eficiente e divertida. Boa sorte em sua jornada no mundo da colaboração de código!



AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão a todos que contribuíram para a realização deste e-book. E a você, caro leitor, por dedicar seu tempo e interesse a este material. Espero que encontre valor e aprendizado em cada página. Obrigada por ler até aqui!

Este e-book é fruto de um desafio de projeto do bootcamp CAIXA - IA Generativa com Microsoft Copilot, oferecido pela DIO. Agradeço à equipe da DIO por proporcionar essa oportunidade de aprendizado e desenvolvimento, e à CAIXA pelo apoio e incentivo à inovação tecnológica.

Nota da Autora

Este e-book foi gerado por uma Inteligência Artificial e cuidadosamente diagramado por um humano. Todo o passo a passo detalhado encontra-se disponível no meu GitHub. Este conteúdo foi criado com fins didáticos de construção, e embora tenha sido revisado por um humano, ainda pode conter alguns erros gerados pela IA. Agradeço pela compreensão e espero que este guia seja uma ferramenta útil no seu aprendizado.

github.com/SMGC-GIT

