



Guia básico para Iniciantes

Aprenda quais são os principais comandos para iniciar
na plataforma GITHUB.

Bruno Rocha

MENU PRINCIPAL

Capítulo 01: O que é o GitHub e por que ele é tão importante?

Capítulo 02: Criando sua conta no GitHub e configurando seu ambiente

Capítulo 03: Criando seu primeiro repositório no GitHub

Capítulo 04: Fazendo alterações e salvando no GitHub

Capítulo 05: Ramificações e mesclagem de código

Capítulo 06: Colaborando em projetos com outros desenvolvedores

Capítulo 07: Resolvendo conflitos de mesclagem

Capítulo 08: Usando issues e pull requests

Capítulo 09: Dicas e truques para se tornar um mestre do GitHub

Agradecimentos

Capítulo 1: O que é o GitHub e por que ele é tão importante?

Capítulo 1: O que é o GitHub e por que ele é tão importante?

Imagine um baú mágico que guarda todos os seus projetos de software, com a segurança de um dragão e a praticidade de um portal interdimensional.

Esse baú mágico é o GitHub! Uma plataforma online onde você pode armazenar, compartilhar e colaborar em projetos de código com outros desenvolvedores do mundo todo. Seja você um cavaleiro solitário ou um membro de uma equipe lendária, o GitHub te ajuda a:

Guardar seus tesouros: Armazene seus projetos de software em repositórios seguros e organizados, como se fossem cofres em seu castelo particular.

Compartilhar sua sabedoria: Mostre seus projetos para o mundo e receba feedback valioso de outros desenvolvedores experientes.

Trabalhar em equipe: Una forças com seus amigos e colegas para construir projetos incríveis, como se fossem um grupo de aventureiros em busca da glória.

Aprender com os melhores: Explore os repositórios de outros desenvolvedores e aprenda com as melhores práticas e técnicas do mercado.

Construir seu portfólio: Mostre suas habilidades e conquistas para o mundo, atraindo oportunidades épicas em sua carreira.

**Capítulo 2:
Criando sua conta
no GitHub e
configurando seu
ambiente**

Capítulo 2: Criando sua conta no GitHub e configurando seu ambiente

Antes de embarcar em sua aventura, é preciso se preparar!

Crie sua conta no GitHub: Acesse <https://github.com/> e siga as instruções para criar sua conta gratuita. ✍️

Instale o Git: O Git é a ferramenta que você usará para controlar as mudanças em seus projetos. Baixe e instale o Git em seu computador a partir de <https://git-scm.com/downloads>.

Configure o Git: Diga ao Git qual o seu nome e e-mail para que ele possa identificar seus commits (salvamentos de alterações) nos projetos. Abra o terminal e digite os comandos:

```
git config --global user.name "Seu Nome"
git config --global user.email SeuEmail@dominio.com
```

Instale um editor de código: Você precisará de um editor de código para escrever e editar seus arquivos de software. Alguns dos mais populares são o Visual Studio Code e o Sublime Text.

Capítulo 3: Criando seu primeiro repositório no GitHub

Capítulo 3: Criando seu primeiro repositório no GitHub

Hora de criar seu primeiro baú do tesouro!

Acesse o GitHub: Entre em sua conta no GitHub e clique no botão verde "+ Novo".

Crie um novo repositório: Digite um nome para seu repositório (por exemplo, "meu-primeiro-projeto"), escolha se deseja que ele seja público ou privado e clique em "Criar repositório".

Clone o repositório: Abra o terminal, navegue até a pasta onde deseja armazenar seus arquivos e digite o seguinte comando, substituindo "seu-nome-usuario" pelo seu nome de usuário no GitHub e "nome-do-repositório" pelo nome do seu repositório:

```
git clone https://github.com/seu-nome-usuario/nome-do-repositorio.git
```

Comece a criar: Abra o editor de código e comece a escrever seus arquivos de software.

Capítulo 4: Fazendo alterações e salvando no GitHub

Capítulo 4: Fazendo alterações e salvando no GitHub

Seu projeto está tomando forma, mas ainda há mais!

Adicione arquivos ao Git: Para que o Git possa rastrear as alterações que você faz, use o comando `git add` seguido dos nomes dos arquivos. *Por exemplo:*

```
git add index.html style.css script.js
```

Comente suas alterações: Descreva as mudanças que você fez em cada commit usando o comando `git commit -m "Sua mensagem de commit"`. *Por exemplo:*

```
git commit -m "Adicionei a página inicial e o CSS básico"
```

Envie suas alterações para o GitHub: Utilize o comando `git push`

Capítulo 5: Ramificações e mesclagem de código

Capítulo 5: Ramificações e mesclagem de código

Imagine trabalhar em um projeto grandioso, como construir um castelo mágico. Para não afetar a estrutura principal enquanto trabalha em novas ideias, você precisa de salas separadas para cada etapa da construção.

No GitHub, as ramificações funcionam como essas salas!

Criando uma ramificação: Crie novas "salas" para suas ideias usando o comando `git branch nome-da-ramificacao`. *Por exemplo:*

```
git branch correcao-bug
```

Trocando de ramificação: Mude para a "sala" que deseja trabalhar usando o comando `git checkout nome-da-ramificacao`. *Por exemplo:*

```
git checkout correcao-bug
```

Capítulo 5: Ramificações e mesclagem de código

Trabalhando na ramificação: Faça suas alterações e commits normalmente dentro da "sala".

Mesclando as alterações: Quando estiver pronto, incorpore as mudanças da sua "sala" de volta na principal usando o comando `git merge nome-da-ramificacao`.

Lembre-se:

Ramificações permitem trabalhar em ideias sem afetar o código principal.

Use nomes descritivos para suas ramificações para facilitar a organização.

Mescle as alterações com cuidado para evitar conflitos.

Capítulo 6: Colaborando em projetos com outros desenvolvedores

Capítulo 7: Colaborando em projetos com outros desenvolvedores

O GitHub te conecta com uma comunidade de aventureiros prontos para colaborar em projetos épicos!

Tornando um repositório público: Abra seu baú do tesouro para o mundo usando a opção "Tornar público" nas configurações do repositório.

Convidando colaboradores: Adicione outros desenvolvedores à sua equipe usando o botão "Convidar colaboradores" nas configurações do repositório.

Trabalhando em equipe: Cada membro da equipe pode criar ramificações, fazer alterações e enviar suas contribuições para o projeto.

Comunicando-se: Use a seção "Issues" para discutir ideias, reportar problemas e se manter atualizado sobre o progresso do projeto.

Dicas para colaboração em equipe:

Defina papéis e responsabilidades para cada membro.

Comunique-se de forma clara e frequente.

Resolva conflitos de forma amigável e profissional.

Agradeça as contribuições de todos!

Capítulo 7: Resolvendo conflitos de mesclagem

Capítulo 7: Resolvendo conflitos de mesclagem

Em um projeto grandioso com vários aventureiros trabalhando juntos, conflitos podem surgir. Mas não se preocupe, o GitHub te ajuda a resolver esses conflitos como um verdadeiro herói!

Identificando conflitos: O GitHub te avisa quando há conflitos de mesclagem.

Comparando as alterações: Abra o editor de código e compare as versões em conflito manualmente.

Editando o código: Edite o código para resolver o conflito, garantindo que as melhores partes de cada versão sejam preservadas.

Resolvendo o conflito: Marque o conflito como resolvido no GitHub para finalizar a mesclagem.

Dicas para evitar conflitos de mesclagem:

Faça commits frequentes e com mensagens claras.

Utilize ramificações para isolar suas alterações.

Comunique-se com seus colegas sobre as mudanças que você está fazendo.

Capítulo 8: Usando issues e pull requests

Capítulo 8: Usando issues e pull requests

O GitHub te oferece ferramentas poderosas para organizar o trabalho em equipe e gerenciar as tarefas do seu projeto.

Criando issues: Utilize a seção "Issues" para registrar tarefas, bugs, ideias ou qualquer outro tópico relacionado ao projeto.

Atribuindo issues: Atribua issues a membros específicos da equipe para definir responsabilidades.

Discutindo issues: Use a seção de comentários para discutir as issues, compartilhar ideias e buscar soluções.

Criando pull requests: Quando estiver pronto para compartilhar suas alterações com a equipe, crie um pull request. Isso notificará os colaboradores sobre suas mudanças e permitirá que eles revisem e comentem o código antes de ser mesclado no projeto principal.

Dicas para usar issues e pull requests:

Seja claro e conciso ao descrever issues e pull requests.

Utilize labels e milestones para organizar as issues.

Responda a comentários e perguntas de forma.

Revise cuidadosamente o código antes de criar um pull request.

Capítulo 9: Dicas e truques para se tornar um mestre do GitHub

Capítulo 9: Dicas e truques para se tornar um mestre do GitHub

Personalize seu perfil: Mostre sua personalidade e suas habilidades no GitHub.

Contribua para projetos de código aberto: Participe de projetos da comunidade para aprimorar suas habilidades e colaborar com outros desenvolvedores.

Utilize ferramentas integradas: Explore as diversas ferramentas disponíveis no GitHub para facilitar seu trabalho, como o GitHub Pages para criar sites e o GitHub Actions para automatizar tarefas.

Mantenha-se atualizado: Fique por dentro das últimas novidades e melhores práticas do GitHub.

Compartilhe seu conhecimento: Ajude outros desenvolvedores a aprenderem o GitHub, criando tutoriais, respondendo perguntas em fóruns online e participando de eventos da comunidade.

Lembre-se:

O GitHub é uma ferramenta poderosa que pode te ajudar a alcançar seus objetivos como desenvolvedor.

Continue aprendendo e praticando para se tornar cada vez mais proficiente na plataforma.

A comunidade do GitHub está sempre disposta a ajudar. Não hesite em pedir ajuda quando precisar.

AGRADECIMENTOS

Agradecimentos

Com dedicação e persistência, você se tornará um verdadeiro mestre do GitHub!

Para saber mais :

**Aprenda comandos Git avançados: Explore comandos mais complexos do Git para realizar tarefas mais elaboradas.
Utilize integrações com outros softwares: Conecte o GitHub a outras ferramentas que você já utiliza, como IDEs e gerenciadores de tarefas.**

Crie seus próprios scripts: Automatize tarefas repetitivas usando scripts em Python ou outras linguagens de programação.

A jornada pelo mundo do GitHub é contínua. Continue explorando, aprendendo e se aprimorando para se tornar um desenvolvedor ainda mais completo e bem-sucedido !

**Livro escrito com o uso de I.A.
Utilizando template do Professor Felipe Aguiar _ DIO
Editado pelo aluno Bruno Rocha _ Bootcamp Santander 2024**