**Hello world**

O Projeto **Hello World** (Olá Mundo) é uma tradição consagrada pelo tempo na programação de computadores. É um exercício simples que você começa quando aprende algo novo. Vamos começar com o GitHub !

Você vai aprender como:

- Criar e usar um repositório

- Iniciar e gerenciar um novo Branch

- Fazer alterações em um arquivo e enviá-las para o GitHub como commits

- Abrir e Mesclar um pull request

## **O que é GitHub?**

O GitHub é uma plataforma de hospedagem de código para controle de versão e colaboração Ele permite que você e outros trabalhem juntos de qualquer lugar.

Este tutorial ensina os fundamentos do GitHub, como repositórios, branches, commits e pull requests. Você criará seu próprio repositório Hello World e aprenderá o fluxo do pull request do GitHub, uma maneira popular de criar e revisar o código.

**Nenhuma Codificação necessária**

Para concluir este tutorial, você precisa de uma conta do GitHub.com e acesso à internet. Você não precisa saber como codificar, usar a linha de comendo ou instalar o Git (O software de controle de versao no qual o GitHub está baseado).

Dica: abra este guia em uma janela separada do navegador (ou guia) para que você possa vê-lo enquanto conclui as etapas do tutorial.

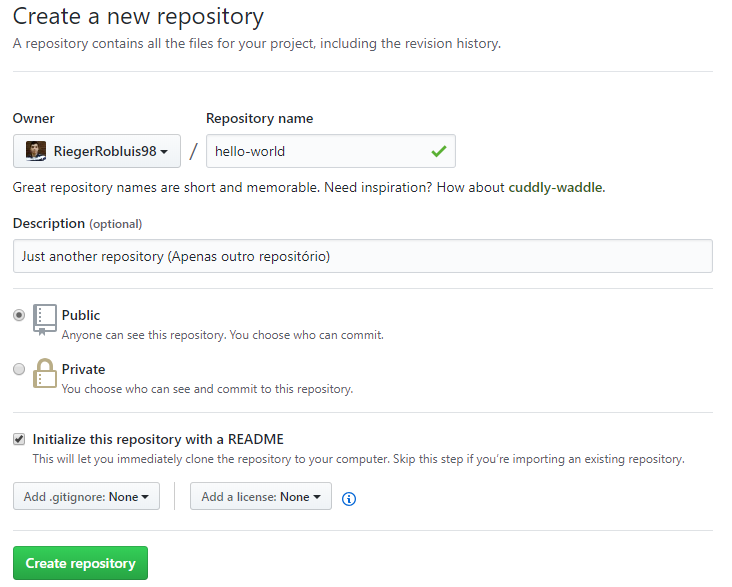
## **Passo 1. Criar um Repositório**

Um **repositório** é geralmente usado para organizar um único projeto. Repositórios podem conter pastas e arquivos, imagens, videos, planilhas e conjuntos de dados - qualquer coisa que seu projeto precisar. Recomendamos incluir um README ou um arquivo com informações sobre seu projeto. O GitHub facilita a adição de um ao mesmo tempo em que você cria seu novo repositório. *Ele também oferece outras opções comuns, como um arquivo de licença*.

Seu repositório *hello-world* pode ser um lugar onde você armazena ideias, recursos ou até mesmo compartilha e discute coisas com outras pessoas.

Para criar um novo repositório

1. No canto superior direito, ao lado do seu avatar ou identicon, cliue em + e selecione **New Repository**
2. Dê nome ao seu repositório *hello-world.*
3. Escreva uma curta descrição
4. Selecione Initialize this Repository with a README.



Clique em Create Repository

## **Passo 2. Criar um Branch (Ramificação)**

O Branch é a maneira de trabalhar em diferentes versões de um repositório de uma só vez.

Por padrão, seu repositório tem um branch chamado *master*, que é considerada a ramificação definitiva. Usamos branches para experimentar e fazer edições antes de enviá-las para o *master*.

Quando você cria um branch fora do branch *master*, está fazendo uma cópia ou instantâneo do *master* como estava naquele momento. Se alguém fizesse alterações no branch *master* enquanto você trabalhava no seu branch, você poderia receber essas atualizações.

Este diagrama mostra:

- O branch *master*

- Um novo branch chamado *feature* (porque estamos fazendo "trabalho de recurso" neste ramo)

- A jornada que esse recurso leva antes de ser mesclado ao *master*



Você já salvou versões diferentes de um arquivo? Algo como:

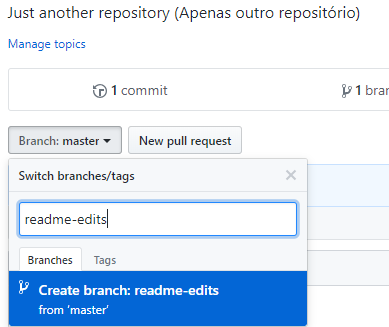
* story.txt
* story-joe-edit.txt
* story-joe-edit-reviewed.txt

Os branches atingem objetivos semelhantes nos repositórios do GitHub.

Aqui no GitHub, nossos desenvolvedores, escritores e designers usam branches para manter correções de bugs e trabalhos de recursos separados de nosso branch *master* (de produção). Quando uma mudança está pronta, eles mesclam seu branch no *master*.

**Para criar um novo branch**

1. Vá ao seu repositório hello-world.
2. Clique na lista suspensa no topo da lista de arquivos que diz branch:master
3. Digite um nome de branch, readme-edits, na caixa de texto new branch
4. Selecione a caixa azul Create branch ou dê Enter em seu teclado.



Agora você tem dois branches, master e readme-edits. Elas parecem exatamente iguais, mas não por muito tempo! Em seguida, adicionaremos nossas alterações à nova ramificação.

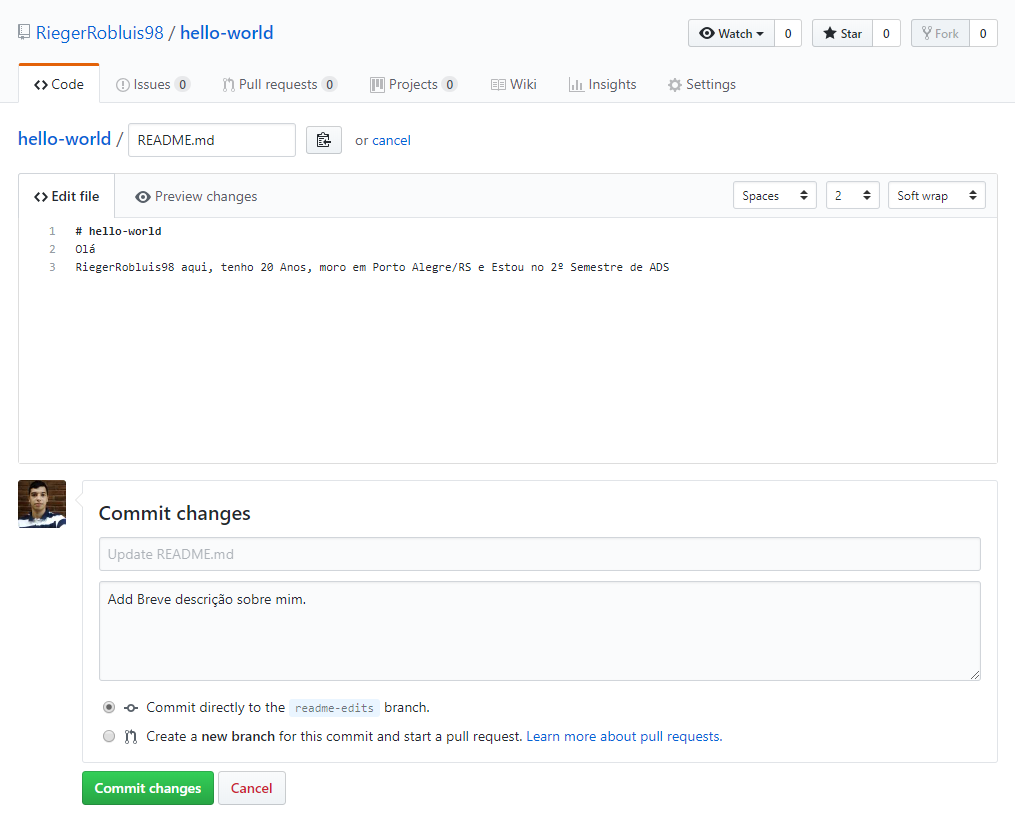
## **Passo 3. Fazer alterações**

Agora, você está na visualização de código do seu branch readme-edits, que é uma cópia do branch master. Vamos fazer algumas alterações.

No GitHub, as alterações salvas são chamadas de commits. Cada commit tem uma mensagem de commit associada, que é uma descrição explicando porque uma mudança em particular foi feita. As mensagens de commit capturam o histórico das suas alterações, para que outros colaboradores possam entender o que você fez e por quê.

**Fazer alterações**

1. Clique no arquivo README.md
2. Clique no ícone de lápis no canto superior direito da visualização de arquivo para editá-lo.
3. No editor, escreva um pouco sobre você.
4. Escreva uma mensagem de confirmação que descreva suas alterações.
5. Clique no botão Commit changes.



Essas alterações serão feitas apenas para o arquivo README no seu branch *readme-edits.* Agora, esse branch contém conteúdo diferente do *master.*

## **Passo 4. Abrir um Pull Request**

Boas alterações! Agora que você tem alterações em um Branch fora do *master*, você pode abrir um pull request.

Pull Requests são o coração da colaboração no GitHub. Quando você abre uma *pull request*, está propondo suas alterações e solicitando que alguém revise e inclua sua contribuição e as una no branch. Pull requests mostram *diffs*, ou diferenças, do conteúdo de ambos os branches. As alterações, adições e subtrações são mostradas em verde e vermelho.

Assim que você fizer um commit, você pode abrir um pull request e iniciar uma discussão, antes mesmo de o código terminar.

Ao usar o sistema @mention do GitHub na sua mensagem pull request, você pode solicitar feedback de pessoas ou equipes específicas, estejam elas no final do corredor ou a 10 fusos horários de distância.

Você pode até mesmo abrir Pull requests em seu próprio repositório e mesclá-las você mesmo. É uma ótima maneira de aprender o fluxo do GitHub antes de trabalhar em projetos maiores.

**Abra um Pull request para alterações no README**

|  |  |
| --- | --- |
| **Passo** | **Screenshot** |
| Clique na guia Pull Request e, na página Pull Request, clique no botão verde New pull request. | git_nPull |
| Na caixa Example Comparisons, selecione o Branch que você fez, readme-edits, para comparar com o master (o original). | Git_readme |
| Analise suas alterações nos *diffs* na página de comparação, verifique se elas são o que você deseja enviar. | Git_Compare |
| Quando estiver satisfeito com as mudanças que deseja enviar, clique no botão verde Create Pull Request | Git_CreatePull |
| Dê ao seu pull request um título e escreva uma breve descrição de suas alterações. | Git_desc |

Quando terminar sua mensagem, clique em Create pull request

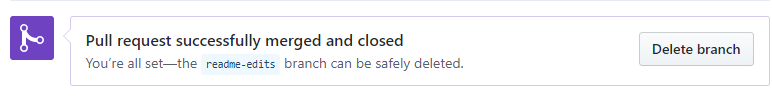
Dica: você pode usar emojis e arrastar e soltar imagens e gifs em comentários e Pull Requests.

## **Passo 5. Mesclar seu Pull Request**

Nesta etapa final, é hora de unir suas alterações - mesclando seu branch readme-edits no branch *master*.

1. Clique no botão verde Merge full request para mesclar as alterações no *master*.
2. Clique Confirm merge
3. Vá em frente e exclua o branch, já que suas alterações foram incorporadas, com o botão Delete Branch na caixa roxa.

### **Git_merge**



### **Celebre!**

Ao concluir este tutorial, você aprendeu a criar um projeto e fazer um pull request no GitHub!

Veja o que você realizou neste tutorial:

Criou um Repositório de código aberto

Iniciou e gerenciou um novo Branch

Alterou um arquivo e confirmou essas alterações no GitHub

Abriu e fundiu uma pull request

Dê uma olhada no seu perfil do GitHub e você verá seus novos quadrados de contribuição!

Para saber mais sobre o poder dos pull request, recomendamos a leitura do guia de fluxo do GitHub. Você também pode visitar o GitHub Explore e participar de um projeto de código aberto

Dica: confira nossos outros Guias, o Canal do YouTube e o Treinamento On-Demand para saber mais sobre como começar a usar o GitHub.