

# LEIA COM ATENÇÃO PARA MENOR NÚMERO DE DÚVIDAS

**Olá. Esse é o resumo que eu (Alison) fiz para vocês de como vamos trabalhar com o GITHUB.**

**Deem zoom em algumas imagens se possível, pois é um pouco difícil encaixar no esquema do WORD/PDF.**

**Essa é a MELHOR plataforma que tem para codarmos em grupo, mas, caso ainda assim vocês achem muita coisa, podemos tentar mudar para alguma extensão mais simples.**

**Nas 3 próximas páginas, irei guardar algumas dicas e comandos importantes, além de um resumo para quando vocês decorarem o processo.**

**Se precisarem de dúvidas, mande no grupo.**

**Nota: Se quiser pular direto para o resumo, precisa fazer pelo menos a Parte 0º e a 1º Parte do processo longo, que é instalar o GIT.**



Alison Christian

<https://alisonchs.github.io>



**Resumo de todo o processo (faça caso já tenha decorado, não tenha algum erro ou ache o processo longo fácil demais).**

## **1. Básico (só a primeira vez que você entrou no GIT, depois nunca mais)**

```
git config --global user.name "Fulano de Tal" // Definir o username (TEM QUE SER O MESMO DO GITHUB!!)
```

```
git config --global user.email fulanodetal@exemplo.br // Definir o e-mail
```

```
git clone github.com/repositorio.git // Clonar repositório
```

```
git remote -v // Mostra os links do repositório ORIGIN e UPSTREAM
```

```
git remote add origin https://github.com/usuario/repositorio.git // No lugar do link, coloque o link do repositório que está no SEU perfil
```

```
git remote add upstream https://github.com/AlisonChs/Projeto-Bikees.git // Comando pronto, pode digitar do jeito que está
```

## **2. Sempre que quiser modificar o código**

```
git pull upstream master // Vai baixar para as suas pastas do projeto qualquer atualização do repositório upstream/oficial que vocês não tenham ainda.
```

Ou seja, vai atualizar o seu código com o que já foi alterado por outras pessoas. (importante sempre atualizar para não bagunçar o código)

### 3. Sempre após editar algum código

```
git add . // Vai adicionar todos os arquivos que você modificou
```

```
git commit -m 'comentario' // Vai começar a atualização de arquivos para o  
sistema remoto, ou seja, o Github
```

```
git push origin main // Envia todos os arquivos modificados de uma vez só para o  
seu repositório do GITHUB.
```

**\*Lembre-se de após fazer o último comando, fazer o PULL request no SEU repositório e avisar no grupo. Se não sabe fazer o PULL Request, pule para o ÚLTIMO PASSO abaixo.\***

## Parte 0º (EXTRA): Projeto, idéias, metas e afins.

Antes de começarmos, devo apresentar esta nova aba pra vocês. Vamos deixar o **Google Drive** para trás, e usar a própria **organização** do **GITHUB** para nos organizarmos.

Seguindo o passo das fotos, vocês vão ver que tem um projeto no meu repositório, e quando vocês fizerem o **FORK/CLONE**, vocês **terão no perfil de vocês TAMBÉM!**

The screenshot shows the GitHub repository page for 'AlisonChs / Projeto-Bikees'. The 'Projects' tab is selected, and a project card titled 'Delete CNAME' is highlighted with red arrows. The card shows a commit history table:

File	Commit Message	Time
docs	Delete CNAME	5 hours ago
src	Criação de arquivos iniciais PT.1	5 hours ago
LICENSE	Initial commit	8 hours ago
README.md	Atualizando o básico do README	8 hours ago

Below the commit history, the 'README.md' content is displayed, showing a Kanban board with the following columns:

- Em progresso.
- Pronto.
- Idéias para o projeto.
- Notas

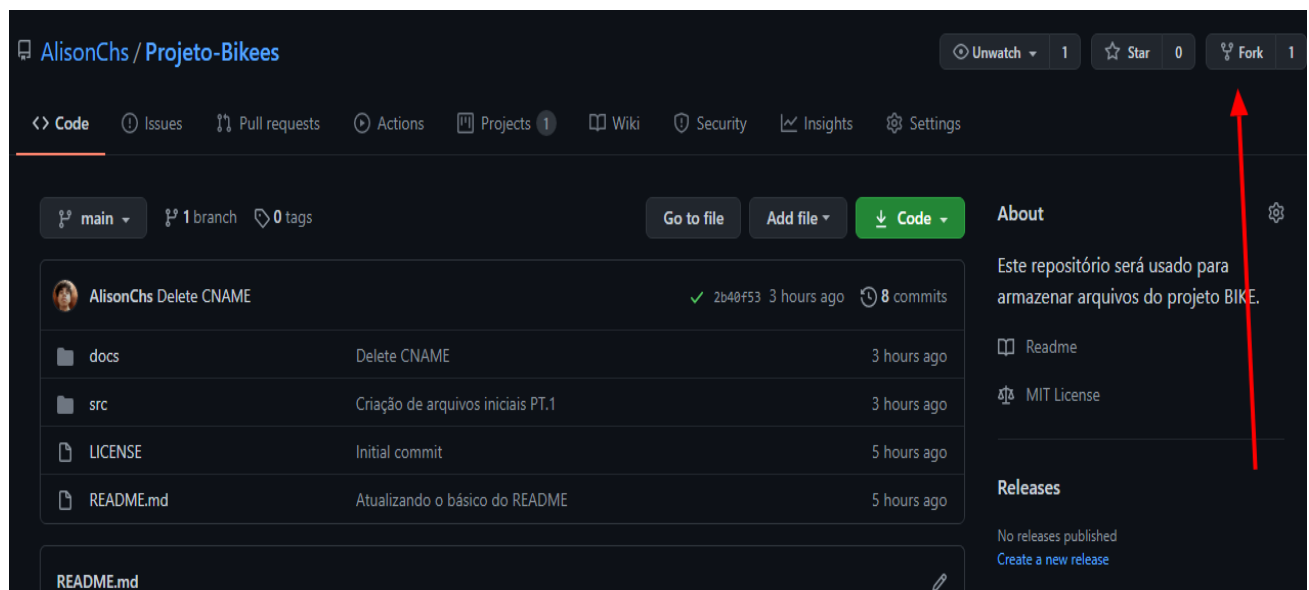
The 'Idéias para o projeto.' column contains a card titled 'Palheta de cores' with a list of color codes:

- #63E849 (Verde)
- #50FF6A (Verde claro)
- #B0FF5D (Verde/Amarelado)
- #D2E849 (Amarelo/Esverdeado)
- #FFF450 (Amarelo)

# 1º Parte: Fork no repositório.

Entrem no site do **GITHUB**, e procurem o meu usuário.

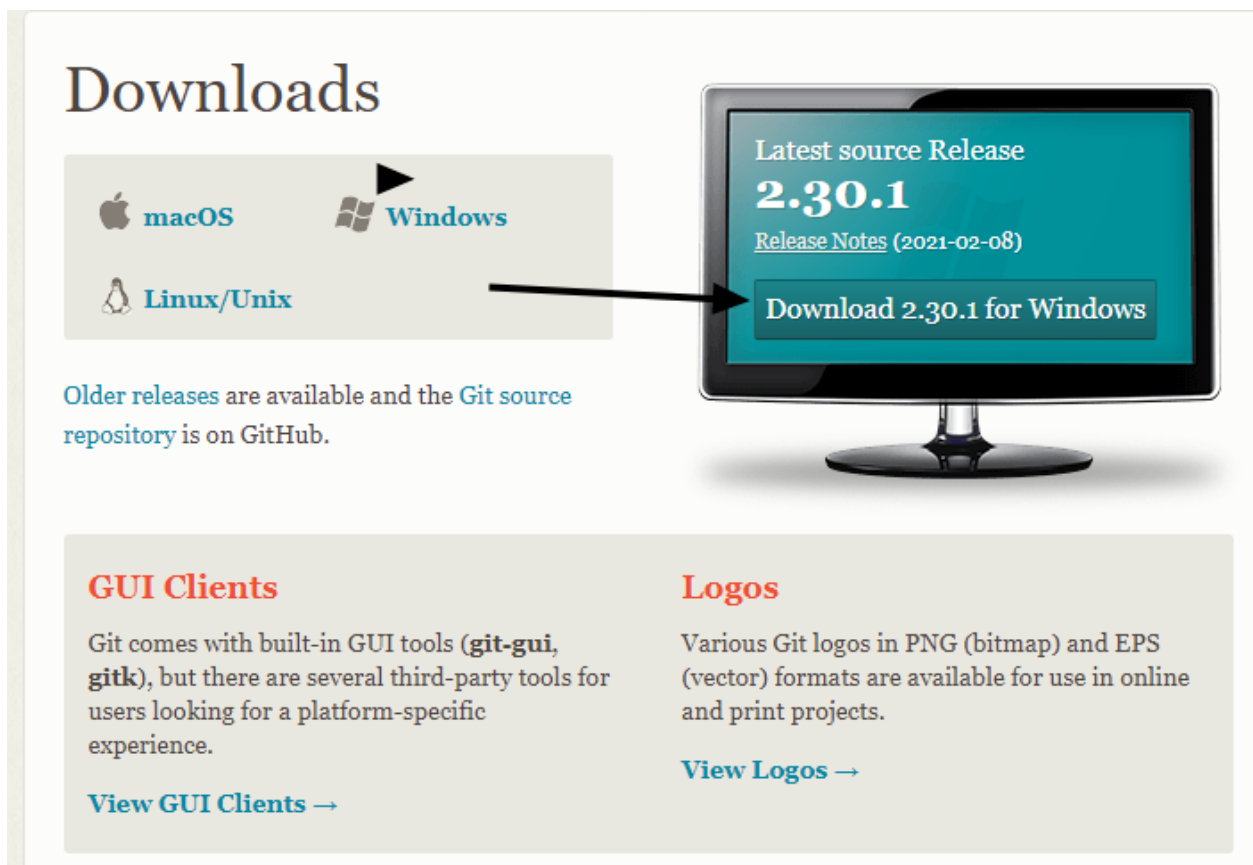
Após terem achado o meu perfil, vá para a parte de cima, e clique no botão “**Fork**”.



## 2º Parte: Instalando o GIT.

Caso vocês usem **Windows x64**, entrem no google e pesquisem “**GIT download**”.

Entrem no site “**git-scm.com**”.

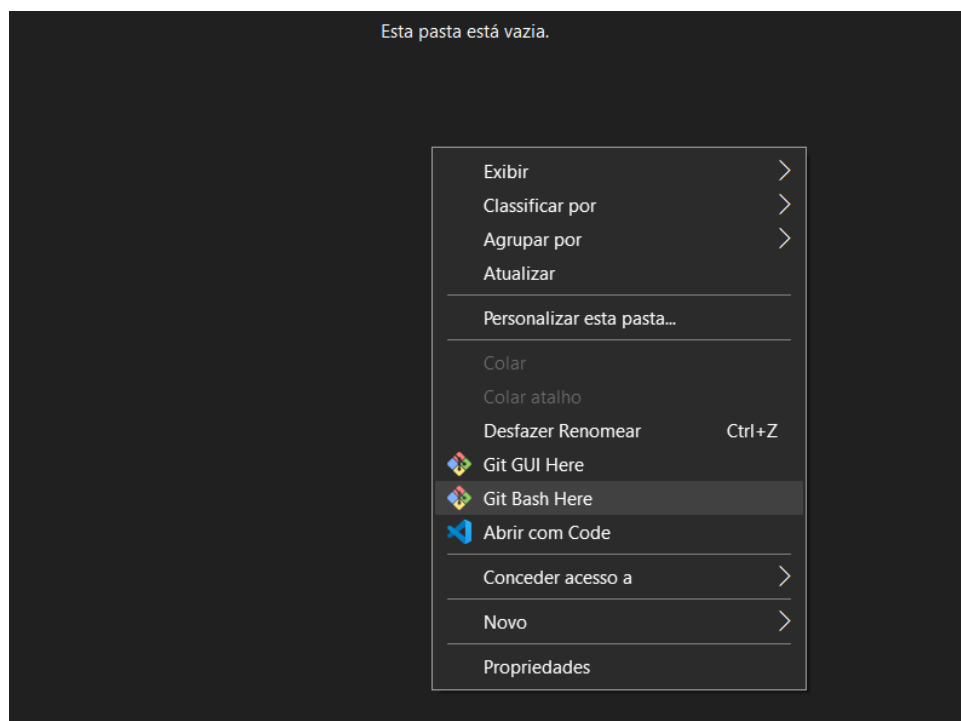


Apertem o **botão** e instalem o **GIT** conforme o padrão, apenas apertando o **NEXT**.

## 3º Parte: Abrindo o GIT e clonando o repositório

Depois de instalado, sempre que você quiser **modificar o projeto, salvar ele** no repositório do **GITHUB**, ou até baixar o projeto da **NUVEM**, aperte com o botão direito e encontre a opção **"GIT BASH"**.

**Sugestão:** Entre na/Crie uma **pasta de projetos** para utilizar deste comando.



No caso, vamos logar na sua conta do **GITHUB** pelo **GIT**. Todo esse processo só vai precisar ser feito **DUAS** vezes.

Depois de utilizar o comando, o **terminal do GIT** será aberto. Dentro ele, utilize os seguintes comandos, usando seu **nome de usuário e seu e-mail**:

```
git config --global user.name "Fulano de Tal"
```

```
git config --global user.email fulanodetal@exemplo.br
```



Se você não se deparou com nenhum **erro** até aqui, você fez o **login** com **sucesso**, e já pode prosseguir para o 4º passo.

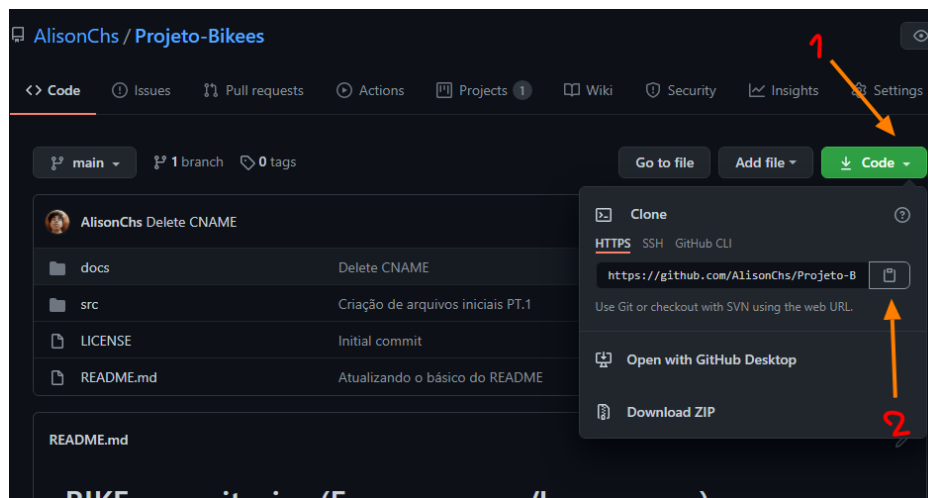
## 4º Parte: Clonando o repositório no GIT.

Conforme vocês assistiram a aula sobre **GIT** na aula do **Maurício** ano passado, o **GITHUB** é uma plataforma para armazenar seus **repositórios** e **projetos**.

O repositório está no perfil de **vocês**. A partir do momento que vocês **clonarem** ele, basicamente terão todo o código para fazerem o que quiser, desde editá-lo, a testá-lo, etc.

Depois que vocês editarem o código, terão que “**enviar**” a sua **atualização**, que é chamada de “**commit**” para o repositório, pelo **GIT**.

Para clonar o repositório, você vai entrar no seu perfil do github, e procurar o repositório do projeto. Após isso, procurará o botão “**Code**” e depois, apertará o botão ao lado para copiar o **link**, conforme na imagem.



Após copiar o link, pode voltar para o console do **GIT**.

Digite o comando (**ainda não aperte o enter!**)

```
git clone
```

Quando terminar de escrever o comando, aperte o **botão direito** e clique em “**paste**”. Depois disso, você já pode dar **enter**.

Cheque a sua pasta de projetos. O projeto vai estar lá, com o código e tudo mais.

Quando você alterar o código novamente, depois da primeira vez, deve seguir a partir do próximo passo.



## 5º Parte: Editando o Código, atualizando na máquina e no remoto.

Agora começa a parte **complicada**.

Após alterar o código, digite o comando

```
cd Projeto-Bikees
```

E use o comando **(O “.” realmente precisa!!)**

```
code .
```

Com este comando, você vai ter aberto o seu **editor de código** padrão/preferido. Edite o código, crie pastas, **seja livre**. Após terminar, abra o console do **GIT** novamente e digite o comando

```
git remote -v
```

Esse comando mostra os links do repositório **ORIGIN** (o seu repositório, o que está na sua conta do GitHub e que você clonou para a sua máquina) e do repositório **UPSTREAM** (o repositório original, que você deu “**fork**”) — ou seja, basicamente estamos dizendo para o terminal qual é o caminho do nosso repositório cópia e qual é o do repositório original.

- Para adicionar um link para o repositório upstream, use o comando  
*git remote add upstream <https://github.com/usuario/repositorio.git>*  
onde “usuario” é o nome de usuário da pessoa que criou o repositório que você deu fork e “repositorio” é o nome do repositório dela.
- Para adicionar um link para o repositório origin, use o comando  
*git remote add origin <https://github.com/usuario/repositorio.git>*  
onde “usuario” é o nome do seu usuário e “repositorio” é o nome do repositório que você deu fork.

Em seguida, digitem

```
git pull upstream main
```

com esse comando, vocês baixarão para as suas pastas do projeto qualquer atualização do repositório upstream que vocês não tenham ainda.

Após usar o comando “**git pull upstream master**”, o seu editor de código é automaticamente aberto e pede para você escrever porque o merge que estamos fazendo é necessário. Escreva um resumo sobre as suas alterações ou teste.

O código do projeto agora está atualizado apenas na **máquina** (ou seja, **localmente**), precisamos deixá-lo atualizado no **GitHub (remotamente)** também. Então ainda no **console**, digitem os comandos (um de cada vez) na seguinte ordem:

```
git add .
```

```
git commit -m 'comentario'
```

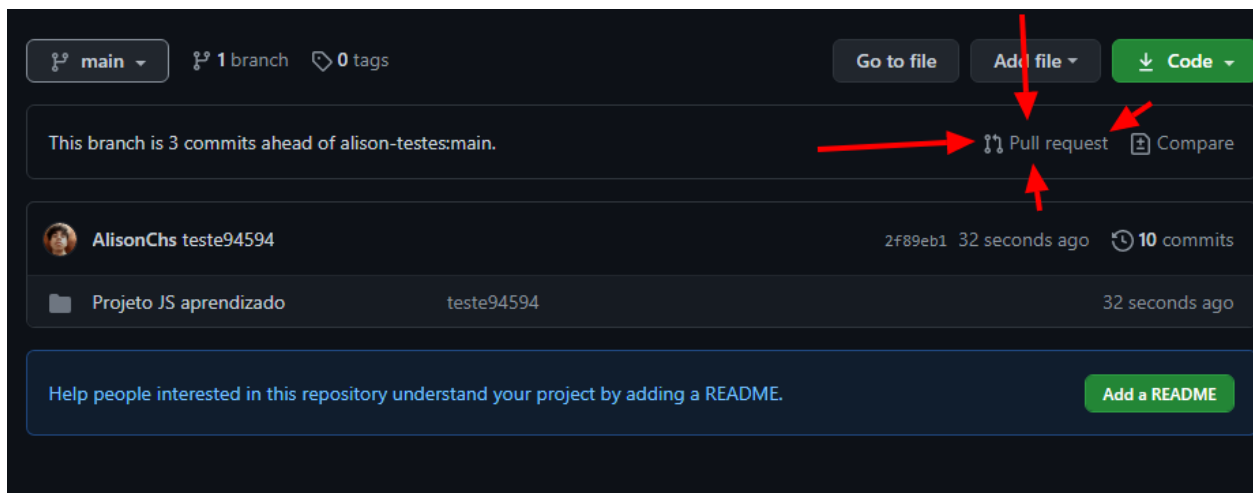
```
git push origin master
```

- Lembre-se de escreverem um **comentário relevante no commit** (por exemplo, se vocês estão subindo para o **GitHub** uma alteração que vocês fizeram em uma parte da **tag nav no HTML**, digitem algo como ‘***git commit -m ‘altera a cor de fundo da nav’***’)

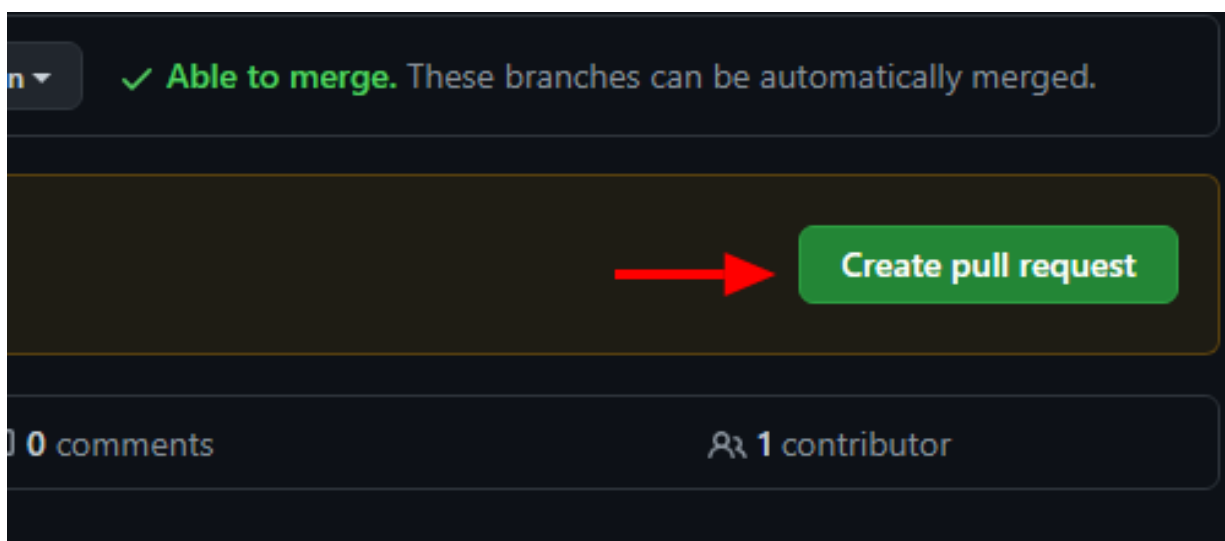
## 6º Parte: Adicionando o PULL Request para todos verem seu código.

Agora, entre no **seu perfil do github**, entre no seu projeto e crie um PULL REQUEST. Siga os passos nas imagens:

1. Aperte o botão 'Pull request'



2. Aperte o botão 'Create pull request'



3. Insira um **título**, que resume o que você alterou. **Exemplo:** “Alteração da cor do menu”

Na aba dos **comentários**, faça um breve **resumo** sobre o que você alterou, **caso você não se lembre**, pode ver suas modificações na **parte de baixo** da página, **conforme a próxima imagem**.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base repository: alison-testes/testes base: main head repository: AlisonChs/testes compare: main ✓ Able to merge. These changes can be merged into the base branch.

Title

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

☒ Allow edits by maintainers

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Showing 1 changed file with 1 addition and 0 deletions.

Projeto JS aprendizado/index.html

```
@@ -3,6 +3,7 @@
3 3 <head>
4 4
5 5 <script type="text/javascript" src="index.js"> </script>
6 + <title> testes35454 </title>
7 7
8 8 </head>
```

Depois de todos estes passos feitos, entre no nosso grupo do **Teams** e diga que terminou suas alterações. **(Não esqueça, pois preciso aprovar seu Pull Request).**

Para dúvidas, estou no Messenger, Teams, através do meu site e brevemente no WPP. Diga a sua dúvida no grupo, pois alguém pode ter a mesma e já ter resolvido e vice-versa.



Alison Christian

<https://alisonchs.github.io>