

## **ENSINO MÉDIO INTEGRADO - INFORMÁTICA**

### **Disciplina de Linguagem de Programação**

#### **Tutorial Básico - Utilizando Git / Github**

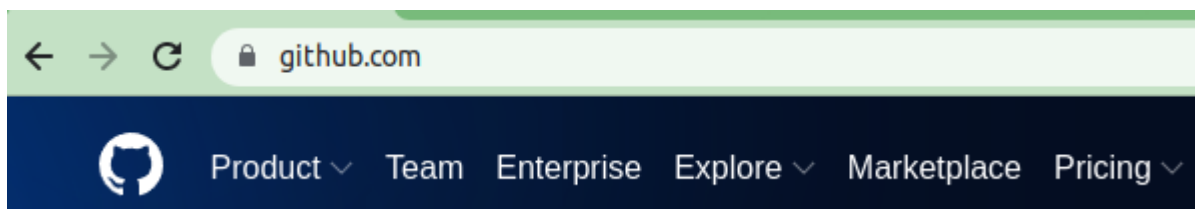
*Gil Eduardo de Andrade*

### **1. Introdução**

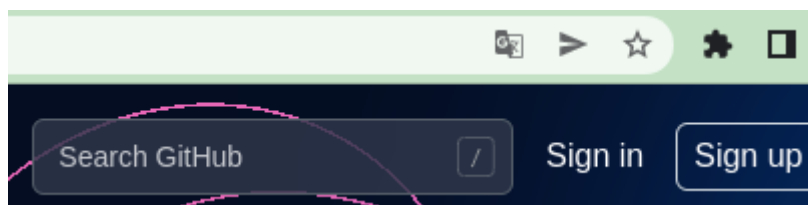
O Git é um sistema de controle de versão de software criado por Linus Torvalds, mesmo criador do sistema operacional Linux. Por outro lado, o GitHub é uma plataforma que possibilita, aos desenvolvedores, gerenciar os códigos das aplicações em desenvolvimento, ao mesmo tempo em que cria um ambiente de colaboração entre desenvolvedores. O Github utiliza o Git como sistema de controle.

### **2. Criando Minha Conta - Github (<https://github.com/>)**

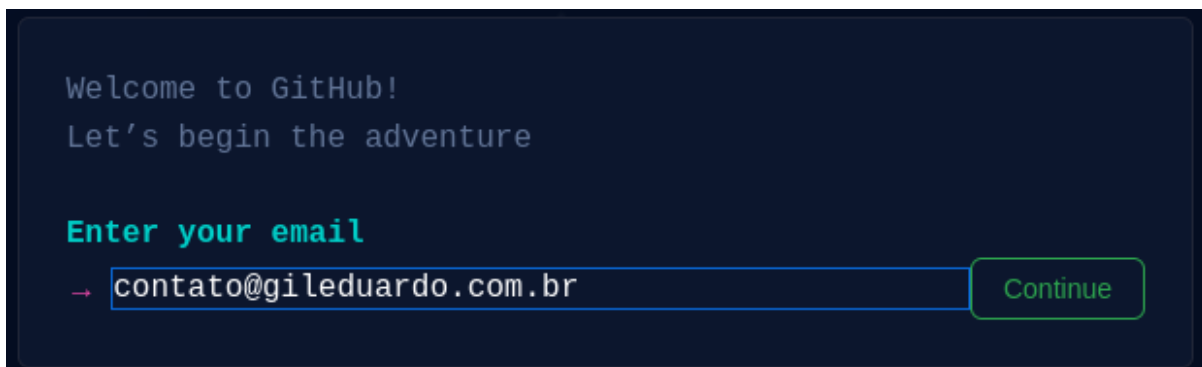
O primeiro passo para utilização do Github é a criação de uma conta na plataforma, que pode ser feita acessando o link <https://github.com/>.



Após acessar o link, escolha a opção “Sign Up” no canto superior direito da tela.



A seguir preencha o seu e-mail e clique em “Continue” (no exemplo estou utilizando o meu e-mail, contudo, você deve utilizar o seu)

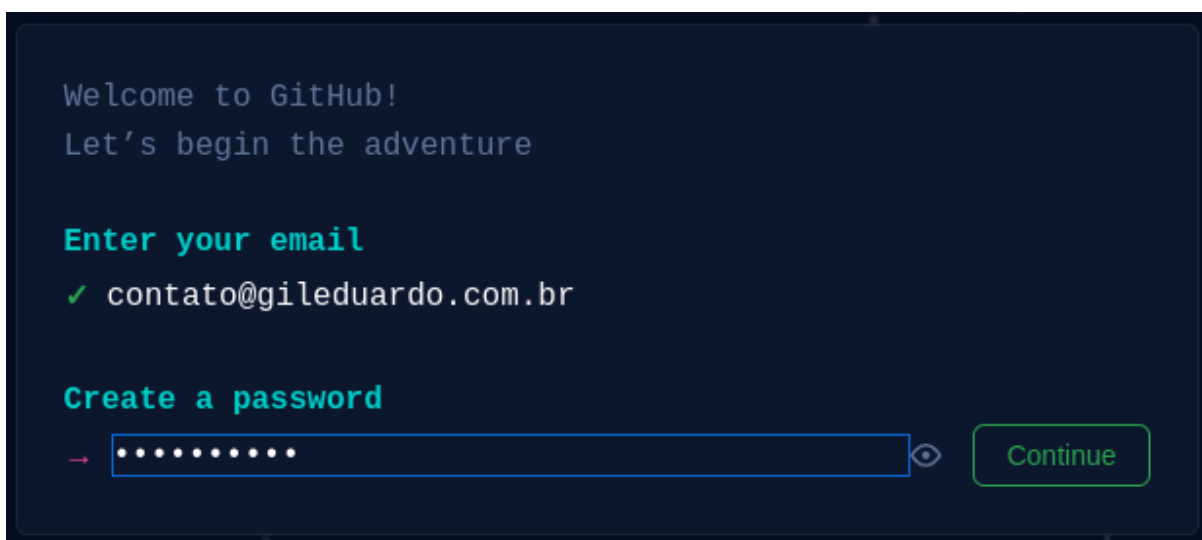


Welcome to GitHub!  
Let's begin the adventure

**Enter your email**

→  [Continue](#)

Após preencher seu e-mail e clicar em “Continue”, defina uma senha para a sua nova conta, clique em “Continue” novamente.



Welcome to GitHub!  
Let's begin the adventure

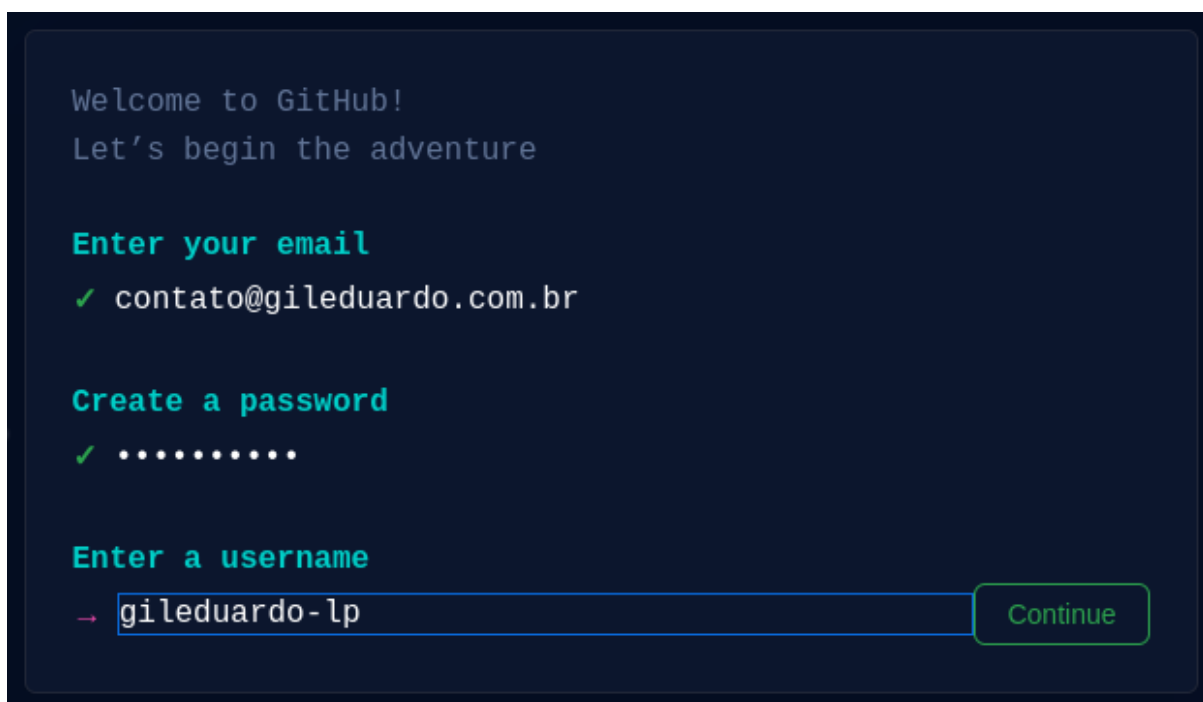
**Enter your email**

✓

**Create a password**

→  [Continue](#)

Após definir sua senha e clicar em “Continue”, escolha um nome de usuário para a sua nova conta, clique em “Continue” novamente.



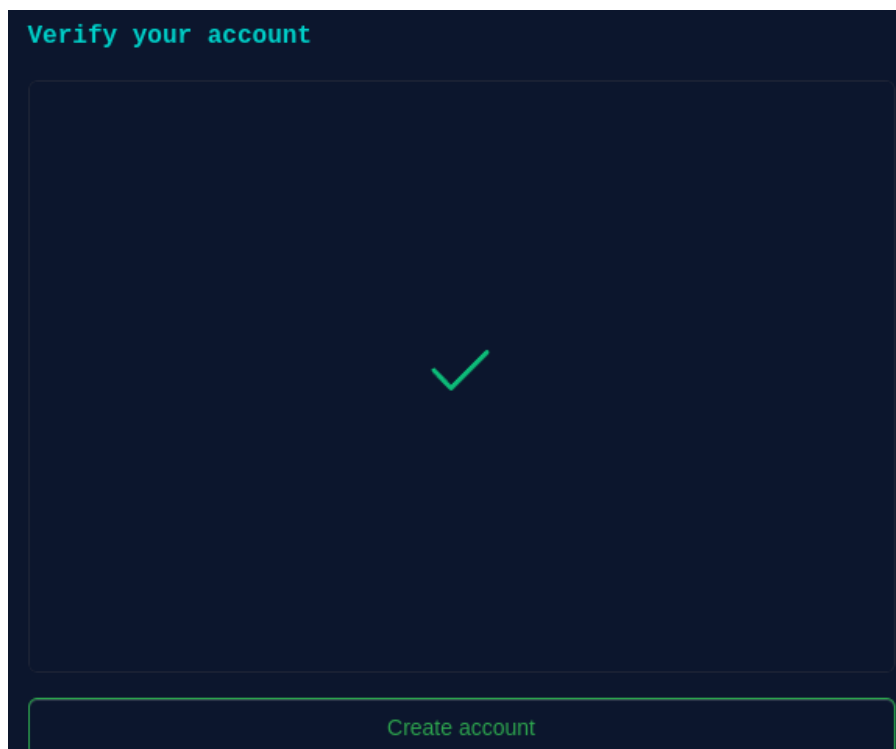
Welcome to GitHub!  
Let's begin the adventure

**Enter your email**  
✓ contato@gileduardo.com.br

**Create a password**  
✓ .....

**Enter a username**  
→ gileduardo-lp Continue

No próximo campo que aparecerá, sobre se você deseja receber notificações, via e-mail, sobre atualizações do Github, fique a vontade para optar por “n” (não) ou “s” (sim). Após isso, resolva o enigma que será mostrado e clique na opção “Create account”.

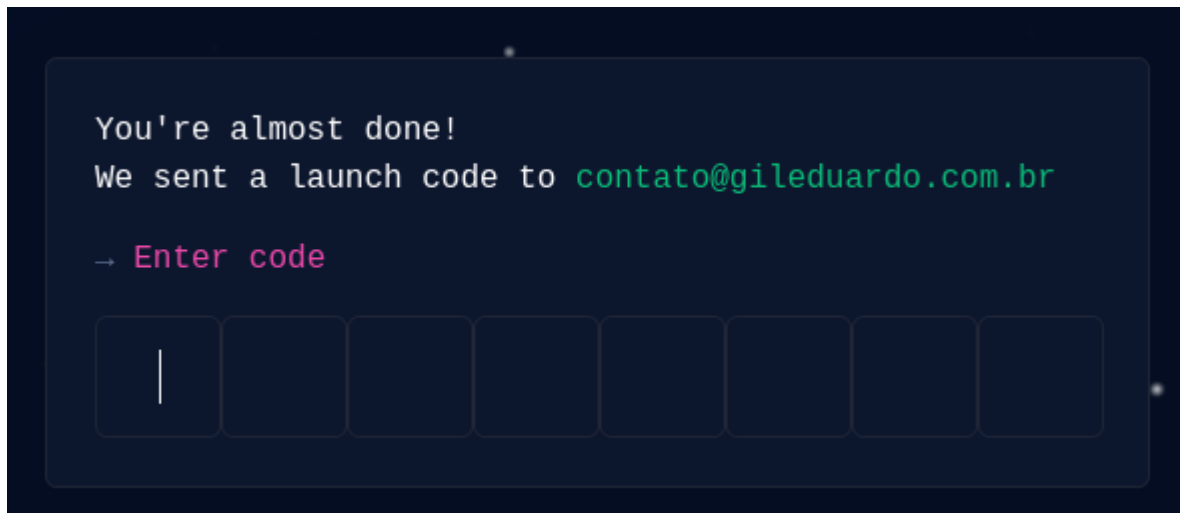


**Verify your account**

✓

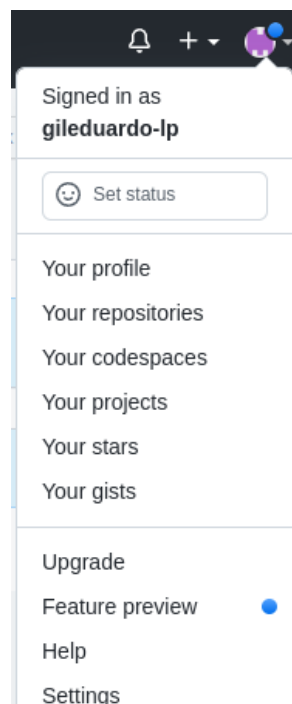
Create account

Um e-mail será enviado para conta informada durante o cadastro, nele você receberá um código, necessário para confirmação/validação da nova conta criada, digite-o como apresentado na tela a seguir



### 3. Gerando o token de acesso

No canto superior direito, clique no ícone da sua conta e selecione a opção “Settings”



Agora selecione a opção “Developer settings”



---

Archives

Security log

Sponsorship log

---

<> Developer settings

Em seguida selecione a opção “Personal access tokens” e o botão “generate new token”

GitHub Apps

OAuth Apps

**Personal access tokens**

---

## Personal access tokens

Generate new token

Na tela seguinte define uma nota (note) para o novo token que está sendo criado e um prazo de expiração para o mesmo, que pode ser “no expiration”.

### Note

Aula

What's this token for?

### Expiration \*

No expiration

The token will never expire!

Selecione também, nos campos abaixo, as permissões vinculadas ao token que está sendo criado. Como teste, pode selecionar todas as permissões. Clique no botão “Generate token” no final da página

Generate token

Cancel

Agora copie o token gerado e guarde em algum lugar seguro, ele será utilizado para podermos enviar nossos arquivos para o repositório que iremos criar na sequência.

#### Personal access tokens

[Generate new token](#)[Revoke all](#)

Tokens you have generated that can be used to access the [GitHub API](#).

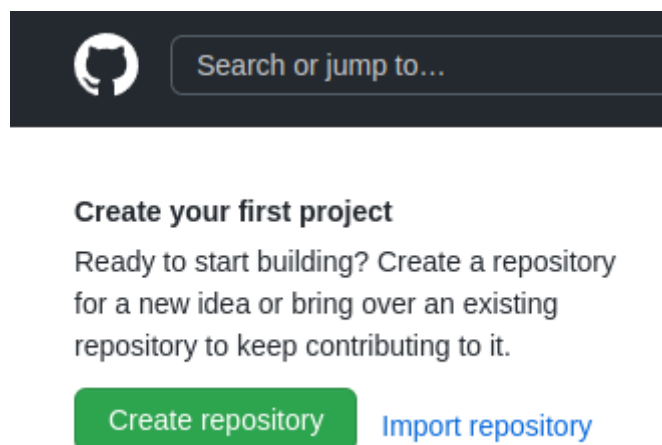
Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp\_

[Delete](#)

## 4. Criando meu Primeiro Repositório no Github

Volte para tela inicial do Github, e no canto superior esquerdo selecione a opção “Create repository”



Agora defina um nome para o seu repositório e especifique se você quer que ele seja público (public) ou privado (private). Os repositórios públicos ficam abertos para visualização de qualquer pessoa, já os privados só podem ser acessados por pessoas autorizadas para tal. Escolha o que achar melhor. Após isso, clique no botão “Create repository”, na parte inferior da página.



**Owner \*** **Repository name \***

gileduardo-lp / exemplo\_aula ✓

Great repository names are short and memorable. Need inspiration? How about [special-doodle?](#)

**Description (optional)**

Códigos em Linguagem C

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

You are creating a public repository in your personal account.

**Create repository**

### Quick setup — if you've done this kind of thing before

or **HTTPS** **SSH** `https://github.com/gileduardo-lp/exemplo_aula.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository inc

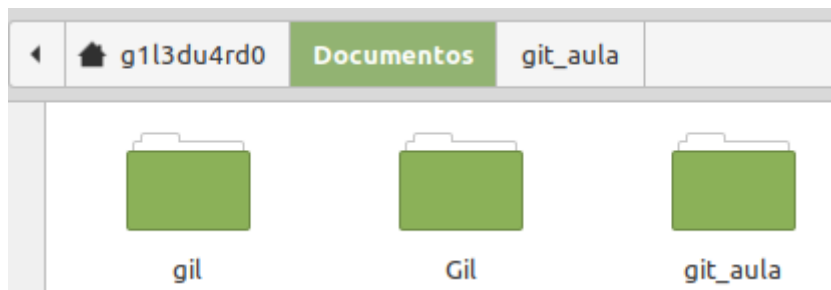
### ...or create a new repository on the command line

```
echo "# exemplo_aula" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/gileduardo-lp/exemplo_aula.git
git push -u origin main
```

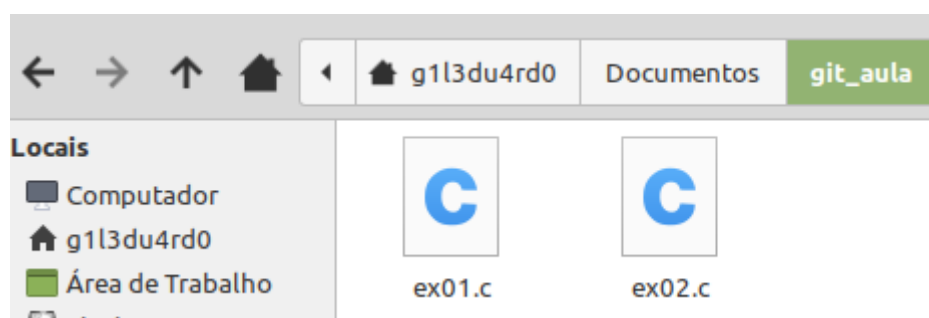
## 5. Criando um diretório na máquina local e gerando um repositório, também local, para ele

Na sua máquina local, crie um diretório com um nome qualquer, nele nós iremos colocar nossos arquivos contendo os códigos em C. Esse diretório nós iremos vincular ao repositório criado no Github, para

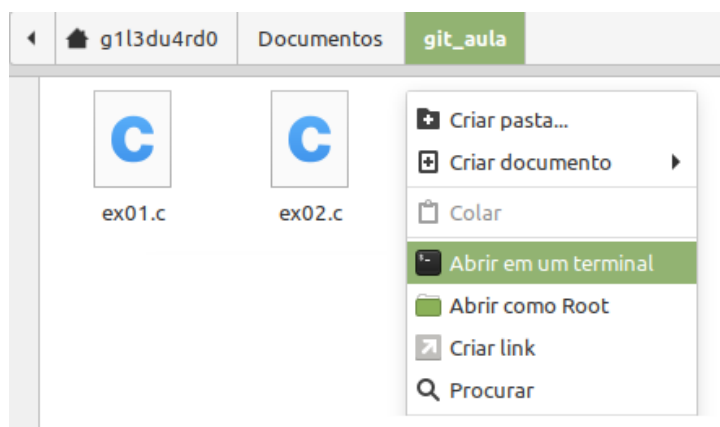
posteriormente enviar os arquivos nele contidos e versionados. Para isso, deixe a tela do navegador no Github minimizada por alguns instantes



Na minha pasta documentos eu criei a pasta “git\_aula”, e dentro dela vou criar dois arquivos “.c” para podermos efetuar alguns testes. Após a criação dos dois arquivos a pasta ficou assim:



Agora vamos abrir um terminal a partir dessa pasta, clicando o botão direito na parte branca da pasta e selecionando a opção “Abrir em um terminal”



Com o terminal aberto vamos utilizar os comandos do Git para criar um repositório (**git init**), adicionar/vincular os arquivos e modificações a ele





(**git add .**), e confirmar as ações de adição e modificação (**git commit -m "mensagem"**).

*Observe que os comandos que iremos utilizar nos próximos passos são praticamente os mesmos que foram disponibilizados na página do Github logo após criarmos o repositório (Imagem abaixo). Ou seja, você pode copiá-los sempre que criar um novo repositório remoto no Github.*

### ...or create a new repository on the command line

```
echo "# exemplo_aula" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/gileduardo-lp/exemplo_aula.git
git push -u origin main
```

### Criando repositório local

Comando: **git init**

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Docur
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$ git init
Repositório vazio Git inicializado em /home/g1l3du4rd0/Docu
mentos/git_aula/.git/
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$
```

### Adicionando os arquivos e alterações

Comando: **git add .**

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Docur
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$ git add .
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$
```



## Verificando o status das alterações

Comando: **git status**

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Docume
Arquivo  Editar  Ver  Terminal  Abas  Ajuda

g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$ git status
No ramo master

No commits yet

Mudanças a serem submetidas:
  (utilize "git rm --cached <arquivo>..." para não apresentar)
    new file:   ex01.c
    new file:   ex02.c

g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$
```

## Confirmando as alterações

Comando: **git commit -m "mensagem"**

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Documentos/git_aula
Arquivo  Editar  Ver  Terminal  Abas  Ajuda

g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$ git commit -m "Primeiro"
[master (root-commit) 7780079] Primeiro
 2 files changed, 28 insertions(+)
 create mode 100644 ex01.c
 create mode 100644 ex02.c

g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$
```

## 6. Vinculando ao repositório local ao remoto (criado no Github) e enviando os arquivos

### Vinculando com o repositório remoto

Comando (copiado do Github):

**git remote add origin https://github.com/gileduardo-lp/exemplo\_aula.git**

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Documentos/git_aula
Arquivo  Editar  Ver  Terminal  Abas  Ajuda

g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$ git remote add origin
https://github.com/gileduardo-lp/exemplo_aula.git
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$
```

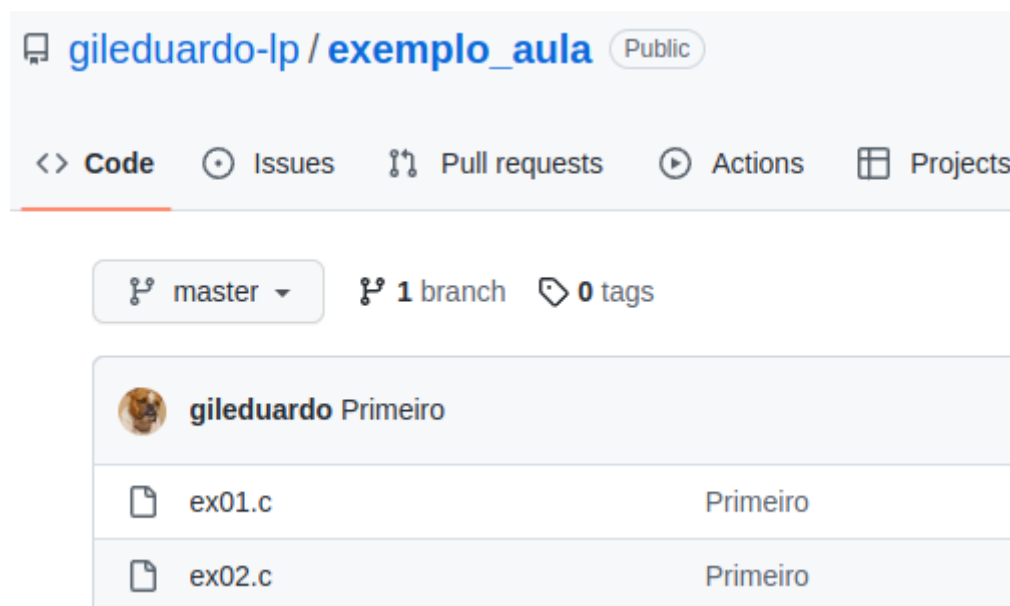
## Efetuando o primeiro envio dos arquivos

Comando: **git push -u origin master**

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Documentos/git_
Arquivo  Editar  Ver  Terminal  Abas  Ajuda

g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$ git push -u origin
master
Username for 'https://github.com': gileduardo-lp
Password for 'https://gileduardo-lp@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 488 bytes | 488.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/gileduardo-lp/exemplo_aula.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$
```

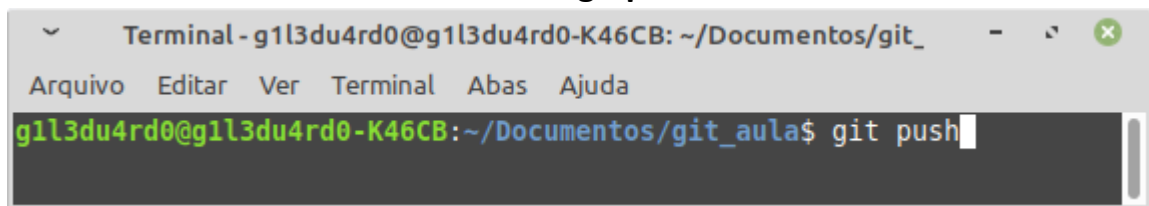
Seu usuário e seu token serão solicitados, digite o seu usuário e pressione ENTER, seu token será solicitado em seguida, copie e cole no terminal (ele não aparecerá escrito), pressionando ENTER novamente. Após isso os arquivos serão enviados para o repositório remoto, e você pode conferir na página do Github se eles realmente estão lá, veja a imagem a seguir.



## Efetuando os próximos envios de arquivos

Para os próximos envios de arquivos o comando push pode ser utilizado de modo simplificado, como você pode ver a seguir:

Comando: **git push**

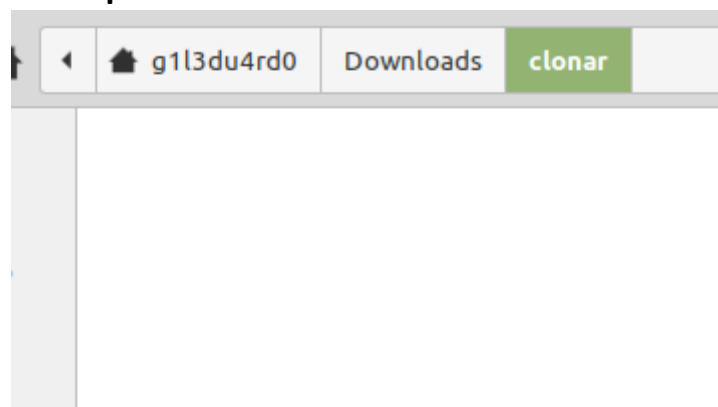


```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Documentos/git_  
Arquivo  Editar  Ver  Terminal  Abas  Ajuda  
g1l3du4rd0@g1l3du4rd0-K46CB:~/Documentos/git_aula$ git push
```

## 7. Clonando meu repositório e atualizando posteriormente

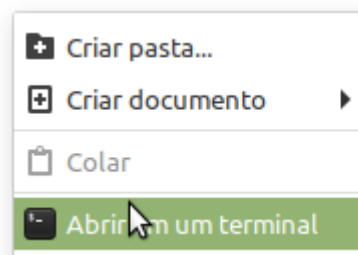
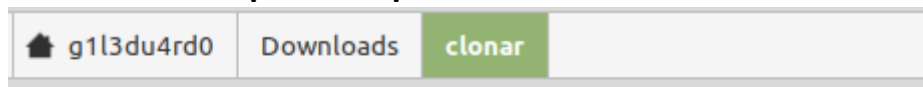
Primeiramente vamos criar um novo diretório na máquina local, para clonar o nosso repositório remoto (Github), e nele colocar os arquivos vindos do servidor. Lembrando que estamos fazendo essa ação considerando que o aluno vai continuar resolvendo as próximas atividades da disciplina e pretende adicionar esses novos arquivos e diretórios no seu repositório remoto.

### Criando uma nova pasta com nome “clonar”

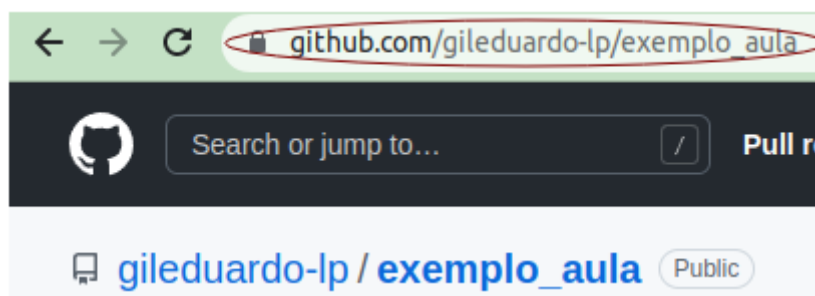


Em seguida, vamos abrir o terminal a partir da pasta criada, utilizando o botão direito do mouse no espaço em branco dentro da pasta. Como mostrado a seguir.

### Abrindo o terminal a partir da pasta “clonar”



Com o terminal aberto utilizamos o comando **git clone link\_repositorio** para poder clonar o nosso repositório para dentro da nossa pasta local. O link a ser utilizado é o link que você visualiza ao acessar o seu repositório via navegador na página do Github (você pode copiar e colar no terminal), como pode ver na imagem a seguir.



Executando o comando no terminal, utilizando o link do repositório.

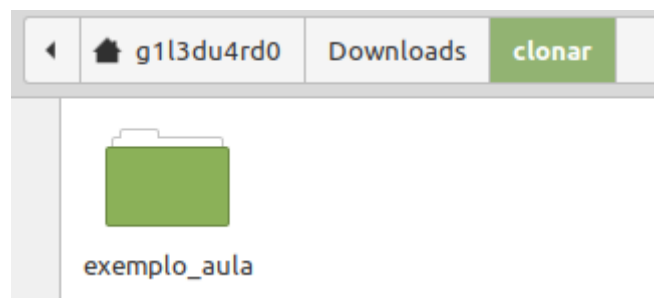
Comando: **git clone https://github.com/gileduardo-lp/exemplo\_aula**



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Downloads/clo
Arquivo  Editar  Ver  Terminal  Abas  Ajuda

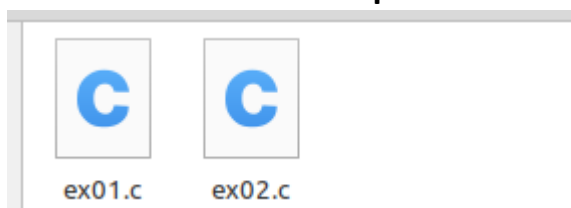
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar$ git clone https://g
ithub.com/gileduardo-lp/exemplo_aula
Cloning into 'exemplo_aula'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 468 bytes | 117.00 KiB/s, done.
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar$
```

Após a clonagem, uma pasta com o mesmo nome do repositório é criada, como pode ser visto a seguir.

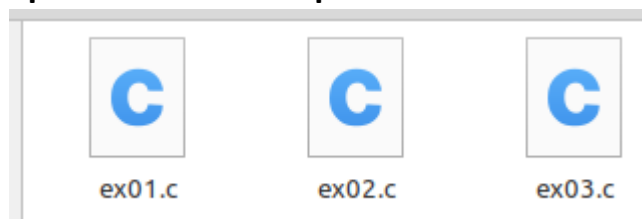


Agora dentro e da pasta “exemplo\_aula” vamos adicionar um novo arquivo, chamado “ex03.c”.

#### Antes de adicionar o arquivo



#### Após adicionar o arquivo





Agora vamos salvar as modificações que fizemos no repositório clonado (na máquina local) e vamos enviar para o Github, nosso repositório remoto.

### Criando repositório local

Comando: ***git init***

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Dow
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar$ git init
Repositório vazio Git inicializado em /home/g1l3du4rd0/Downloads/clonar/.git/
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar$
```

### Adicionando os arquivos e alterações

Comando: ***git add .***

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Downloads/clonar/
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar/exemplo_aula$ git add .
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar/exemplo_aula$
```

### Verificando o status das alterações

Comando: ***git status***

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Downloads/clonar/ex
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar/exemplo_aula$ git status
No ramo master
Your branch is up to date with 'origin/master'.

Mudanças a serem submetidas:
  (use "git restore --staged <file>..." to unstage)
    new file:   ex03.c
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar/exemplo_aula$
```

### Confirmando as alterações

Comando: ***git commit -m "mensagem"***









```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Downloads/clonar/ex
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar/exemplo_aula$ git commit
-m "novo arquivo apos clonar"
[master b802f7f] novo arquivo apos clonar
1 file changed, 14 insertions(+)
create mode 100644 ex03.c
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar/exemplo_aula$
```

## Efetando novo envio dos arquivos

Comando: **git push**

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: ~/Downloads/clonar/ex
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar/exemplo_aula$ git push
Username for 'https://github.com': gileduardo
Password for 'https://gileduardo@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 420 bytes | 420.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/gileduardo-lp/exemplo_aula
7780079..b802f7f master -> master
g1l3du4rd0@g1l3du4rd0-K46CB:~/Downloads/clonar/exemplo_aula$
```

## Verificando o repositório do Github (se o novo arquivo “ex03.c” está lá)

	gileduardo novo arquivo apos clonar	
	ex01.c	Primeiro
	ex02.c	Primeiro
	ex03.c	novo arquivo apos clonar