



# IFRN

## PROGRAMAÇÃO BÁSICA EM C#

---

Guia de Uso do GitHub

Prof. Gilbert Azevedo

# Objetivos

- Conhecer o GitHub
- Utilizar o GitHub com o repl.it

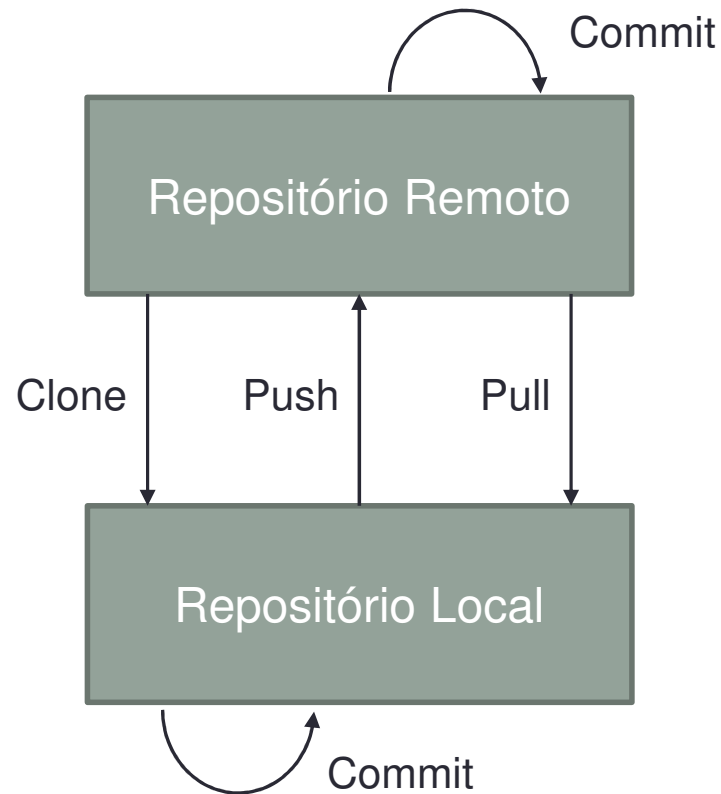
# GitHub

- O GitHub é uma plataforma de hospedagem utilizada para controle de versão e desenvolvimento colaborativo
  - Repositório de documentos e código-fonte
  - Controle de versão
  - Registro de projetos e tarefas
  - Fórum de discussão
  - Relatórios de atividades

# GitHub – Repositório Pessoal

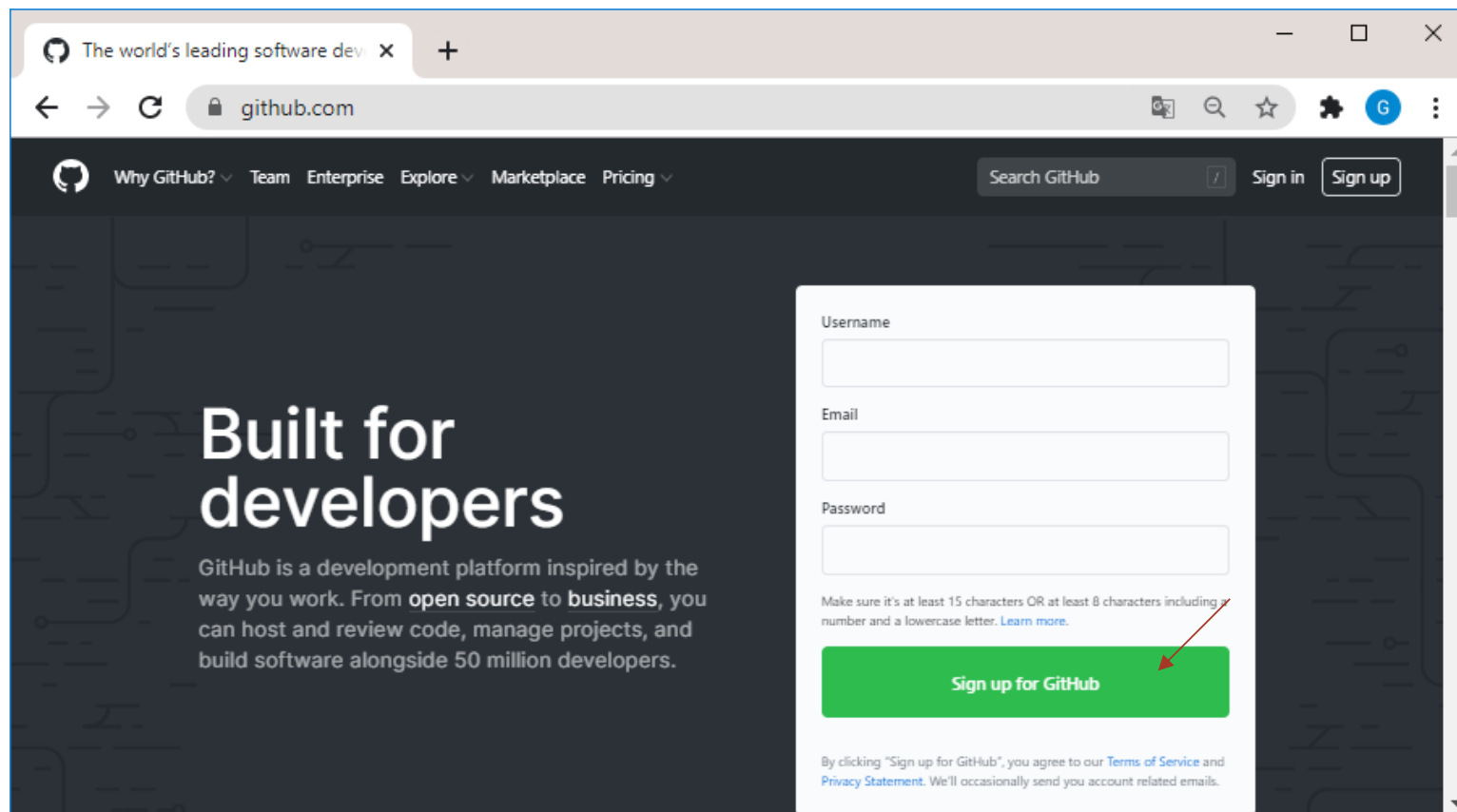
- Passos para armazenamento de projetos no GitHub integrado ao repl.it
  - Registro, login, criação do repositório (remoto) no site do GitHub
  - Criação de pastas e arquivos no GitHub
  - Clonagem do repositório no repl.it
  - Configuração do repl.it para execução de programas copiados do repositório
  - Criação de novos programas no repositório local com o repl.it
  - Confirmação de alterações no repositório local
  - Sincronização das alterações com o repositório (remoto)

# GitHub – Repositório Pessoal



# Registro no GitHub

- <https://github.com/>



The image shows a web browser window displaying the GitHub homepage. The browser's address bar shows 'github.com'. The page features a dark background with the text 'Built for developers' and a description of GitHub as a development platform. A white sign-up modal is overlaid on the right side of the page. The modal contains input fields for 'Username', 'Email', and 'Password'. Below the password field, there is a note about password requirements: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)'. A green button labeled 'Sign up for GitHub' is at the bottom of the modal, with a red arrow pointing to it. At the very bottom of the modal, there is a small disclaimer: 'By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.'

Username

Email

Password

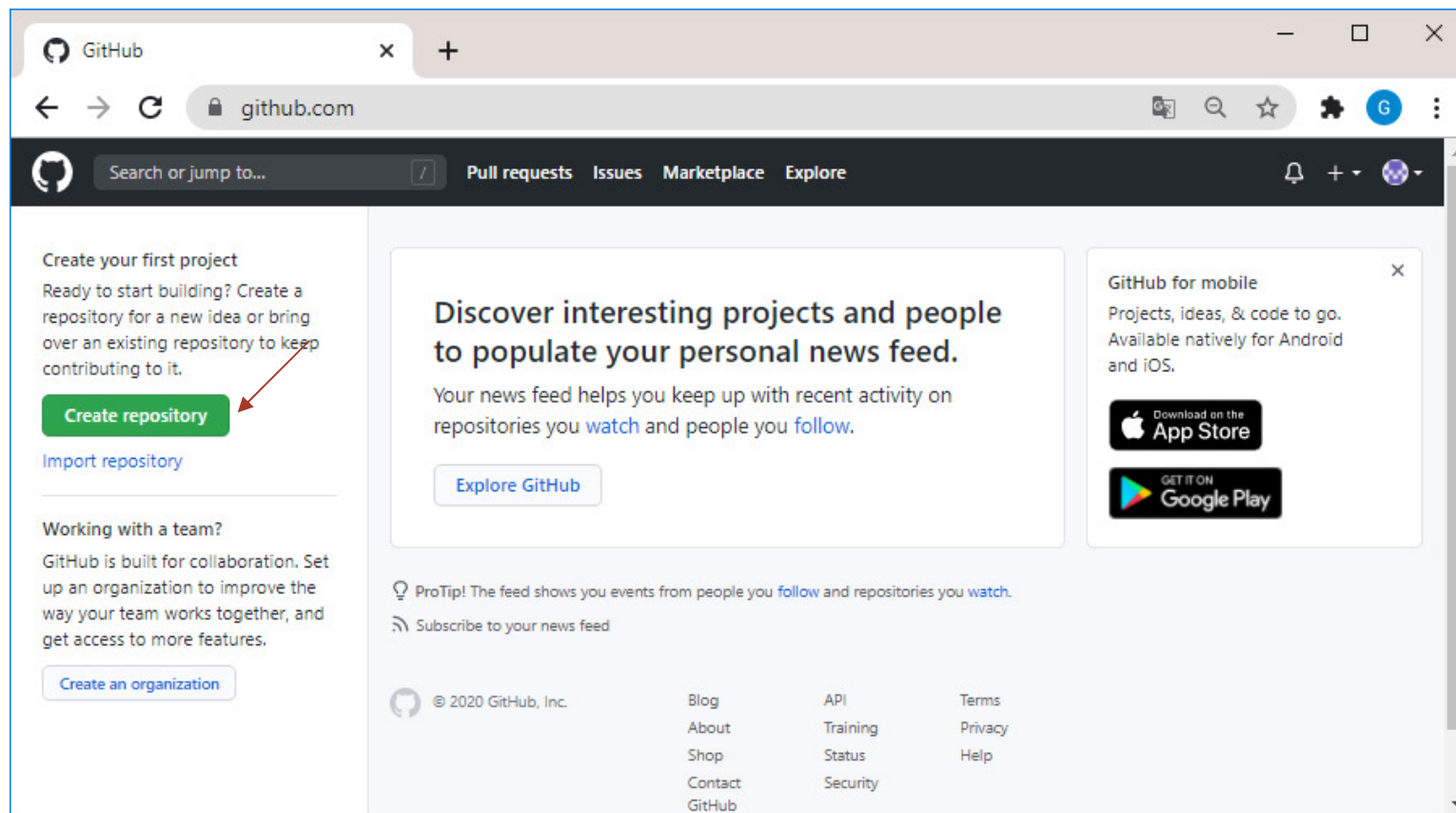
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

# Novo Repositório

- Crie um novo repositório no GitHub, selecionado *Create repository*



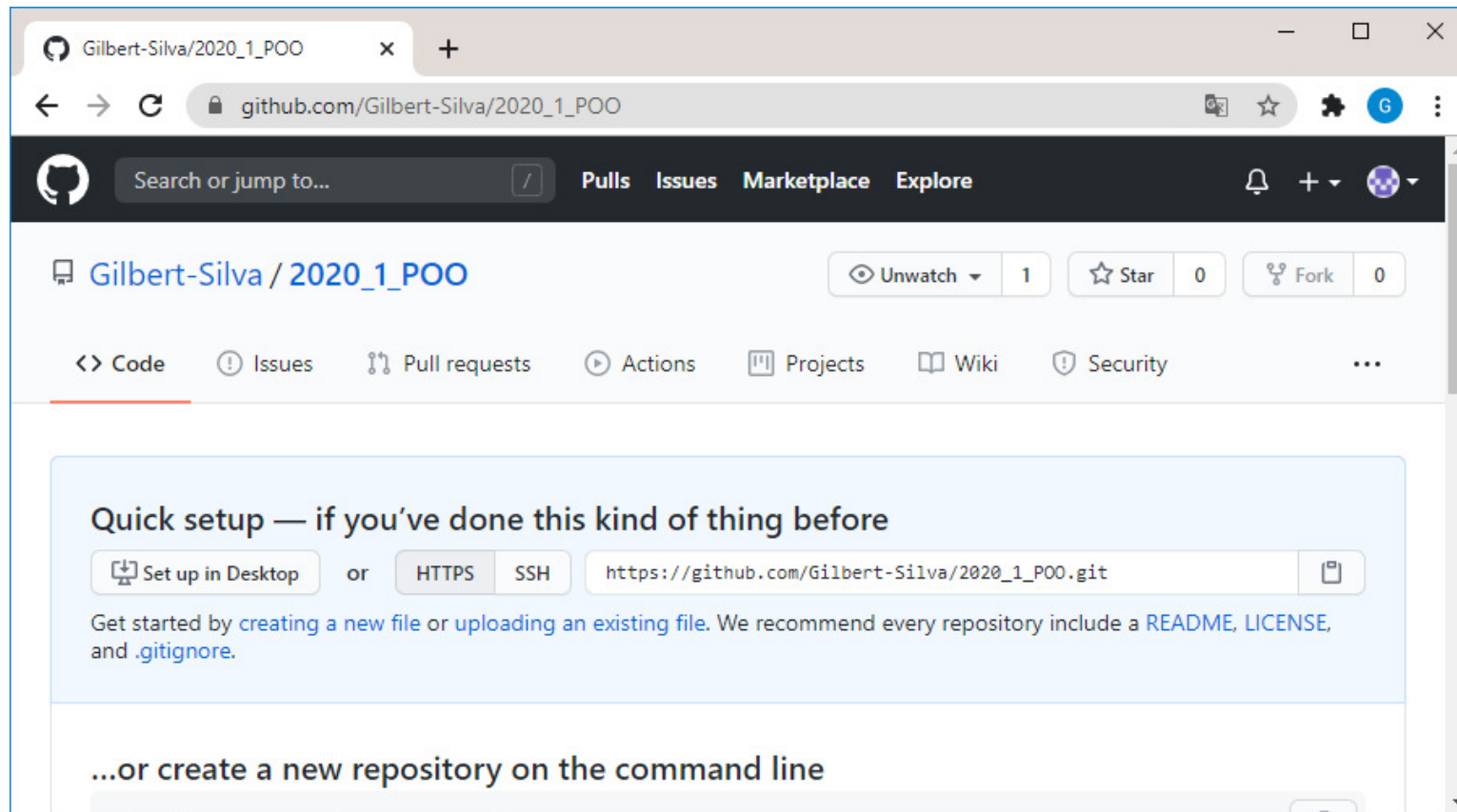
# Nomeando o Repositório

- Defina um nome para o repositório



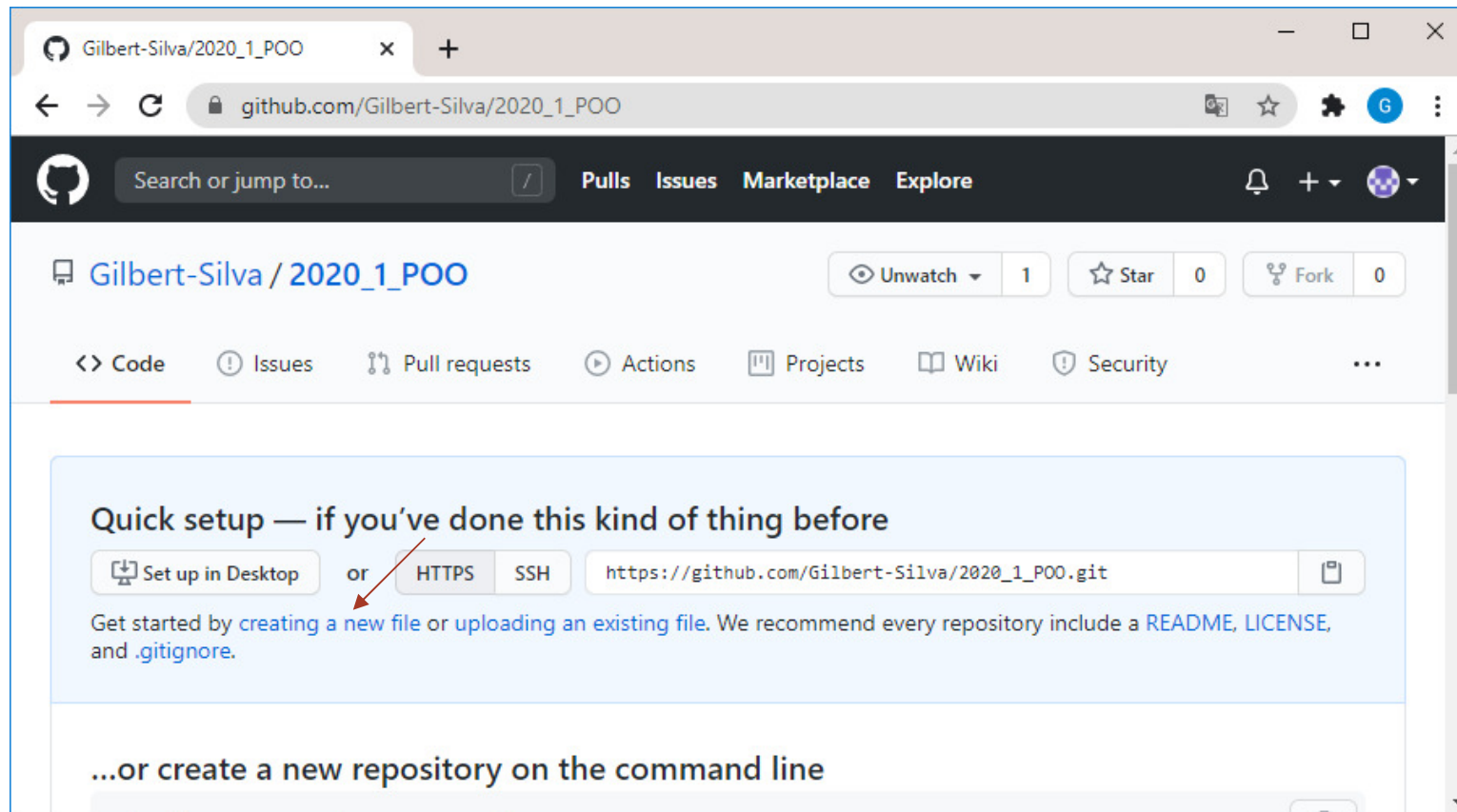
# Página do Repositório

- A página mostra os dados do repositório: código, tarefas (questões), solicitações Pull, ações, projetos, Wiki, ...



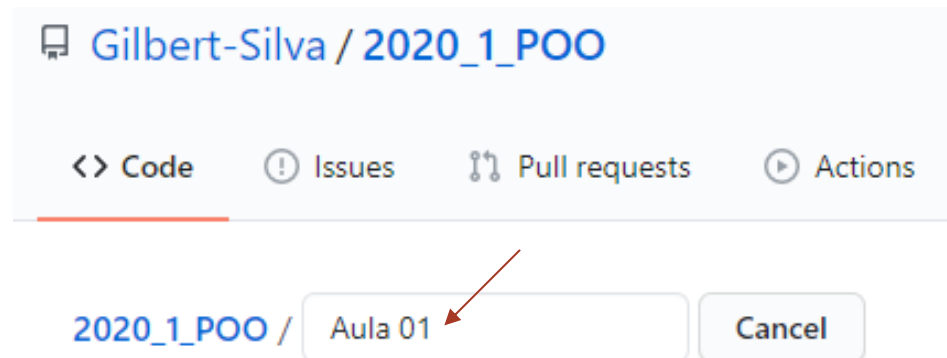
# Criando uma Pasta

- Selecione *creating a new file* para adicionar pastas e arquivos ao repositório

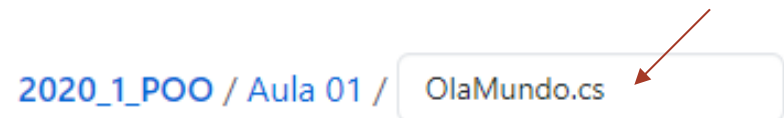


# Nomeando a Pasta e o Arquivo

- Na caixa de texto, digite “*Aula 01*”. A pasta é criada quando uma “/” for digitada.

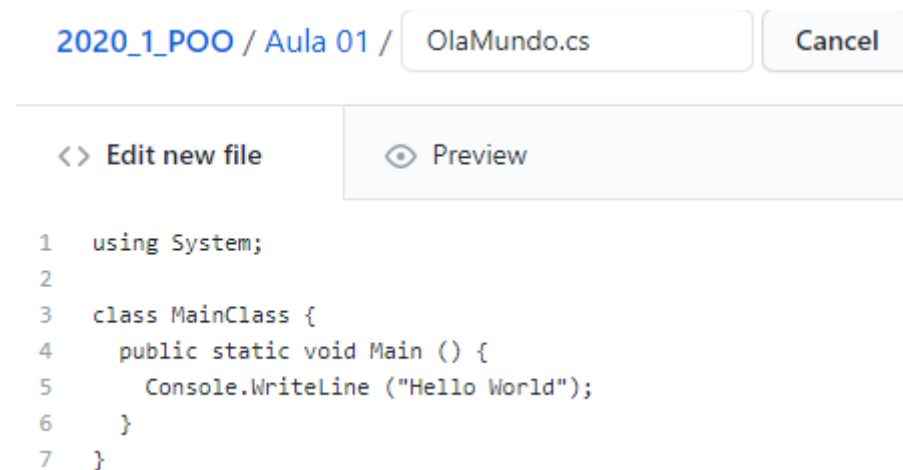


- Digite o nome do arquivo “*OlaMundo.cs*”.



# Digitando o Arquivo

- Digite o código do *OlaMundo.cs* no editor

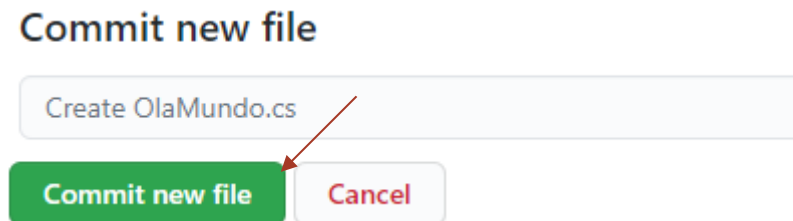


2020\_1\_POO / Aula 01 / OlaMundo.cs Cancel

<> Edit new file Preview

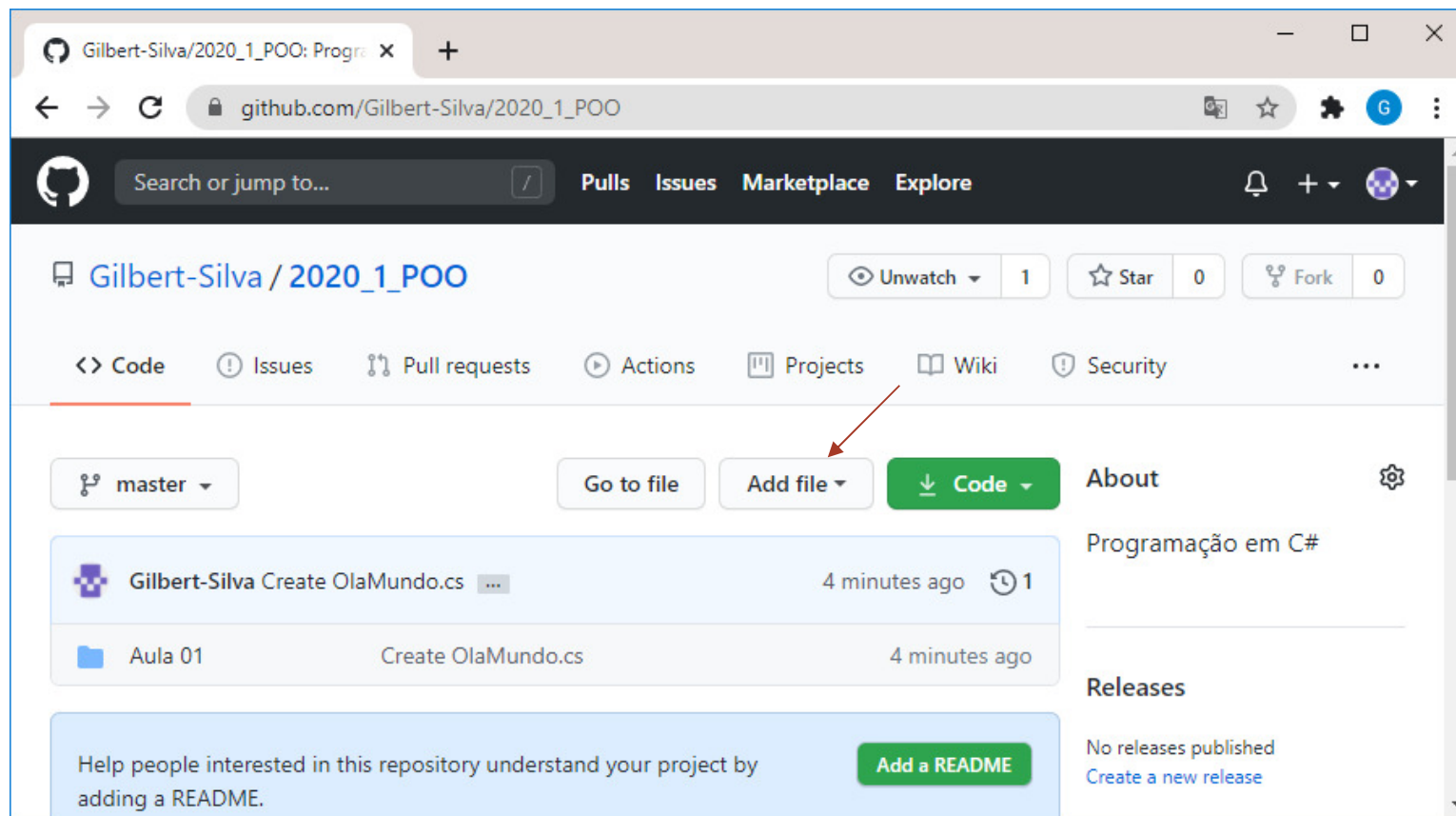
```
1 using System;
2
3 class MainClass {
4     public static void Main () {
5         Console.WriteLine ("Hello World");
6     }
7 }
```

- Confirme a criação do arquivo no botão *Commit*



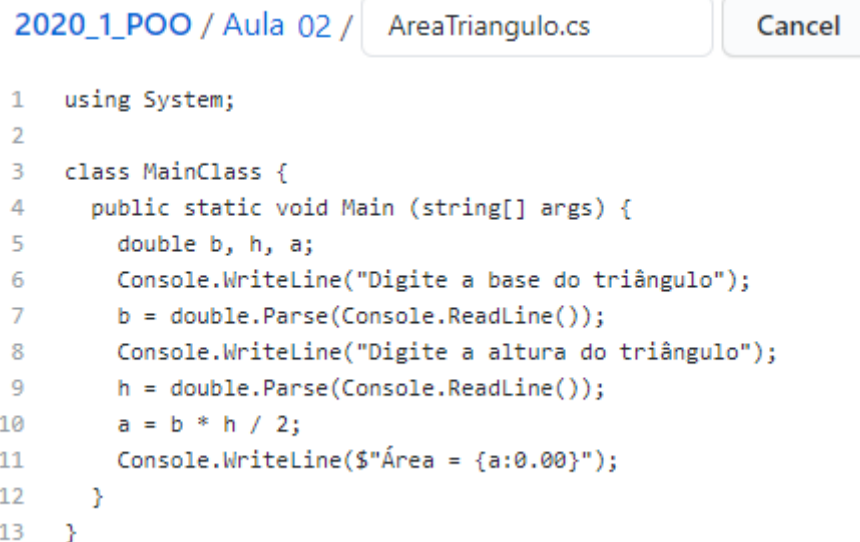
# Visualizando o Repositório

- Selecione *Add file/Create new file* para adicionar um novo arquivo



# Adicionando outro Arquivo

- Repita o procedimento anterior. Crie a pasta “*Aula 02*” e o arquivo “*AreaTriangulo.cs*”



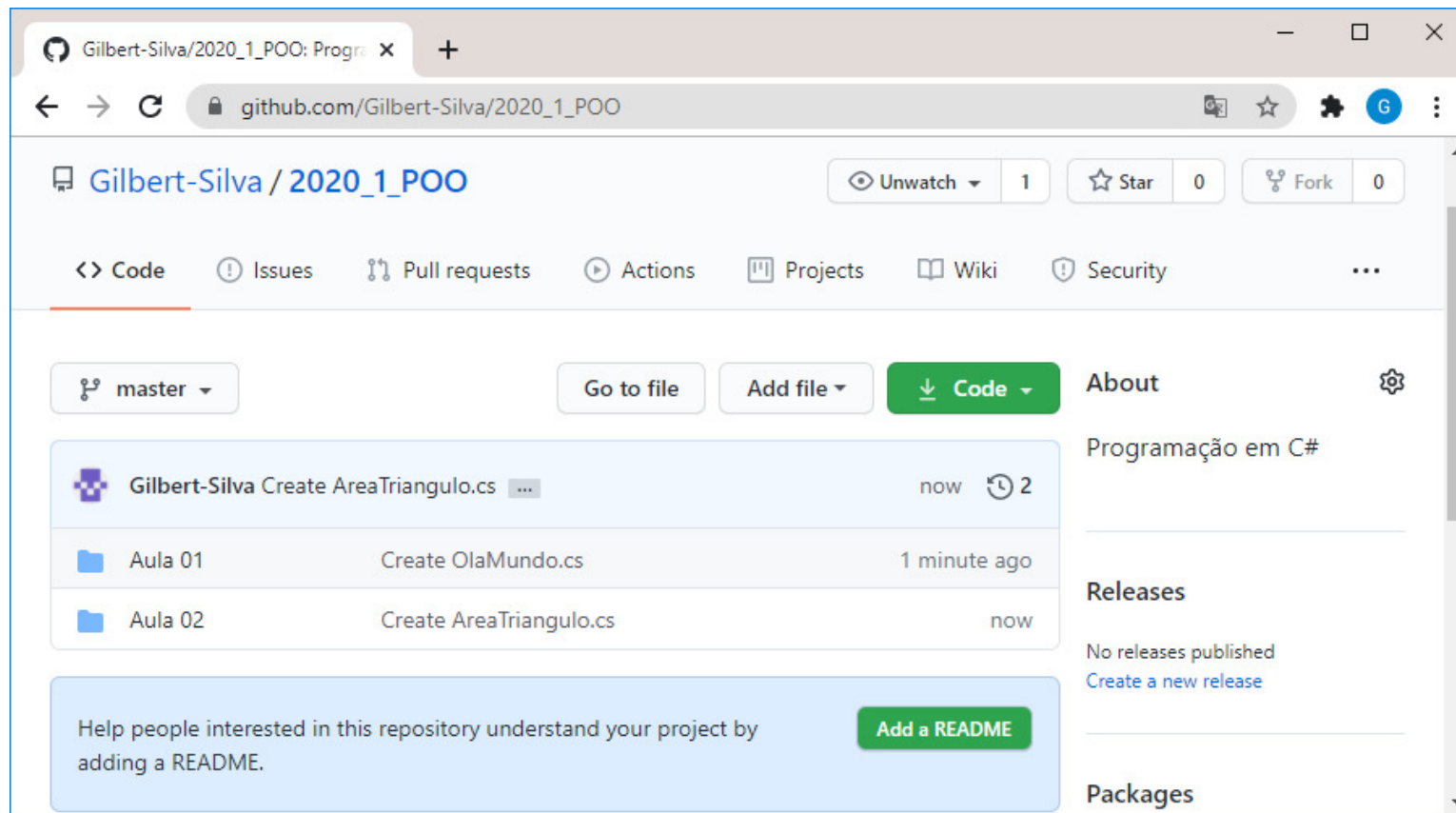
```
2020_1_POO / Aula 02 / AreaTriangulo.cs Cancel

1  using System;
2
3  class MainClass {
4      public static void Main (string[] args) {
5          double b, h, a;
6          Console.WriteLine("Digite a base do triângulo");
7          b = double.Parse(Console.ReadLine());
8          Console.WriteLine("Digite a altura do triângulo");
9          h = double.Parse(Console.ReadLine());
10         a = b * h / 2;
11         Console.WriteLine($"Área = {a:0.00}");
12     }
13 }
```

- Confirme a criação do arquivo no botão *Commit*

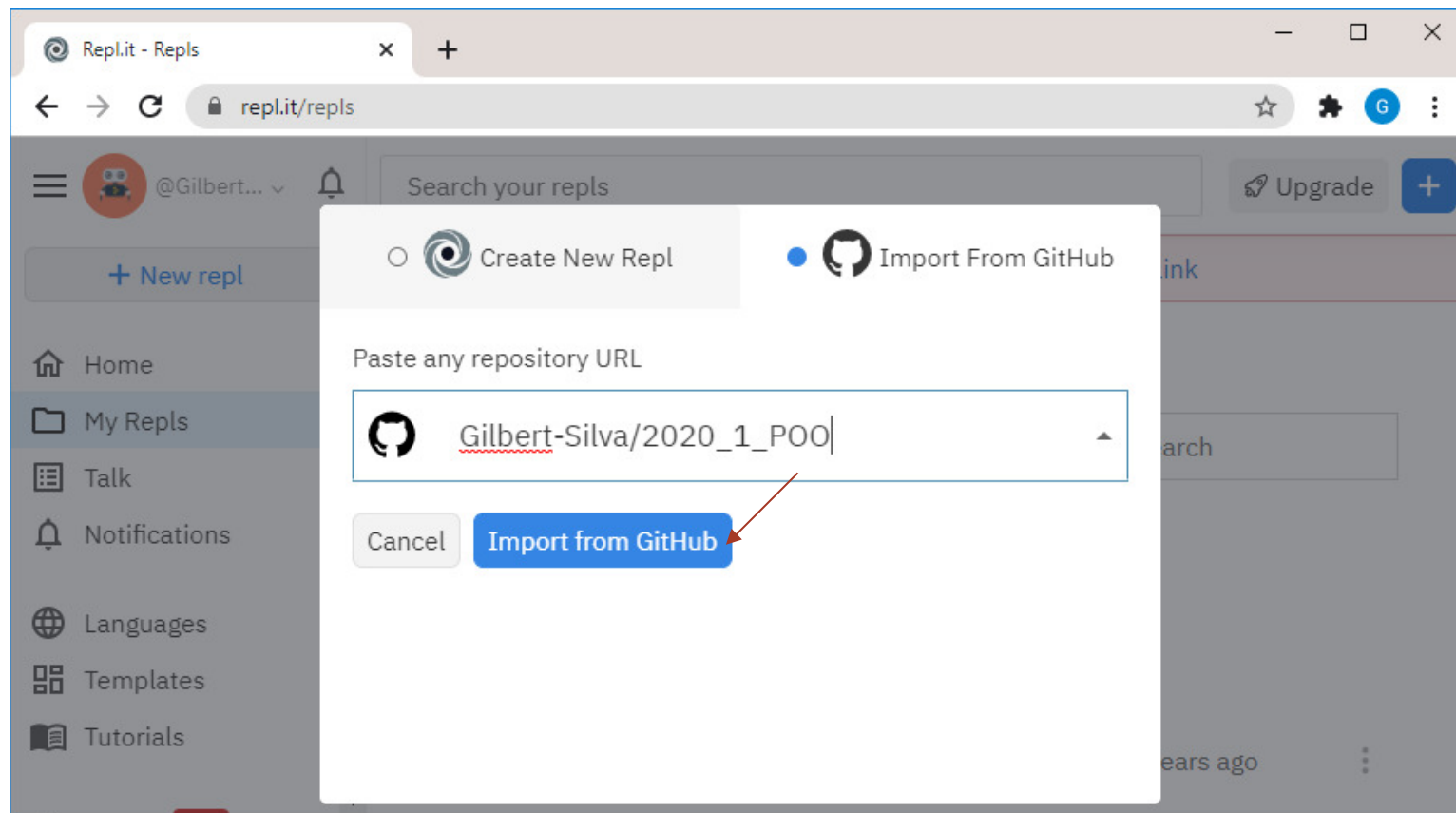
# Visualizando o Repositório

- As pastas *Aula 01* e *Aula 02* são apresentadas no repositório



# Clonando o Repositório no repl.it

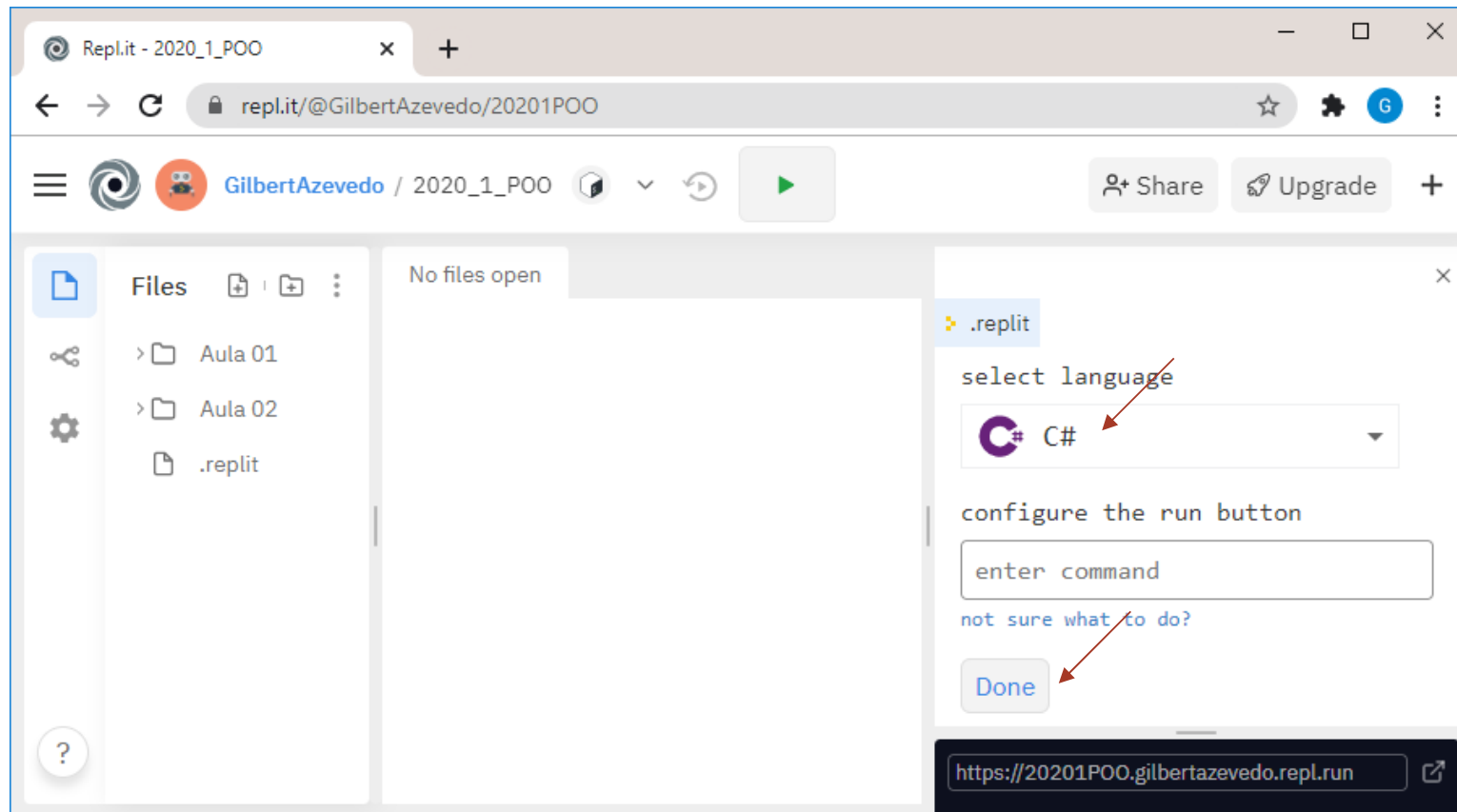
- No repl.it, selecione *New repl* e em seguida *Import From GitHub*
  - Informe o repositório a ser clonado





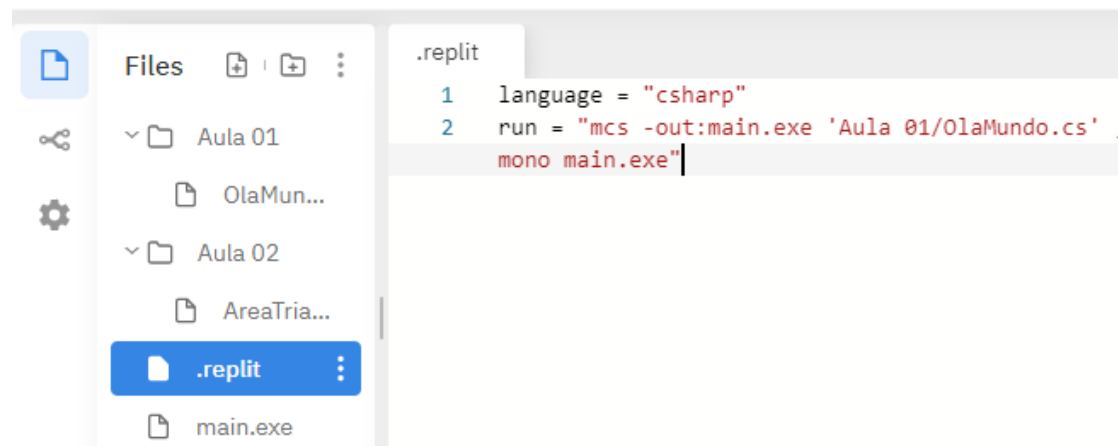
# Configurando a Linguagem

- Informe a linguagem de programação e selecione *Done*
  - O repositório é apresentado na página *Files* do repl.it



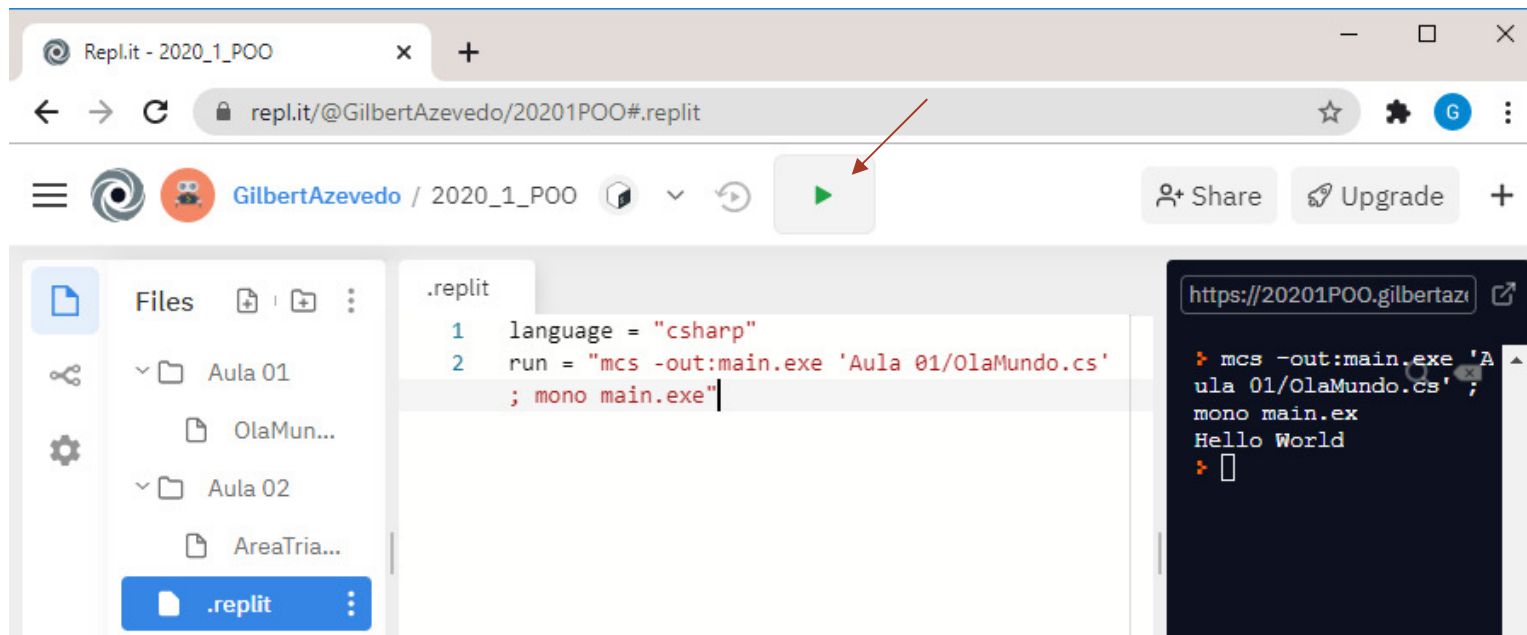
# Configurando o Comando Run

- Informe no comando *run* do arquivo *.replit* do repositório, o arquivo do repositório que vai ser executado
  - `language = "csharp"`
  - `run = "mcs -out:main.exe 'Aula 01/OlaMundo.cs' ; mono main.exe"`



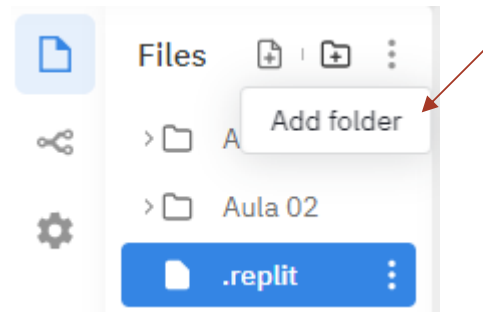
# Executando o Programa

- Após informar o arquivo no *run*, é só executar o programa

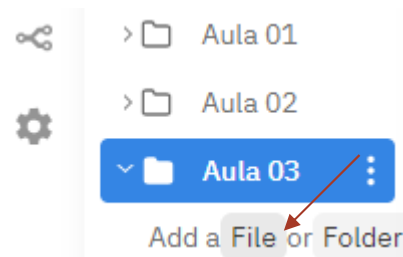


# Criando uma Pasta no repl.it

- Selecione *Add folder* para inserir uma nova pasta *Aula 03*

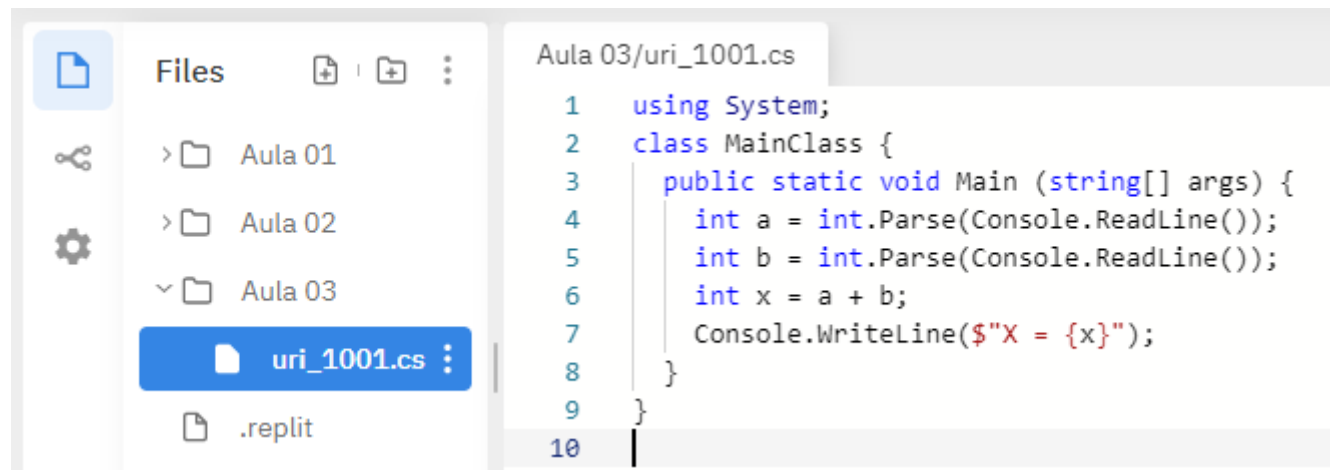


- Na pasta *Aula 03*, selecione *Add a File*



# Criando um Arquivo no repl.it

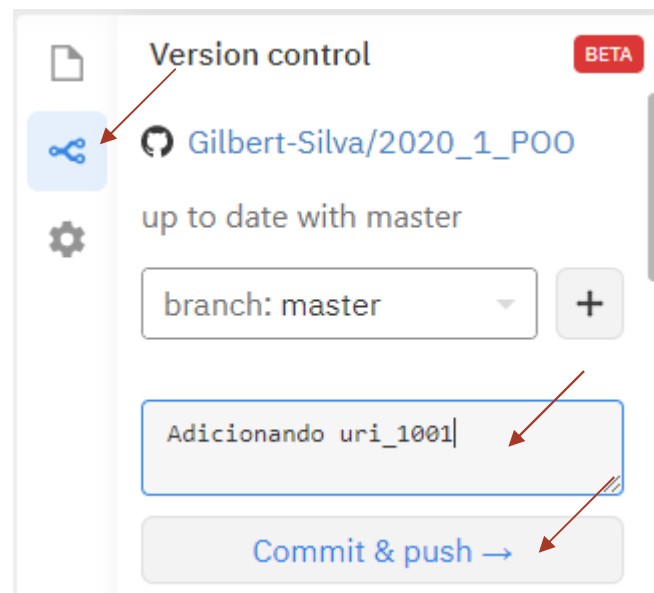
- Adicione o arquivo “*uri\_1001.cs*” e digite o código do programa



- Ajuste o comando *run* no arquivo *.replit* e execute o programa
  - `run = "mcs -out:main.exe 'Aula 03/uri_1001.cs' ; mono main.exe"`

# Confirmando as Alterações

- Selecione o botão *Version control* para confirmar as alterações.
  - Informe um texto com as alterações realizadas no repositório
  - Selecione o botão *Commit & push* para confirmar



# Enviando as Alterações

- O *commit* deixa a versão no repl.it uma versão a frente do GitHub



- Para atualizar o GitHub, execute o comando *push* no terminal do repl.it

- `git push`

```
https://20201POO.gilbertazevedo.repl.run
> git push
Username for 'https://github.com':
Password for 'https://
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 1.68 KiB | 1.68 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/Gilbert-Silva/2020_1_POO
 e871c4e..19e783a master -> master
>
```

# GitHub Atualizado

The screenshot shows a web browser window displaying the GitHub repository page for `Gilbert-Silva/2020_1_POO`. The browser's address bar shows the URL `github.com/Gilbert-Silva/2020_1_POO`. The repository page header includes the repository name, a navigation bar with links for `Code`, `Issues`, `Pull requests`, `Actions`, `Projects`, `Wiki`, and `Security`, and interaction buttons for `Unwatch` (1), `Star` (0), and `Fork` (0). Below the header, there are buttons for `master`, `Go to file`, `Add file`, and `Code`. The main content area displays a list of files and folders, with the most recent commit by `GilbertAzevedo` titled `Adicionando uri_1001` (8 minutes ago, 3 commits). The file list includes:

File/Folder	Commit Message	Time Ago
Aula 01	Create OlaMundo.cs	1 hour ago
Aula 02	Create AreaTriangulo.cs	1 hour ago
Aula 03	Adicionando uri_1001	8 minutes ago
.replit	Adicionando uri_1001	8 minutes ago
main.exe	Adicionando uri_1001	8 minutes ago

On the right side of the repository page, there are sections for `About` (Programação em C#), `Releases` (No releases published, [Create a new release](#)), and `Packages`.



# Referências

- GitHub: <https://github.com/>
- GitHub Guides: <https://guides.github.com/activities/hello-world/>
- GitHub Learning Lab: <https://lab.github.com/>

# Fim

- Tarefa
  - Questionário
- Próxima Aula
  - Estruturas de Decisão