



# Programação de jogos digitais

Diretoria de Educação  
Gerência de Educação Profissional  
Abril de 2020





# Git

Controle de  
versão

# Controle de versão e Github

Git

Controle de  
versão

## Usando o controle de versão com o Github

Vamos começar a usar o controle de versão em conjunto com o serviço de hospedagem Github. Isso atua como um banco de dados online usado para hospedar projetos e interage muito bem com o controle de versão. Neste ponto, aprendemos como adicionar controle de versão a um projeto, salvar arquivos por meio de confirmações, criar ramificações e mesclar versões de arquivos. Agora vamos fazer todas essas coisas e muito mais com um repositório online, em vez de local. Isso exigirá algumas etapas extras, como clonar o repositório online, extrair arquivos do repositório e, finalmente, enviar arquivos para o repositório.

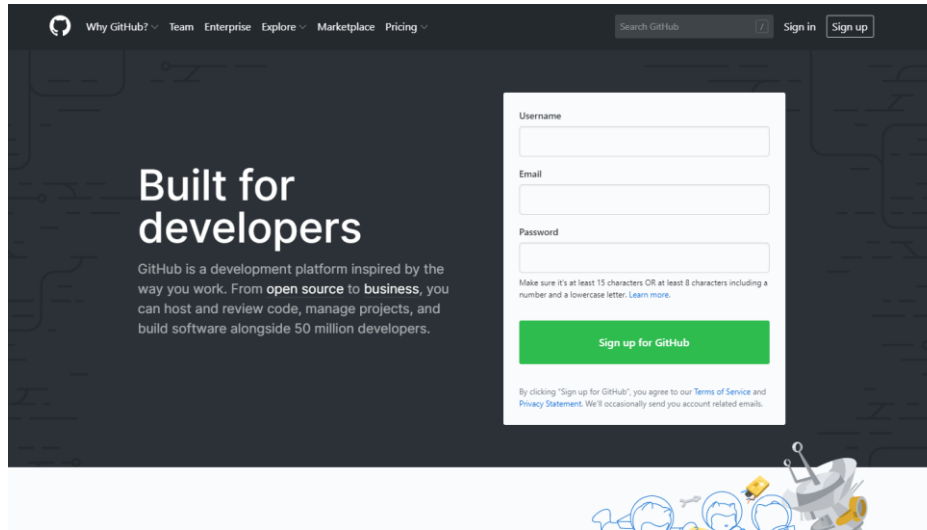
# Controle de versão e Github

# Git

Controle de  
versão

## Usando o controle de versão com o Github

A primeira coisa que precisamos fazer é configurar uma conta do Github. Este é um processo fácil e é totalmente gratuito. Acesse: <https://github.com/>



The image shows the GitHub sign-up page. The background is dark with a faint pattern of code. On the left, the text 'Built for developers' is prominent, followed by a description of GitHub as a development platform. On the right, there is a white sign-up form with fields for 'Username', 'Email', and 'Password'. Below the password field, there is a note about password requirements and a link to 'Learn more'. A green 'Sign up for GitHub' button is at the bottom of the form. At the very bottom of the page, there is a small illustration of a rocket ship and some code icons.

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search GitHub / Sign in Sign up

**Built for developers**

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 50 million developers.

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.

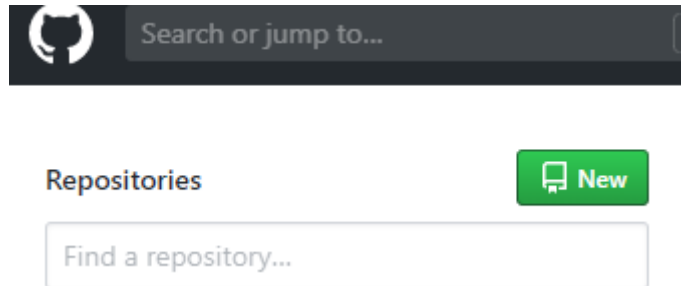
# Controle de versão e Github

Git

Controle de  
versão

## Usando o controle de versão com o Github

Depois de entrar, queremos criar um novo repositório para o nosso projeto. Pressione o botão "novo":



# Controle de versão e Github

Git  
Controle de  
versão

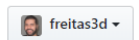
## Usando o controle de versão com o Github

A partir daqui, preencha os campos com as informações do seu projeto:

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Owner



Repository name \*

/ FlappyBird ✓

Great repository names are short and memorable. Need inspiration? How about [expert-octo-journey](#)?

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: Unity ▾

Add a license: None ▾



Create repository

# Controle de versão e Github

# Git

Controle de  
versão

## Usando o controle de versão com o Github

Depois que o repositório for criado, será algo parecido com isto:

The screenshot shows the GitHub interface for a newly created private repository named 'FlappyBird' by user 'freitas3d'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with tabs for '<> Code', 'Issues 0', 'Pull requests 0', 'Actions', 'Projects 0', 'Security 0', 'Insights', and 'Settings'. The main content area has a placeholder description 'No description, website, or topics provided.' with an 'Edit' button. Below the description, it shows repository statistics: '1 commit', '1 branch', '0 packages', and '0 releases'. A row of buttons includes 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. The commit history shows an 'Initial commit' by 'freitas3d' 5 minutes ago, with a file named '.gitignore'. At the bottom, there is a blue banner with the text 'Add a README with an overview of your project.' and a green 'Add a README' button.

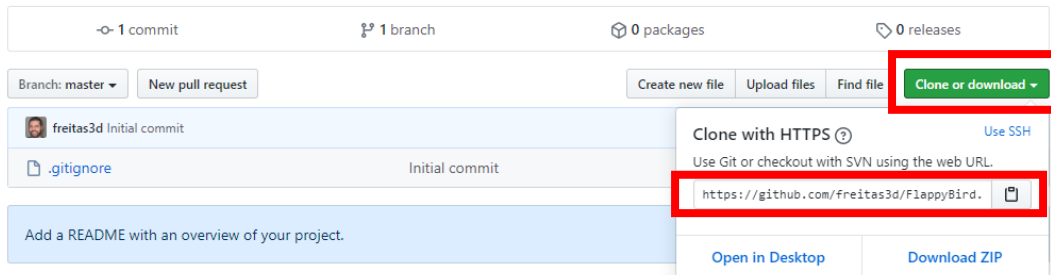
# Controle de versão e Github

# Git

Controle de  
versão

## Clonando um repositório Online

Agora, queremos ter acesso a este repositório do Github em nosso computador, para que possamos enviar nossos arquivos de projeto para ele. Isso é conhecido como "clonar" um repositório e é a segunda maneira de adicionar controle de versão a um repositório local. Queremos buscar o link para o repositório online, clicando no botão "Clonar ou baixar" e copiando o link da seguinte forma:





# Controle de versão e Github

Git

Controle de  
versão

## Clonando um repositório Online

Em seguida, queremos criar um novo diretório local para interagir com o github. Acessei a pasta do projeto FlappyBird. Se você quiser criar uma pasta usando o terminal use o comando **mkdir <nome do diretório>** que criará um diretório chamado <nome do diretório> no diretório de trabalho atual. Use o comando **cd <nome do diretório>** para acessar a pasta do projeto.

```
Cleber@CleberPC MINGW64 ~  
$ cd G:/UNITY/Projeto2DSenai  
  
Cleber@CleberPC MINGW64 /g/UNITY/Projeto2DSenai  
$
```

# Controle de versão e Github

Git

Controle de  
versão

## Clonando um repositório Online

Agora, executamos o comando **git clone <repository-link>**, em que o <repository-link> é substituído pelo link que você clonou. Ele conterá um diretório com tudo o que está atualmente no repositório do github que, neste momento, é apenas um arquivo  
Veja:

```
Cleber@CleberPC MINGW64 /g/UNITY/Projeto2DSenai
$ git clone https://github.com/freitas3d/FlappyBird.git
Cloning into 'FlappyBird'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
Cleber@CleberPC MINGW64 /g/UNITY/Projeto2DSenai
$ |
```

# Controle de versão e Github

Git

Controle de  
versão

## Clonando um repositório Online

Agora copie todo o conteúdo da pasta baixada para a raiz da pasta do projeto. Em seguida, por meio do comando:

```
git status
```

Você perceberá que há arquivos (os do projeto) que não estão adicionados, portanto, utilize o comando:

```
git add .
```

Para adicionar todos eles. Em seguida, faça um commit para comprometer todos arquivos adicionados ao Git usando o comando:

```
git commit -m "Primeiro commit"
```

Você notará que os arquivos ainda não estão no servidor github. Isso ocorre porque nós os comprometemos apenas localmente e ainda precisamos empurrá-los para o github.

# Controle de versão e Github

Git

Controle de  
versão

## Enviar para o Github

Para enviar os arquivos confirmados ao github, execute o comando:

**git push**

Isso adicionará todos os arquivos que foram confirmados (no diretório atual) ao github. Você precisará fornecer seu nome de usuário e senha do github. Se for bem-sucedido, você deverá ver algo assim:

```
Cleber@CleberPC MINGW64 /g/UNITY/Projeto2DSenai (master)
$ git push
Enumerating objects: 68, done.
Counting objects: 100% (68/68), done.
Delta compression using up to 4 threads
Compressing objects: 100% (66/66), done.
Writing objects: 100% (67/67), 144.22 KiB | 3.70 MiB/s, done.
Total 67 (delta 14), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (14/14), done.
To https://github.com/freitas3d/FlappyBird.git
   eeab128..34c4818  master -> master
```

# Controle de versão e Github

# Git

Controle de  
versão

## Enviar para o Github

Após atualizar a página do github, você deverá ver os arquivos do seu projeto lá. Ele também mostrará o commit mais recente com mais informações, se você clicar nele:

The screenshot shows the GitHub interface for a repository named 'freitas3d / FlappyBird'. At the top, there are buttons for 'Unwatch', 'Star', and 'Fork'. Below this is a navigation bar with links to 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The main content area shows the repository's description (empty), a table of commits, and a list of files. The commit table shows the latest commit '34c4818' from 'freitas3d' 3 minutes ago. The file list includes 'Assets', 'Packages', 'ProjectSettings', and '.gitignore', all with their respective commit hashes and timestamps.

File	Commit	Time
Assets	Primeiro commit	3 minutes ago
Packages	Primeiro commit	3 minutes ago
ProjectSettings	Primeiro commit	3 minutes ago
.gitignore	Initial commit	29 minutes ago

# Controle de versão e Github

Git

Controle de  
versão

## Puxando do Github

Se você quiser recuperar as versões atuais dos arquivos no repositório do github, precisará puxá-las com o comando:

**git pull**

Isso buscará tudo o que está atualmente armazenado nesse repositório, a menos que não tenha havido alterações (como no meu):

```
Cleber@CleberPC MINGW64 /g/UNITY/Projeto2DSenai (master)
$ git pull
Already up to date.

Cleber@CleberPC MINGW64 /g/UNITY/Projeto2DSenai (master)
$
```

# Controle de versão e Github

Git

Controle de  
versão

## Gitignore

Se houver algum arquivo que não desejamos enviar ou puxar, podemos usar um arquivo gitignore. Eu recomendo usar arquivo gerado pelo Github específico para a Unity, conforme fizemos no momento da criação do projeto.

# Controle de versão e Github

Git

Controle de  
versão

## Novas ramificações no Github

Muitas vezes, queremos criar novas ramificações no github, assim como fizemos com o controle de versão local. Isso garante que os novos recursos ou versões do aplicativo não afetem a versão atual. Podemos usar os mesmos comandos de antes, mas se queremos enviar para um ramo diferente, precisamos especificar para qual ramo estamos enviando. Para isso, usamos o comando:

```
git push origin <branch-name>
```

Substituindo <branch-name> pelo nome do ramo que acabamos de criar.



# Controle de versão e Github

Git

Controle de  
versão

## Novas ramificações no Github

Por exemplo, se eu quisesse um novo ramo chamado “NovaMecanica” e quisesse enviar meus arquivos para ele, faria algo assim:

```
Cleber@CleberPC MINGW64 /g/UNITY/Projeto2DSenai (master)
$ git checkout -b NovaMecanica
Switched to a new branch 'NovaMecanica'

Cleber@CleberPC MINGW64 /g/UNITY/Projeto2DSenai (NovaMecanica)
$ git branch
* NovaMecanica
  master

Cleber@CleberPC MINGW64 /g/UNITY/Projeto2DSenai (NovaMecanica)
$ git push origin NovaMecanica
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'NovaMecanica' on GitHub by visiting:
remote:   https://github.com/freitas3d/FlappyBird/pull/new/NovaMecanica
remote:
To https://github.com/freitas3d/FlappyBird.git
 * [new branch]      NovaMecanica -> NovaMecanica

Cleber@CleberPC MINGW64 /g/UNITY/Projeto2DSenai (NovaMecanica)
$ |
```

# Controle de versão e Github

# Git

Controle de  
versão

## Novas ramificações no Github

Note que agora a página do repositório do github possui duas ramificações

The screenshot shows the GitHub repository page for 'freitas3d / FlappyBird'. The repository is private and has 1 watch, 0 stars, and 0 forks. The 'Code' tab is selected, showing options for cloning or downloading. The 'Branches' tab is also visible, showing a list of branches. The 'master' branch is the default, and there is a new branch named 'NovaMecanica'.

freitas3d / FlappyBird Private

Unwatch 1 Star 0 Fork 0

< Code Issues 0 Pull requests 0 Actions Projects 0 Security 0 Insights Settings

No description, website, or topics provided. Edit

Manage topics

2 commits 2 branches 0 packages 0 releases

Branch: master New pull request Create new file Upload files Find file Clone or download

Switch branches/tags Latest commit 34c4818 16 minutes ago

Find or create a branch...

Branches	Tags
✓ master default	
NovaMecanica	

Commit	Time
Primeiro commit	16 minutes ago
Primeiro commit	16 minutes ago
Primeiro commit	16 minutes ago
Initial commit	42 minutes ago

Add a README with an overview of your project. Add a README

# Controle de versão e Github

Git

Controle de  
versão

## Solicitações Pull

Depois de fazer alterações no repositório do github, você provavelmente desejará contar aos outros membros da sua equipe. Você pode fazer isso através de "solicitações de recebimento". Isso permite que outros membros da equipe revisem as alterações, adicionem comentários e aprovem ou desaprovem as alterações.

Para mostrar como criar corretamente uma solicitação pull, modifiquei um arquivo do projeto para mostrar algumas alterações. Depois, enviei as alterações apenas para o ramo "NovaMecanica".

# Controle de versão e Github

Git

Controle de  
versão

## Solicitações Pull

Agora vamos criar uma solicitação de recebimento. Vamos para a página da web do github, para a guia "Solicitações de recebimento" e pressionamos o botão "Nova solicitação de recebimento". Escolhemos a base como "master" e a comparação como "NovaMecanica" porque queremos mesclar o "desenvolvimento" em "master". Em seguida, pressionamos o botão "Criar solicitação de recebimento":

