

Introdução ao github

Olá alunos(as),

Bem-vindos(as) a nossa última aula da disciplina de Gerência de Configuração. Nesta aula, vamos falar sobre o Github, que é um serviço muito importante de hospedagem de repositórios do Git. Veremos um pouco da história dessa ferramenta e como criar um repositório e fazer um upload dos commits para esse repositório.

Assim, não se esqueçam de ler esta aula, assistir aos nossos vídeos e fazer as atividades. Se tiverem alguma dúvida, vocês podem usar o nosso quadro de avisos, que eu responderei.

⚡ Bons estudos!

Objetivos de aprendizagem

Ao término desta aula, vocês serão capazes de:

- entender o que é o Github;
- conhecer a importância dessa ferramenta;
- saber como criar um repositório para essa ferramenta.

Seções de estudo

- 1 – O que é o github
- 2 – Como criar o seu repositório no github

1 - O que é o github

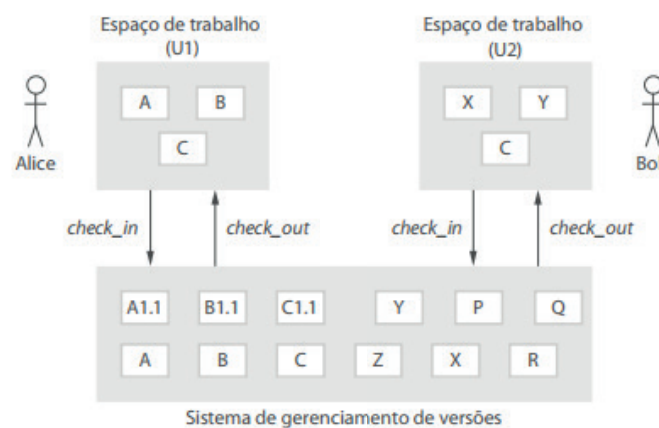
Na aula anterior, você viu como criar um repositório no Git, e criar os seus primeiros commits. Você já deu um grande progresso ao entender como funciona o Git. Mas, o controle de versionamento pode nos oferecer algo a mais.

Nas primeiras aulas da nossa matéria, quando nós

visitamos as quatro atividades da Gerência de Configuração, vimos que, na atividade de Gerenciamento de Versões, você que podemos criar repositórios locais em cada uma das máquinas em que o código está sendo desenvolvido. Porém, para que tudo siga em harmonia, é necessário um repositório central para centralizar as alterações feitas, para que todos estejam no mesmo lugar.

Vamos lembrar aqui o que vimos a respeito disso. Suponha que dois programadores estejam trabalhando em um projeto. Esse projeto está centralizado em um repositório centralizado. A partir desse repositório centralizado, os programadores atualizam os seus repositórios locais realizando o *check_out* e enviam as suas alterações fazendo o *check_in* dos commits para esse repositório.

Figura 1 – Check-in e Check-out a partir de um repositório de versões



Fonte: SOMMERVILLE, 2011, p. 484

Mas, você deve estar se perguntando: onde posso abrigar esse repositório central? Você pode criar esse repositório central de maneira manual, usando algum computador ou servidor encaalhado para isso. Porém, como em todo sistema de cliente-servidor, isso demanda uma configuração inicial do servidor, exigindo algum tempo para isso.

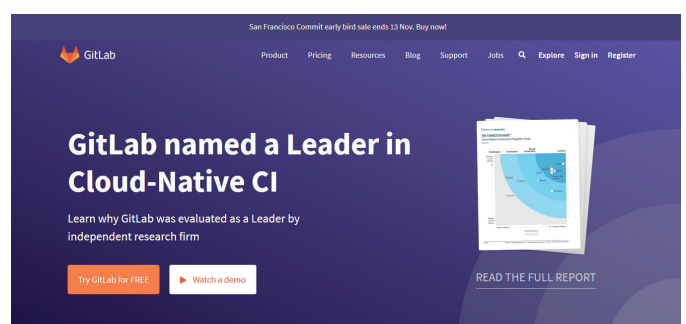
Caso você esteja interessado em saber como funciona a instalação e a configuração de um servidor Git, seguem, para o seu estudo, alguns links de artigos (sendo que alguns deles são da própria documentação oficial do Git) que falam a respeito desse processo de configuração.

- <https://git-scm.com/book/pt-br/v1/Git-no-Servidor-Configurando-o-Servidor>
- <https://blog.pantuza.com/tutoriais/criando-um-servidor-git>

Para facilitar a vida dos programadores, algumas empresas desenvolveram soluções que permitem a hospedagem de repositórios Git de forma online, oferecendo algo a mais

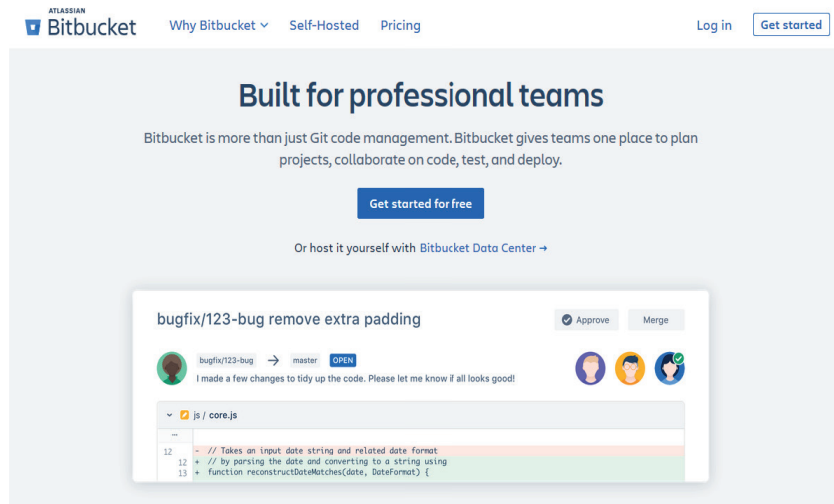
em sua cartela de serviços. São exemplos disso o GitLab (<https://about.gitlab.com/>) e o Bitbucket (<https://bitbucket.org/product/>), que dão a possibilidade de seus usuários para hospedar os seus repositórios do Git, sem complicação:

Figura 2 – Página de apresentação do Gitlab



Fonte: Disponível em: <https://about.gitlab.com>. Acesso em 18 nov. 2019.

Figura 3 – Página de apresentação do Bitbucket.



Fonte: Disponível em: <https://bitbucket.org/product/>. Acesso em 18 nov. 2019.

Mas, o principal deles é o Github (<https://github.com/>).

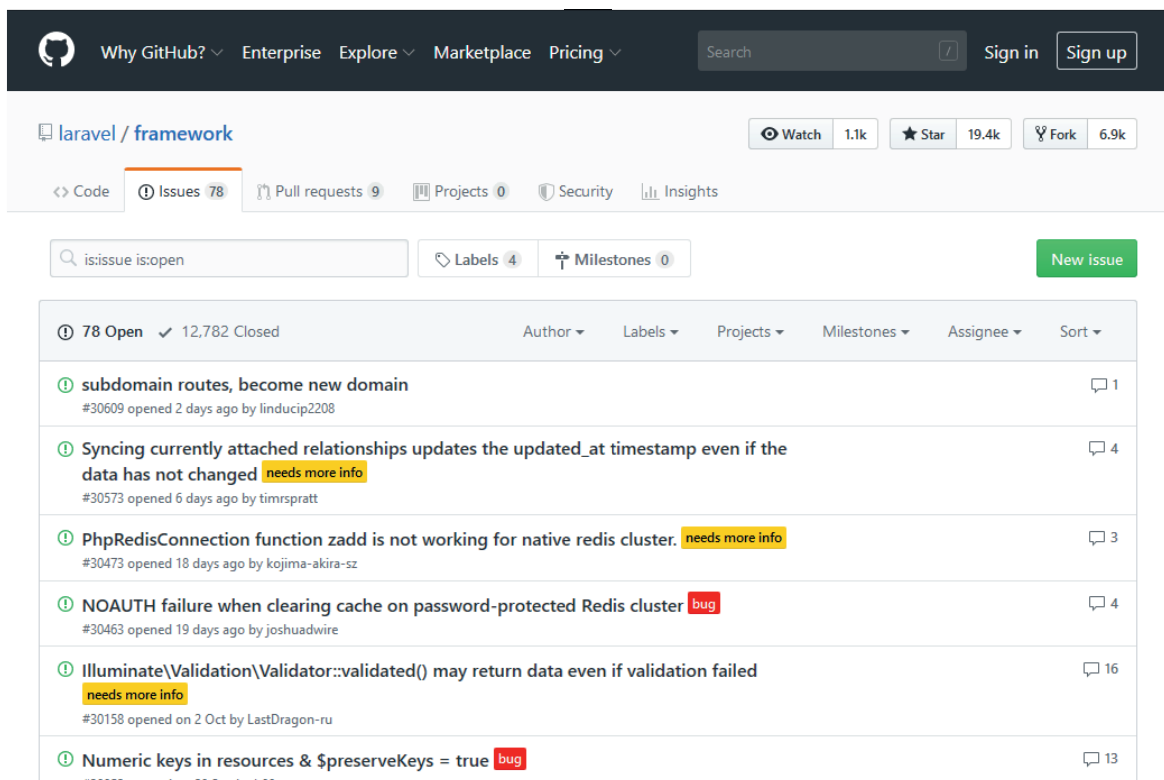
O GitHub é uma plataforma onde você pode armazenar seus projetos. É como se fosse uma rede social, só que de códigos, onde seus desenvolvedores podem disponibilizá-los para outras pessoas verem.

Quando seu projeto está no GitHub, você pode facilmente baixar uma cópia em outro computador. É uma plataforma gratuita e armazena milhões de projetos, tanto open source, pessoais e até mesmo comerciais. (GUEDES, 2019)

Como já dissemos, o Github não tem apenas a utilidade de hospedar repositórios Git. Ele oferece uma série de recursos que podem cobrir ou auxiliar todas as demais atividades da Gerência de Configuração:

- Cada repositório (se o mantenedor permitir) possui uma página denominada **issues**, que permite que os usuários da plataforma registrem bugs ou proponham novas funcionalidades. Os demais usuários podem contribuir na discussão em um modelo similar a um fórum online. Ou seja, essa página pode contribuir para o “Gerenciamento de Mudanças”.

Figura 4 – Exemplo de uma página de issues, do repositório do framework do Laravel

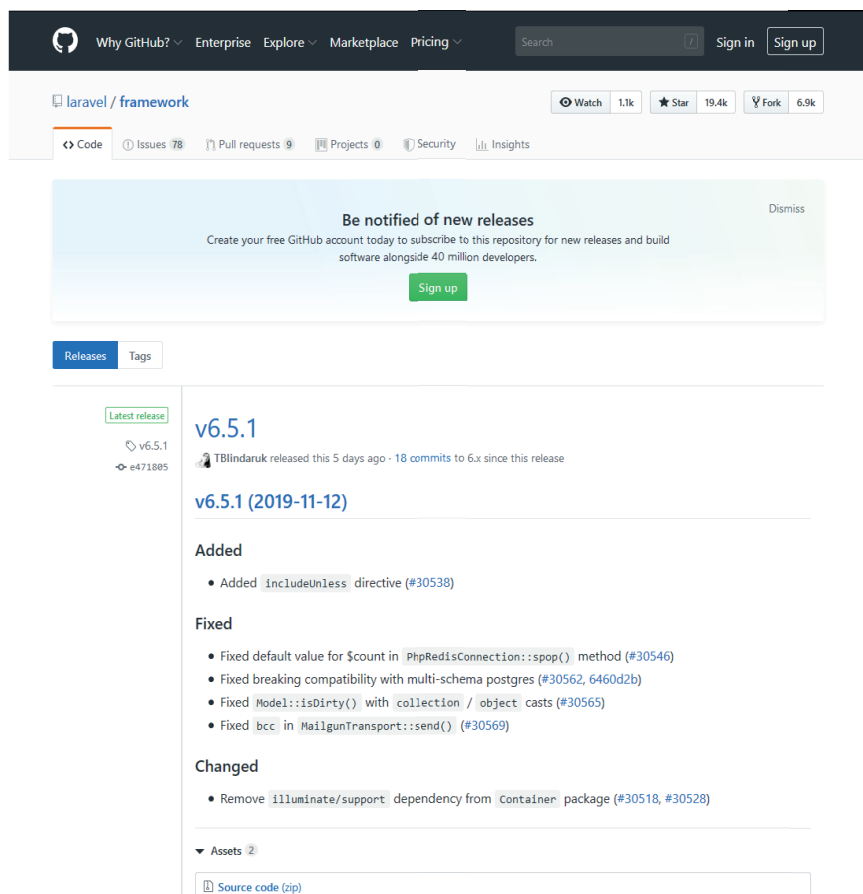


Fonte: Extraída da página: <https://github.com/laravel/framework/issues>. Acesso em 18 nov. 2019.

- Os mantenedores do repositório podem, a partir de um determinado commit, declarar que este commit é um release do sistema desenvolvido. Os releases são listados em uma página própria, onde o visitante pode baixar o arquivo ZIP com o código-fonte do

projeto no momento em que o release foi declarado. Além disso, os mantenedores do repositório podem disponibilizar arquivos binários complementares, que serão listados entre os arquivos de download do release.

Figura 5 – Exemplo de uma página de releases, do repositório do framework do Laravel

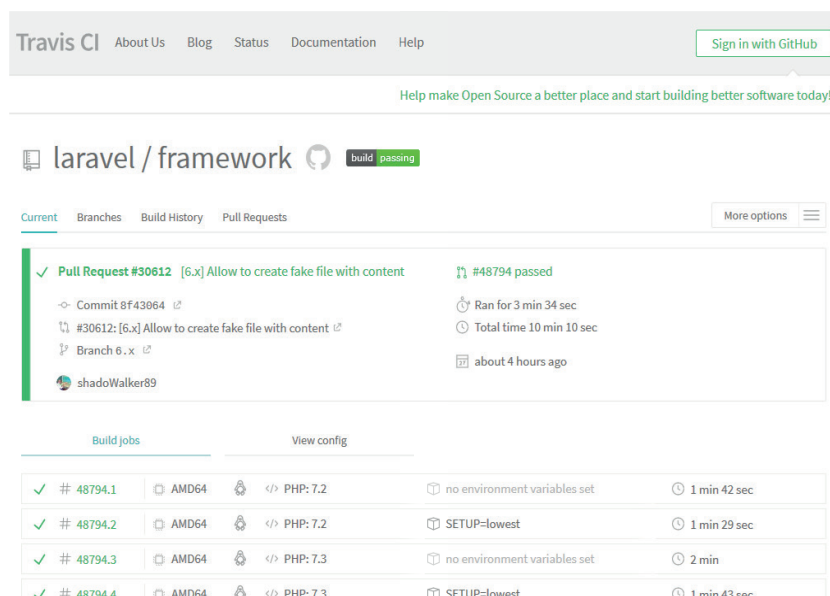


Fonte: Extraída da página: <https://github.com/laravel/framework/releases>. Acesso em 18 nov. 2019.

- Além disso, ter o repositório disponibilizado no Github permite que o projeto possa ser usado em vários sistemas que apoiam na atividade de

“Construção do Sistema” como, por exemplo, o Travis-CI (<https://travis-ci.org>).

Figura 6 – Exemplo de uma página de testes executados pela ferramenta Travis CI em relação ao repositório do framework do Laravel



Fonte: Extraída da página: <https://travis-ci.org/laravel/framework>. Acesso em 18 nov. 2019.

Bem, se esses argumentos não foram o suficiente, vamos dar mais um para convencer você a usar o Github, caso ainda não tenha usado: Em decorrência do número elevado de usuários (mais de 40 milhões de usuários na estatística do Github em 2019, sendo que mais de 10 milhões eram novos usuários naquele ano – GITHUB, 2019), e do elevado número de repositórios, sendo que a maioria deles são de código fonte aberto, abrigando projetos famosos como o Laravel, JQuery, Vue, entre outros, **várias empresas estão utilizando o Github como item a ser avaliado para selecionar candidatos a vagas de emprego.** Ou seja, o Github se tornou para muitos programadores, um portfólio de seus projetos. (ŽIVKOVIĆ, 2018; SCHMITZ, 2015, GUEDES, 2019)

Assim, é muito importante você já ter um repositório do Github para que você possa ter contato com esse novo

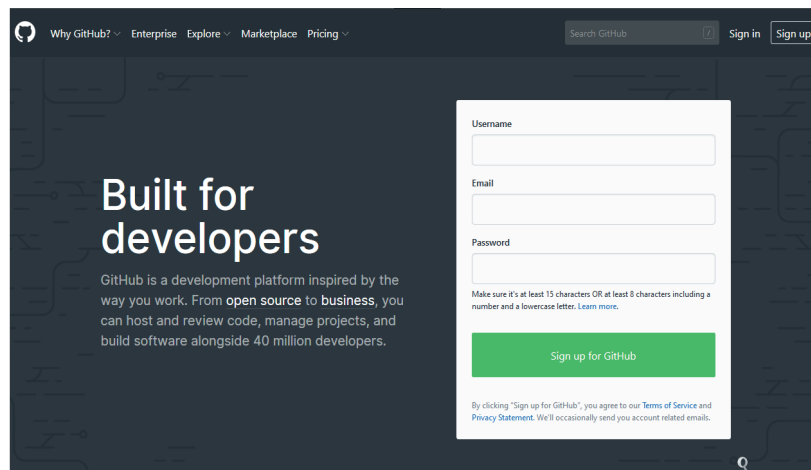
mundo. A seguir, vamos detalhar os passos para a criação da sua conta no Github. Vamos lá?

1.1 – Criando uma conta no Github

Para criar a sua conta no Github, é necessário entrar no site oficial da plataforma: <https://github.com/>. No lado direito da janela, você perceberá um formulário com três campos, a saber:

- Seu nome de usuário – que deve ser único entre todos os usuários do Github;
- Se e-mail válido;
- A senha – que deve ter no mínimo 15 caracteres. Caso inclua um número e uma letra minúscula, o tamanho mínimo cai para 8 caracteres.

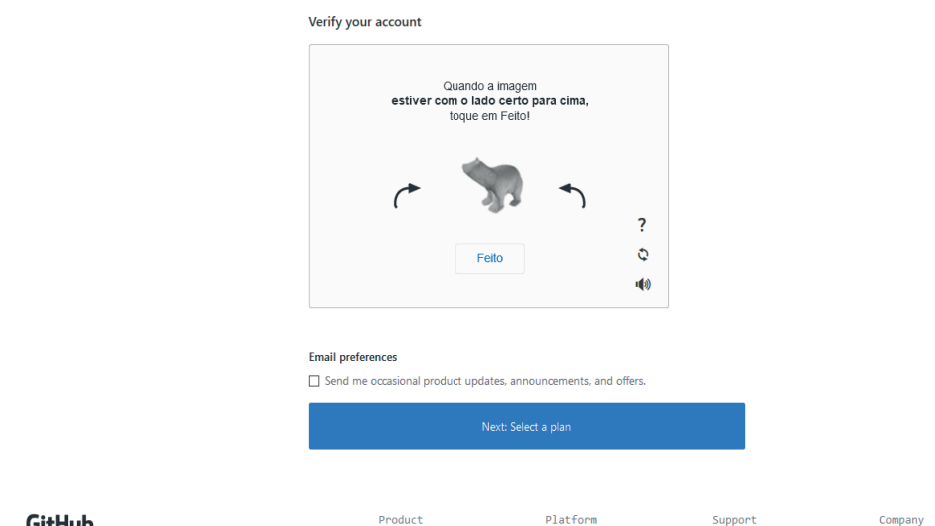
Figura 7 – Página inicial do Github



Fonte: Extraído da página: <https://github.com/>. Acesso em 18 nov. 2019.

Preencha esses campos e clique em “Sign up for Github”. Em seguida, será apresentado um teste para provar que você não é um robô. Faça-o, e depois clique em “Select a plan”.

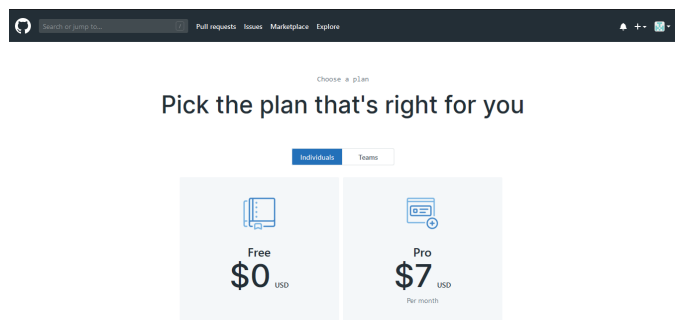
Figura 8 – Verificação da conta



Fonte: Extraído da página: <https://github.com/>. Acesso em 18 nov. 2019.

Em seguida, o Github vai pedir para que você selecione um plano dentre as opções “Free” e “Pro”. Selecione a opção “Free”, que vai oferecer uma série de recursos interessante, para aqueles que são novatos na plataforma.

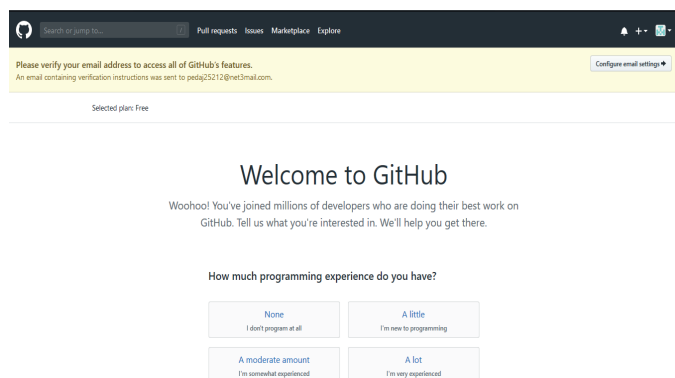
Figura 9 – Tela de seleção do perfil



Fonte: Extraído da página: <https://github.com/>. Acesso em 18 nov. 2019.

Depois, responda a pesquisa curta de perfil do Github.

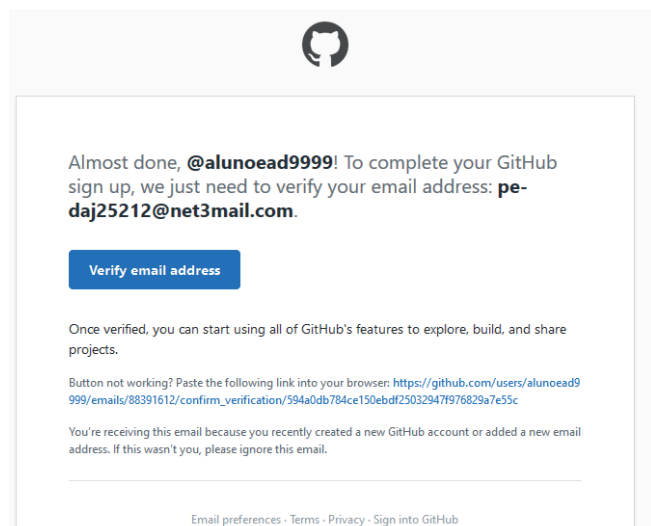
Figura 10 – Pesquisa de coleta de Perfil do Github



Fonte: Extraído da página: <https://github.com/>. Acesso em 18 nov. 2019.

Ao final, mais um passo deve ser seguido. Você deve acessar o e-mail cadastrado no Github e procurar um e-mail deles contendo algo similar ao texto “verify your email address” (verifique o seu endereço de e-mail). Ao achar o e-mail, clique no botão para verificar o seu e-mail e ativar a conta.

Figura 11 – E-mail de verificação do Github.

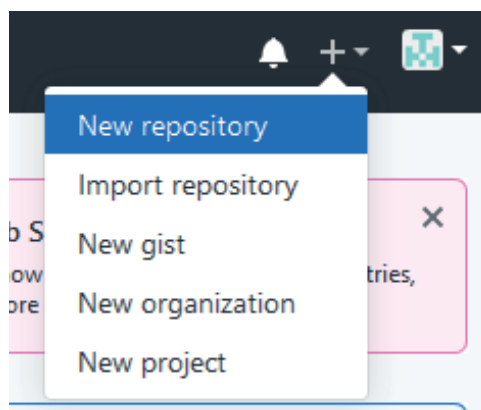


Fonte: Acervo Pessoal.

Com isso feito, estamos prontos para criar o seu primeiro repositório, cujos detalhes descreveremos a seguir. Vamos lá?

2 - Como criar o seu repositório no github

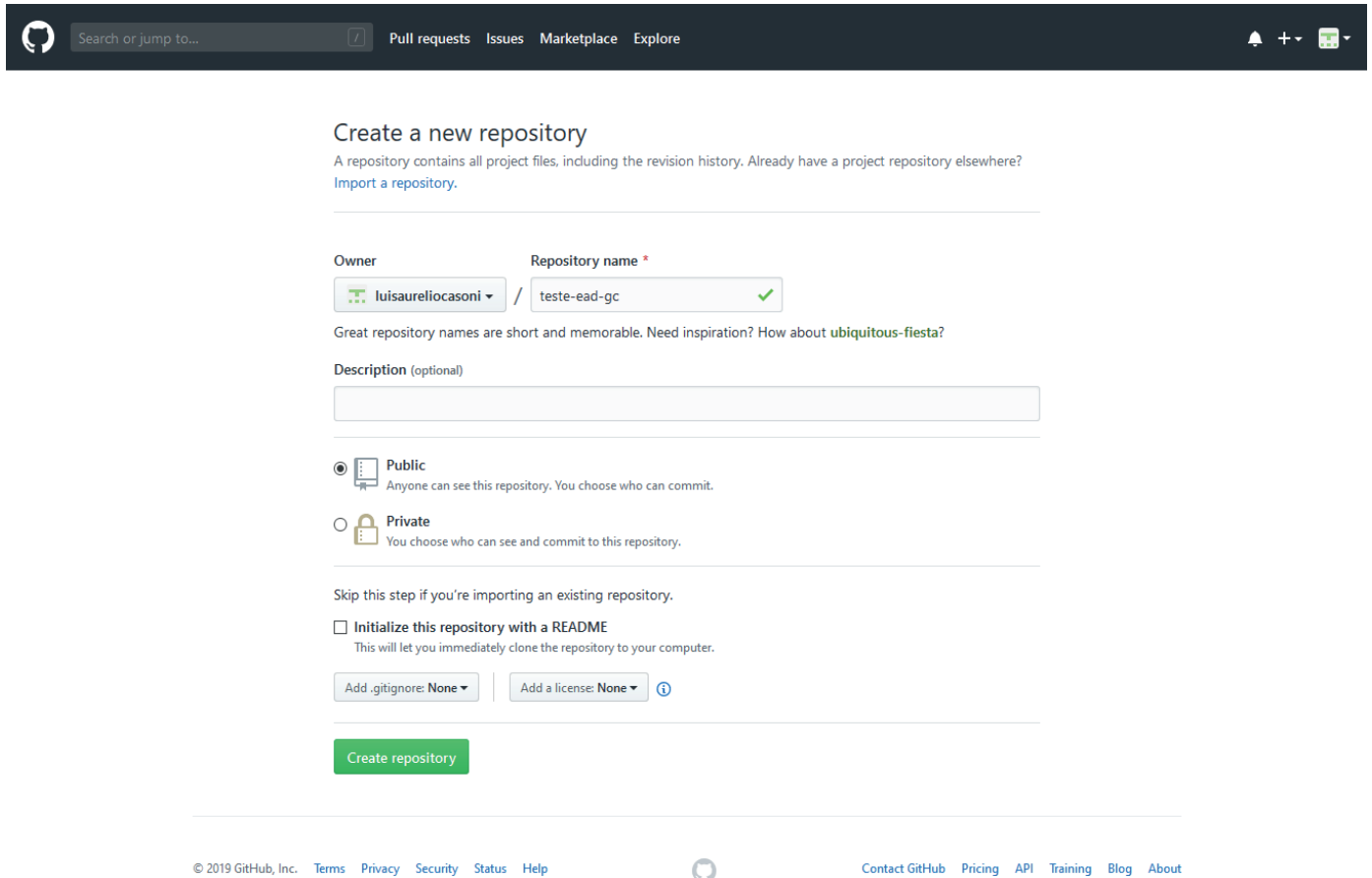
Vamos agora mostrar como criamos um repositório online e fizemos o upload dos nossos commits (check-in) para a plataforma Github. Ao verificar o seu endereço de e-mail, o Github já deve abrir de cara uma tela de criação para o novo repositório. Caso você já tenha fechado a aba do link de ativação, saiba que a criação de um novo repositório online é simples. Basta apenas na barra superior escura da janela, clicar no símbolo “+” e depois clicar em “New Repository”, como está indicado na figura a seguir:



Em seguida, o Github te pedirá algumas informações:

- **Repository Name:** o nome do seu repositório. Ele deve conter letras, números e traços. Você pode digitar espaços, mas eles serão convertidos para traços;
- **Description:** um breve texto descrevendo sobre o repositório;
- **Visibilidade:** indica qual é visibilidade do repositório. Se marcar como Public, qualquer pessoa pode ver e pesquisar pelo repositório, sendo até indexado pelos mecanismos de busca. Se você marcar como Private, o acesso aos arquivos é restrito, e o repositório será oculto da pesquisa do Github. Em qualquer um dos casos, você escolhe quais pessoas poderão alterar o repositório – se o repositório for público, quantas você quiser e se o repositório for privado, até três.



Escolha um nome qualquer (no meu caso, vou colocar “teste-ead-gc”) e marque o repositório como público. Não altere os itens que se referem ao README, .gitignore ou License (licença). Em seguida, clique no botão “Create Repository”.



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 ***

 luisaureliocasoni / teste-ead-gc 

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


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.

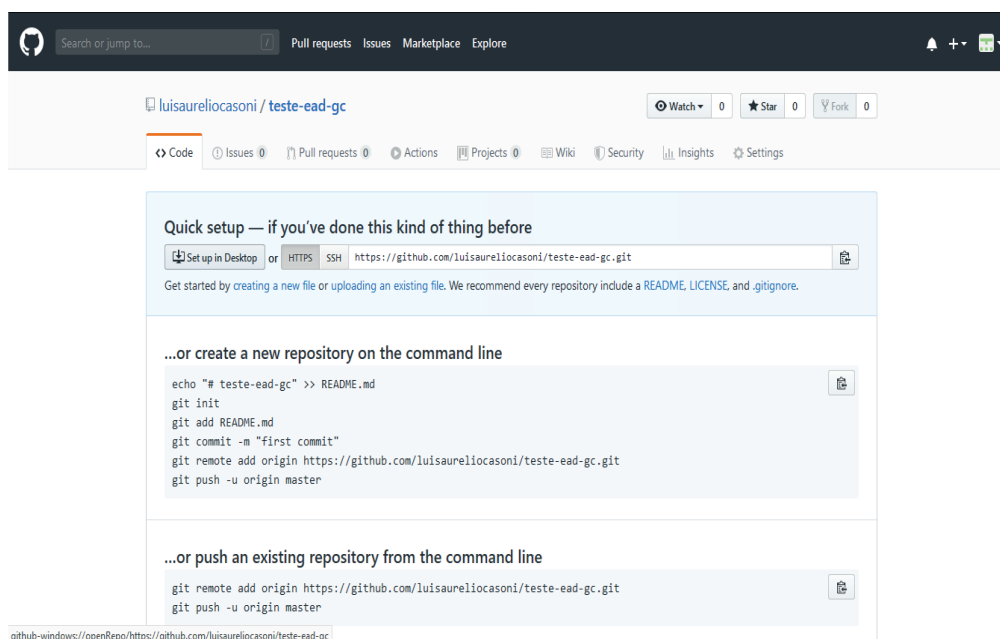
☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.




Add .gitignore: **None** | Add a license: **None** 

Create repository

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)





O repositório será criado. Ao terminar a criação, perceba que o Github vai te mostrar algumas instruções. No nosso caso, interessa apenas o link de acesso ao repositório – que sempre termina com `.git`. Para acessá-lo, procure na barra azul um texto que começa com “https://”. Copie toda essa URL



[luisaureliocasoni / teste-ead-gc](#)  0  0  0

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)


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

 Set up in Desktop or  HTTPS  SSH `https://github.com/luisaureliocasoni/teste-ead-gc.git` 

Get started by creating a new file or uploading an existing file. We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


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

```
echo "# teste-ead-gc" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/luisaureliocasoni/teste-ead-gc.git
git push -u origin master
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/luisaureliocasoni/teste-ead-gc.git
git push -u origin master
```



github-windows/openRepo/https://github.com/luisaureliocasoni/teste-ead-gc

Em seguida, abra a nossa pasta do repositório e abra um terminal. Vamos digitar dois comandos, sendo um para linkar o nosso repositório local ao remoto e o segundo para fazer o push, que é a nomenclatura que o Git dá para quando o programador faz o *check-in*.

```
git remote add origin https://github.com/  
luisaureliocasoni/teste-ead-gc.git
```

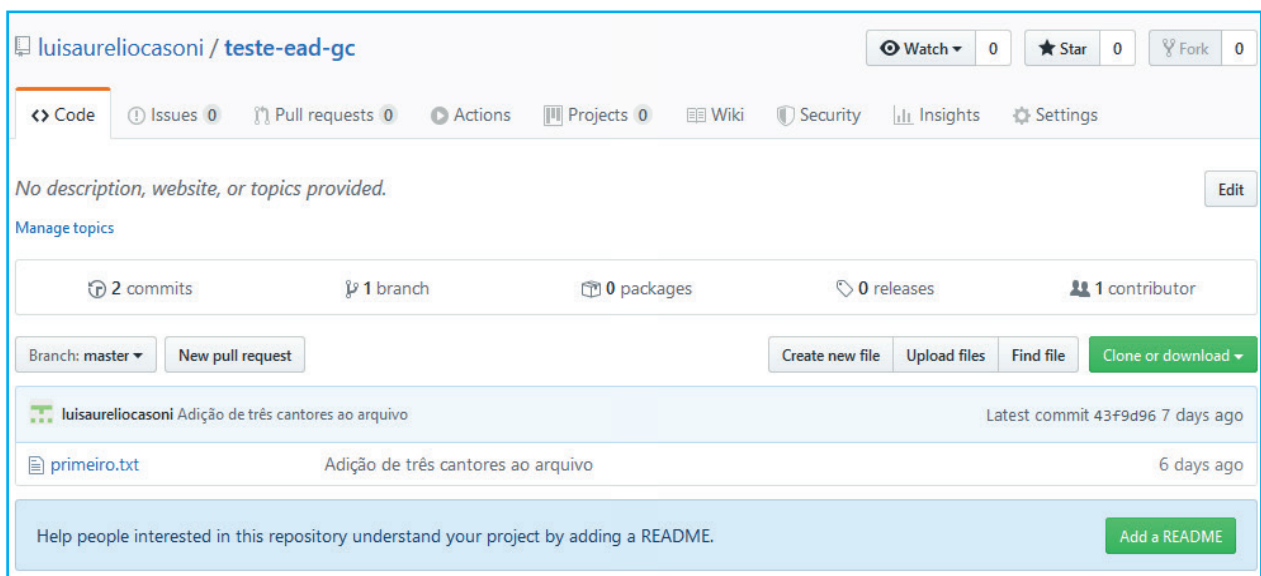
```
git push -u origin master
```

Lembrando que o primeiro comando depende da URL em que o repositório se encontra. Aguarde o final da operação.

Talvez o Git solicite as suas credenciais no Github, para fazer a operação.

```
PS D:\Desktop\primeiro-repositorio> git remote add origin https://github.com/luisaureliocasoni/teste-ead-gc.git  
PS D:\Desktop\primeiro-repositorio> git push -u origin master  
fatal: HttpRequestException encountered.  
Username for 'https://github.com': luisaureliocasoni  
Password for 'https://luisaureliocasoni@github.com':  
Counting objects: 6, done.  
Delta compression using up to 8 threads.  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (6/6), 597 bytes | 0 bytes/s, done.  
Total 6 (delta 0), reused 0 (delta 0)  
To https://github.com/luisaureliocasoni/teste-ead-gc.git  
* [new branch] master -> master  
Branch master set up to track remote branch master from origin.  
PS D:\Desktop\primeiro-repositorio>
```

Ao terminar, atualize a página do repositório em seu navegador. Veja que o nosso arquivo, em que editamos na aula 07, já está lá!



E, com isso, encerramos a nossa última aula. Não se esqueça de estudar os materiais complementares. E sempre tenha curiosidade! Existem muitas ferramentas que podem ser úteis na sua carreira. Sucesso!

Retomando a aula

Chegamos ao final da nossa última aula. Vamos relembrar?

1 – O que é o github

Você viu na nossa primeira aula que o GitHub é uma plataforma onde você pode armazenar seus projetos. É como se fosse uma rede social, só que de códigos, onde seus desenvolvedores podem disponibilizá-los para outras pessoas verem.

2 – Como criar o seu repositório no github

Nesta seção, você aprendeu como criar o seu repositório no Github e como fazer o *check-in* de seus commits presentes no repositório local para o repositório remoto.

Referências

- 4LINUX. O que é DevOps. s.d. Disponível em: <https://www.4linux.com.br/o-que-e-devops>. Acesso em 03 nov. 2019.
- APACHE. Apache Maven Project. Disponível em: <https://maven.apache.org/>. Acesso em 03 nov. 2019.
- AQUILLES, Alexandre; FERREIRA, Rodrigo. *Controlando versões com Git e GitHub*. São Paulo: Casa do Código, 2017.
- ATLASSIAN. Git Branch. s.d. Disponível em: <https://www.atlassian.com/br/git/tutorials/using-branches>. Acesso em 23 out. 2019.
- AWS. O que significa integração contínua? s.d. Disponível em: <https://aws.amazon.com/pt/devops/continuous-integration/>. Acesso em 03 nov. 2019.
- BERALDO, Roberto. Composer: Agilidade no Gerenciamento de Dependências com PHP. Profissionais TI, 2015. Disponível em: <https://www.profissionaisiti.com.br/2015/06/composer-agilidade-no-gerenciamento-de-dependencias-com-php/>. Acesso em 29 out. 2019.
- COMPOSER. Disponível em: <https://getcomposer.org/>. Acesso em 29 out. 2019.
- COMPOSER. Introduction. Disponível em: <https://getcomposer.org/doc/00-intro.md>. Acesso em 29 out. 2019.
- DANTAS, Cristine. Gerência de Configuração de Software. Devmedia, 2009. Disponível em: <https://www.devmedia.com.br/gerencia-de-configuracao-de-software/9145>. Acesso em: 23 out. 2019.
- DIOLINUX. Linux 5.2 chega com quase 600 mil novas linhas de código. 2019. Disponível em: <https://www.diolinux.com.br/2019/07/kernel-52-chega-com-quase-600-mil-novas-linhas-de-codigo.html>. Acesso em 19 nov. 2019.
- DUDLER, Roger. git - guia prático. 2019. Disponível em: https://rogerdudler.github.io/git-guide/index.pt_BR.html. Acesso em 19 nov. 2019.
- ENGHOLM JR., Hélio. *Engenharia de software na prática*. São Paulo: Novatec, 2010.
- FERNANDES, Diego. Git & Github: O que é? Por que? Como iniciar? Rocketseat, 2017. Disponível em: <https://blog.rocketseat.com.br/iniciando-com-git-github/>. Acesso em 19 nov. 2019.
- FREE SOFTWARE FOUNDATION. GNU RCS. Disponível em: <https://www.gnu.org/software/rcs/>. Acesso em 03 nov. 2019.
- GIT. .3 Primeiros passos - Noções Básicas de Git. Disponível em: <https://git-scm.com/book/pt-br/v1/Primeiros-passos-No%C3%A7%C3%B5es-B%C3%A1sicas-de-Git>. Acesso em 19 nov. 2019.
- GIT. Básico de Branch e Merge. s.d. Disponível em: <https://git-scm.com/book/pt-br/v1/Ramifica%C3%A7%C3%A3o-Branching-no-Git-B%C3%A1sico-de-Branch-e-Merge>. Acesso em: 23 out. 2019.
- GIT. Uma Breve História do Git. s.d. Disponível em: <https://git-scm.com/book/pt-br/v1/Primeiros-passos-Uma-Breve-Hist%C3%B3ria-do-Git>. Acesso em 19 nov. 2019.
- GIT. Uma Breve História do Git. s.d. Disponível em: <https://git-scm.com/book/pt-br/v1/Primeiros-passos-Uma-Breve-Hist%C3%B3ria-do-Git>. Acesso em: 03 nov. 2019.
- GITHUB. The State of the Octoverse. Github, 2019. Disponível em: <https://octoverse.github.com/>. Acesso em 19 nov. 2019.
- GROW UNDER. Git: manual de utilização e vantagens e desvantagens. 2016. Disponível em: <https://www.growunder.com/pt/blog/dicas/65-git-como-funciona-e-quais-as-vantagens-e-desvantagens>. Acesso em 19 nov. 2019.
- GUEDES, Marylene. Git e GitHub: quais as diferenças? TreinaWeb, 2019. Disponível em: <https://www.treinaweb.com.br/blog/git-e-github-quais-as-diferencas>. Acesso em 19 nov. 2019.
- H., Rafael. Tutorial do GIT Básico – Introdução ao GIT. Hostinger, 2019. Disponível em: https://rogerdudler.github.io/git-guide/index.pt_BR.html. Acesso em 19 nov. 2019.
- IEEE COMPUTER SOCIETY. *IEEE Standard for Configuration Management in Systems and Software Engineering*. New Jersey, p. 71. 2012.
- JANUSKA, Antonin. My Personal Git Tricks Cheatsheet. DEV, 2019. Disponível em: <https://dev.to/antjanus/my-personal-git-tricks-cheatsheet-23j1>. Acesso em 19 nov. 2019.
- JENKINS. Disponível em: <https://jenkins.io/>. Acesso em 03 nov. 2019.
- LWN. Git 1.0.0 released. 2005. Disponível em: <https://lwn.net/Articles/165127/>. Acesso em 19 nov. 2019.
- MATOLA, Rodrigo. Github: mostre o seu trabalho e seja contratado!. Geekhunter, 2017. Disponível em: <https://blog.geekhunter.com.br/github-uma-oportunidade-de-mostrar-o-seu-trabalho-e-ser-contratado/>. Acesso em 19 nov. 2019.
- OMBREDANNE, Philippe. Software Dependencies: A not-too-technical introduction. NEXB, 2016. Disponível em: https://www.nexb.com/blog/introduction_to_software_dependencies.html. Acesso em 29 out. 2019.
- ORACLE. Chapter 5 SCCS Source Code Control System. s.d. Disponível em: <https://docs.oracle.com/cd/E19504-01/802-5880/6i9k05dhp/index.html>. Acesso em 03 nov. 2019.
- OTTERO, Rodrigo. Introdução ao Maven. DevMedia, 2012. Disponível em: <https://www.devmedia.com.br/introducao-ao-maven/25128>. Acesso em 03 nov. 2019.
- PACKAGIST. Disponível em: <https://packagist.org/>. Acesso em 29 out. 2019.
- PACKAGIST. Statistics. Disponível em: <https://packagist.org/statistics/>. Acesso em 29 out. 2019.
- PRESTON-WERNER, Tom. Versionamento Semântico 2.0.0. 2014. Disponível em: <https://semver.org/lang/pt-BR/>. Acesso em 18 nov. 2019.
- ROCHA, Fábio Gomes. Integração contínua: uma introdução ao assunto. DevMedia, 2013. Disponível em: <https://www.devmedia.com.br/integracao-continua-uma-introducao-ao-assunto/28002>. Acesso em 03 nov. 2019.
- SCHMITZ, Daniel. Tudo que você queria saber sobre Git e GitHub, mas tinha vergonha de perguntar. Tableless, 2015. Disponível em: <https://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-github-mas-tinha-vergonha-de>

perguntar/. Acesso em 19 nov. 2019.

SINK, Eric. Version Control by Example. 2011. Disponível em: <https://ericsink.com/vcbe/html/index.html>. Acesso em: 03 nov. 2019

SOMMERVILLE, Ian. *Engenharia de Software*. 9. ed. São Paulo: Pearson, 2011.

TRAVIS CI. Disponível em: <https://travis-ci.org/>. Acesso em 03 nov. 2019.

VIEIRA, Rafael Nunes. Gerenciadores de dependências. Medium, 2016. Disponível em: <https://medium.com/@nvieirarafael/gerenciadores-de-depend%C3%Aancias-bed99b01e94a>. Acesso em 29 out. 2019

WARNER. Thank you for 100 million repositories. Github, 2018. Disponível em: <https://github.blog/2018-11-08-100m-repos/>. Acesso em 19 nov. 2019.

Minhas anotações