# Tema 03 – Tutorial GitHub Desktop

## Introdução

Olá meus querides alunes, sejam-bem vindes à nossa aula remota de Programação Web do Instituto da Oportunidade Social. Nessa aula, você vai ver um tutorial de como utilizar o GitHub Desktop para criar repositórios e fazer upload dos seus códigos no GitHub.

## Sobre o GitHub

O GitHub é um repositório que fornece a hospedagem para o desenvolvimento de software e controle de versão usando Git. Ele oferece funcionalidades do Git como: controle distribuído de versões e um gerenciamento de código-fonte (SCM, *Source Code Management*) e, também, funcionalidades próprias como: rastreamento de erros, controle de acesso de colaboração, requisição de recursos, gerenciamento de tarefas, wikis, etc.

Ele foi criado por Chris Wanstrath, P. J. Hyett, Tom Preston-Werner e Scott Chacon usando o framework de aplicações web Ruby on Rails em fevereiro de 2008, and started in February 2008. Juntos, eles criaram a empresa GitHub Inc., que existia desde 2007 em São Francisco. Desde 2012, a Microsoft era um grande usuário do GitHub e em outubro de 2018, ela anunciou a compra da empresa GitHub Inc., que gerencia o repositório.

O site para você criar uma conta no GitHub (Figura 1) é: <https://github.com/>, para criar sua conta você deve clicar em **Sign up**.

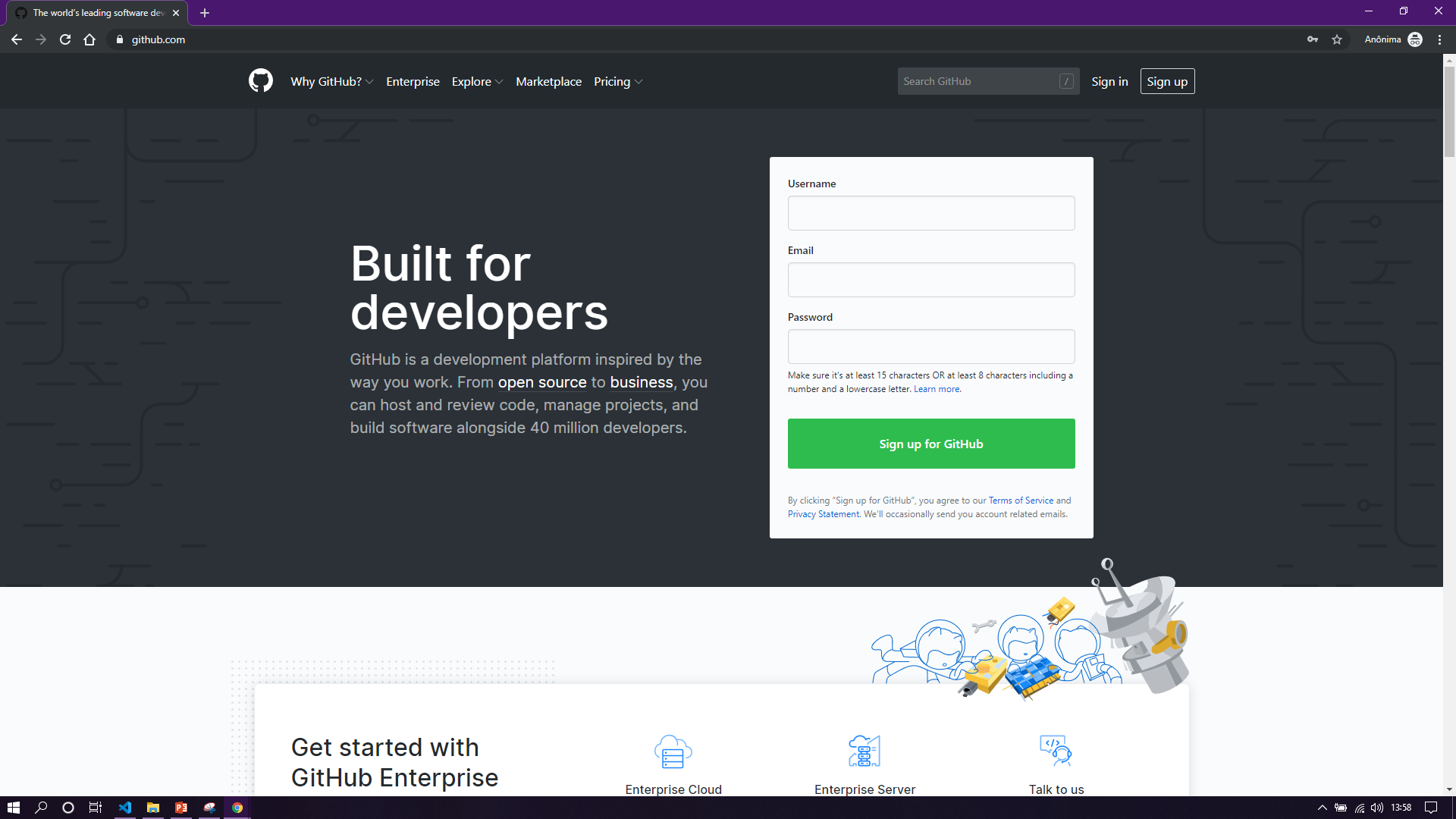


Figura 1. Site do GitHub.

Você deverá preencher os campos da Figura 2: Nome do Usuário (Username), endereço de e-mail (Email address) e senha (Password). O campo Nome do Usuário deve ser um nome inédito, que não existe no GitHub (O próprio GitHub faz essa verificação). o endereço de e-mail deve ser um e-mail válido e ativo e a senha deve conter pelo menos 15 caracteres ou 8 caracteres incluindo um número e letras maiúsculas. Com tudo preenchido, faça a verificação de pessoa, para ter certeza de que você não é um bot e clique no botão **Create account**.

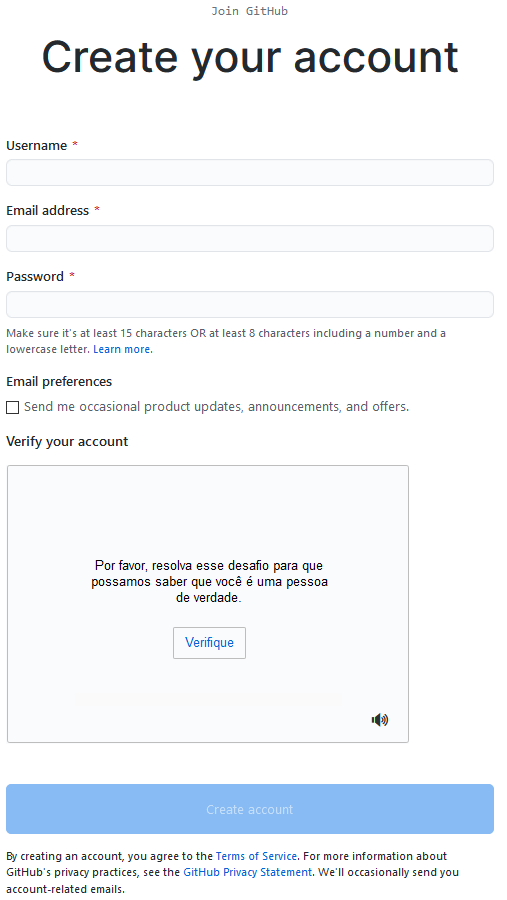


Figura 2. Campos para criação de conta no GitHub.

Na próxima página, O GitHub solicitará a você escolher o plano de subscrição (Figura 3). As opções disponíveis são:

* Free: repositórios públicos e privados ilimitados, até três colaboradores, fórum de problemas (issues), monitoramento de bugs e ferramentas de gerenciamento de projetos.
* Pro: acesso ilimitado a todos os repositórios, número ilimitado de colaboradores, fórum de problemas (issues), monitoramento de bugs e ferramentas avançadas.
* Team: todas as funcionalidades mencionadas anteriormente, controle de acesso de equipe e gerenciamento de usuários.
* Enterprise: todas as funcionalidades do plano Team, hospedagem própria ou em nuvem, prioridade na prestação de suporte técnico, login único (single sign-on) e mais.

Para um usuário inicial e não coorporativo é melhor optar pela opção gratuita.

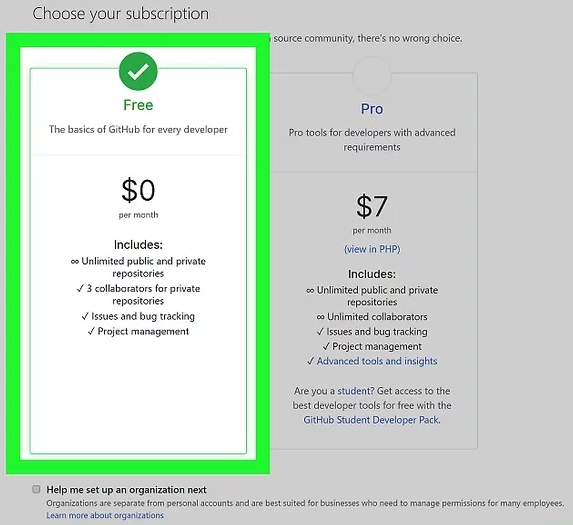


Figura 3. Opção de planos do GitHub.

Na página seguinte, você deverá completar seu cadastro, clicando em um botão para verificar seu endereço de e-mail (Figura 4).

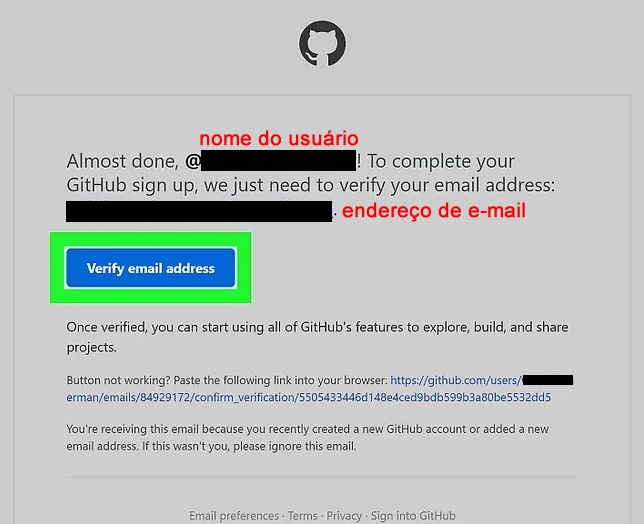


Figura 4. Verificação do E-mail.

Você receberá um e-mail de confirmação e assim que você realizar essas operações, você deverá selecionar as suas preferências ou pular esse passo (Figura 5). Então, seu cadastro estará finalizado.

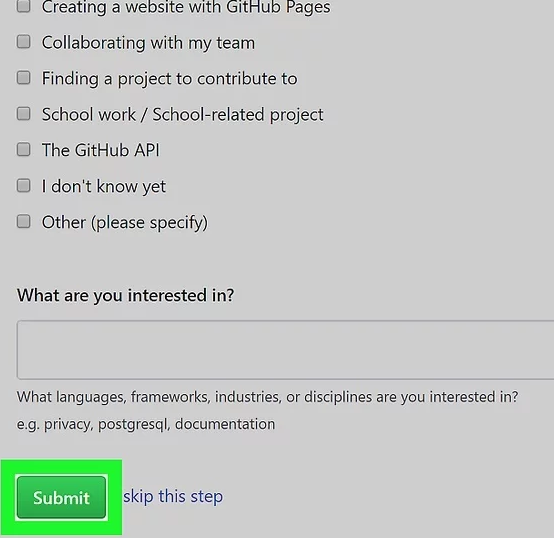


Figura 5. Preferências do Usuário.

## Instalando o Git em seu computador

Essa instalação é necessária se você quiser usar o VS Code para gerenciar os seus arquivos no GitHub. Caso você queria usar apenas o GitHub Desktop, que é uma ferramenta mais simples para iniciantes, pode saltar para a seção 3 dessa aula. Siga os passos para instalar o Git no seu computador com sistema operacional Windows.

1. Acesse o site: <https://git-scm.com/> e faça o download da versão mais atual do Git.
2. Execute o arquivo, que você baixou. No Windows, aparecerá a janela **Information** mostrada na Figura 6. Clique no botão **Next**.

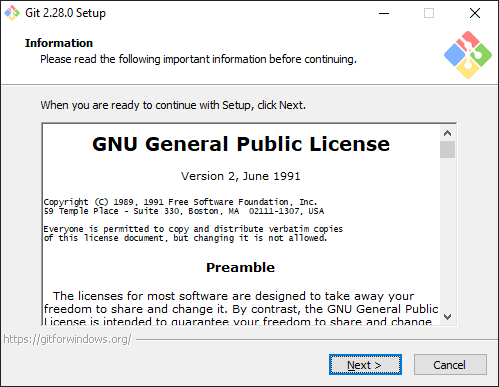


Figura 6. Instalação do Git no Windows.

1. Na janela **Select Destination Location**, selecione o diretório de instalação (Figura 7), você pode deixar como o diretório recomendado ou alterar se desejar. Clique no botão **Next**.

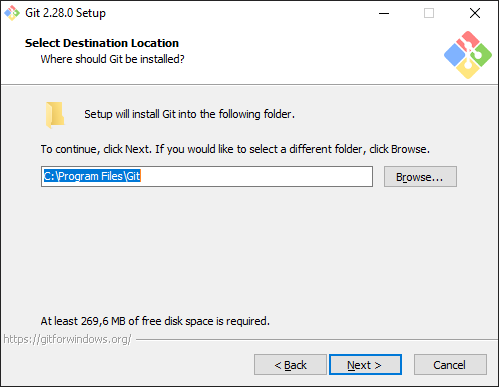


Figura 7. Seleção do diretório de instalação.

1. Na janela **Select Components**, marque os componentes que deseja instalar (Figura 8). Clique no botão **Next**.

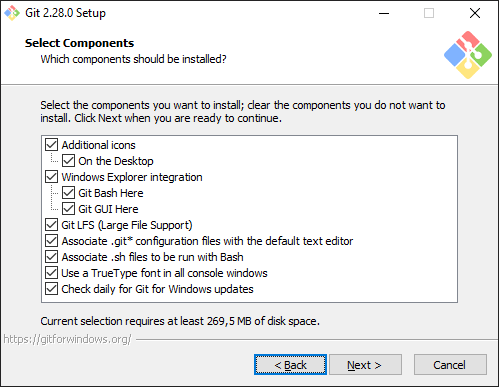


Figura 8. Escolha dos componentes para serem instalados.

1. Na janela **Select Star Menu Folder**, escolha se você quer criar um atalho no menu Iniciar (Figura 9). Clique no botão **Next**.

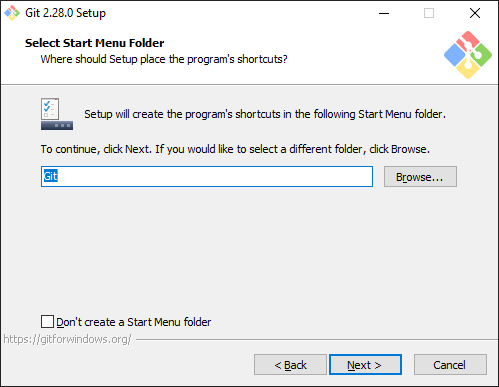


Figura 9. Criar atalho no Menu Iniciar.

1. Na janela **Choosing the default editor used by Git**, selecione o **Visual Studio Code** como seu editor padrão (Figura 10). Clique no botão **Next**.

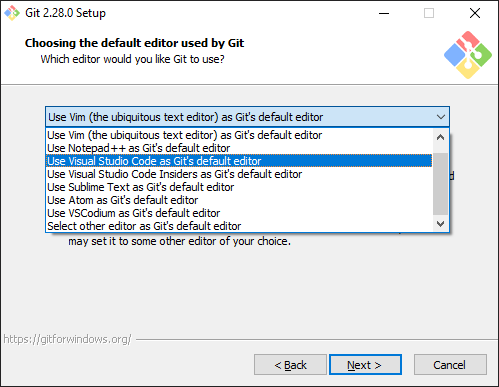


Figura 10. Escolha do Editor Padrão.

1. Na janela **Adjusting your Path enviroment**, deixe as opções padrão selecionadas (Figura 11) e clique no botão **Next**.

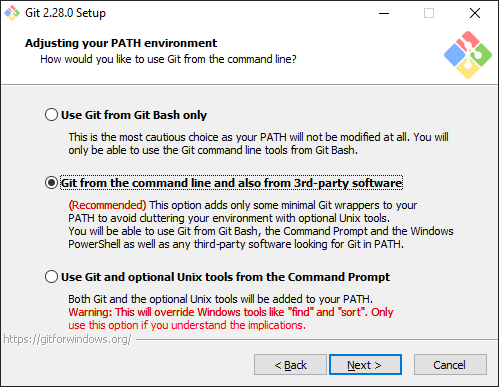


Figura 11. Ajuste do caminho no ambiente de desenvolvimento.

1. Na janela **Choosing HTTPS transport backend**, deixe as opções padrão selecionadas (Figura 12) e clique no botão **Next**.

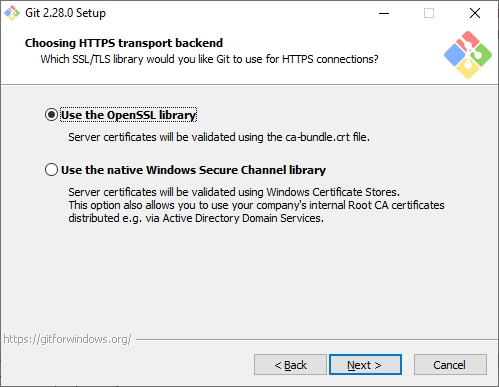


Figura 12. Escolha do backend para o transporte de HTTPs.

1. Na janela **Configuring the line ending conversions**, deixe as opções padrão selecionadas (Figura 13) e clique no botão **Next**.

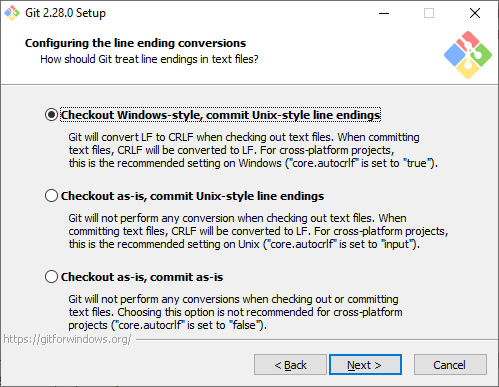


Figura 13. Configuring the line ending conversions.

1. Na janela **Configuring the terminal emulator to use with Git Bash**, deixe as opções padrão selecionadas (Figura 14) e clique no botão **Next**.

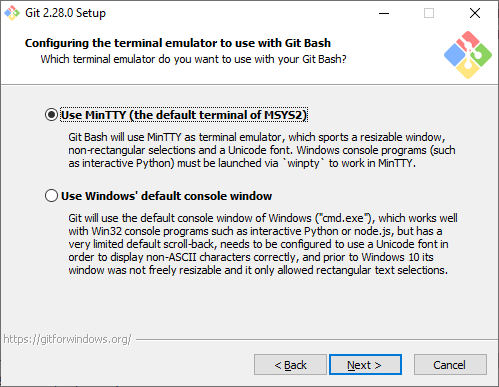


Figura 14. Configuring the terminal emulator to use with Git Bash.

1. Na janela **Choose the default behavior of ‘git pull’**, deixe as opções padrão selecionadas (Figura 15) e clique no botão **Next**.

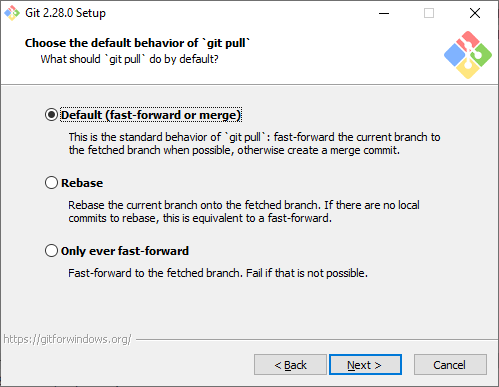


Figura 15. Choose the default behavior of ‘git pull’.

1. Na janela **Choose a credential helper**, deixe as opções padrão selecionadas (Figura 16) e clique no botão **Next**.

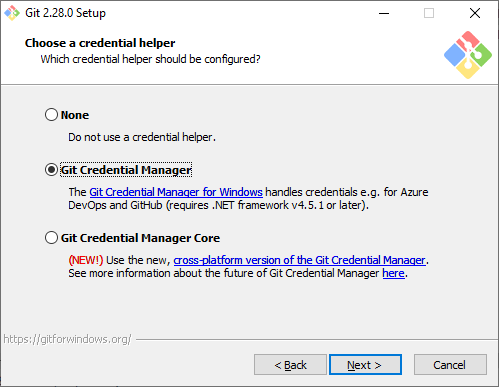


Figura 16. Choose a credential helper.

1. Na janela **Configuring extra options**, deixe as opções padrão selecionadas (Figura 17) e clique no botão **Next**.

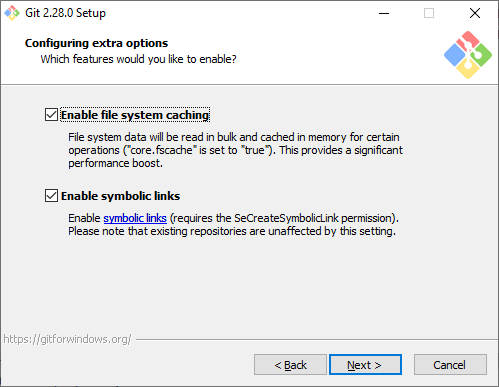


Figura 17. Configuring extra options.

1. Na janela **Configuring experimental options**, deixe as opções padrão selecionadas (Figura 18) e clique no botão **Next**.

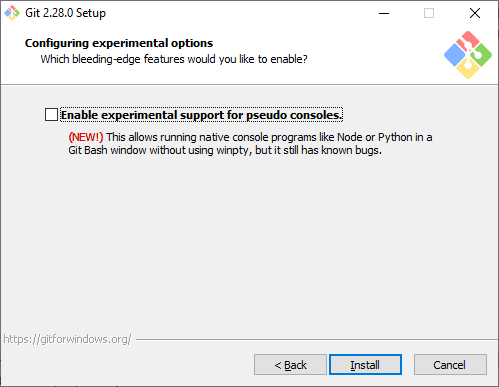


Figura 18. Configuring experimental options.

1. Na janela **Completing the Git Setup Wizard**, deixe as opções padrão selecionadas (Figura 19) e clique no botão **Next**.

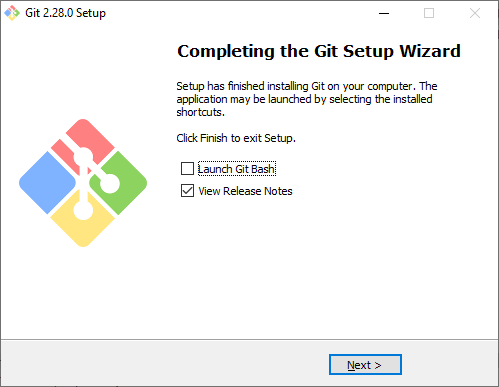


Figura 19. Completing the Git Setup Wizard.

1. Agora está parcialmente certo para você usar o VS Code para gerenciar seus arquivos no GitHub.

## Tutorial do GitHub Desktop

Uma vez que você já possui uma conta ativa e tem o sistema operacional Windows ou MacOS instalado no seu computador, é possível utilizar o GitHub Desktop (Figura 14) para gerenciar, por meio de um ambiente gráfico seus arquivos e repositórios no GitHub. Ele te dá acesso a todos os recursos e ferramentas disponíveis no GitHub e é uma solução mais amigável para um iniciante em Git.

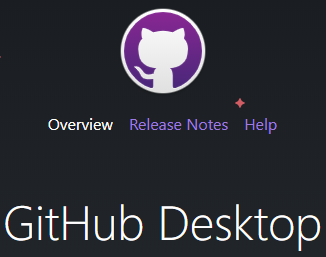


Figura 20. GitHub Desktop.

Para utilizar o GitHub desktop, você deverá seguir as seguintes etapas:

### Etapa 1. Instalar e entrar no GitHub Desktop.

1. Primeiro, você precisa baixar e instalar o software: <https://desktop.github.com/>
   1. O GitHub Desktop dá suporte às versões mais recentes do Windows e do MacOS.
2. Ao iniciar a instalação, você poderá realizar o login com sua conta, criar nova conta ou pular esse passo (Figura 3).

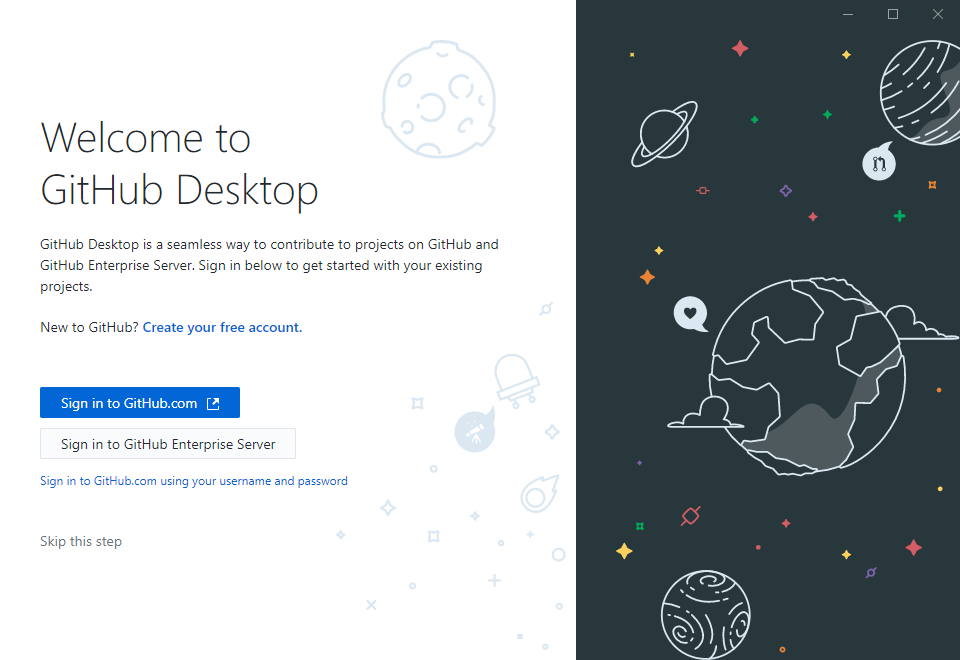


Figura 21. Tela inicial de instalação.

1. Como nós já criamos uma conta anteriormente, clique em Sign in to GitHub.com e faça o login na sua conta.
2. Você será direcionado para o site, realizará o login a sua conta do site e clique no botão de **Authorize desktop** (Figura 16).

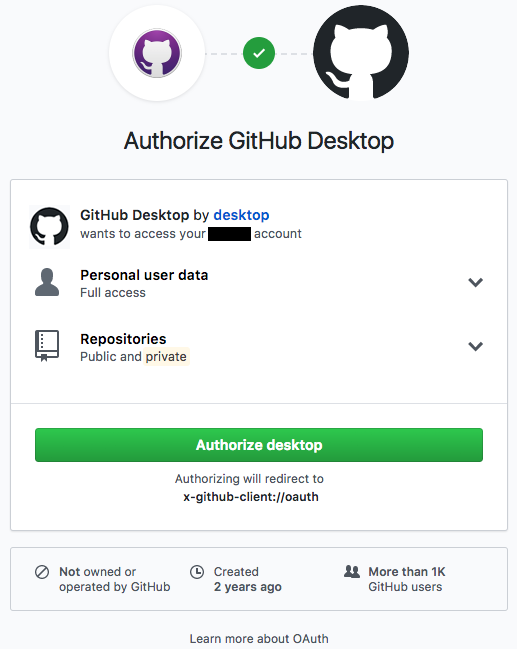


Figura 22. Autorização de acesso do GitHub Desktop.

1. Na janela *Configure Git*, você deve preencher/confirmar os seus dados Name e Email e clicar no botão **Continue**.

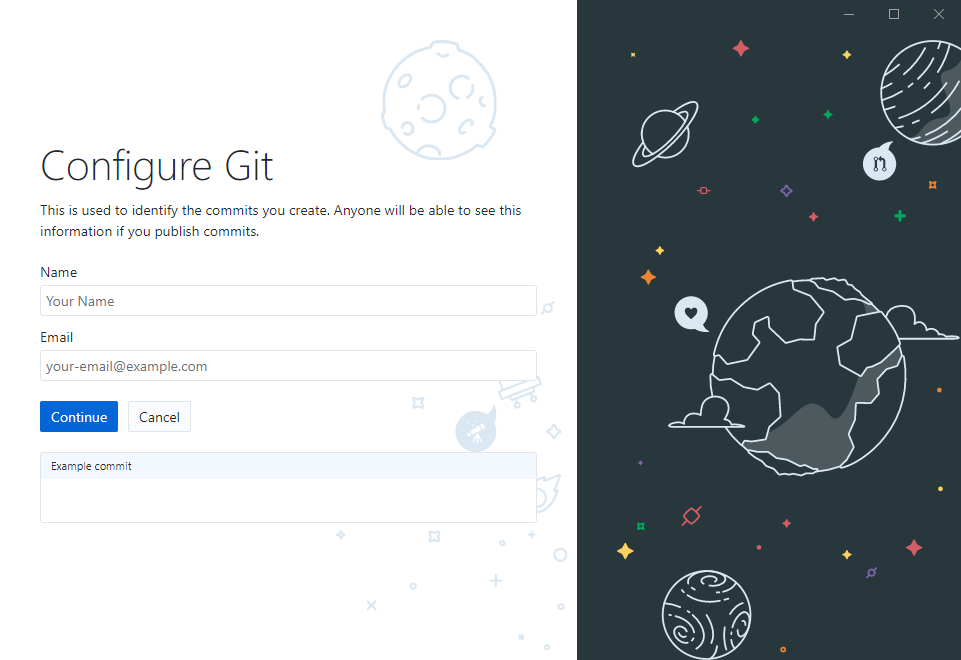


Figura 23. Janela Configure Git.

1. Na janela final, você pode optar ou não por enviar dados estatísticos para o GitHub e deverá clicar no botão **Finish**.

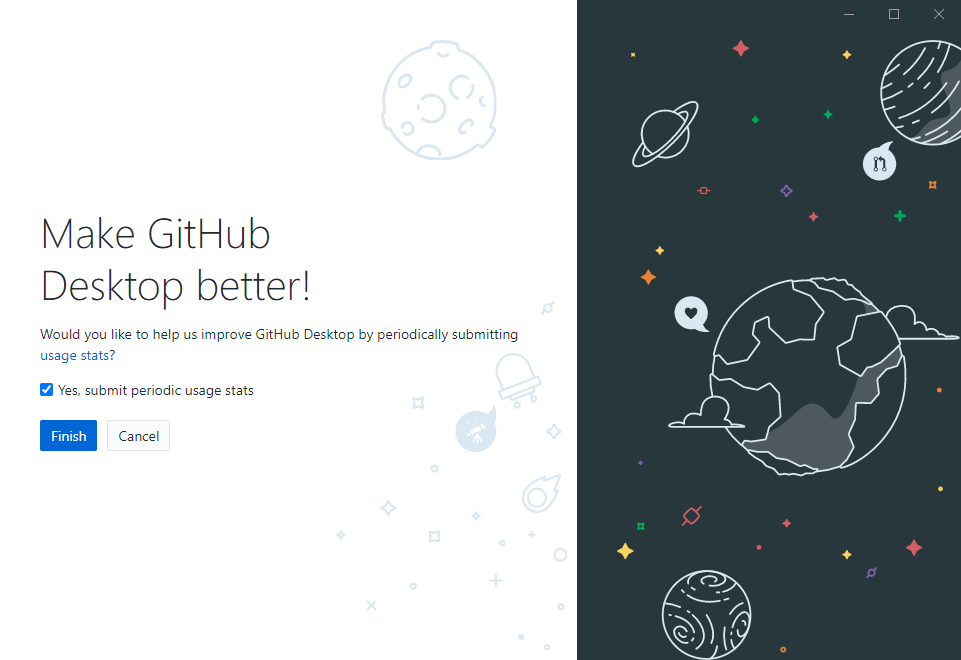


Figura 24. Janela de finalização da instalação/configuração.

### Etapa 2. Criar um repositório.

1. Após a instalação e a configuração do GitHub Desktop, a interface Let's get started!" (Vamos começar!) do GitHub Desktop (Figura 19) irá ser aberta automaticamente (Caso não ocorra, você poderá fazer isso manualmente pelo atalho criado na área de trabalho). Nessa tela, é possível optar por:
   1. criar e clonar um repositório de tutorial;
   2. clonar um repositório existente;
   3. criar novo repositório no seu disco rígido;
   4. adicionar um repositório existente do seu disco rígido.

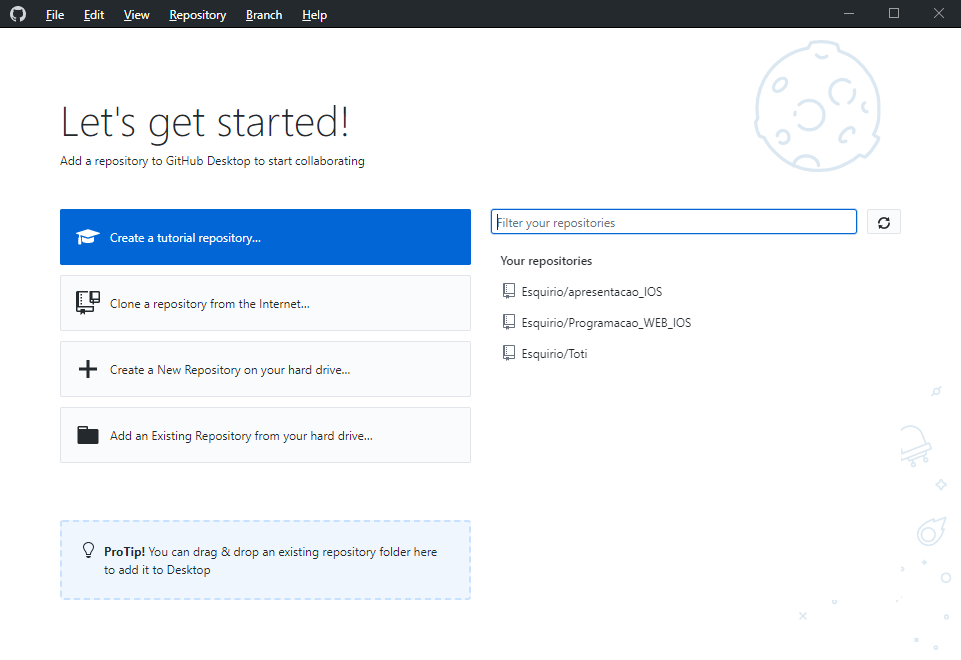
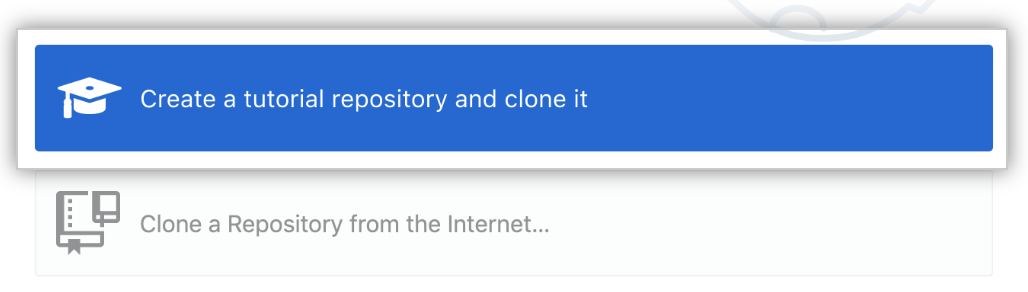


Figura 25. Interface GitHub Desktop.

1. Você pode criar um repositório pelo tutorial do GitHub. Para isso, clique em **Create a tutorial repository and clone it** (Criar um repositório de tutorial e cloná-lo).



* 1. Um tutorial será aberto para ajudar você a criar o seu repositório (Figura 20). Clique no botão **Continue**.

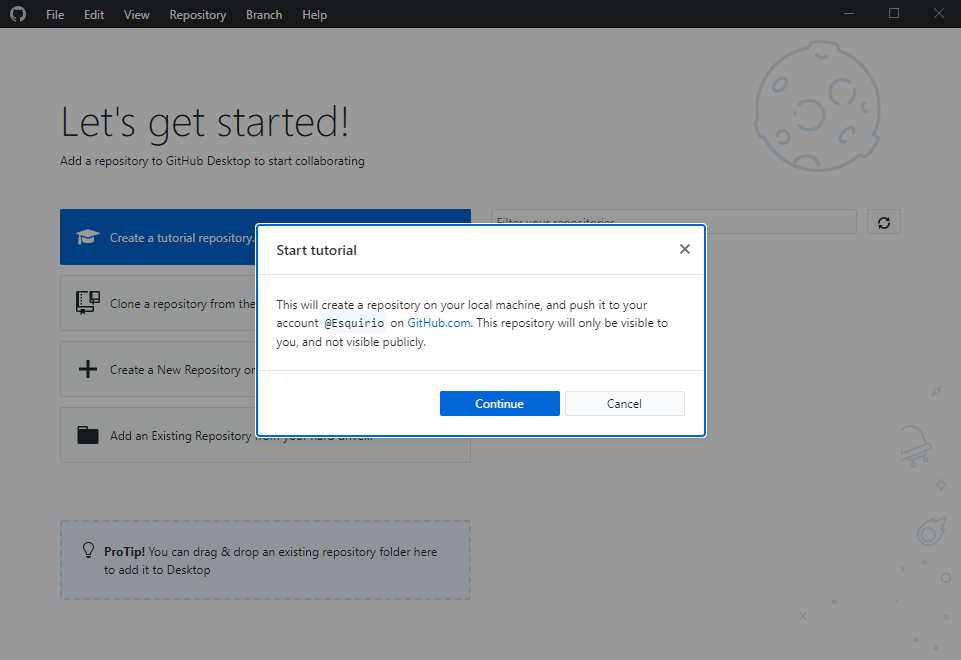


Figura 26. Tutorial para criar um novo repositório.

* 1. Você verá que o GitHub Desktop criará um repositório privado com o nome desktop-tutorial (Figura 21). Esse repositório criado já estará disponível na Nuvem, ou seja, no seu GitHub online, mas somente você tem o acesso a ele nesse momento.

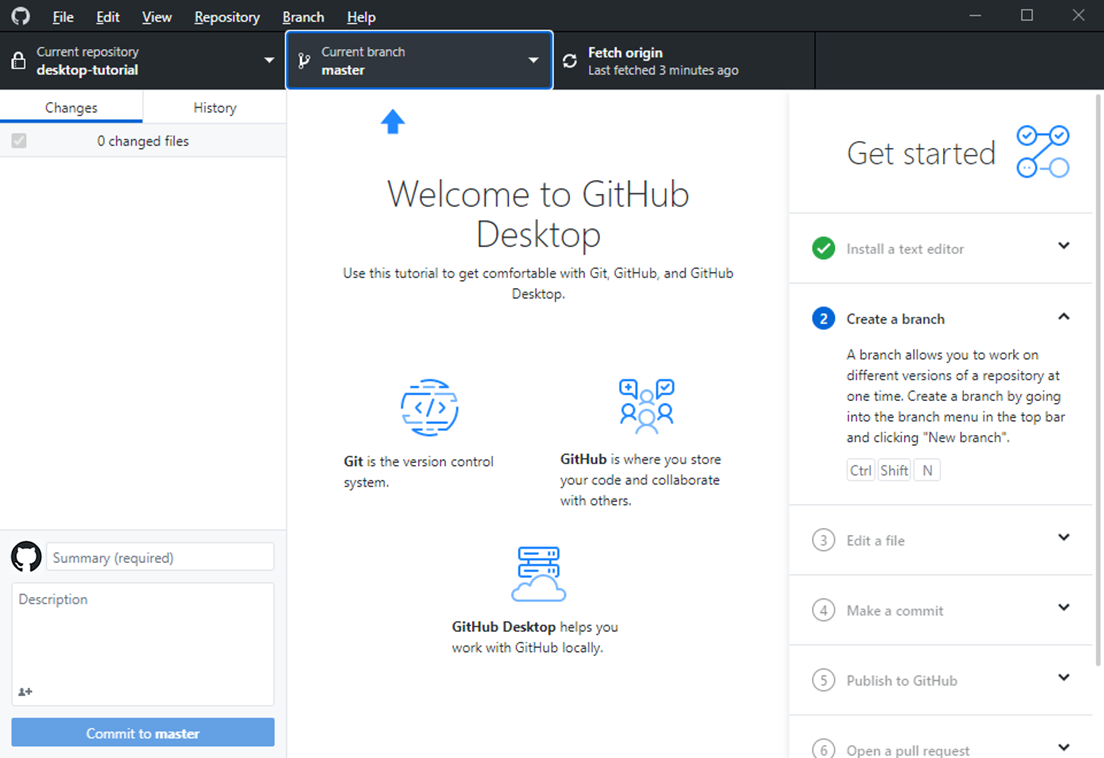


Figura 27. Repositório desktop-tutorial.

1. Você pode também criar um repositório no seu HD, para isso clique em **Create a New Repository on your hard drive** (Criar um repositório no disco rígido) ou acesse o Menu File e escolha a opção **New Repository** (Figura 22).

|  |  |
| --- | --- |
| Criar um repositório |  |

Figura 28. Criar um novo repositório no HD.

* 1. Nesse caso, o GitHub Desktop deixará você editar as opções (Figura 23):
     + Name: define o nome do repositório local (no seu HD). O nome do repositório não pode ter espaços e nem caracteres especiais.
     + Description: é um campo opcional com mais informações sobre o propósito do repositório.
     + Local path: define o local do repositório no computador. Você pode deixar o padrão dentro do diretório Documentos ou escolher outro local para armazenar seus arquivos.
     + Se você deseja inicializar o repositório com um arquivo README (leia-me). Arquivos LEIAME ajudam as pessoas a entenderem o objetivo do seu projeto.
     + Git ignore: permite incluir um arquivo personalizado para ignorar determinados arquivos no repositório local, isto é, informações que você não quer armazenar no controle de versão.
     + License: permite adicionar uma licença de código aberto a um arquivo LICENSE no repositório.

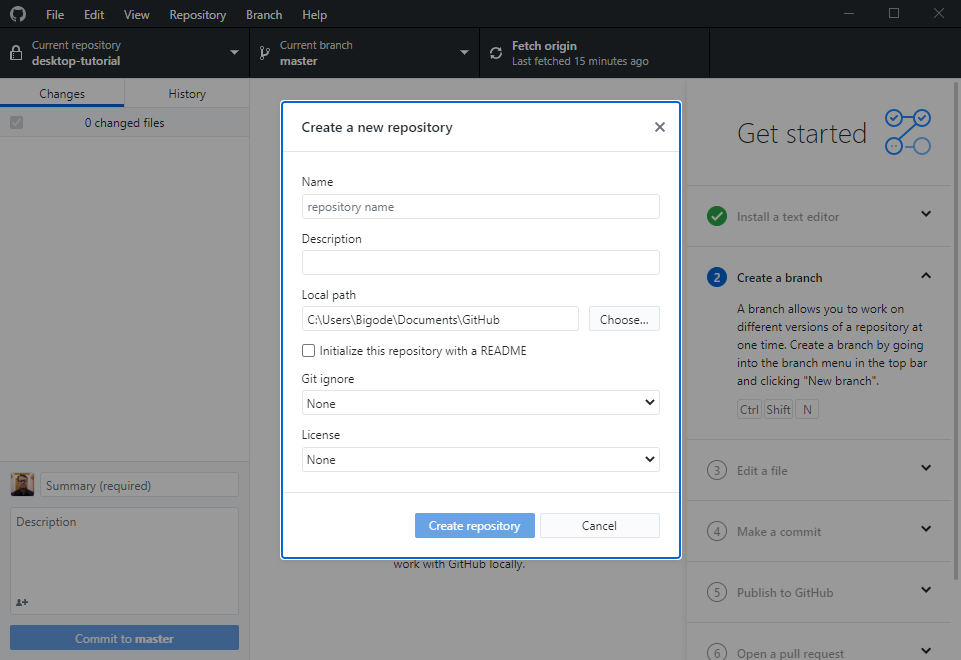


Figura 29. Campos para criação de novo repositório.

* 1. Preencha os campos como desejar e clique em **Create repository**.
  2. Se você quiser publicar o seu repositório criado no HD, pois até então ele está apenas disponível no seu computador local. Então, você precisará clicar no botão **Publish repositor**.

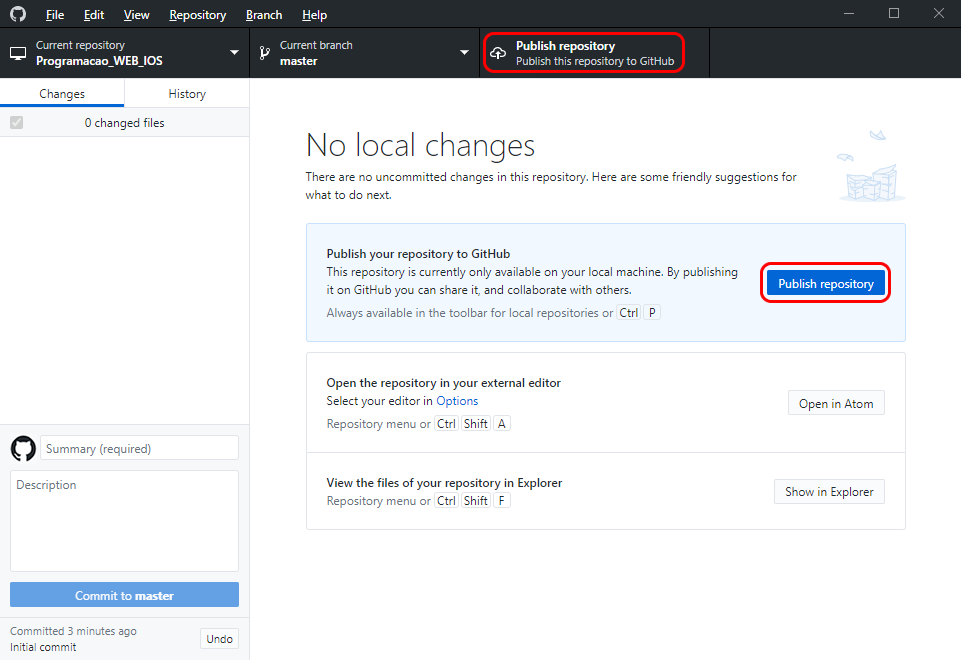


Figura 30. Publicar seu repositório no GitHub.

* 1. Uma mensagem para publicação do repositório irá aparecer, confira as informações e estando tudo certo, clique em **Publish repositor** (Figura 25).

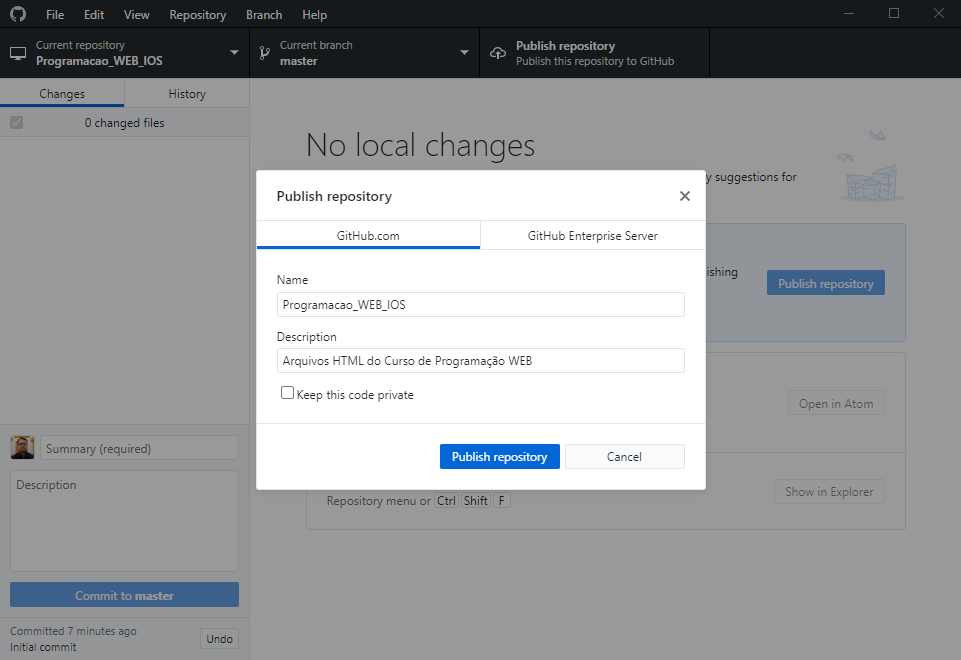


Figura 31. Configurações da publicação do Repositório.

* 1. No curso, vamos optar por deixar o repositório público, pois assim os alunos poderão acessar e pegar os arquivos do curso. Assim, deixaremos . O endereço do repositório é: <https://github.com/Esquirio/Programacao_WEB_IOS>

### Etapa 3. Explorar o GitHub Desktop

1. Se você completou a Etapa 2, você tem agora dois repositórios (Figura 26): o desktop-tutorial criado pelo tutorial e o segundo repositório (no exemplo Programacao\_IOS) criado e publicado manualmente.

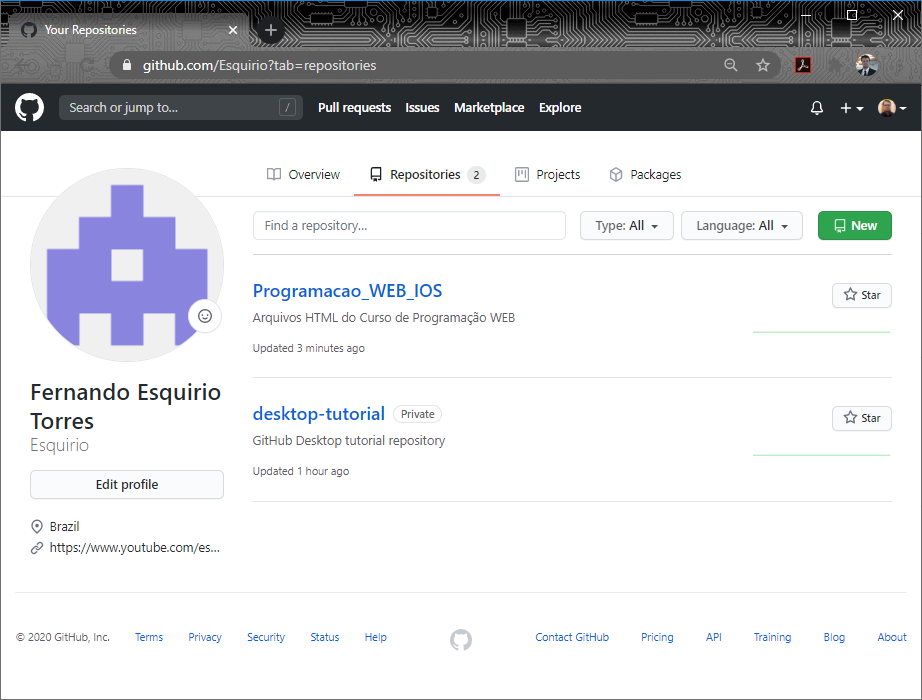


Figura 32. Repositórios disponíveis no GitHub.

1. Abaixo do menu, uma barra de opções do repositório atual com o estado do mesmo no GitHub Desktop (Figura 27). Como você pode observar, a barra possui três opções:
   1. Current repository (Repositório atual): mostra o nome do repositório em que você está trabalhando. Você pode clicar em Current repository (Repositório atual) para alternar entre repositórios no GitHub Desktop.
   2. Current branch (Branch atual): mostra o nome do branch em que você está trabalhando. Você pode clicar em Current branch (Branch atual) para exibir todos os branches do repositório, alternar entre branches ou criar um branch. Depois de criar pull requests no repositório, você também poderá exibi-las clicando em Current branch (Branch atual).
   3. A opção Fetch origin (Atualizar repositório remoto): busca as modificações no diretório remoto.

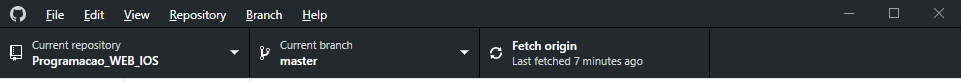


Figura 33. Barra de opções do repositório corrente.

1. Na barra lateral à esquerda, você verá Changes (Alterações) e  History (Histórico) (Figura 28).
   1. A opção Changes (Alterações): mostra as mudanças que você fez nos arquivos do branch atual, mas que ainda estão sem commit no repositório local. Na parte inferior, você verá as caixas "Summary" (Resumo) e "Description" (Descrição), além do botão Commit to master (Fazer commit para o mestre).
   2. É nessa área que você fará o commit das novas alterações.
   3. O botão Commit (Fazer commit) mostra para qual branch você fará o commit das alterações.

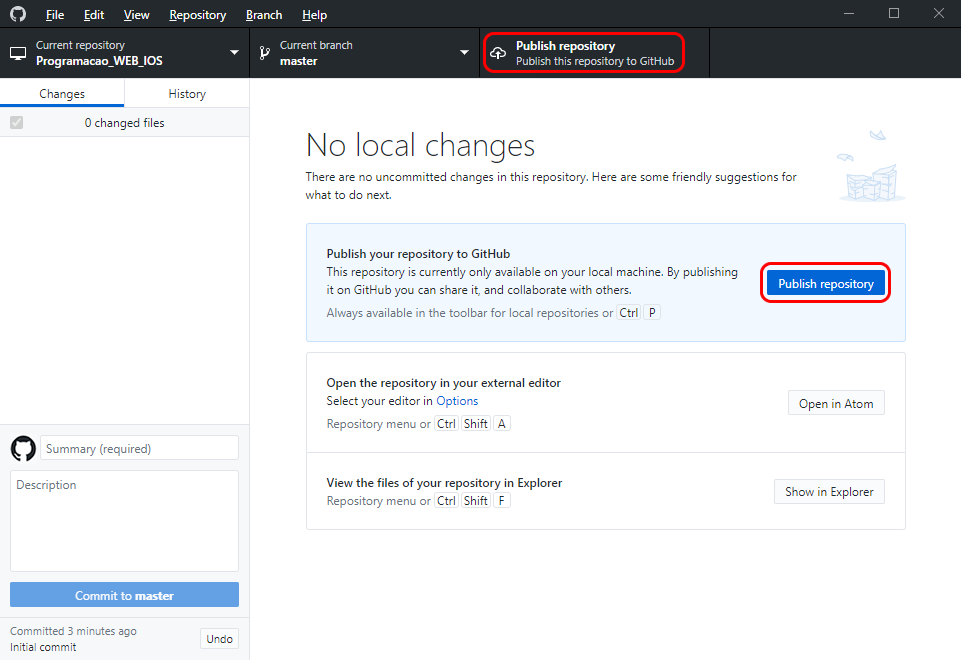


Figura 34. Barra lateral com as opções de Change e History.

* 1. A opção History (Histórico, Figura 29): mostra os commits anteriores no branch atual do repositório. Provavelmente você verá um "Initial commit" (Commit inicial) criado pelo GitHub Desktop quando você criou o repositório.
  2. À direita do commit, dependendo das opções escolhidas durante a criação do repositório, você poderá ver arquivos .gitattributes, .gitignore, LICENÇA ou LEIAME. Ao clicar em cada arquivo você verá o diff, que consiste no registro das alterações feitas no arquivo do commit em questão.
  3. O diff não mostra todo o conteúdo do arquivo, mas somente as partes que foram alteradas.

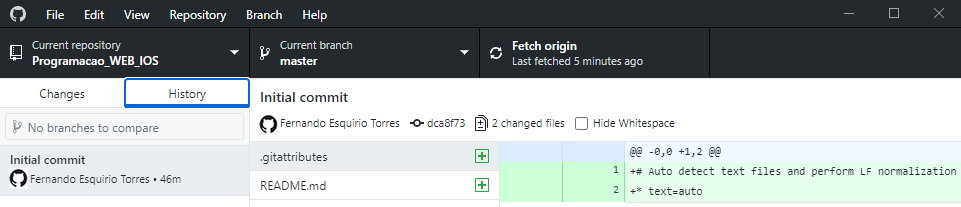


Figura 35. Opção History.

### Etapa 4. Visualizar repositório remoto no GitHub

1. Você pode visualizar o seu repositório remoto, basta acessar o Menu **Repository** e selecionar a opção **View on GitHub** (Figura 30).

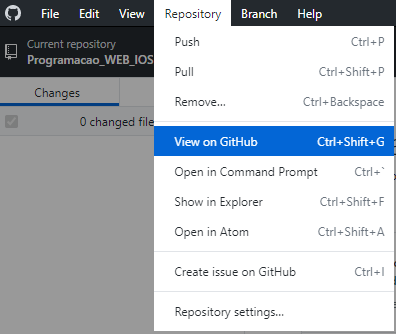


Figura 36. Visualizar o repositório remoto no GitHub.

### Etapa 5. Configurar um editor de texto

1. A Figura 31 mostra como acessar as configurações para escolher um editor de texto. Clique em **File** (Arquivo) e selecione a opção **Options** (Opções).

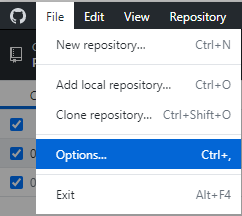


Figura 37. Acesso as opções do GitHub Desktop.

1. Use o menu lateral **Integrations** e selecione um editor externo. A lista mostrará todos os editores instalados (Figura 32). Se a lista estiver vazia, instale um editor compatível, como o Atom, Notepad++, Visul Studio Code, etc e refaça os passos da Etapa 5. Para obter uma lista dos editores compatíveis, consulte o site: <https://docs.github.com/pt/desktop/installing-and-configuring-github-desktop/configuring-a-default-editor>.

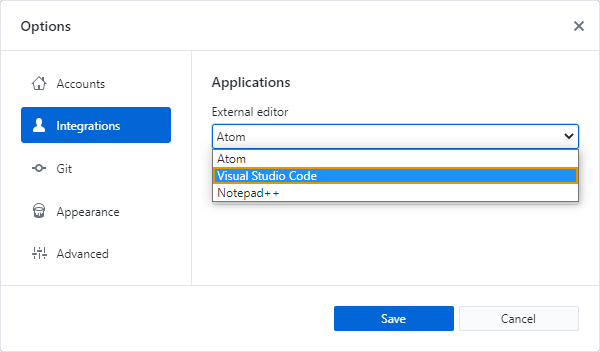


Figura 38. Lista de editores instalados no computador.

### Etapa 6. Fazer commit e push das alterações:

1. Para iniciar o editor externo pelo GitHub Desktop, clique em **Repository** e em Open in **NOME\_DO\_EDITOR** (Figura 33).

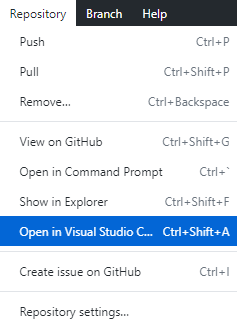


Figura 39. Iniciar um Editor Externo no GitHub Desktop.

* 1. Assim que você abrir o Visual Studio Code, irá aparecer na janela lateral esquerda do programa o seu repositório (Figura 34a). Se aparecer uma mensagem no canto inferior direito para você instalar a extensão do git.path (Figura 34b) significa que você não seguiu corretamente a seção de instalação do Git em seu computador.

|  |  |
| --- | --- |
|  |  |

Figura 40. (a) Janela lateral com o reposítório e (b) mensagem de aviso para a instalação do git.path.

1. Se você não tiver nenhum arquivo no seu repositório local, comece fazendo alterações no arquivo README.md. Adicione informações que descrevam o projeto, como a função, a relevância e a utilidade dele. Lembre-se de que esta será a primeira interação das pessoas com o seu projeto. Depois disso, estará tudo pronto para fazer o seu primeiro commit!
2. Saia do editor de texto, volte para o GitHub Desktop e navegue até a guia Changes (Figura 35). Na lista de arquivos, você verá o README.md alterado. A marca de verificação no arquivo README.md indica que as alterações feitas serão parte do seu commit. Talvez você queira fazer alterações em vários arquivos no futuro, mas sem fazer o commit das alterações de todos eles. O GitHub Desktop permite selecionar somente as alterações que você pretende incluir no commit.

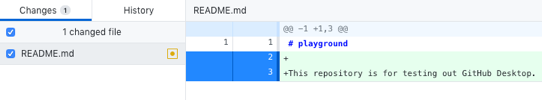


Figura 41. GitHub Desktop mostrando que existem mudanças no repositório local.

1. Na parte inferior da lista Changes (Alterações), adicione uma mensagem ao commit. À direita da sua foto de perfil, digite uma breve descrição do commit. Já que estamos alterando o arquivo README.md, algo como "Adicionar informações sobre o propósito do projeto" seria um bom resumo.

Abaixo do resumo, o campo de texto "Description" (Descrição) permite digitar uma descrição mais longa das alterações feitas no commit.

Essa descrição pode ser útil para analisar o histórico de um projeto e entender o motivo das alterações. Como estamos fazendo uma atualização básica do arquivo README.md, fique à vontade para ignorar a descrição.

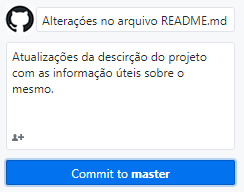


Figura 42. Resumo e descrição das alterações no repositório local.

1. Clique em **Commit to master**. O botão do commit mostra o branch atual (mestre, neste caso) para que você tenha certeza de fazer commit para o branch desejado.
2. Para fazer **push** das alterações no repositório remoto no GitHub, clique em **Push origin** (Figura 37).

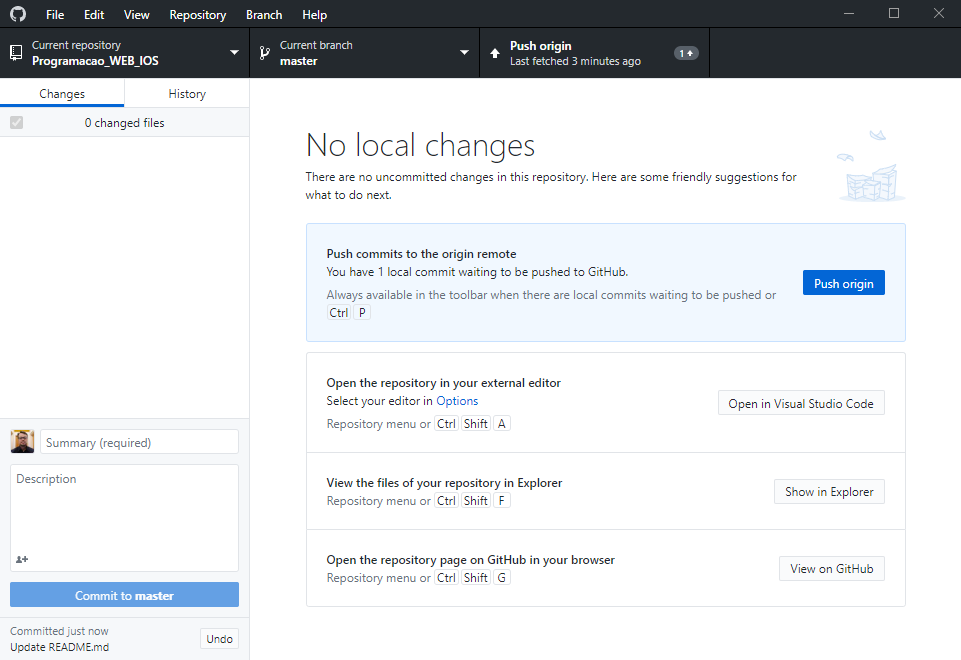


Figura 43. Opção de push para o repositório remoto.

1. Para abrir o repositório no GitHub no navegador, clique em **View on GitHub** (Exibir no GitHub) e, assim, você pode conferir o repositório remoto atualizado (Figura 38).

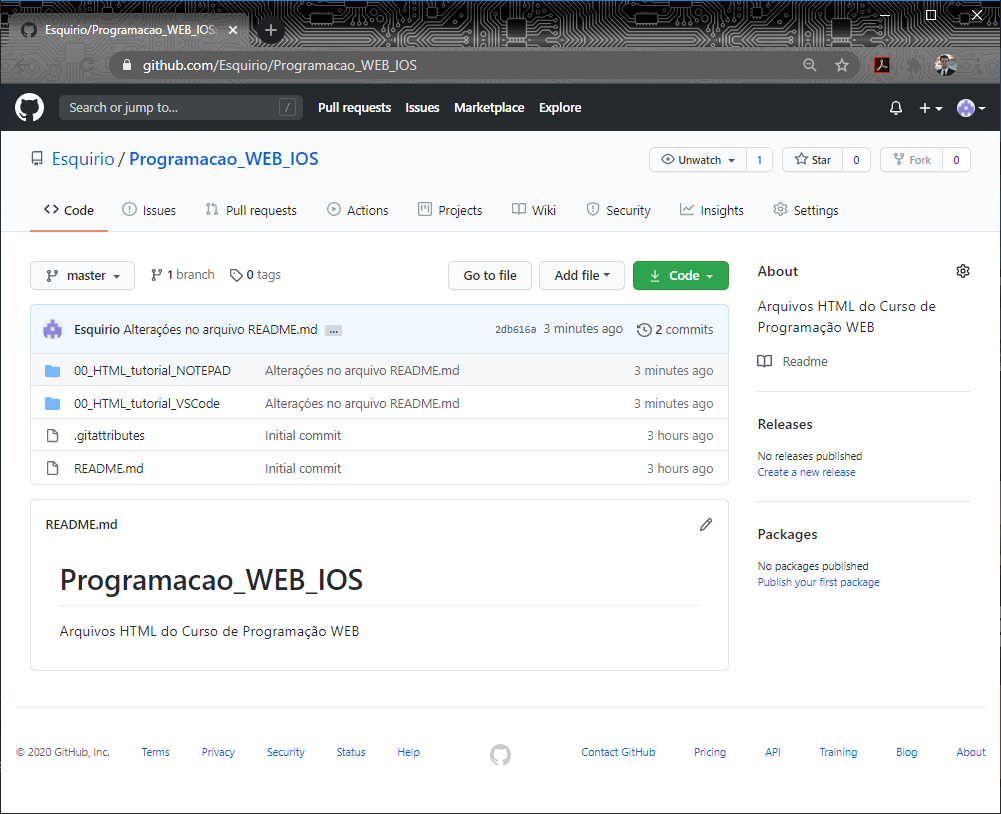


Figura 44. Repositório remoto atualizado.

1. Não se preocupe se seu repositório estiver diferente do mostrado na Figura 38, pois você poderá colocar os seus códigos desenvolvidos nos cursos à medida que você vai evoluindo seu aprendizado.