

Aplicações para Internet

- ✓ Exercício em Laboratório
- ✓ GitHub
- ✓ GitHub Pages
- ✓ Hospedagem em servidores reais
- ✓ FTP com FileZilla ou WebFTP

✔ Git

- Pela [documentação oficial](#), Git é um sistema de controle de versão distribuído de código aberto e gratuito, projetado para lidar com tudo, de projetos pequenos a grandes. O que isso significa? Significa que com o Git é possível manter um **histórico das alterações** dos seus arquivos, sabendo **quem, por que e quando** um arquivo foi editado.
- Baixe e instale: <https://git-scm.com/>

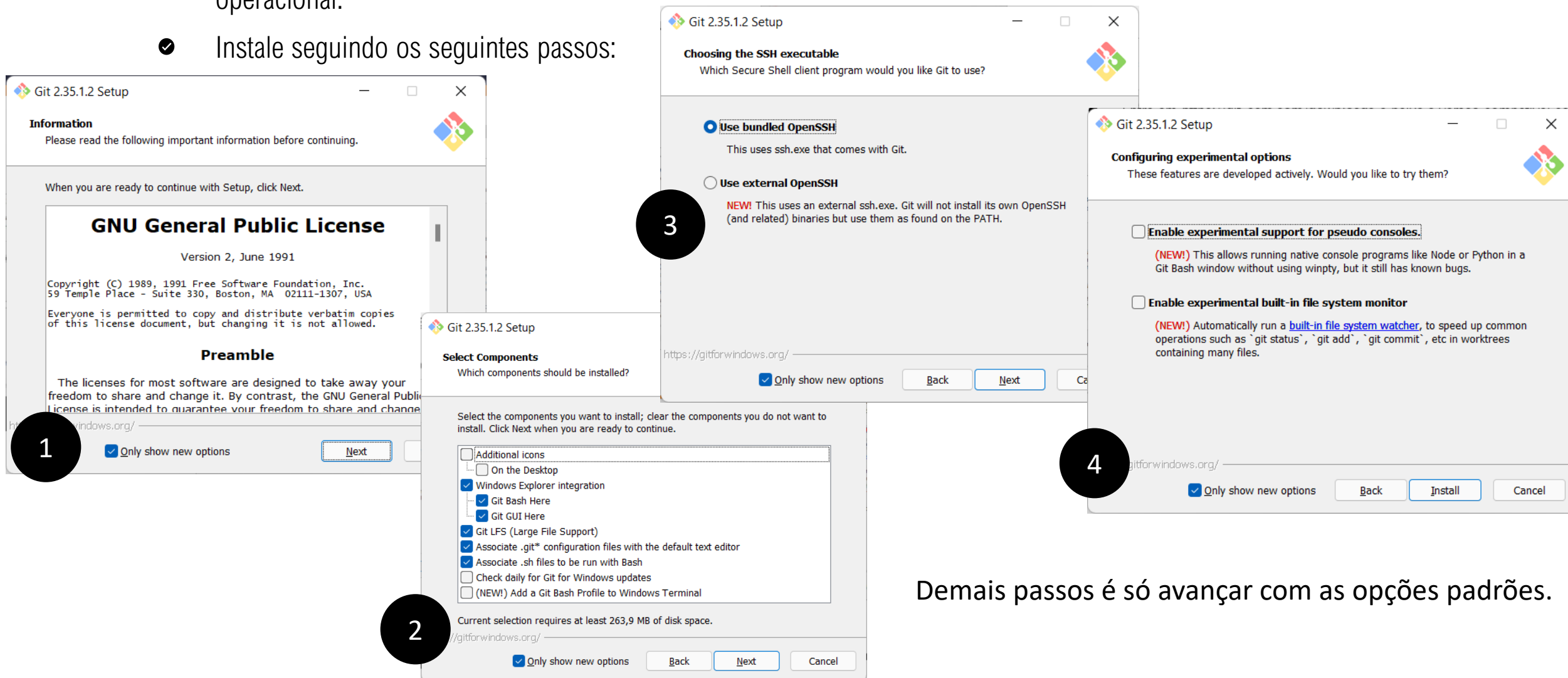
✔ GitHub

- Github é uma plataformas de hospedagem de código-fonte, é útil para gerenciar seu código e ainda permite a colaboração entre devs, ele utiliza o Git como sistema de controle de versão.
- Nessa plataforma, **cada projeto** contendo um código-fonte **é considerado um repositório**.
- Crie sua conta gratuita em: <https://github.com/>

✔ GitHub Pages


- Conforme definição no site oficial, o GitHub Pages está disponível para repositórios públicos para a conta do GitHub Free e para repositórios públicos e privados para a conta do GitHub Pro, Team, Enterprise.
- **CUIDADO:** leia as diretrizes e limites de uso em: <https://docs.github.com/pt/github/working-with-github-pages/about-github-pages> , **destaco: limite de 1GB, banda larga de 100GB por mês, proibido conteúdos ilegais e o sistema não foi feito para hospedagem gratuita na web ou revenda.**

- ✓ Entre em <https://git-scm.com/downloads> e baixe a versão compatível com o seu Windows ou seu sistema operacional.
- ✓ Instale seguindo os seguintes passos:



Demais passos é só avançar com as opções padrões.

- ✓ Acesse <https://github.com/> para criar a conta no github



Join GitHub

Create your account

Username *

Email address *

Password *

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

Email preferences

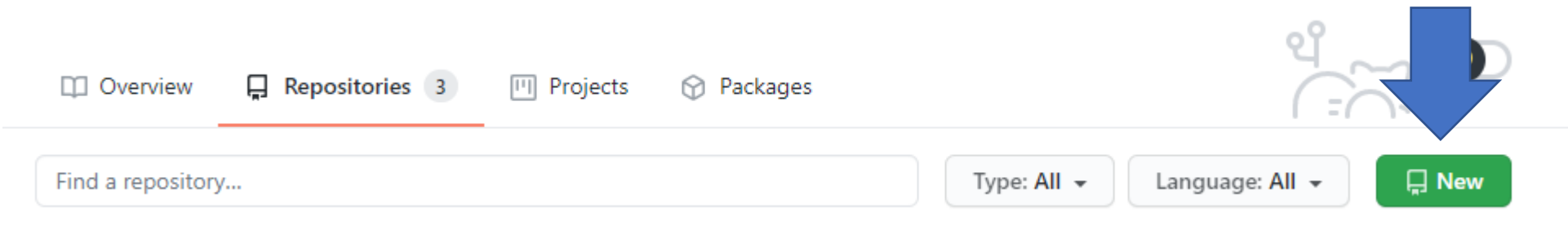
☐ Send me occasional product updates, announcements, and offers.

Verify your account

- ✔ No menu Terminal, abra um novo Terminal e vamos configurar os dados da nossa conta do GitHub digitando:
 - `git config --global user.email "seu-email@example.com"`
 - `git config --global user.name "Seu Usuário"`
 - OBS: estas configurações são globais, ou seja, são usadas por todos os nossos repositórios.

- ✔ Você também pode fazer as configurações acima para um repositório específico, basta tirar o `--global`
 - `git config user.email "seu-email@example.com"`
 - `git config user.name "Seu Usuário"`

- ✓ Faremos um projeto em sala para um professor que ministra aulas de programação com diferentes linguagens, o site deverá divulgar os serviços e aulas oferecidas e formulário de contato.
- ✓ Após criar sua conta e logar no GitHub, crie um repositório chamado **profcode**, siga os passos abaixo:





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 *

 alcidestbj / profcode 

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

Description (optional)


Projeto do prof de programação de computadores

- ☒  **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.

Initialize this repository with:

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

- ☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)
- ☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)
- ☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  main as the default branch. Change the default name in your [settings](#).

Create repository

Nome do repositório

Descrição do repositório

Público

Adicione um arquivo README e depois poderá inserir mais detalhes do projeto.

```
git add <arquivos...>
```

Este comando adiciona os arquivos em um lugar que chamamos de index, como uma área temporária. Este comando não está adicionando um arquivo novo ao repositório, mas sim dizendo que o arquivo está sendo preparado para entrar na próxima revisão do repositório. Um exemplo comum de uso é adicionar todos os arquivos com a instrução “**git add .**”

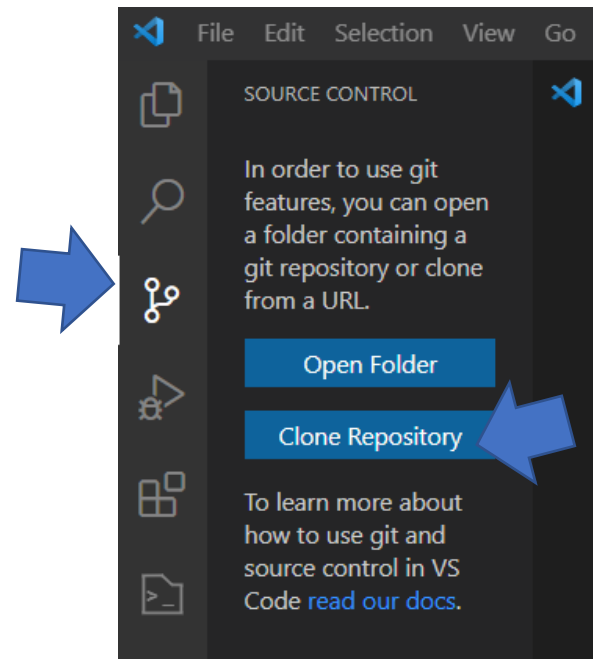
```
git commit -m "comentário qualquer"
```

Este comando realiza o “commit”, que significa pegar todos os arquivos sinalizados pelo comando add e criar uma revisão com um número e um comentário, que será vista por todos.

```
git push
```

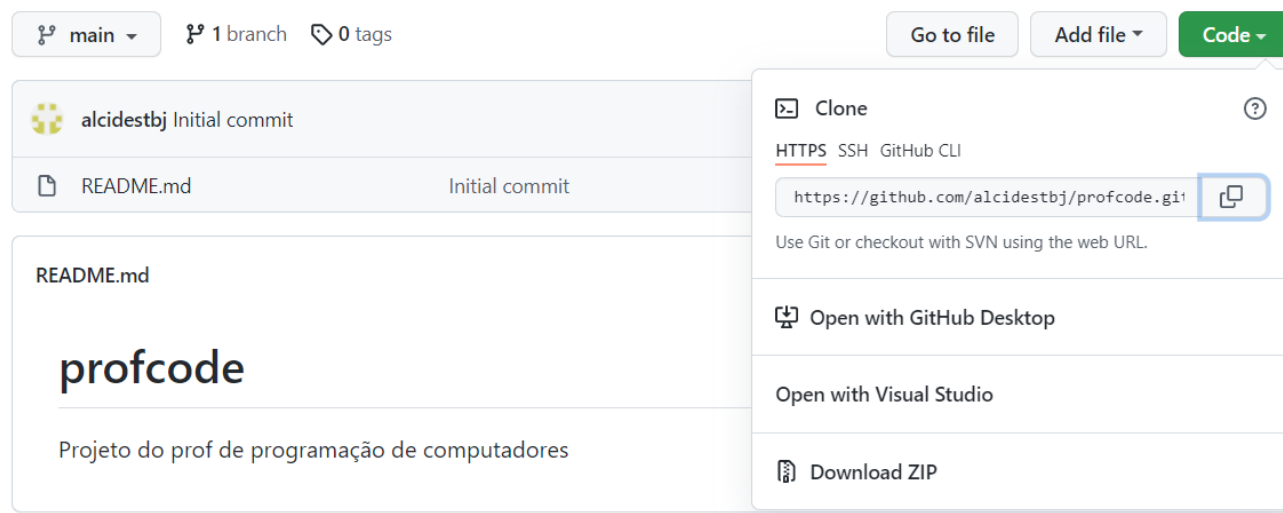
Push (empurrar) este comando publica todos os seus commits para o GitHub. Neste momento, será pedido a sua senha.

- ✓ Abra o VSCode e siga os passos abaixo, esses passos serão usados para o repositório profcode, mas você poderá aproveitar a mesma ideia para outros repositórios:
- ✓ 1 – Com o VSCode aberto, feche todas as pasta/projetos clicando em File>>Close Folder ou Workstation.
- ✓ 2 – Clique no ícone do Git e na opção Clone Repository, está opção é equivalente ao **git clone <<seu endereço do repositório>>** que pode ser realizada no terminal.

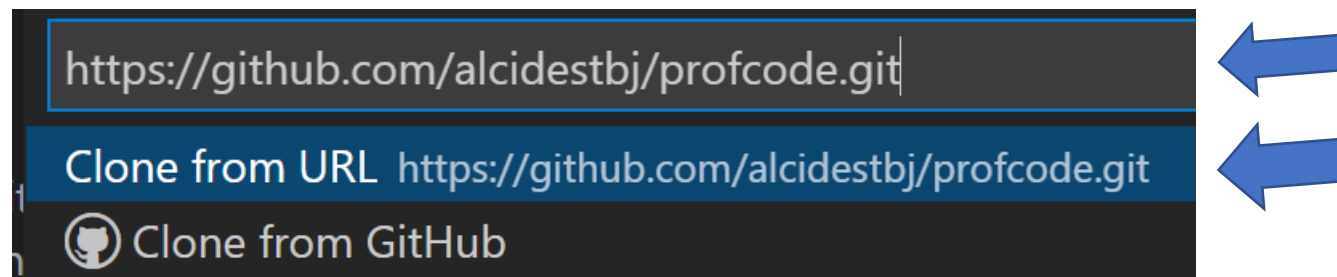


OBS: depois de ter feito o clone, nas próximas vezes, pode abrir a pasta do repositório normalmente no VSCode. Basta clicar no ícone do GIT e na opção Open Folder.

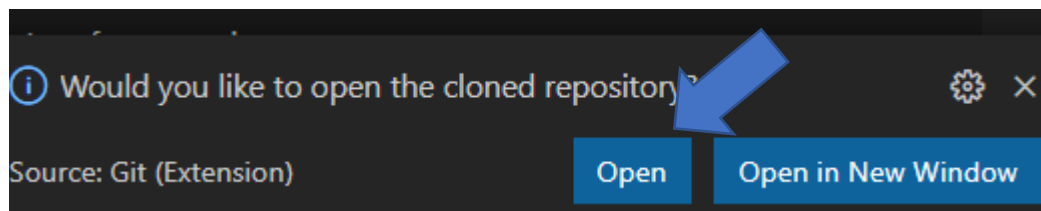
- ✓ 3 – Copie o endereço do seu repositório



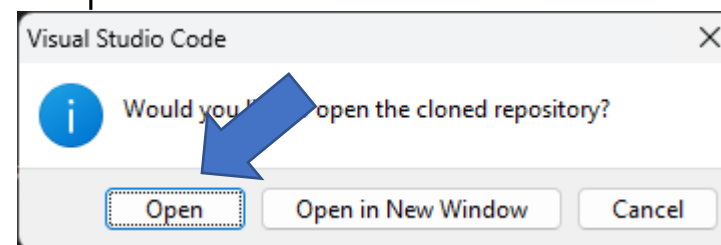
- ✓ 4 – Cole o endereço copiado na barra de pesquisa aberta e clique no link Clone From



- ✓ 5 – Selecione a pasta de destino da clonagem, após este passo, ao terminar o clone, o VSCode mostrará uma caixa no canto inferior direito ou um popup, clique me Open



ou



- ✓ 6 – Entre na pasta clonada e verá nosso arquivo README, na listagem geral do Windows, verá também uma pasta .git, ela é utilizada pelo próprio git, **não há necessidade de mexer.**
- ✓ 7 – Crie um arquivo index.html com o conteúdo
 - `<h1>ProfCode</h1>`
- ✓ 8 – Salve o arquivo
- ✓ 9 – Precisamos subir para o GitHub este arquivo ou outros que criarmos, **este processo também se repete sempre que alterarmos o arquivo. Execute estes comandos no terminal (Ctrl + Shift + ').**
 - `git add .`
 - `git commit -m "arquivo index com teste inicial"`
 - `git push`

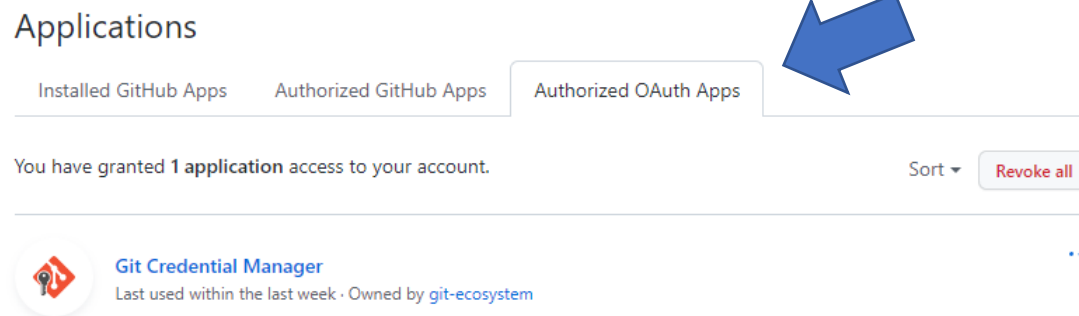
- ✔ Sempre que adicionarmos arquivos ou fizermos modificações em algum já existente, devemos fazer o add, commit e push vistos no passo 9. OBS: faremos isso geralmente no final da aula para atualizar o GitHub.

- ✓ Caso encontre problemas em executar os comandos no laboratório, basta acessar a página do seu GitHub em qualquer navegador, escolher o repositório do projeto que está trabalhando e clicar no botão Add File >> Upload Files
- ✓ Arraste as pastas e arquivos para a região indicada na página, escreva um texto referente as alterações e clique em Commit Changes

The screenshot shows the GitHub 'Commit changes' interface. At the top, there is a large rectangular area for dragging files, with the text 'Drag files here to add them to your repository' and a link 'Or choose your files'. A blue arrow points to this area. Below this is the 'Commit changes' section, which includes a text input field for 'Add an optional extended description...'. A blue arrow points to this field. At the bottom of the 'Commit changes' section, there are two radio buttons: 'Commit directly to the main branch.' (selected) and 'Create a new branch for this commit and start a pull request.' A blue arrow points to the 'Commit directly to the main branch.' option. At the very bottom, there are two buttons: 'Commit changes' (green) and 'Cancel' (red). A blue arrow points to the 'Commit changes' button.

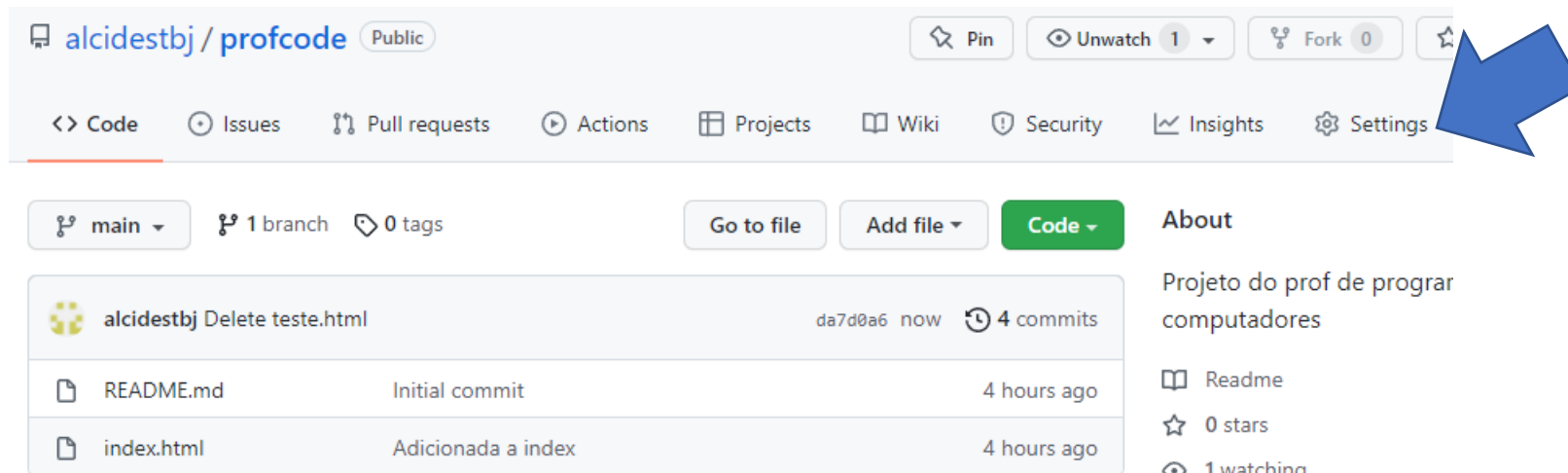
- ✓ Quando usar máquinas compartilhadas com outros usuários, é interessante **remover suas identificações e revogar o acesso do VSCode ao Git**, para isso, siga os passos abaixo:

- No terminal do VSCode
 - `git config --global --unset-all user.name`
 - `git config --global --unset-all user.email`
- Acesse a página do GitHub, clique no ícone do seu perfil
- Clique em Settings
- Clique no item Applications (aplicações no menu esquerdo)
- Clique na aba Authorized OAuth Apps



Clique aqui e escolha a opção Revoke, na janela que aparecerá, confirme que está revogando o acesso.

- ✓ Podemos de forma temporária, disponibilizar nosso projeto ProfCode no GitHub Pages, para isso, siga os passos:
- ✓ 1 – Após finalizar o projeto, faça o push completo dos arquivos para o GitHub.
- ✓ 2 – Clique na opção Settings do seu repositório:



- ✓ 3 – Na opção GitHub Pages, configure o Source para MAIN e depois clique em SAVE

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Build and deployment

Source

Deploy from a branch ▾

Branch

Your GitHub Pages site is currently being built from the main branch. [Learn more.](#)

main ▾

/ (root) ▾

Save

Learn how to [add a Jekyll theme](#) to your site.

- ✓ 4 – Feito isso, insira um domínio para seu site e salve.

Custom domain

Custom domain


Custom domains allow you to serve your site from a domain other than `alcidestbj.github.io`. [Learn more.](#)

profcode

Save

DNS records should point to the [internationalized domain name](#).

- ✓ 5 – Aguarde alguns minutos e aparecerá o link para você acessar.

Your site is live at <https://alcidestbj.github.io/profcode/>
Last deployed by  alcidestbj 2 minutes ago

Visit site

...

OBS: Lembre-se que o uso do GitHub Pages é limitado.

1. Criar o site localmente
2. Registrar um domínio (existem domínios gerais gratuitos).
3. Contratar um serviço de hospedagem (existem opções de hospedagem gratuitas).
4. Configurar os servidores de DNS no domínio criado com base no servidor de hospedagem.
5. Fazer o upload dos arquivos de seu site para o servidor (geralmente é feito utilizando FTP).

Gratuitos

- Não permite gerenciamento do domínio
- Ferramentas limitadas
- A maioria possuem propagandas
- Sem suporte (geralmente)
- Servidores instáveis
- Sem backups
- Recursos limitados
- Tráfego limitado

Para sites pessoais
ou trabalhos
simples

Pagos

- Gerenciamento completo do domínio
- Diversas ferramentas para gerenciamento, criação de sites etc.
- Suporte
- Servidores estáveis
- Backups dos dados
- Recursos ilimitados (geralmente)
- Tráfego ilimitados (geralmente)

Para sites e
serviços
profissionais

- ✔ Gratuitos (algumas opções)



<https://byet.host/>



<https://www.000webhost.com/>

- ✓ Após instalar o FileZilla, vamos abrir a ferramenta e configurar a conexão de teste da aula, as informações passadas durante a aula serão válidas somente até seu término.
- ✓ Caso o FileZilla não esteja instalado, utilize o endereço webftp informado pelo professor ou o informado no seu cadastro do 000webhost.
<https://webftp.locaweb.com.br/login>
- ✓ OBS: quando criar sua conta no 000webhost, utilize os dados que eles fornecerem.

AULA DE APLICAÇÕES PARA INTERNET (SOMENTE TESTE)

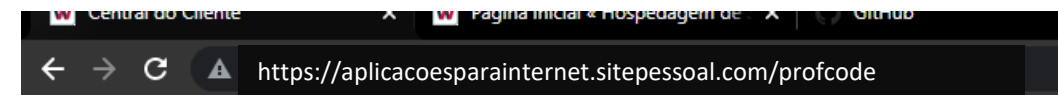
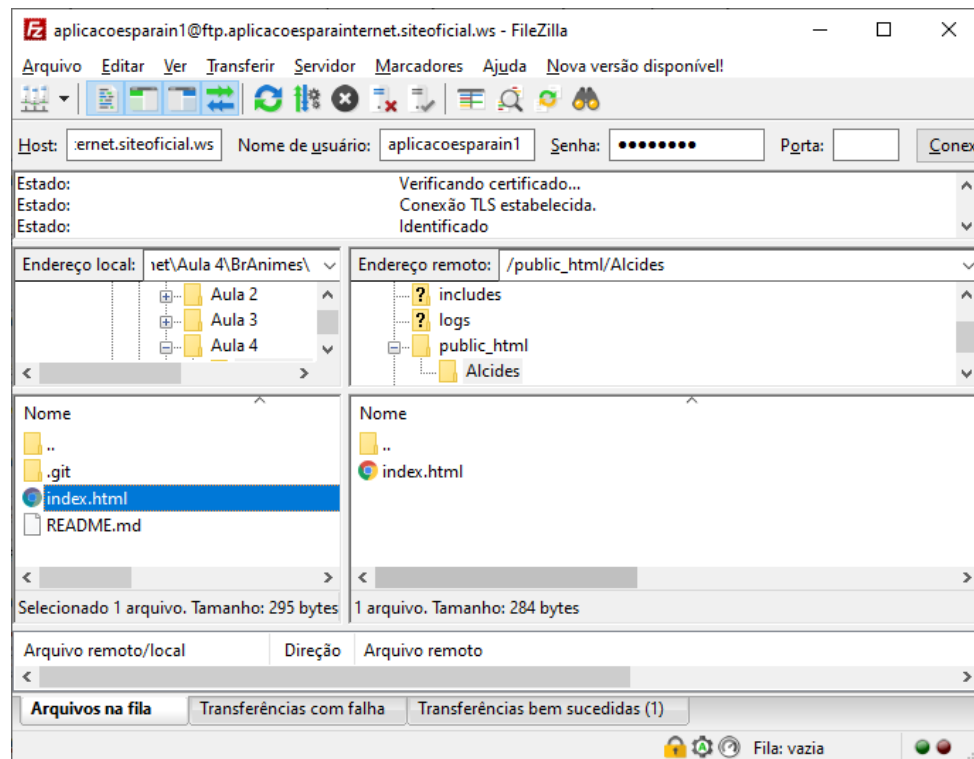
URL: aplicacoesparainternet.sitepessoal.com

FTP: [ftp.aplicacoesparainternet.sitepessoal.com](ftp://aplicacoesparainternet.sitepessoal.com)

Usuário de FTP: aplicacoesparain2

Senha de FTP: senha será informada na aula (válida somente para o período da aula do dia)

- ✓ Acesse sempre a pasta public_html
- ✓ Crie uma pasta com seu nome e depois entre nela
- ✓ Copie os arquivos do projeto para sua pasta e teste no navegador
 - EX: <http://aplicacoesparainternet.siteoficial.com/Seunome/>



ProfCode

- ✔ Vamos agora desenvolver todo o projeto do ProfCode conforme especificações do cliente.
- ✔ Após finalizarmos o projeto teremos que ter:
 - Uma copia no GitHub e o link do seu repositório no GitHub Pages

Para os projetos mais elaborados do semestre (HTML+CSS), sugestão:

- Contrate um servidor de hospedagem **grátis** ou pago (o mais barato possível) para hospedar os projetos do semestre.





That's all Folks!

E-mail: alcides.junior@cruzeirodosul.edu.br