# **PLANO DE AULA**

Aula 01 | Tempo estimado: 1 hora e 30 minutos | Web Starter

Tipo da atividade: Offline

Ferramenta(s): Computador, Git, nodeJS e VS Code

Conteúdos

* Github e HTML.

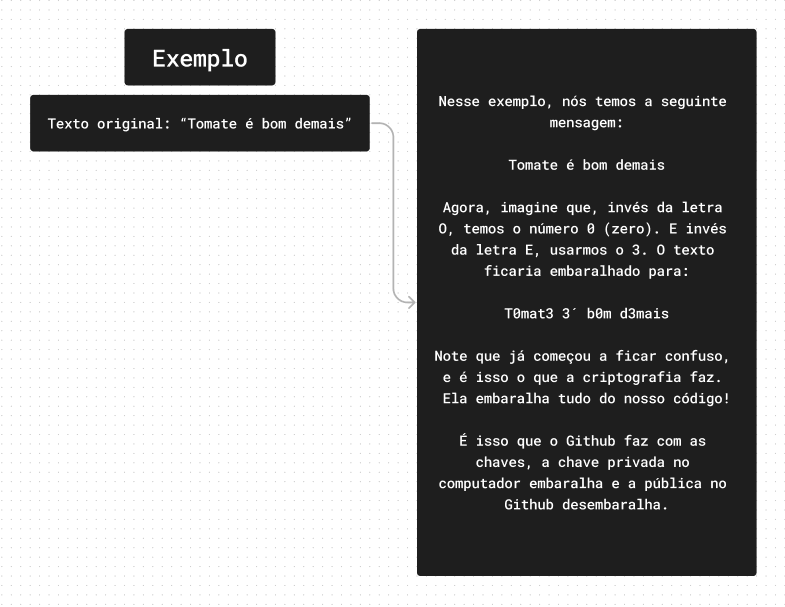
Objetivos

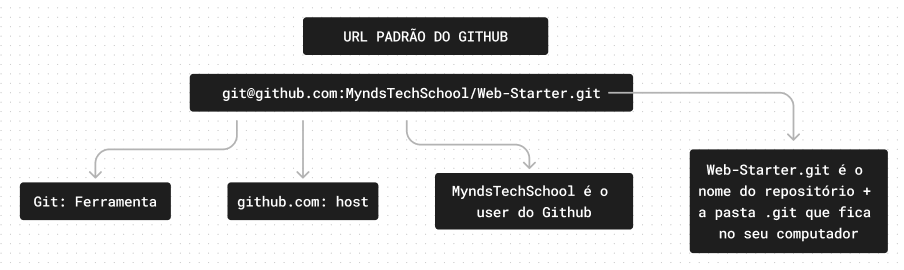
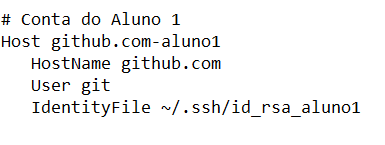
* Criar Conta Github;
* Entender como a Web funciona, o que é a Web e quais as vantagens de usar o Github. Se ambientar com o Github.

Estratégias e atividades

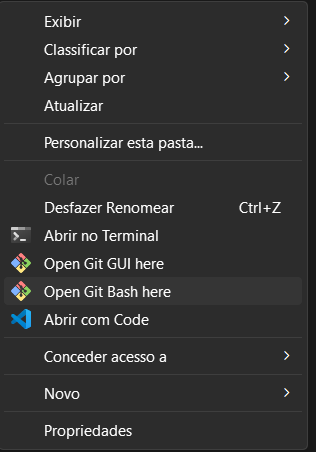
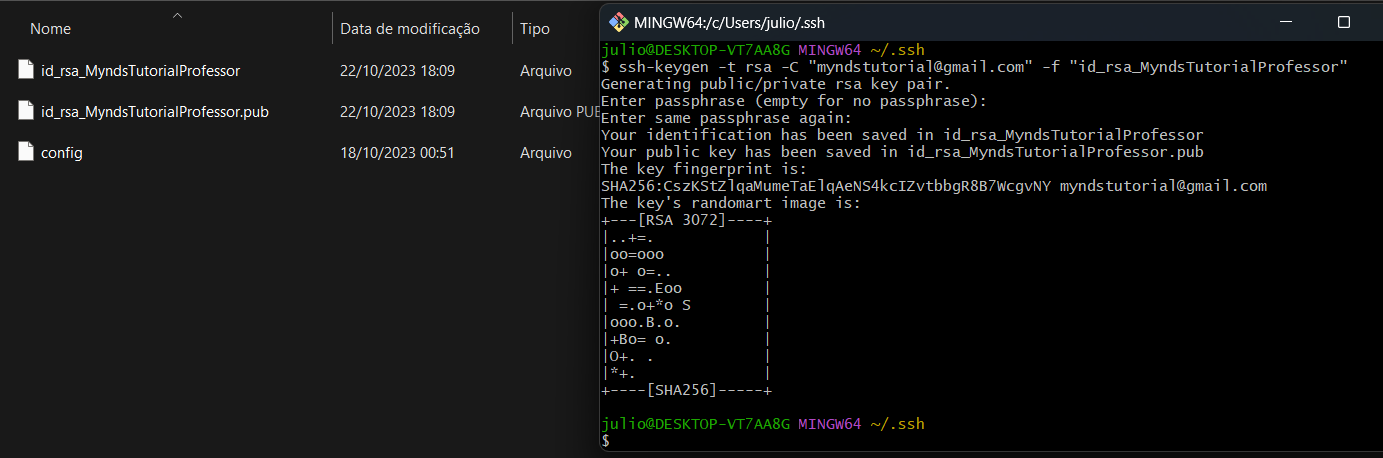
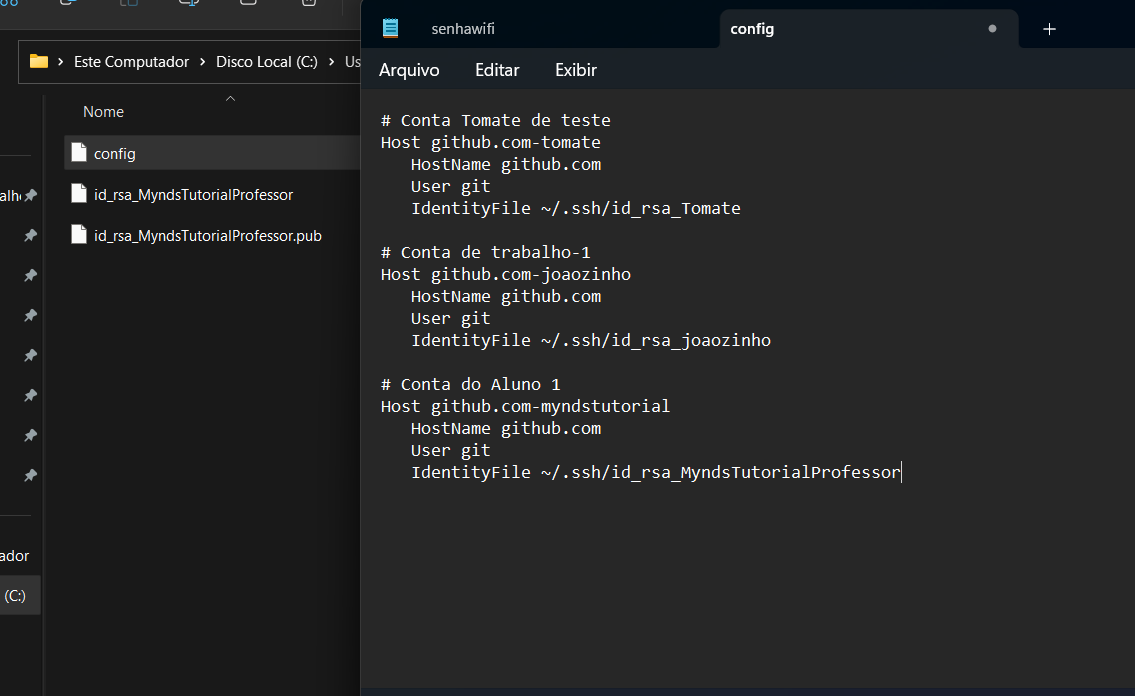
1. Explicar sobre Web e como Web é importante para o mundo, não apenas se resumindo a sites.
2. Criar conta no site [Github](https://github.com/). Cada aluno terá um gmail, onde ele vai criar a conta. É extremamente importante que o aluno tenha sempre acesso ao gmail, pois se ele perder a senha vai perder a conta do Github.
3. Ativar a autenticação de 2 fatores do Github no [Extensão chrome](https://chromewebstore.google.com/detail/2fa-authenticator-app/gmohoglkppnemohbcgjakmgengkeaphi?hl=pt-br), desta forma sempre que o aluno estiver logado com o gmail dele poderá ver o número de autenticação.
4. Após criar a conta no site, o professor explicará o que é Git e Github, e quais as vantagens de se criar um Github.
5. Entendendo sobre Criptografia: nesta parte o aluno irá entender sobre criptografia, e porque é importante criptografar o código.

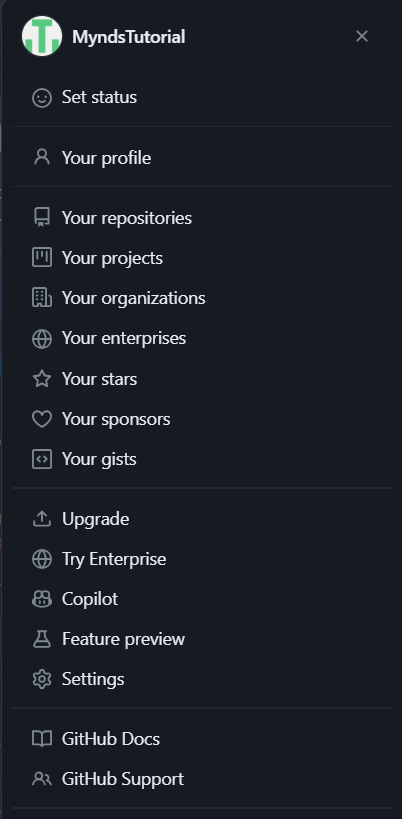
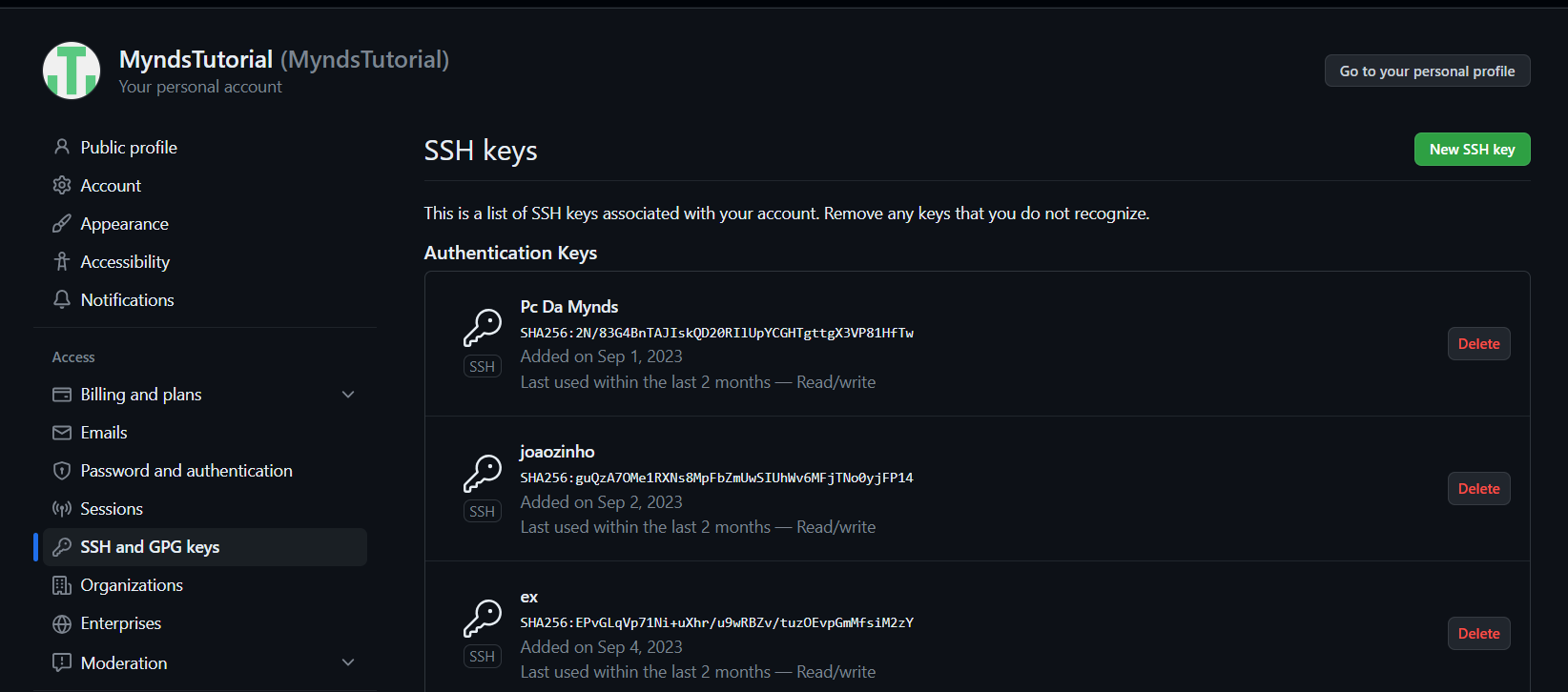
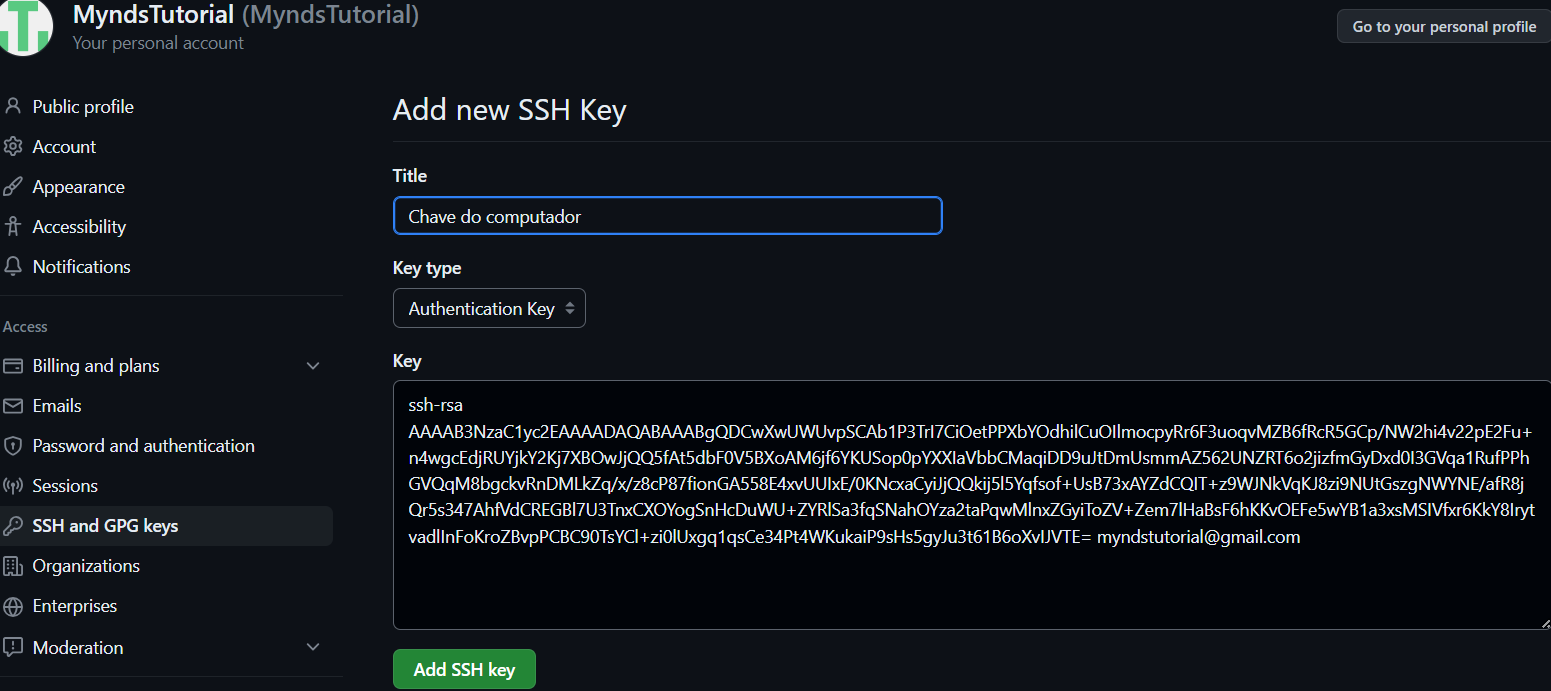
* **Criptografia:**
  + Chave pública: A chave pública tem como tarefa desembaralhar o código criptografado, pois ela sabe como traduzir o código para o original.
  + Chave privada: A chave privada criptografa o código e faz com que o arquivo se “embaralhe”, fazendo com que apenas quem sabe a sequência possa traduzir.

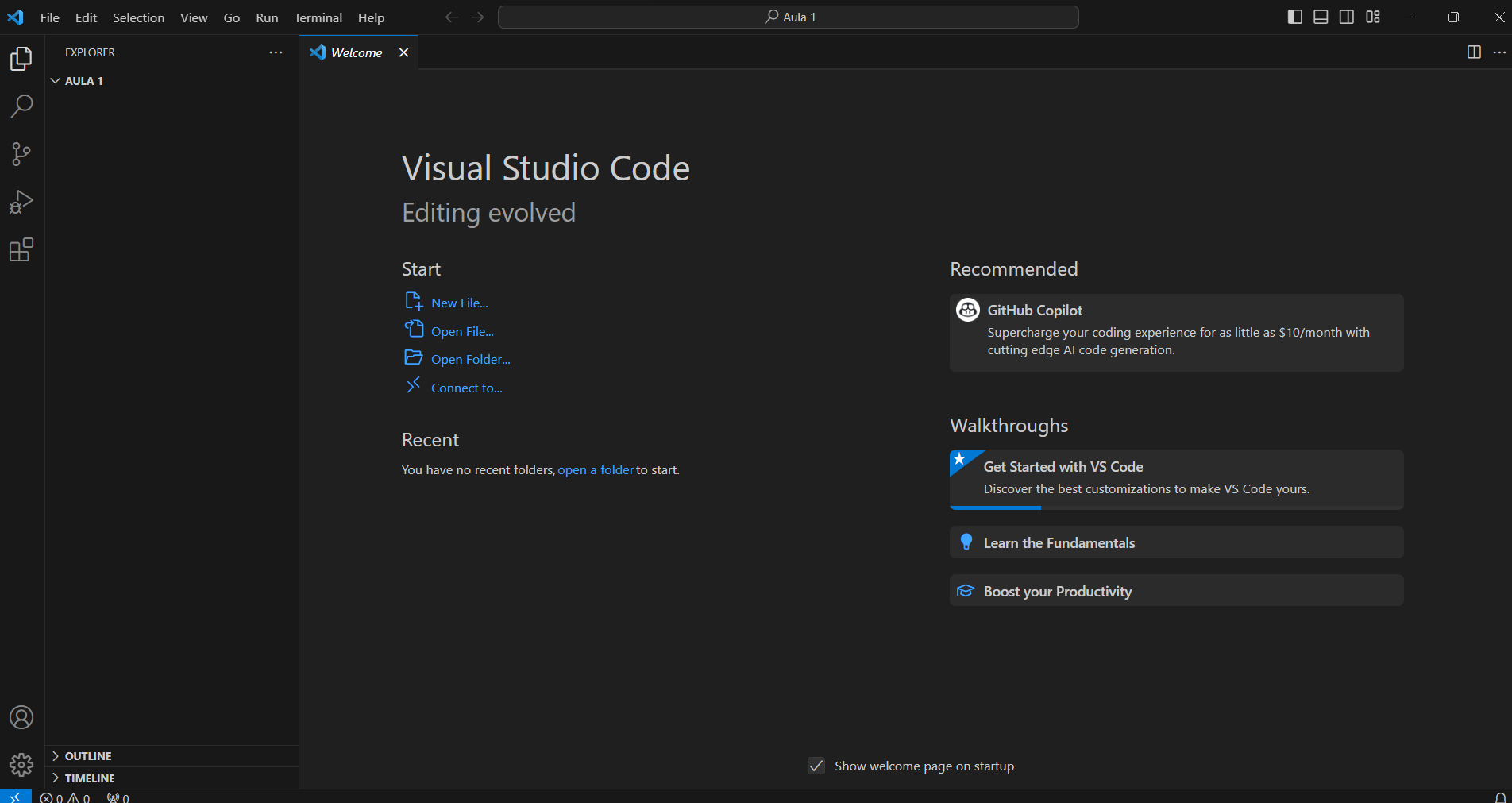
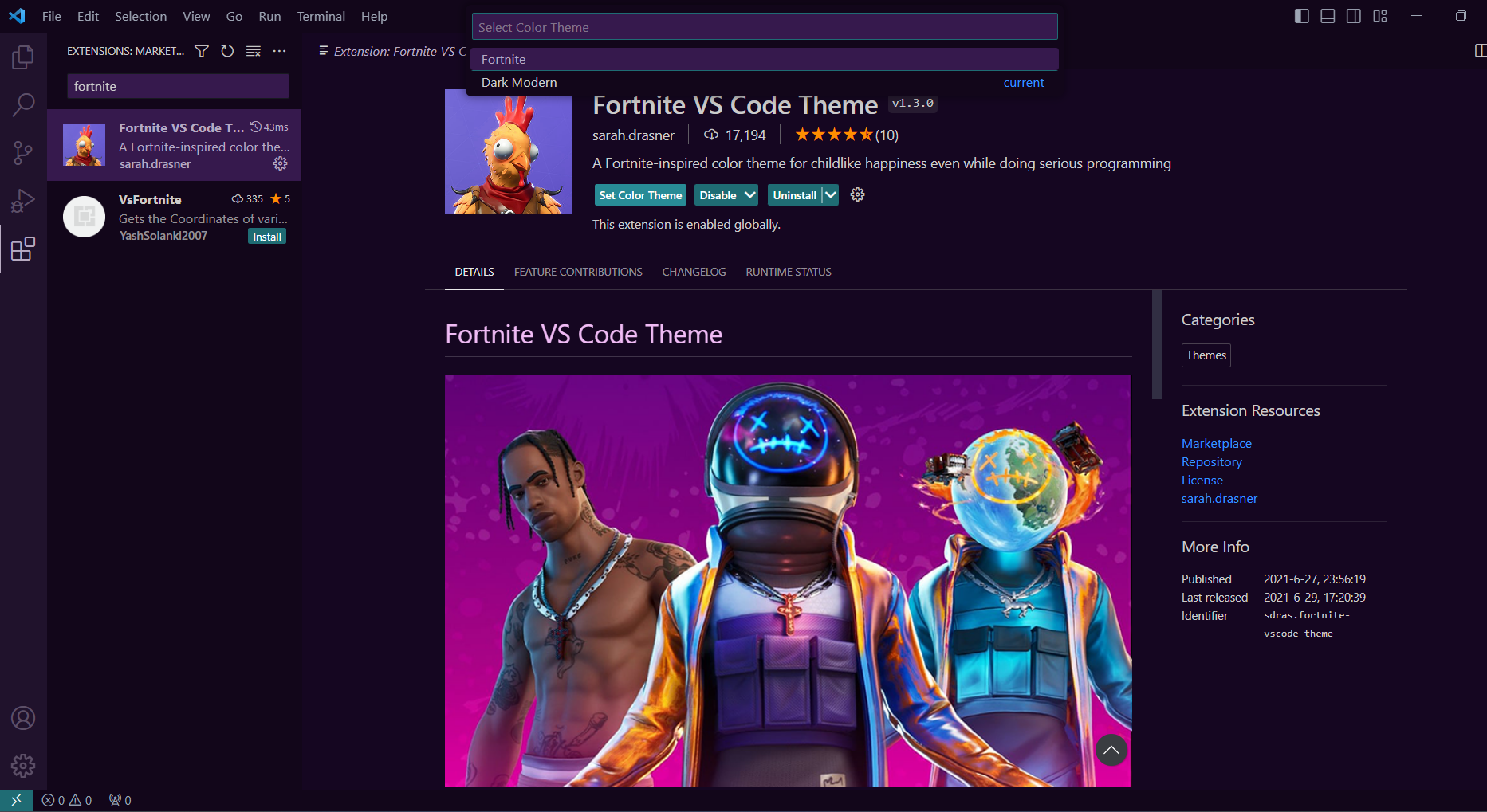
Exemplo de criptografia:

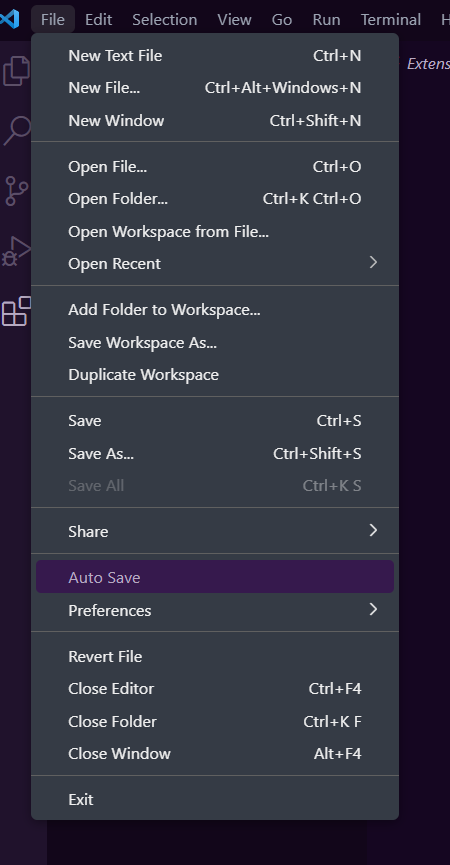
1. Criar o Config dentro de C:\Users\nomeDoUsuario\.ssh (caso não tenha uma pasta chamada .ssh, crie uma.
   1. Como o Windows não permite mais criar um executável por questões de segurança, disponibilizamos o arquivo para configurar múltiplas chaves ssh neste [link](https://drive.google.com/drive/folders/10l1Bk1Hs9XEgI1t65k0Uwaf5G9108iZs?usp=share_link), nele existe um arquivo chamado **config**.
   2. Como funciona este arquivo: é importante primeiro entender sobre como o Github funciona e interage com a ferramenta Git, baixada para iniciar a aula. Entendendo a URL de um repositório:  
      
   3. Agora, o importante é entender que a ideia central do Git é ter uma única chave ssh no computador, porém nós estamos usando o computador da escola, então para isto será criado um arquivo de configuração. Esse arquivo que está no link acima, tem vários templates de chaves ssh:  
        
      Se repararmos, nele existem algumas informações importantes. A primeira é o host, se lembrarmos a URL do Github pede um host, então com isto podemos customizar os hosts dos alunos.

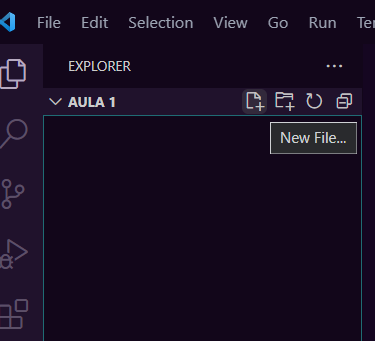
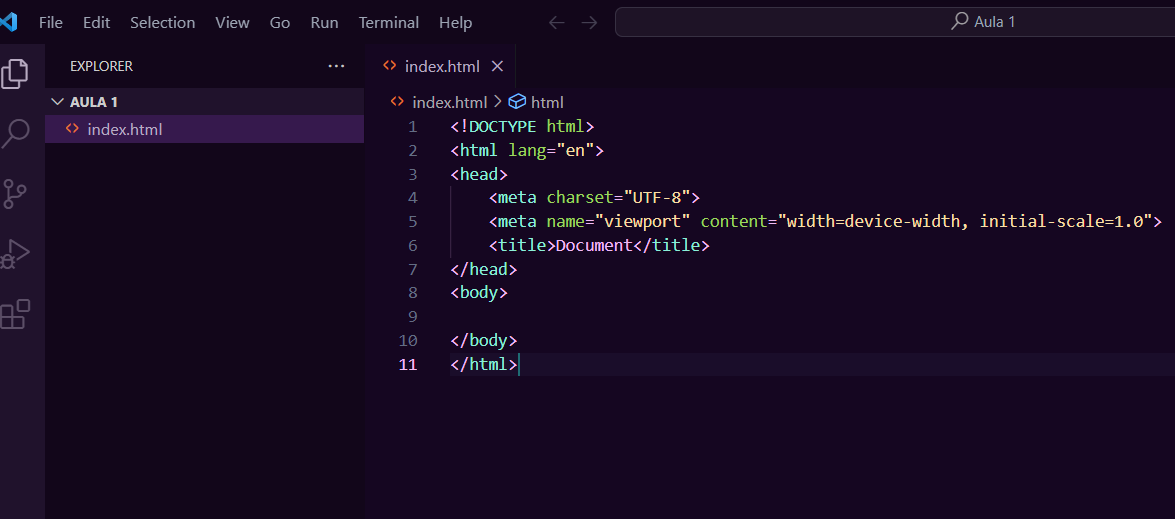
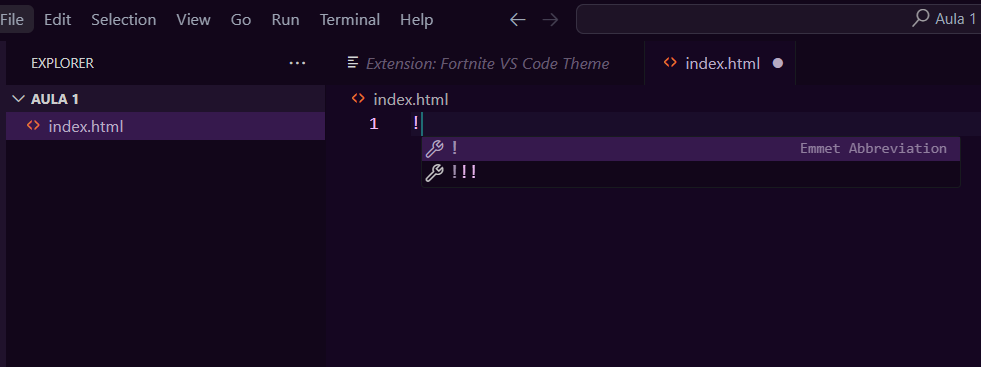
O segundo argumento importante é o IdentityFile que mostra onde está salvo a chave ssh do aluno, contendo a pasta .ssh/nomeDaChaveNãoPublica

1. Criar a chave ssh:
   1. Dentro da pasta .ssh clique com o botão direito e vá em **Open Git Bash Here**:  
      
   2. Clique nele e digite:   
      Lembre-se de alterar o gmail para o gmail do Github e o id\_rsa\_NomeDoAluno. Caso o aluno vá fazer esse processo em casa, ele não precisará de ter também o arquivo de config, a única coisa que ele precisará fazer em casa é baixar o Git Bash e digitar:  
        
      ssh-keygen -t ed25519 -C "[your\_email@example.com](mailto:your_email@example.com)"  
        
      Após isto, será criada a chave ssh. Dê enter várias vezes. Se você reparar verá que foram criadas duas chaves: uma .pub e uma sem extensão, a com .pub você irá abrir em um bloco de notas:
   3. Reparando bem, você irá usar as duas chaves (na escola) e em casa o aluno só irá precisar da .pub que vai estar no passo
   4. Agora, abra o arquivo de config e coloque o nome da chave, criando um host pro aluno parecido com este exemplo:  
      São alteradas apenas duas partes: o nome do Host e o IndentityFile que é o caminho para o arquivo da chave. Por isto o nome tem que ser igual. Depois é só dar ctrl + S e ir para o último passo:
   5. Abra o arquivo com .pub clicando no botão direito do mouse e abrindo com o bloco de notas, dê Ctrl + A e Ctrl + C para copiar e vamos até o Github para salvar a chave. Clique no ícone do lado direito da tela e vá em settings:

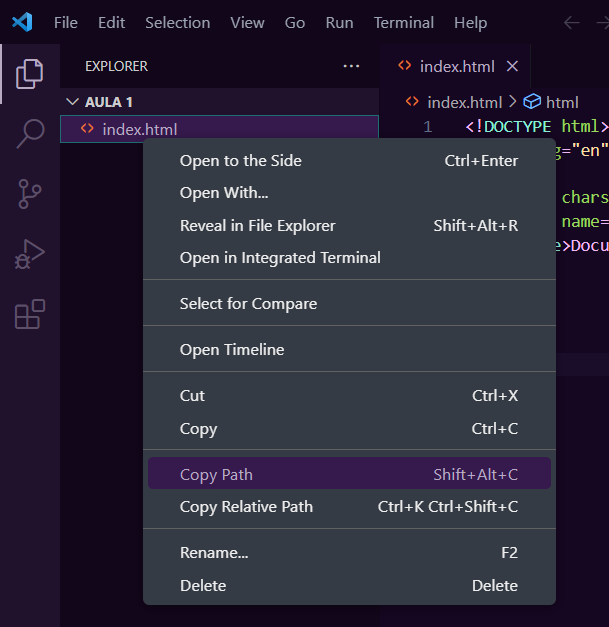
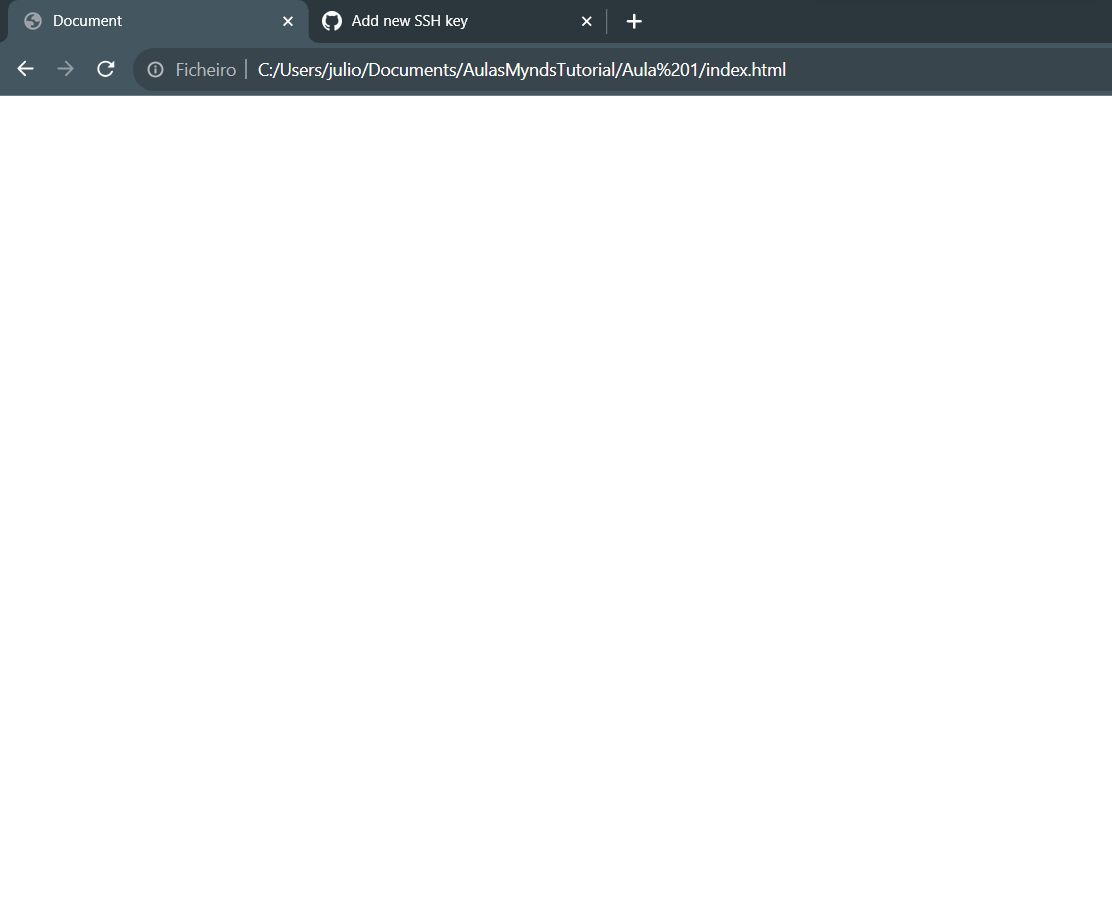
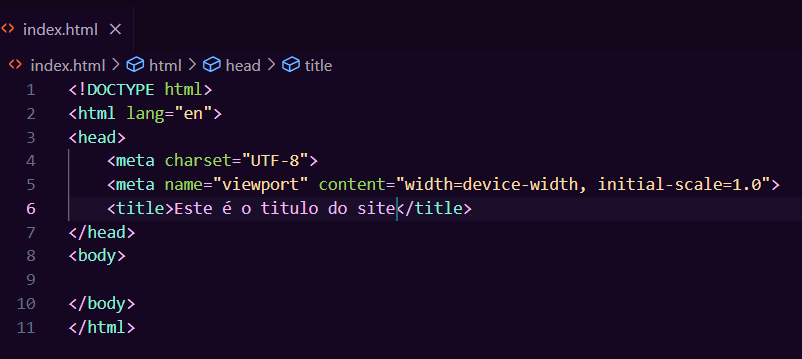
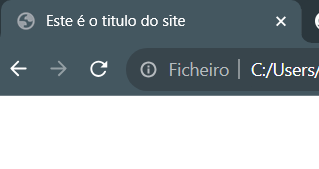
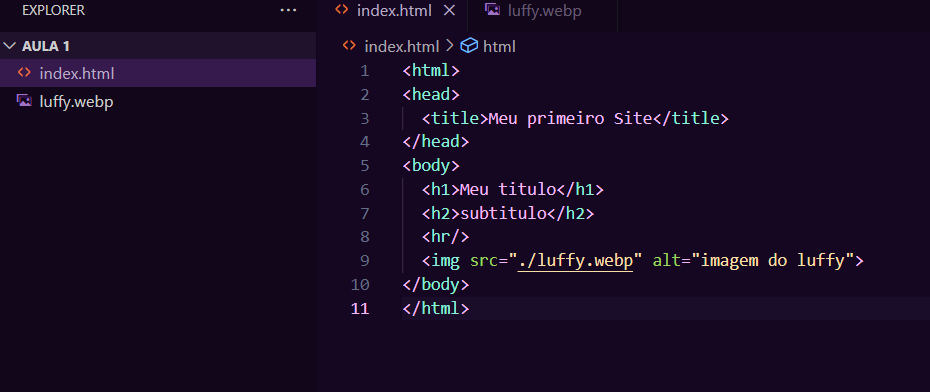
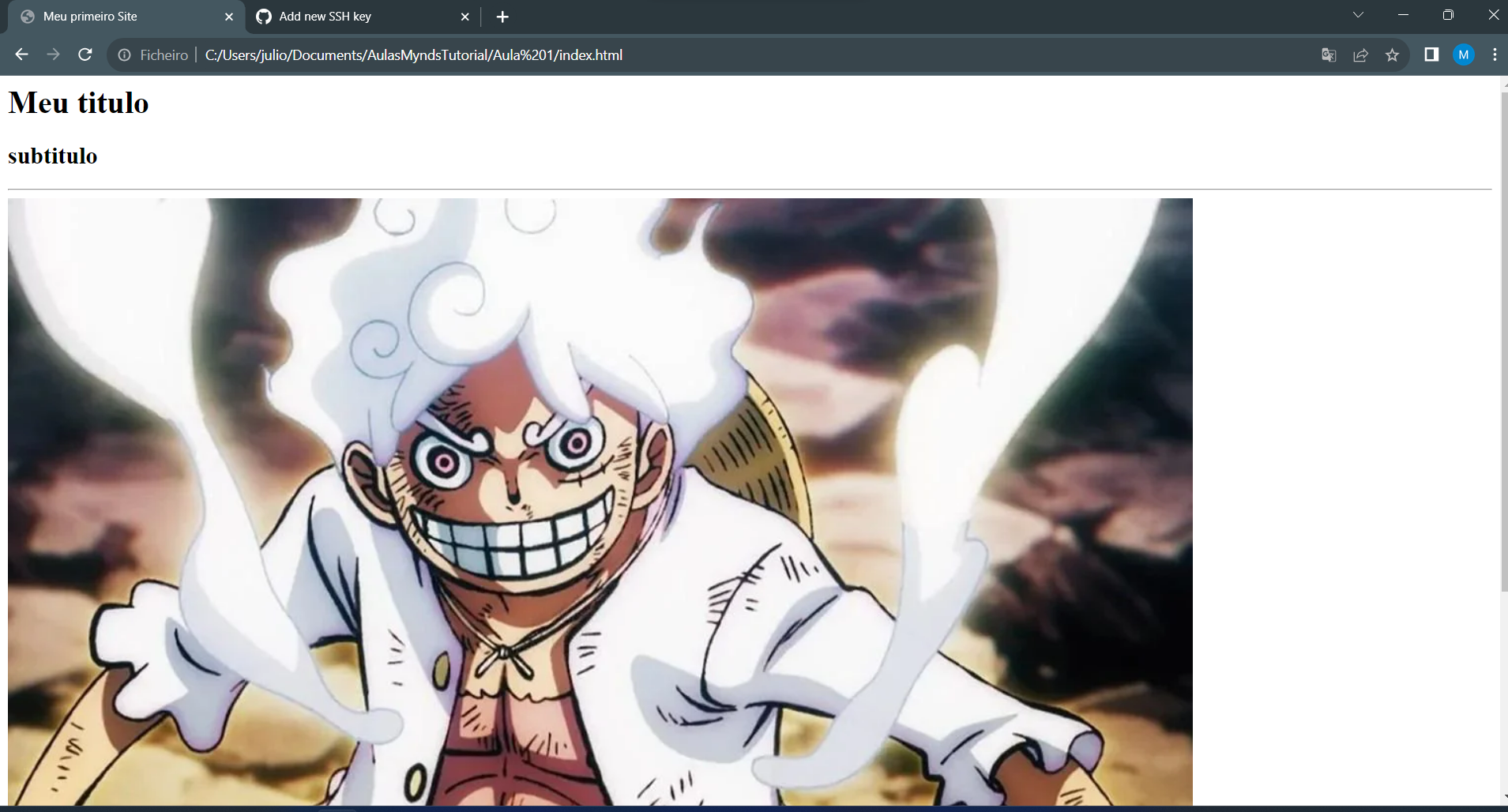
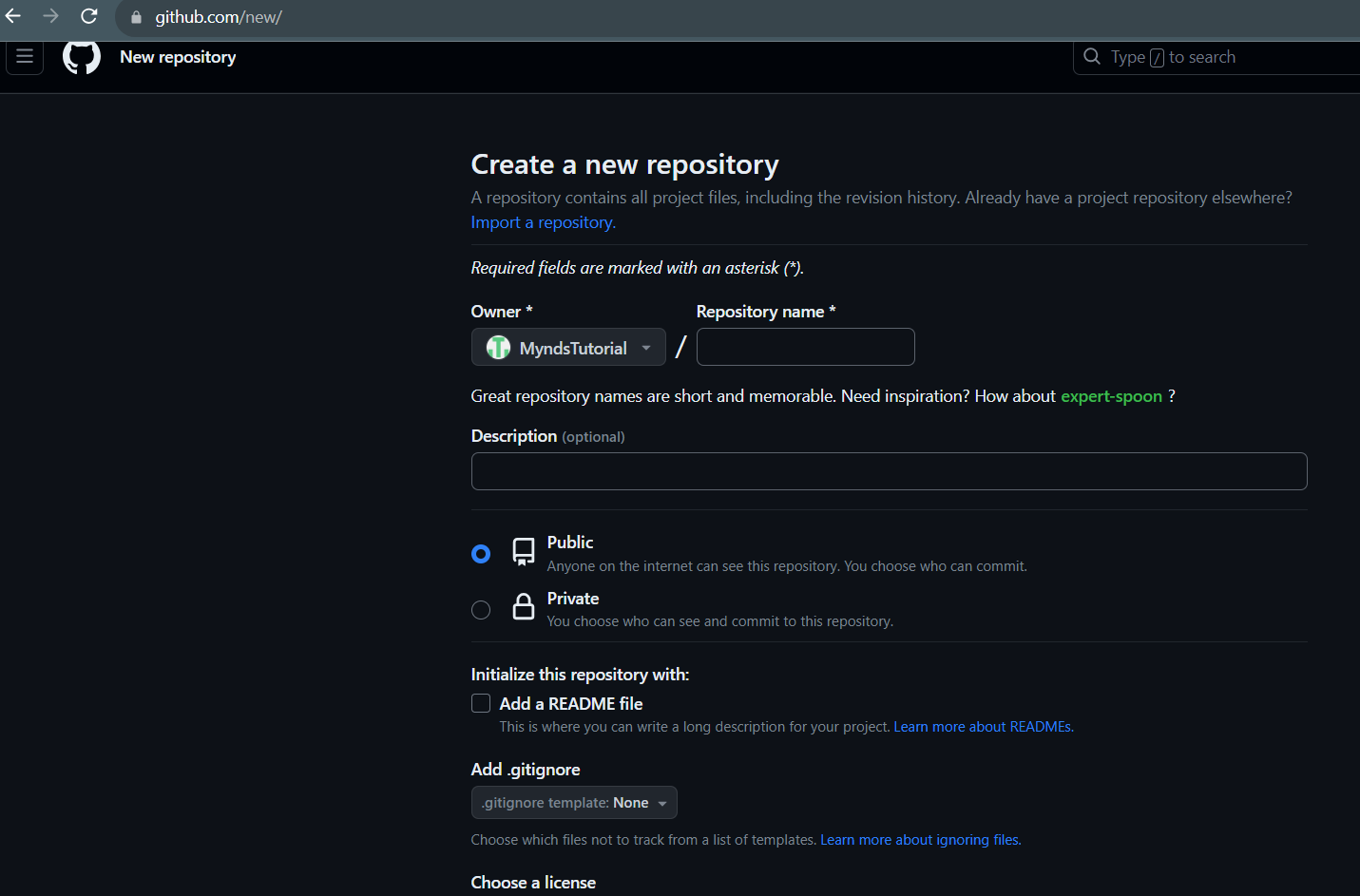
  
Após isso vá em SSH and GPC keys e adicione uma nova chave ssh:  
  
  
Agora que adicionamos a chave ssh, podemos ir para o código da aula, pois o Github já está configurado.

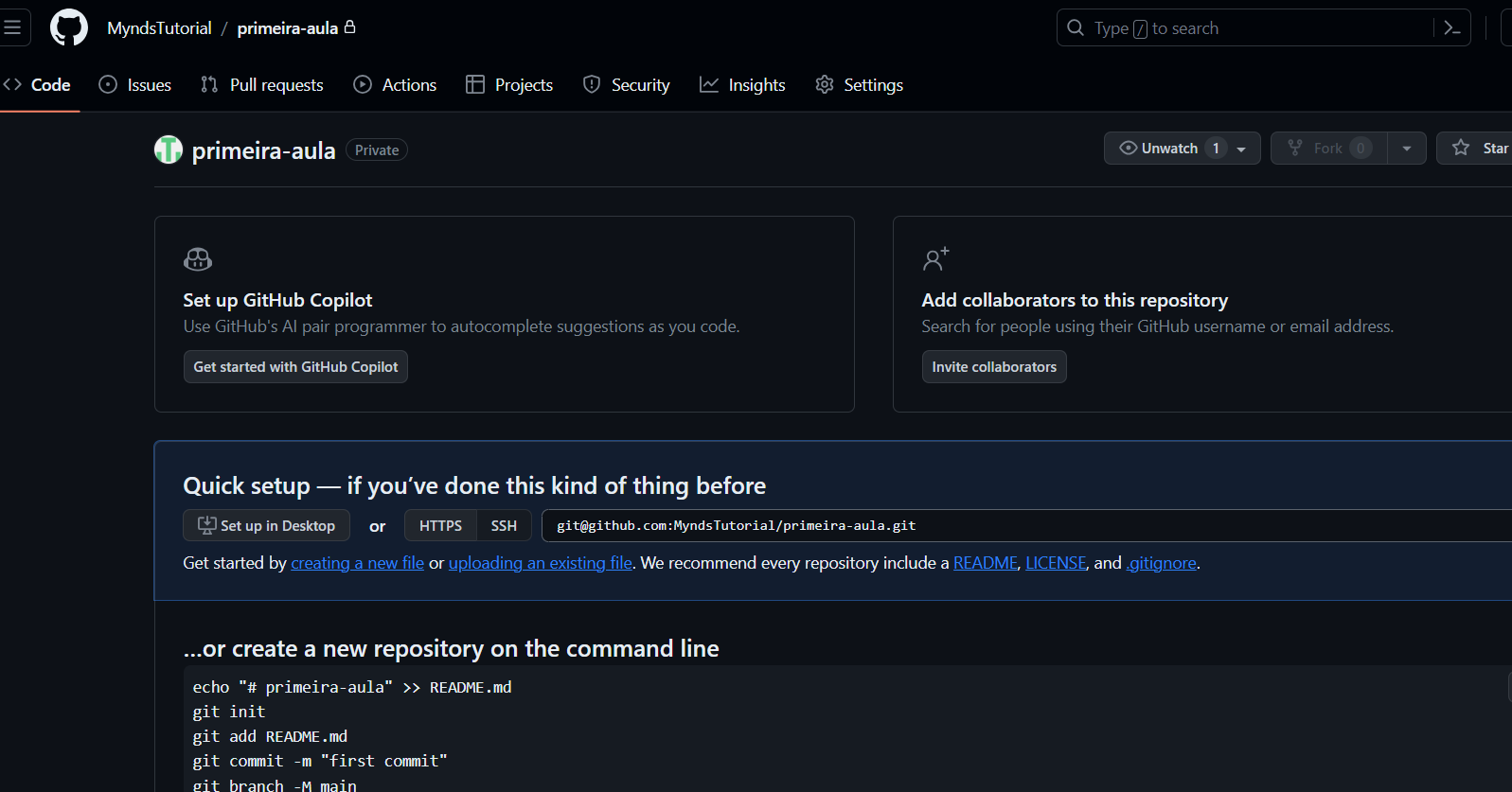
1. Crie uma pasta no computador e abra-a no Visual Studio Code:  
   
2. Opções legais para ensinar aos alunos. O *file*, na aba superior é onde você pode abrir os arquivos e ir até as pastas. Existe uma opção chamada extensões, onde o aluno pode instalar as extensões visuais do Visual Studio Code. Vou instalar a Fortnite no exemplo, porque gosto da cor roxa, mas o aluno pode procurar várias, exemplos de nomes:
   1. Dark
   2. Fortnite
   3. Naruto theme
   4. Panda Theme
   5. Dracula
   6. Monokai
3. Exemplo Fortnite:
4. Para garantir que o projeto seja sempre salvo, clique na opção **Auto Save** do VS Code: File > Auto Save

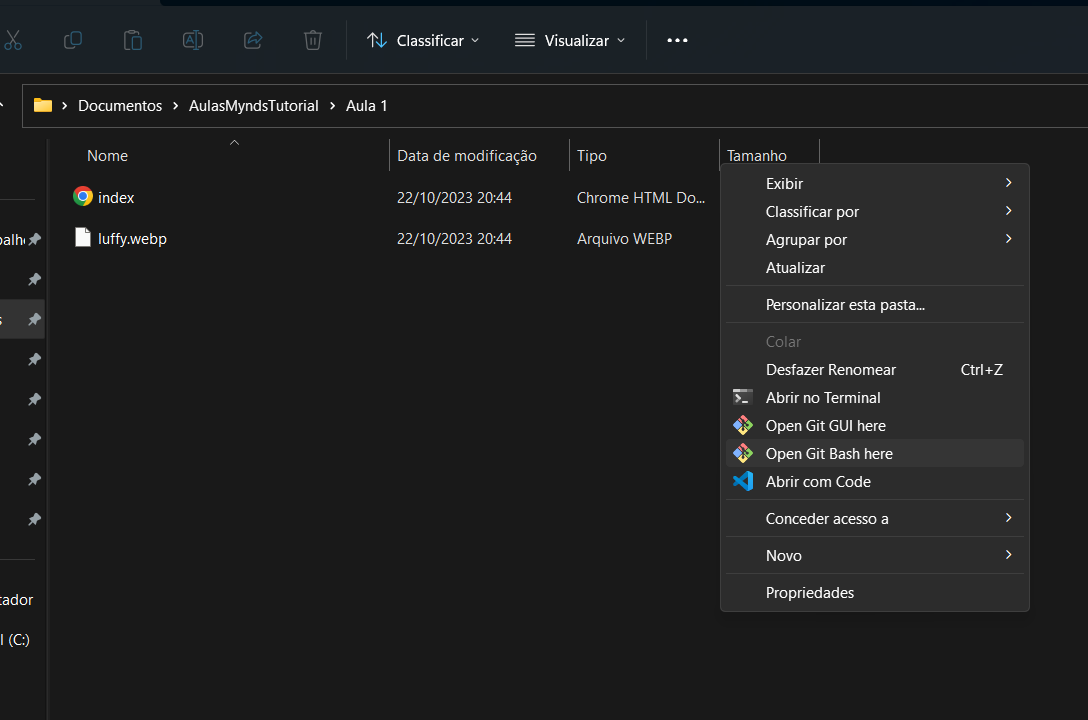
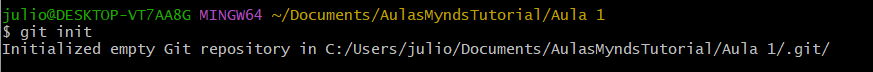
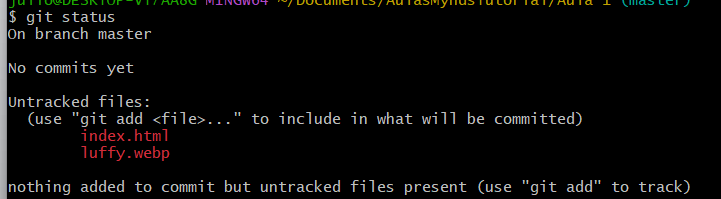
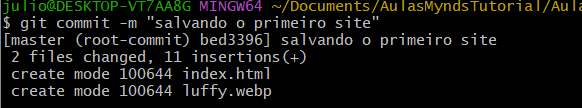
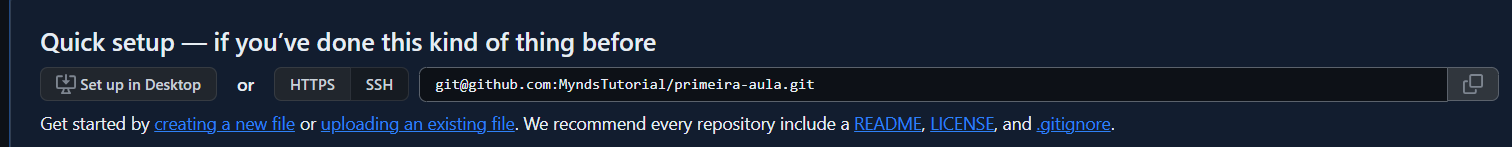
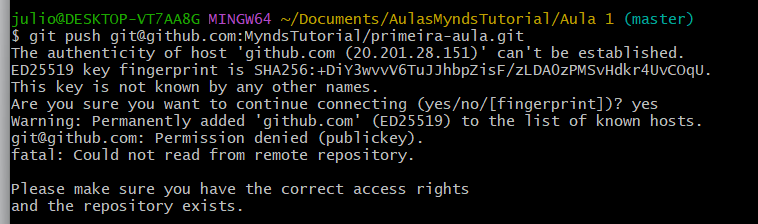
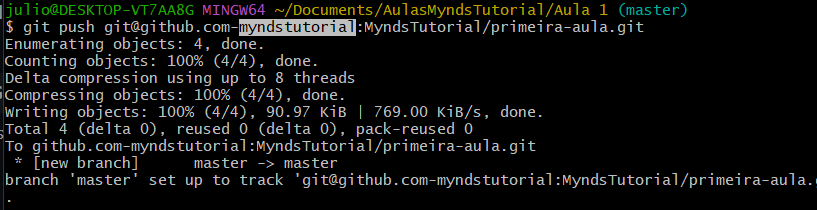
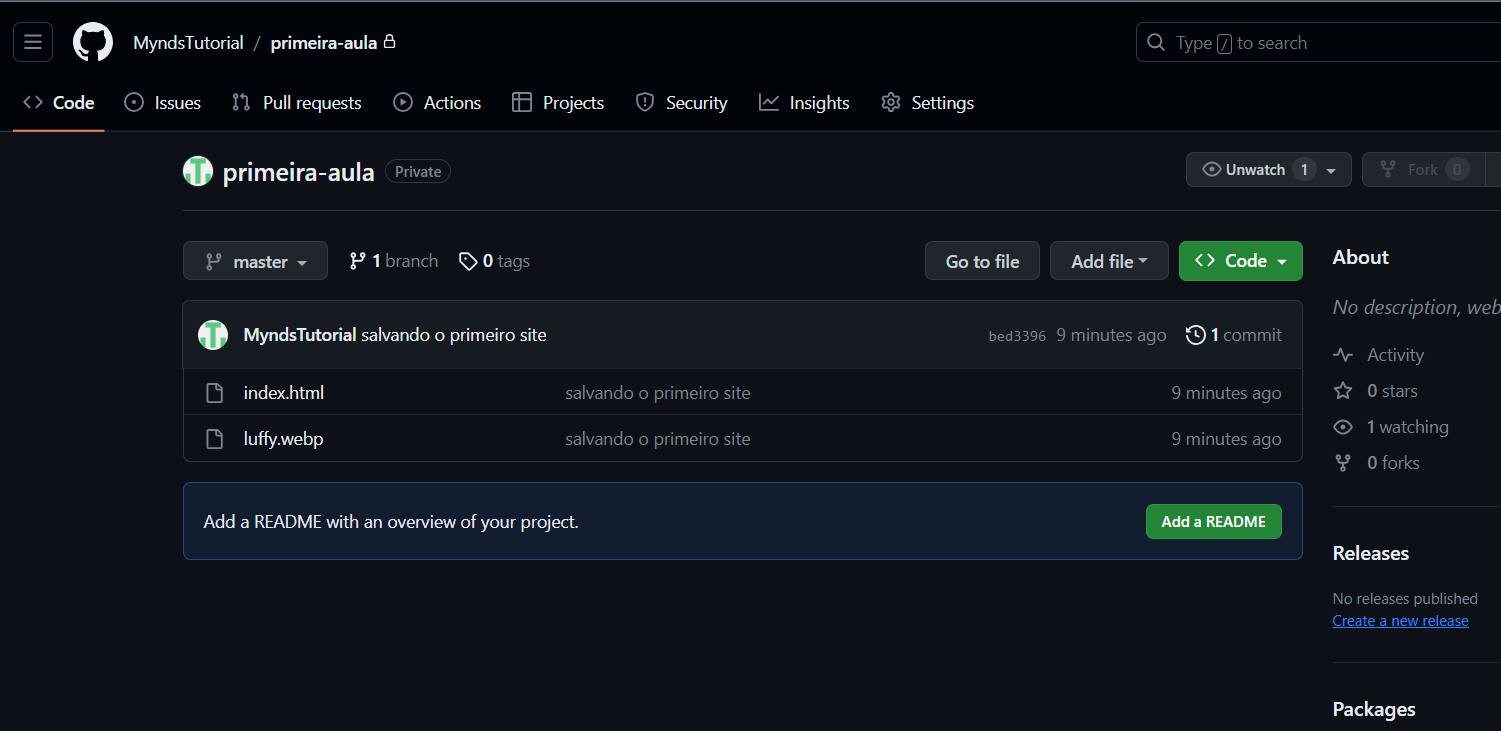
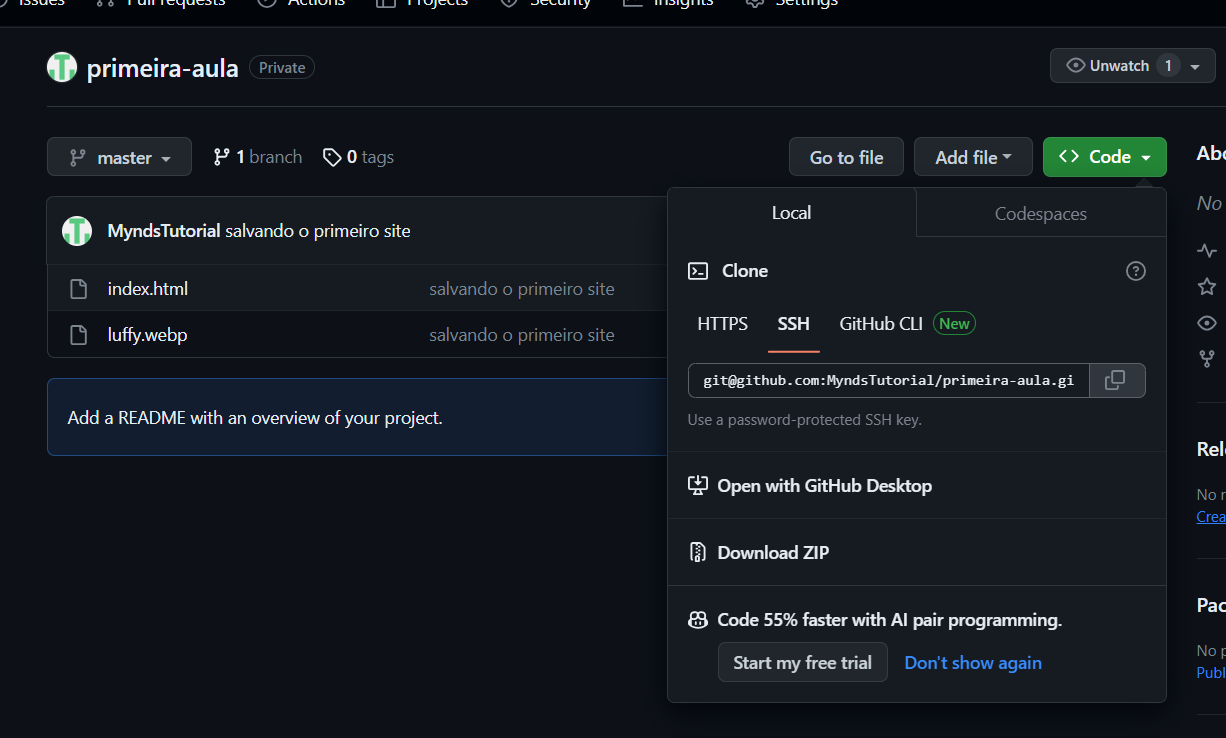
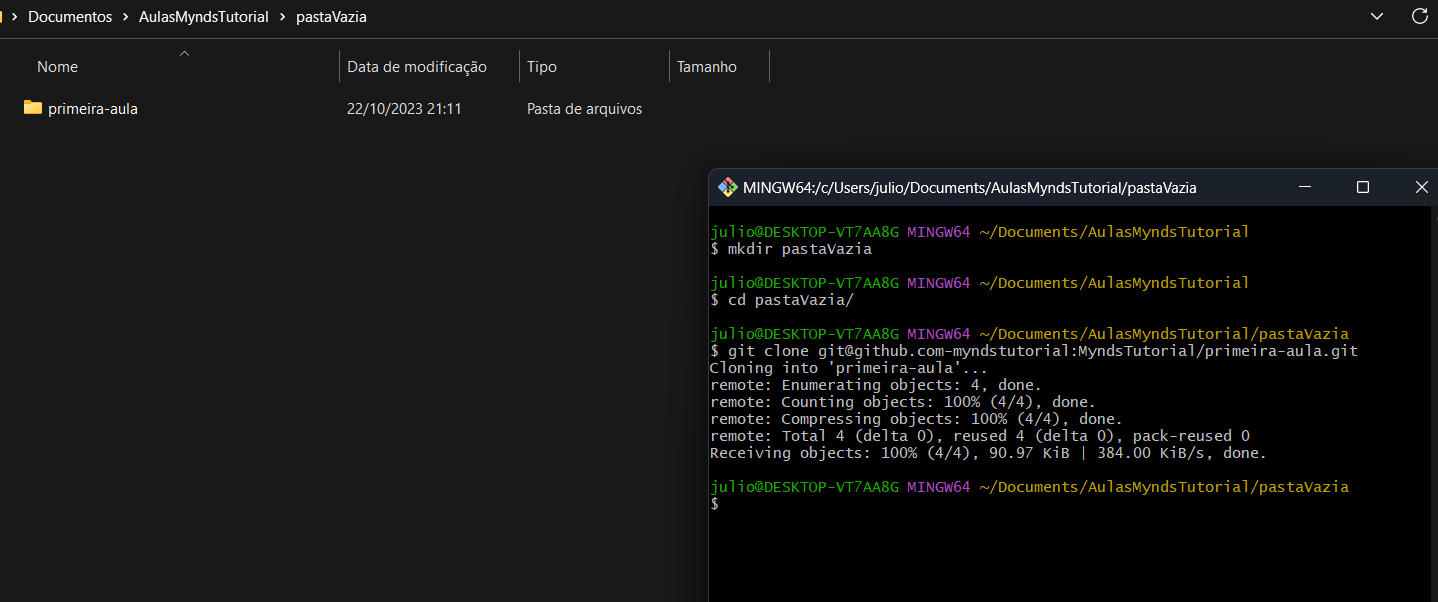


1. Agora, vamos criar um arquivo chamado index.html:  
   
2. Este primeiro ícone irá criar um arquivo, chamaremos de index.html. Repare que se eu digitar **!** ele vai me dar um template para poder programar:  
   
3. Vamos entender como funciona o HTML:

HTML é uma linguagem de marcação, então ele não serve para poder “programar” sites, ele apenas mostra uma estrutura e nós seguimos essa estrutura. Vamos começar com o início, todo site é dividido em duas partes, o **<head> </head>** e o **<body> </body>**. Dentro do **head** nós vamos colocar tudo que não é referente às informações que não fazem parte do corpo do site, como por exemplo, nomes, informações extras como links ou meta-dados, Já dentro do **body**, nós colocamos o corpo do site, como as imagens, os botões entre todas as outras coisas que existem em um site.

1. Tags: se você reparar, a maioria das palavras no html começam com < e terminam com > isso se chama marcação, toda palavra no html que for ser parte do nosso site vai ser desta maneira:  
   <tag>Conteudo que não faz parte do html </tag>
2. Vamos abrir o site no navegador, clique no index.html com o botão direito e clique nesta opção: CopyPath
3. Cole no navegador, que irá abrir uma tela em branco com o nome Document na parte superior:
4. Vamos começar alterando o nome do site:  
     
   Ao dar F5 e reiniciar o site verá que o título mudou:  
   
5. Agora vamos às Tags que aprenderemos na aula de hoje:
   1. **<h1> </h1>**  
      É o título maior do site. A onde ele sempre terá o maior tamanho por padrão.
   2. **<h2> </h2>**   
      É o subtítulo. A mudança entre ele e o título é o tamanho da letra.
   3. **<hr/>**  
      Linha de separação onde faz uma linha no site.
   4. **<img src="./nomeDaImagem.png" alt="descrição">**Tag de imagem onde nós precisamos colocar onde ela está salva. Ou seja, ‘./’ representa que está salva na mesma pasta do projeto. Ao final, é o nome do arquivo de imagem que queremos mostrar.
6. Então este código:  
   
7. Vira este site:  
   
8. Vamos postar este site no Github. Se você digitar na parte superior, na URL, **repo.new**, onde se coloca os sites, você irá parar nesta tela:
9. As únicas coisas que são preciso fazer nesta tela são: o nome do repositório e clicar no botão de Create Repository, então irá aparecer esta tela:



1. É importante saber que este site que estamos usando o Github.com é como se fosse uma pasta online, onde podemos salvar o projeto que estamos utilizando. Cada “salve” que damos no nosso código nós chamamos de commit.
2. Abra o Git Bash Here na pasta onde está salvo o seu código:  
   
3. Comandos do Git:
   1. git config --global user.email "[myndstutorial@gmail.com](mailto:myndstutorial@gmail.com)": configura seu email para os commits
   2. git config --global user.name "MyndsTutorial": configura o nome do usuário que vai aparecer no commit, é obrigatório ser o nome do Github: 
   3. git config --global push.autoSetupRemote true: este comando **desativa a opção de upstream facilita os** **push**:  
      
   4. git init: inicia uma pasta chamada .git que serve para salvar seus arquivos. Esta pasta é obrigatória para poder salvar o código:
   5. git status: ele mostra para você a situação atual do projeto:  
        
      Repare que ele diz que nós precisamos adicionar as modificações feitas no código.
   6. git add .: este comando significa que vamos adicionar todas as modificações:
   7. git commit -m “salvando o site da primeira aula”: este comando serve para salvar o código e guardar uma mensagem com o salve:
   8. git push <link do repositório>: este código serve para enviar o link do repositório, porém é extremamente importante lembrar que quando estiver na escola você precisa configurar o host. Para pegar o link do repositório você coloca em **ssh e copia**:  
      Agora é só dar o comando e:  
      Repare que mesmo eu dando “yes”, foi negado o acesso e isso é porque eu não coloquei o host que configurei lá atrás no ‘config’, então na escola é necessário:  
        
      E repare que agora deu tudo certo, e temos agora um site online:  
        
      Mas agora como funciona para pegar este site online e colocar no nosso computador de casa?
   9. Primeiro copiamos o link do repositório:  
      
   10. E depois na nossa pasta nova, clonamos o projeto com o comando:  
       **git clone <link-do-repositorio>**  
         
       Agora temos a nossa pasta clonada em outro computador.

De lembrete mais importante sobre o Github, lembrem-se:

Chave ssh com -nomeDoAluno só na escola.

Em casa não precisa ter o -nomeDoAluno, porque o comando do tutorial já faz automaticamente a configuração.

Recursos

* [Git - Downloads](https://git-scm.com/downloads)
* [Download Visual Studio Code - Mac, Linux, Windows](https://code.visualstudio.com/download)

Observação

* Todos os arquivos de configuração já estão disponíveis e o código está no repositório da Mynds, existe um Github chamado GithubTutorial onde o professor pode entender 100% de todos os detalhes e configurações necessárias a ferramenta Git.
* Na hora de explicar pros alunos sobre criptografia, vale a pena mostrar um pedaço deste vídeo (em torno de 4 minutos) sobre criptografia do cellbit:  
  [CICADA 3301 - PARTE 1 - CELLBIT](https://www.youtube.com/watch?v=Ep5qn8pLCMA)
* Links Extras de estudo sobre o Github:  
  [O QUE É GIT E GITHUB? - definição e conceitos importantes 1/2](https://youtu.be/DqTITcMq68k?si=0gEyQuywjrcMM4jI)  
  [COMO USAR GIT E GITHUB NA PRÁTICA! - desde o primeiro commit até o pull request! 2/2](https://youtu.be/UBAX-13g8OM?si=9LqJceHGIfj-osHz)

Tarefas

* Instalar o Visual Studio Code em casa e tentar montar um site simples com o código parecido:  
  