Git e GitHub

* O que é GitHub?

O GitHub nada mais é uma plataforma/Rede social para programadores, a base da plataforma é de armazenar e compartilhar códigos de programação. ou seja, é uma ótima plataforma para divulgar seus projetos, tanto pessoais quanto profissionais.

* O que é GIT?

O GIT nada mais é que um sistema de versionamento, um sistema de controle que tem a função de registrar toda e qualquer alterações que o código possa sofrer, armazenando essas informações, assim caso algum problema futuro o desenvolvedor pode regredir a versões anteriores de sua aplicação de modo rápido e simples.

* A diferencia entre o GIT e GitHub

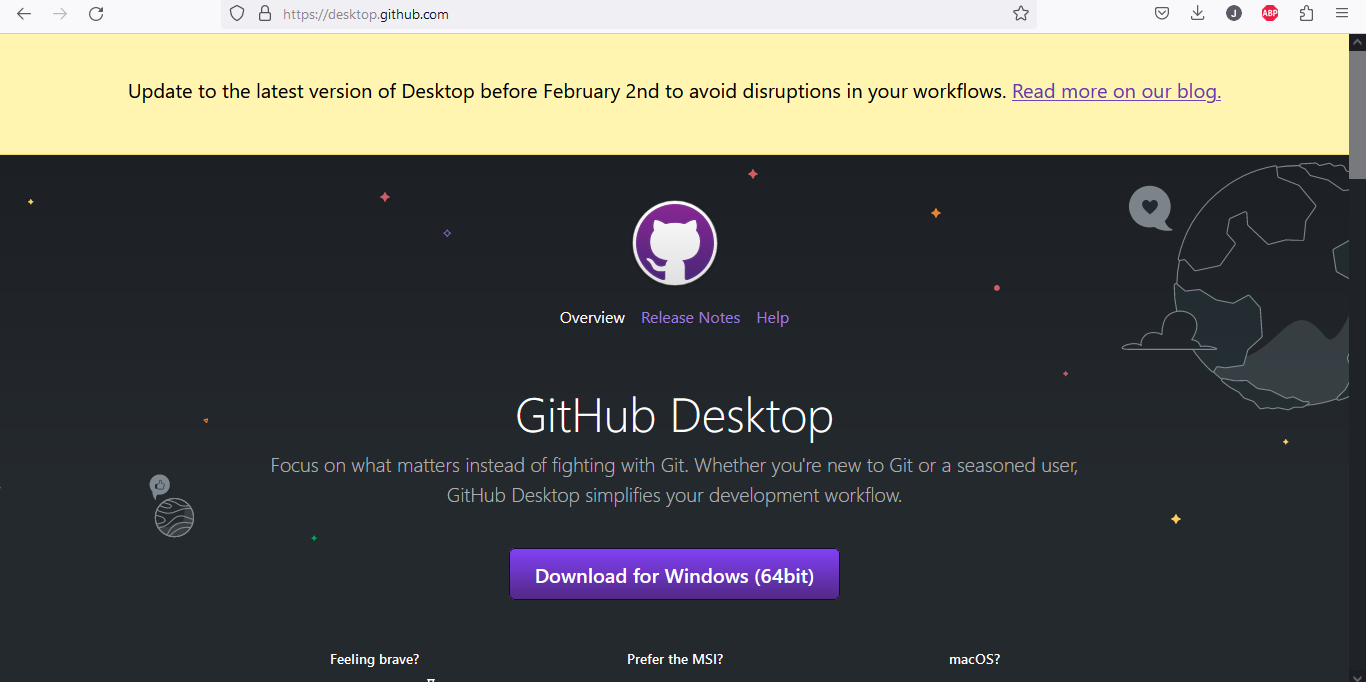
Enquanto o GitHub funciona de forma semelhante uma rede social, o GIT é um sistema para versionamento, uma espécie de biblioteca do seu próprio código.

* Relação entre o mercado e GitHub

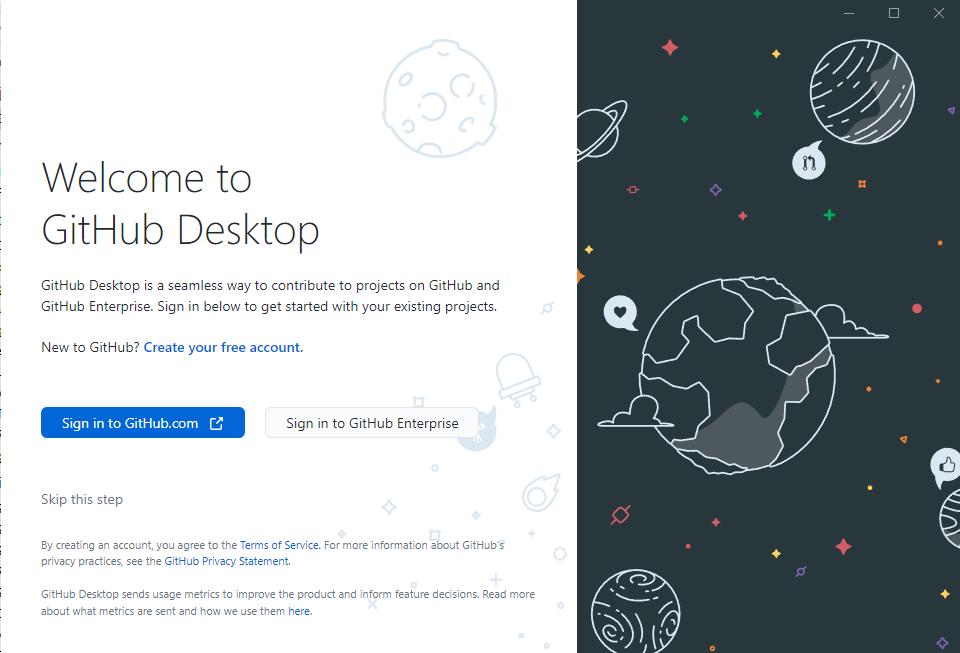
Como dito anteriormente, o GitHub é uma plataforma para que desenvolvedores e estudantes possam expor seus projetos, sendo eles estudos ou trabalhos, servindo como portifólio, esse é o motivo de ser tão requisitado em vagas para programação, seja front-end, back-end ou full Stack.

Tutorial de como usar o GitHub (Desktop)

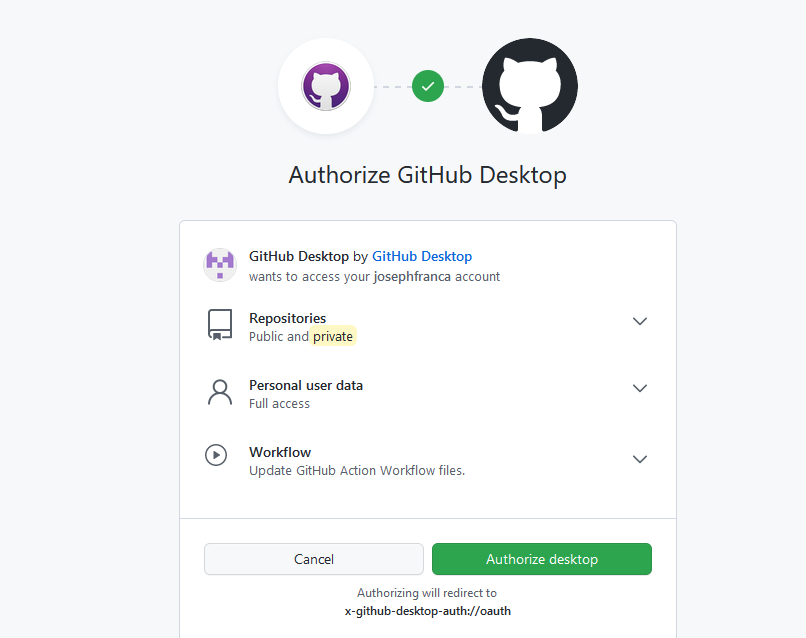
1 a primeira coisa a se fazer é baixar o programa, o link para download é: <https://desktop.github.com/>



Esse é o site de onde você deve fazer o download.

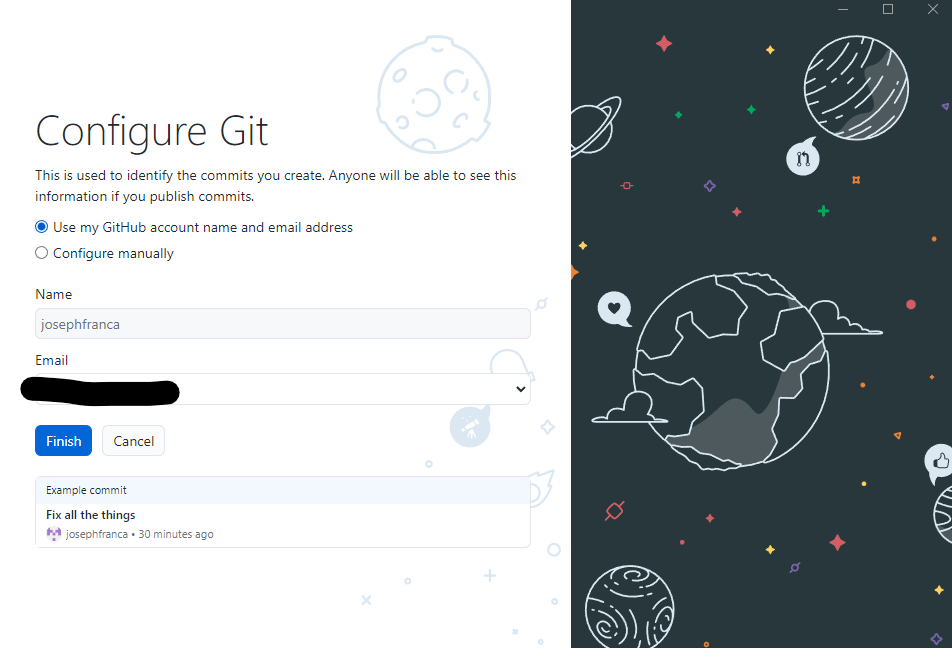
2 com o arquivo executável GitHubDesktopsetup -x64 de um duplo clique, ao abrir o programa essa é a tela que irá aparecer: 

Que conta com alguns botões, sendo um deles para logar em uma conta GitHub convencional e a outra para logar em uma conta empresarial.

Assim que você clicar para logar com sua conta GitHub essa tela aparecerá: 

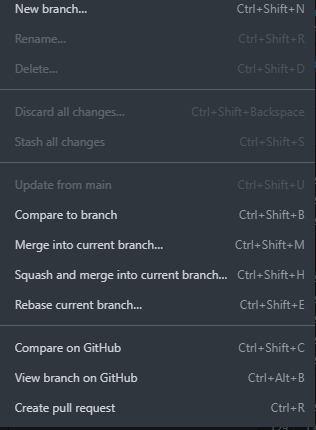
Pedindo que você autorize o seu desktop.

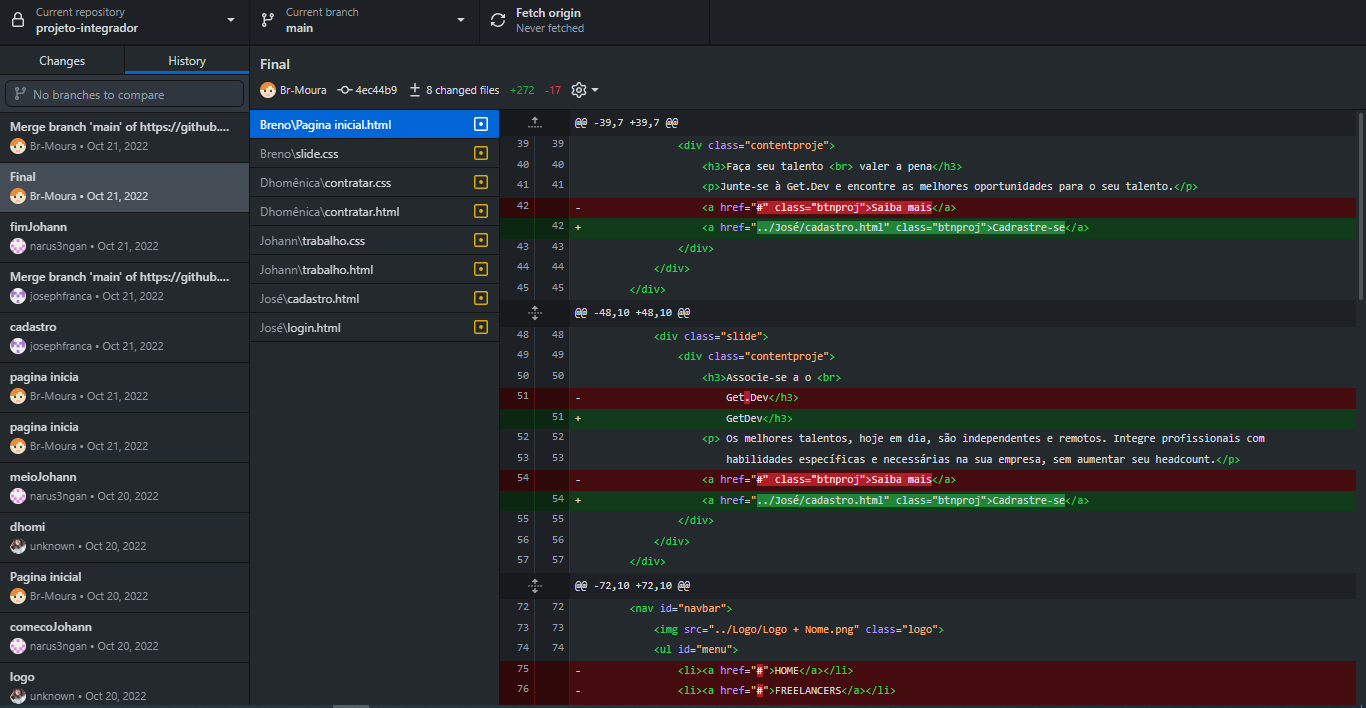
Logo em seguida aparece a tela para configurar o GIT



Na aba Arquivos (ou files, eu não consegui encontrar a opção para que o programa fique em português brasileiro), encontra-se três opções: “new repository” que é para criar um novo repositório, “add local repository “para adicionar um repositório local e por fim, “clone repository” para clonar um repository já existente. (um repositório contém todos os arquivos do projeto e seu histórico de alterações de cada arquivo, assim podendo ser gerenciado pelos desenvolvedores).

Na aba Branch você encontrará opções para um novo Branch, renomear, deletar, comparar os branchs, comparar no GitHub entre outros.





Essa print é de um projeto que eu e meu antigo grupo fizemos meses atrás, na versão desktop, o histórico de Branch.

Branch é como uma ramificação do seu projeto que aponta as alterações feitas nos arquivos de projeto como se fosse uma linha do tempo, assim quando necessário fazer uma alteração ou adição é gerado um novo Branch, caso você volte atrás ou de alguma maneira precise voltar para versão anterior será usado a versão anterior do seu código através da linha do tempo anterior.

* Comandos importantes e suas funcionalidades

Caso por algum motivo você use a versão em linha de comando no lugar da versão GUI.

Para criar um novo repositório: git init

Para adicionar um diretório especifico: git add nome\_diretorio

Para adicionar todos os arquivos e/ou diretórios: git add

Para comitar arquivo/diretório: git commit meu\_arquivo

Para remover arquivo: git rm nome\_arquivo

Para remover o diretório: git rm -r nome\_diretorio

Para visualizar histórico: git log

Para exibir os repositórios remotos: git remote -v

Para enviar arquivos/diretórios para o repositório remoto: git push -u master (depois desse primeiro push não é mais necessário colocar -u master)

Para atualizar os arquivos no Branch atual: git pull