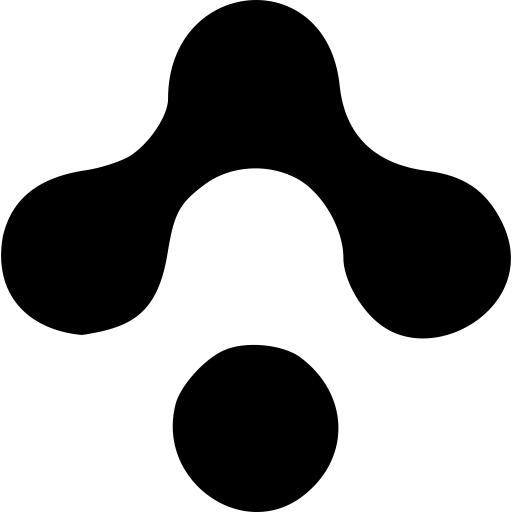


Técnico de Gestão e Programação de Sistemas Informáticos

Prova de Aptidão Profissional

**Jet Hire**



Cantanhede

Tomé Almeida, 12º TGPSI, Nº 15

2022 / 2025



Escola Técnico Profissional de Cantanhede

Técnico de Gestão e Programação de Sistemas Informáticos

Prova de Aptidão Profissional: Jet Hire

Equipa de Acompanhamento: Michael Teixeira, Elisabete Cavaleiro e Ana Marques

Cantanhede

2024/2025

CRIATIVIDADE É SÓ CONECTAR AS COISAS.

***Steve Jobs***

**Agradecimentos**

**Índice Geral**

**Índice de Figuras**

**Índice de Tabelas**

# Introdução (Português)

- Nota introdutória (PAP)

- Descrição Detalhada do Projeto

- Explicação das Fases do Projeto (3 Fases) e das atividades definidas em cada fase.

- Explicação do Relatório (Referir o nome dos cabeçalhos e explicar o que se pode encontrar em cada um.

# Introduction (Inglês)

# Planeamento

## Fundamentação

Razões que levaram à escolha do tema.

|  |  |
| --- | --- |
| **Objetivos** | |
| **Gerais** | **Específicos** |
| Terminar o Curso | * Ter boa nota na PAP; * Terminar o curso com uma boa média. |
| Desenvolvimento da PAP | * Aumentar os meus conhecimentos em Programação; * Melhorar as minhas competências de concretização de projetos; * Adquirir conhecimentos na Linguagens (Quais?) * Aplicar os conhecimentos adquiridos ao longo dos três anos do curso. * Explorar novas tecnologias utilizadas para o desenvolvimento da PAP. * Coloca a aplicação na Play Store. * Colocar o site online para que seja visível por muitos utilizadores. * Desenvolver competências na área de desenvolvimento de projetos de software. |

## Recursos

### Hardware

Desktop pessoal

- AMD Ryzen 5 7600X @ 4.7Ghz

- RAM 32Gb DDR5 5600Mhz

- Radeon RX 6700 10Gb

- Disco SSD 512Gb

- Disco HDD 2Tb

- Linux Ubuntu 22.04.1

Portátil pessoal

- Apple Silicon M3 Chip

- RAM 16Gb

- Disco SSD 512Gb

- macOS Sequoia 15.01

### Software

- Microsoft Office 2021 Pro Plus

Requisitos Mínimos

- Sistema Operativo – Windows 10+ ou macOS 11.6+

- Processador – 1.6 GHz ou superior

- Memória RAM – 4Gb (64 bits)

- Disco – 4Gb

- Resolução Mínima – 1280px X 768px

- Visual Studio Code

Requisitos Mínimos

- Processador – 1.6 GHz ou superior

- Memória RAM – 1Gb

- Github Desktop

Requisitos Mínimos

- Sistema Operativo – Windows 10+ (64 bits) ou macOS 10.5+

- Arc

Requisitos Mínimos

- Sistema Operativo – Windows 10 19H1+ ou macOS 13+

- Google Chrome

Requisitos Mínimos

- Sistema Operativo – Windows 7+ ou macOS 10.9+

- Processador – 2 GHz ou superior

- Memória RAM – 2Gb

- Krita

Requisitos Mínimos

- Sistema Operativo – Windows 8.1+

- Processador – 1.6 GHz ou superior

- Memória RAM – 4Gb

- Disco – 1Gb

- MongoDB Compass

Requisitos Mínimos

- Sistema Operativo – Windows 10+ ou macOS 11+ ou Ubuntu 16.04+

## Metodologia

Para desenvolver a Jet Hire optei por usar a metodologia Kanban. Esta metodologia é muito visual e facilita o controlo da gestão de tarefas tanto como garante um fluxo contínuo do desenvolvimento.

A metodologia Kanban consiste na utilização de um quadro de tarefas para monitorizar o fluxo de trabalho. As tarefas são categorizadas e assim distribuídas respetivamente.

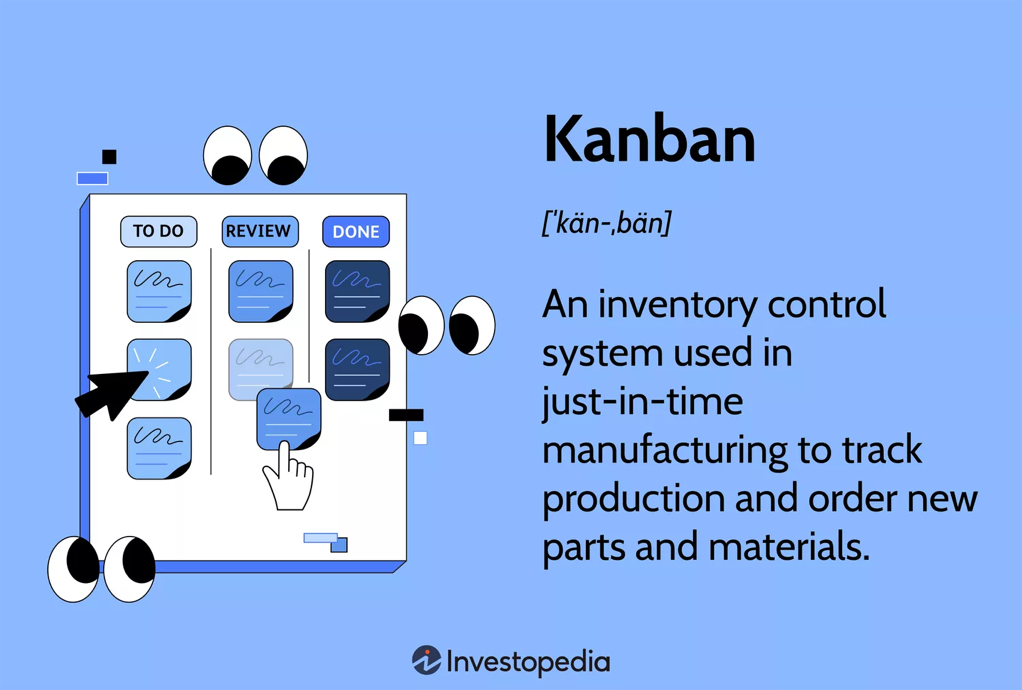


Figura 1 - Metodologia Kanban

*Fonte:* [*https://www.investopedia.com/terms/k/kanban.asp*](https://www.investopedia.com/terms/k/kanban.asp)

Escolhi esta metodologia porque ajuda a adaptar o trabalho conforme as necessidades mudam e dá uma visão clara do que se está a fazer, do que já se fez e do que se vai fazer. Como os requisitos da aplicação podem evoluir ao longo do tempo, o Kanban permite fazer estes ajustes de maneira fácil. Além disso, esta metodologia ajuda a identificar e resolver problemas rapidamente, garantindo que o projeto avança de forma organizada e eficiente.

## Atividades

A Prova de Aptidão Profissional irá ser desenvolvida em três fases. Cada fase é constituída por várias atividades. A primeira fase, a Fase de Definição tem como principais atividades: pesquisa bibliográfica, análise do sistema, análise dos requisitos e planeamento do projeto. A segunda fase, a Fase de Desenvolvimento é constituída pelas seguintes atividades, desenho, codificação e testes. A terceira e última fase, a Fase de Manutenção é onde encontramos as atividades de correção, adaptação e evolução.

### Fase de Definição

► Pesquisa Bibliográfica – etapa inicial do trabalho com o objetivo de reunir todas as informações necessárias para o desenvolvimento do projeto.

► Análise do Sistema – atividade onde se realizou o levantamento de todas as funcionalidades do projeto e a forma como elas vão funcionar. Nesta atividade foi também realizada a escolha das tecnologias para a realização do mesmo.

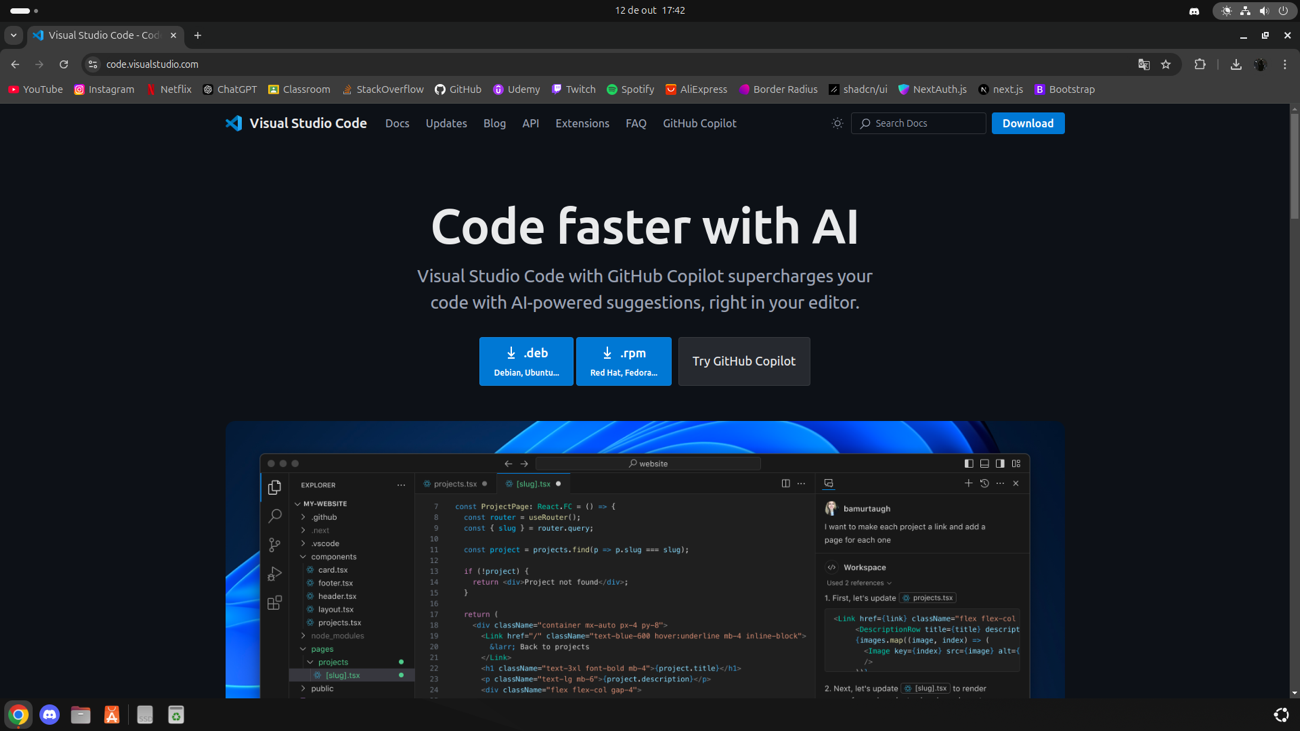
► Análise dos Requisitos – nesta atividade foi necessário fazer a verificação dos requisitos mínimos do Software necessário para implementação do projeto.

► Planeamento do Projeto – com esta atividade foi desenvolvido o cronograma em que se estabelece as balizas temporais para cada uma das fases do projeto.

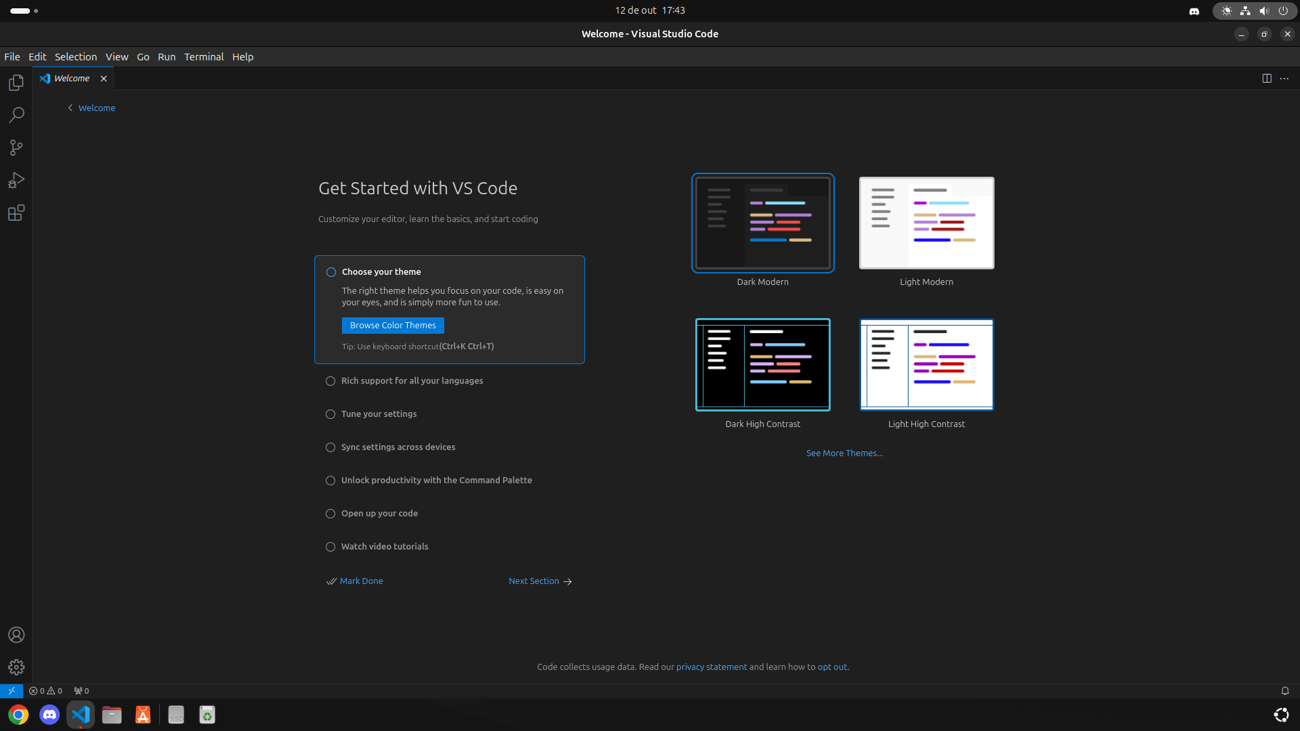
**Instalações**

Visual Studio Code

Para começar a instalação do Visual Studio Code é preciso fazer *download* do instalador através do [site oficial](https://code.visualstudio.com/).



Depois de executar o instalador e a instalação estiver pronta, a página inicial abre.

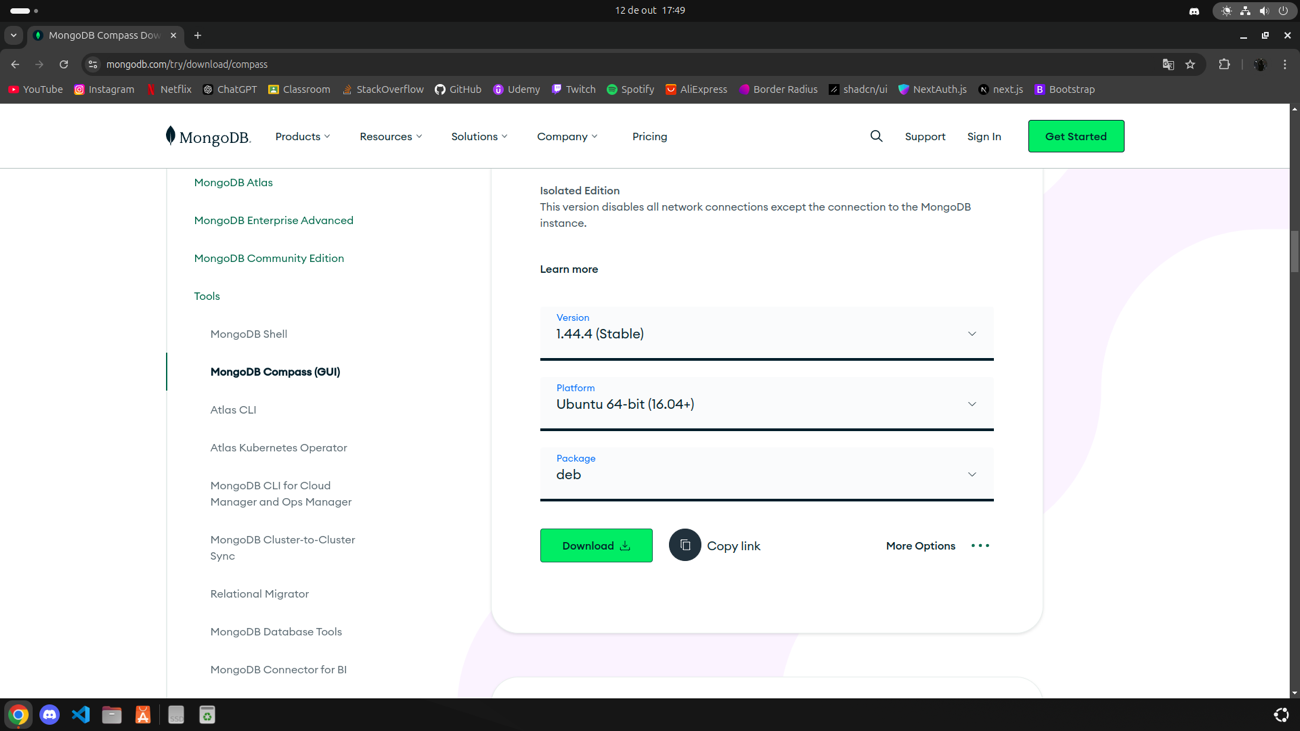


Para personalizar o Visual Studio Code a meu gosto e de acordo com as minhas necessidades adicionei as seguintes extensões:

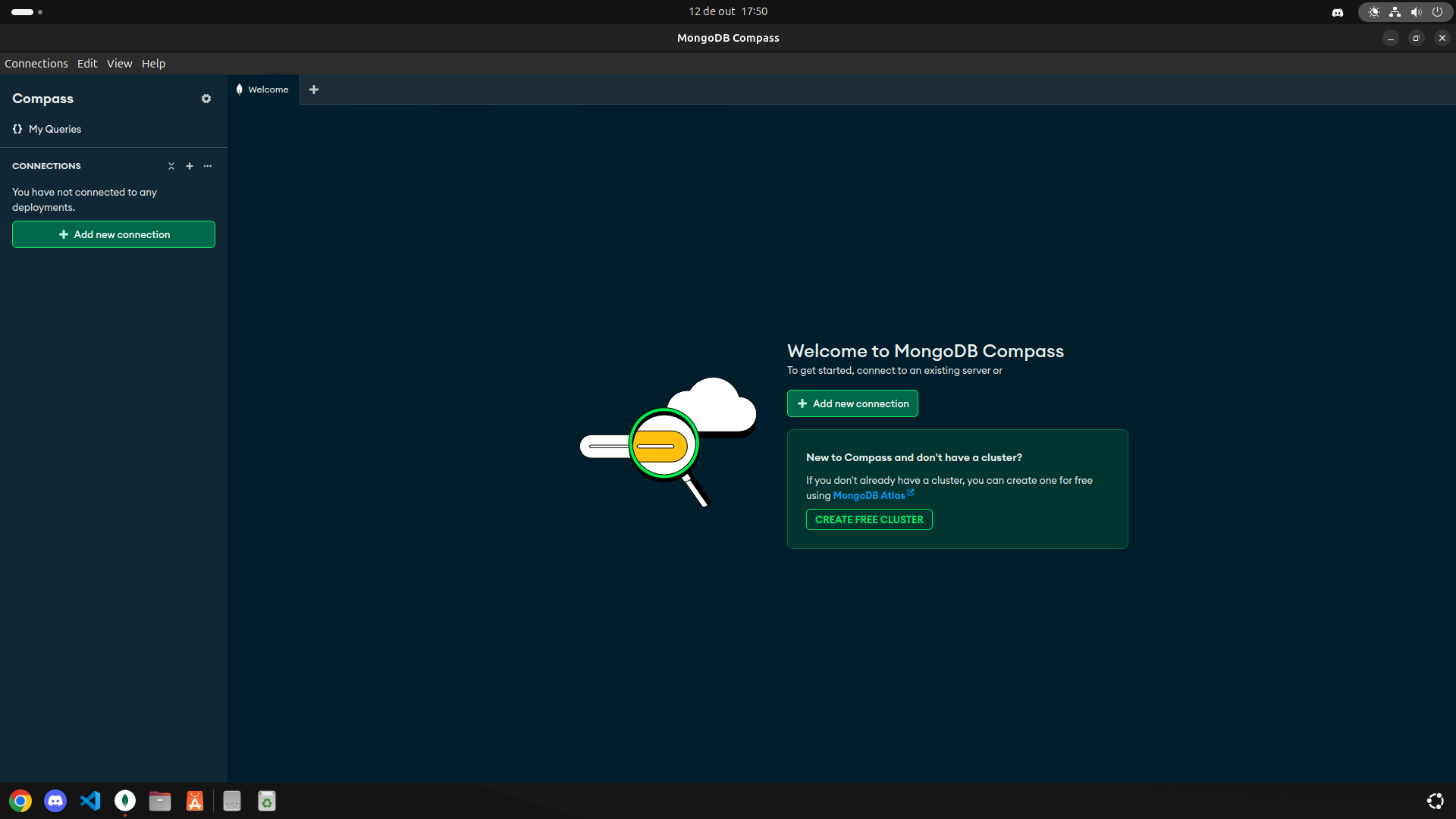
|  |  |
| --- | --- |
| * **Auto Rename Tag** * **Error Lens** * **Min Theme** * **MongoDB for VS Code** * **Symbols** |  |

MongoDB Compass

Tal como o VS Code é necessário fazer *download* do instalador através do [site oficial](https://www.mongodb.com/try/download/compass).



De seguida, é só executar o instalador.

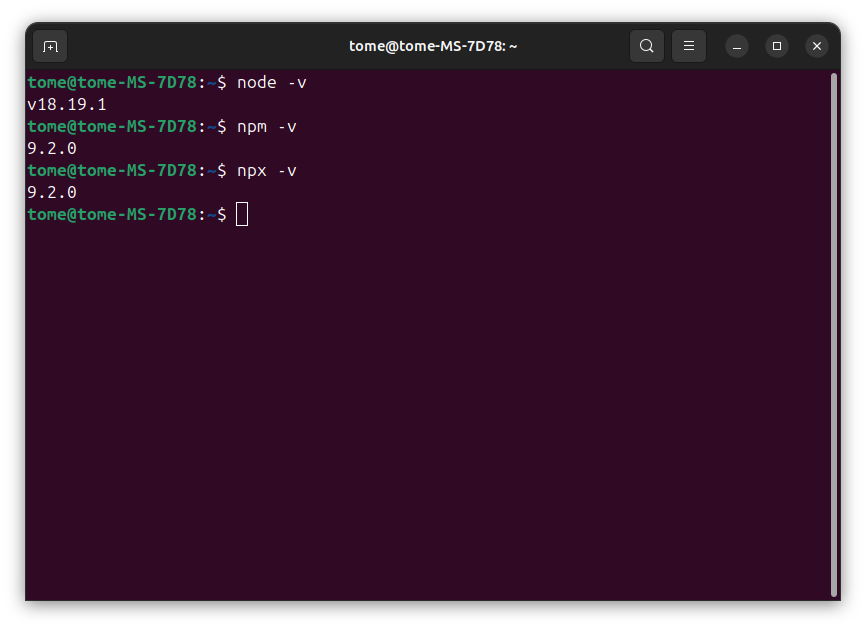


Node.js (node / npm / npx)

No terminal do Ubuntu apenas é preciso utilizar os seguintes comandos para instalar o node e o npm:

|  |  |
| --- | --- |
| ***sudo apt install nodejs*** | ***sudo apt install npm*** |
|  |  |

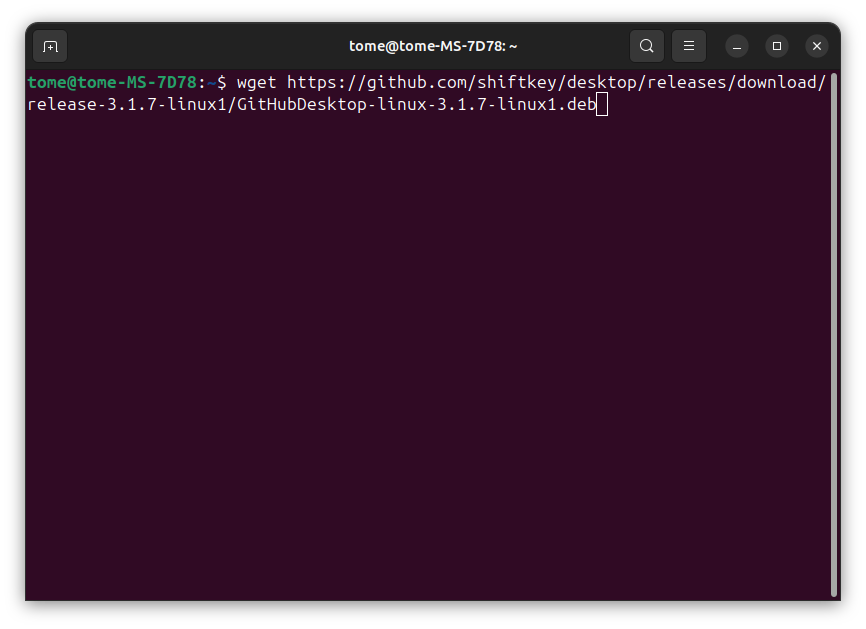
Para verificar se foi instalado corretamente executei os comandos para verificar a respetiva versão:



Github Desktop

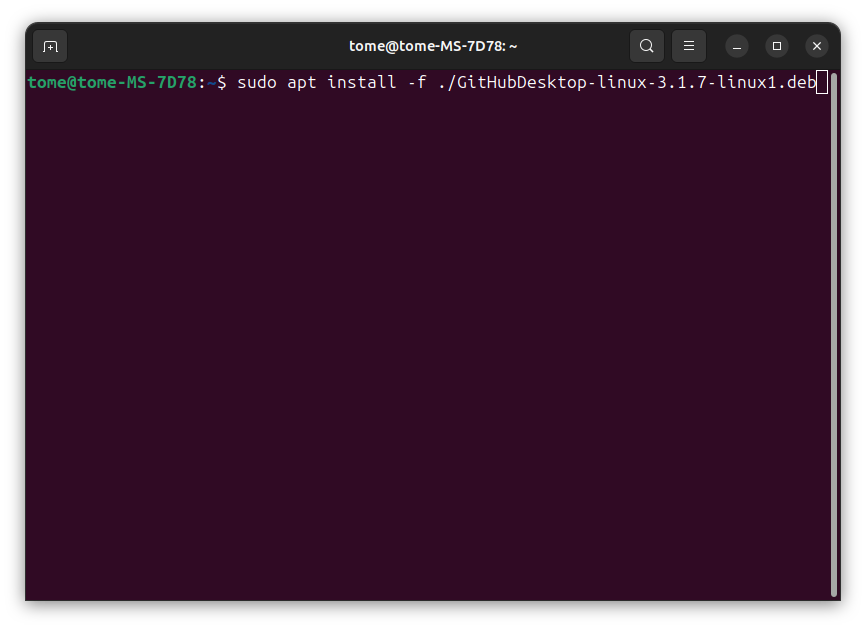
Para instalar o Github Desktop no Ubuntu é necessário fazer *download* através do terminal, visto que não existe instalador no site oficial. Para isso é preciso utilizar o seguinte comando:

*wget https://github.com/shiftkey/desktop/releases/download/release-3.1.7-linux1/GitHubDesktop-linux-3.1.7-linux1.deb*

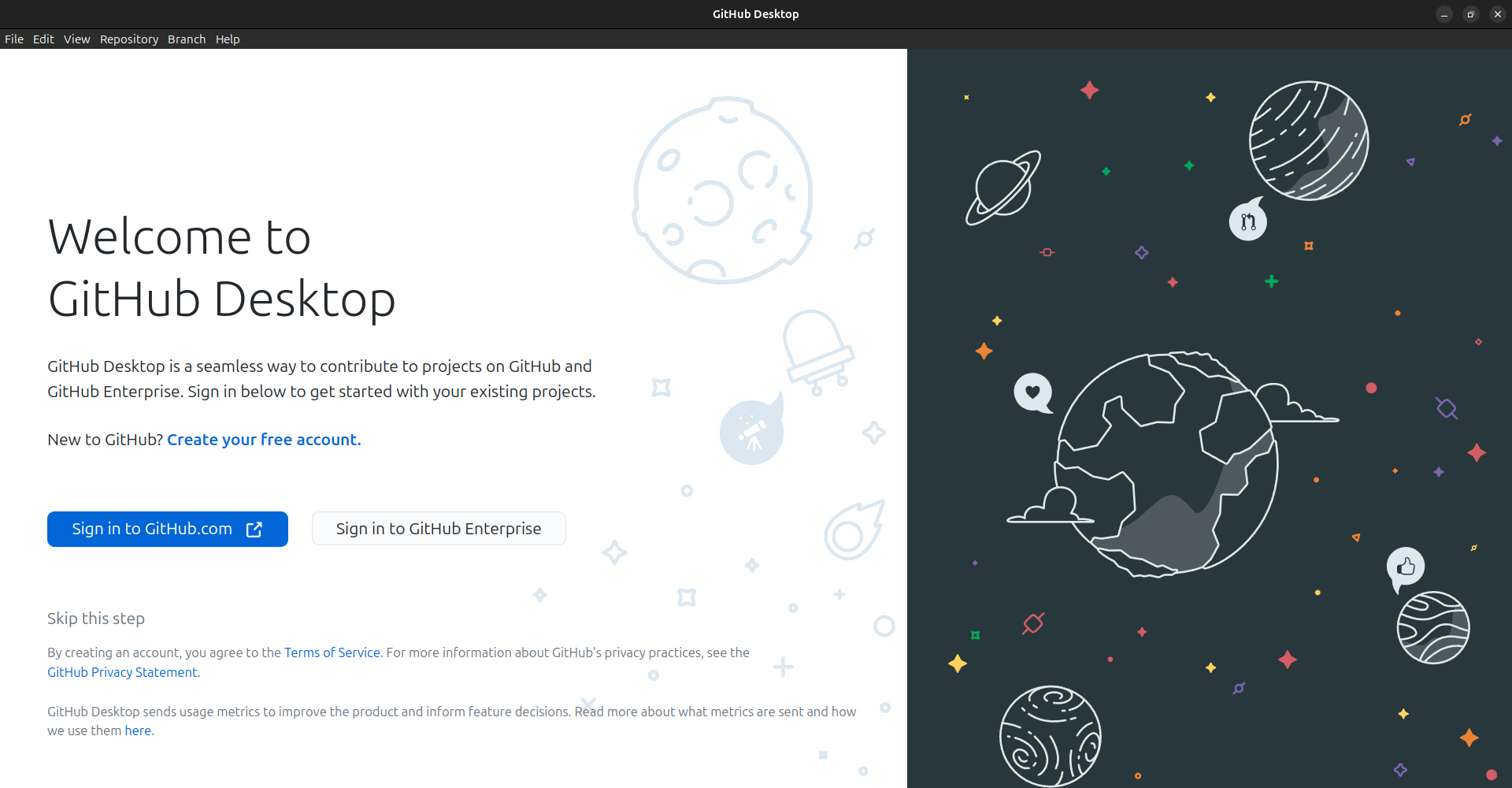


De seguida, é só executar o instalador com o comando:

*sudo apt install -f ./GitHubDesktop-linux-3.1.7-linux1.deb*

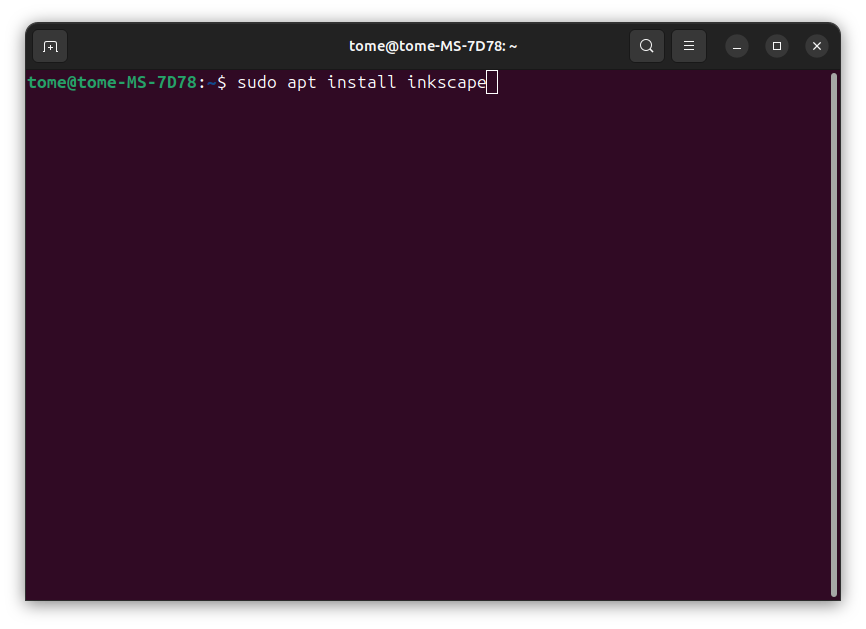


No fim da execução do instalador só falta abrir a aplicação.

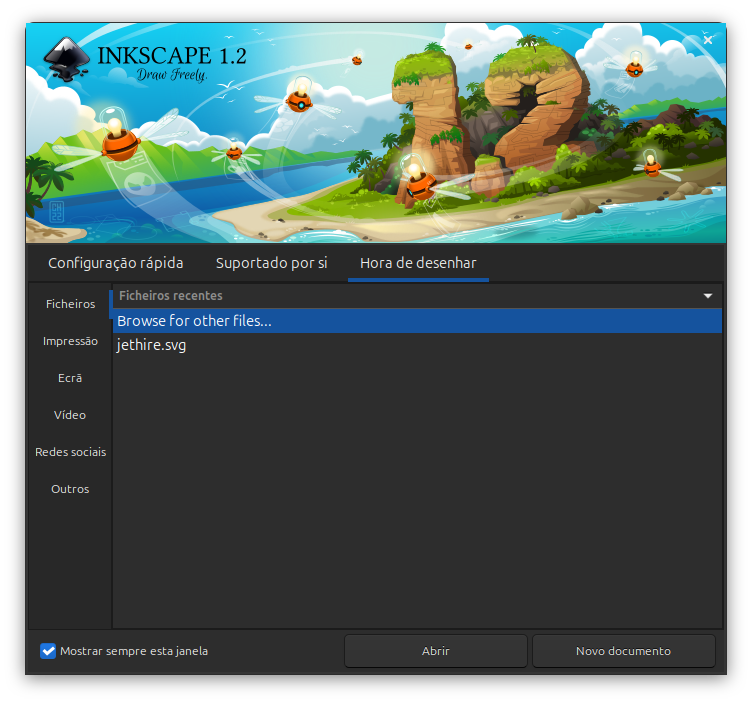


Inkscape

Para a instalação do Inkscape só foi preciso utilizar o comando *sudo apt install inkscape* no terminal.

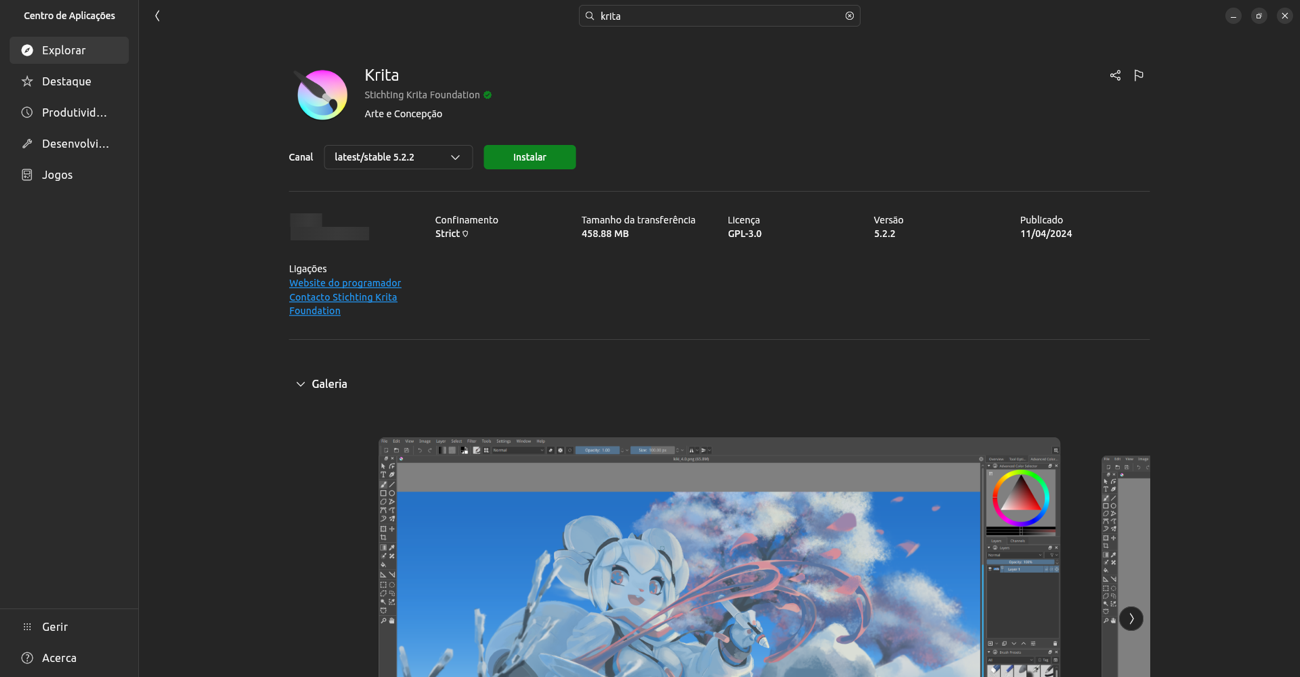


A instalação está concluída e pronta a ser utilizada.

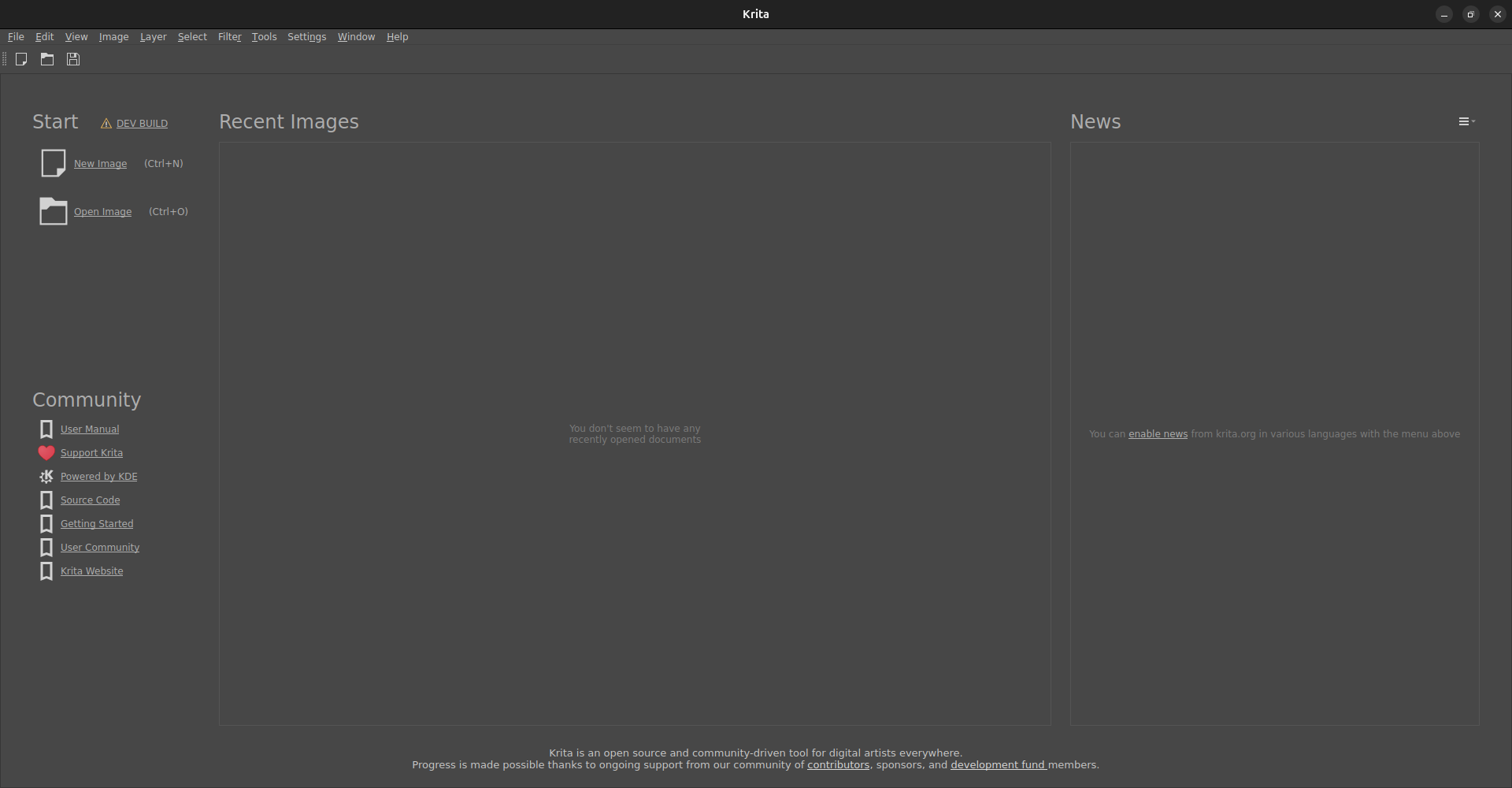


Krita

Como o Krita está disponível no Centro de Aplicações do Ubuntu apenas foi preciso instalar através do mesmo.



A aplicação está pronta a ser utilizada.



**Logótipo**

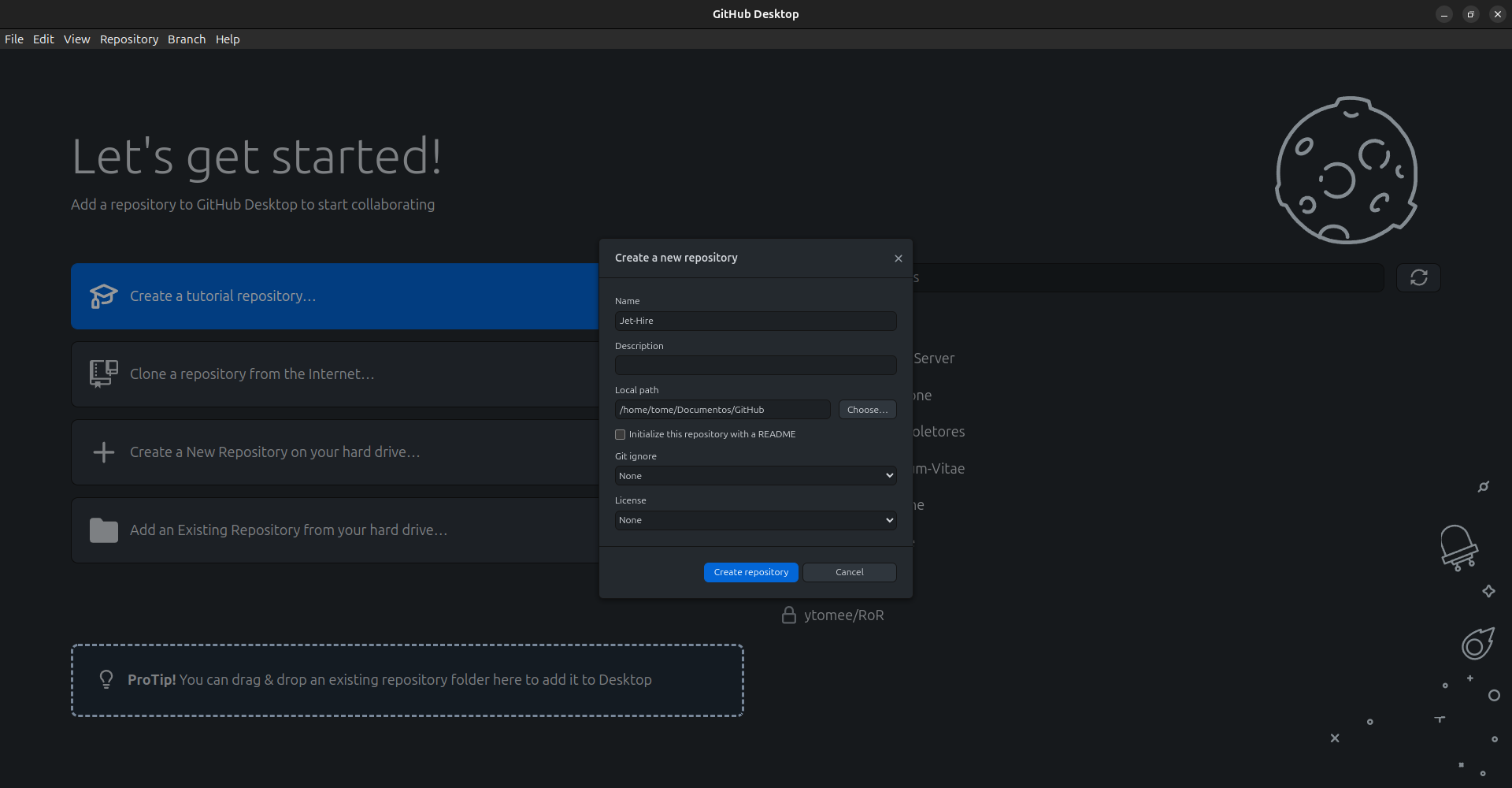
|  |  |
| --- | --- |
| Decidi criar o logótipo de uma forma completa, mas como o meu projeto consiste na inserção de novos elementos numa empresa, separei um dos quatro elementos para destacar o elemento sozinho que vai posteriormente integrar a empresa para a complementar. |  |

**Sincronização entre dois computadores**

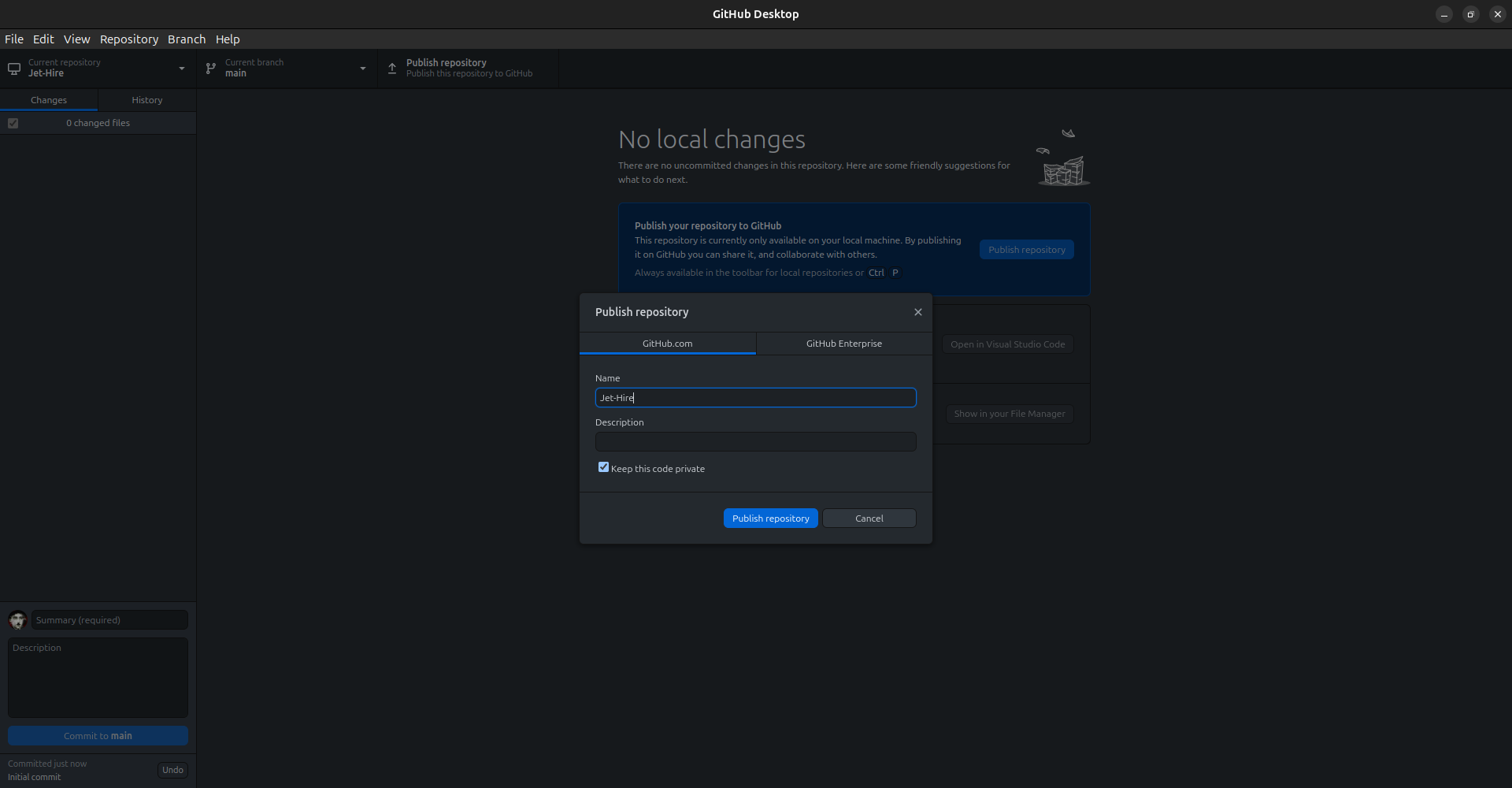
Como utilizei dois computadores para a realização projeto utilizei o GitHub Desktop para fazer esta sincronização.

**Criação do repositório**

Para começar, é preciso criar o respetivo repositório onde se vai guardar todos os ficheiros:

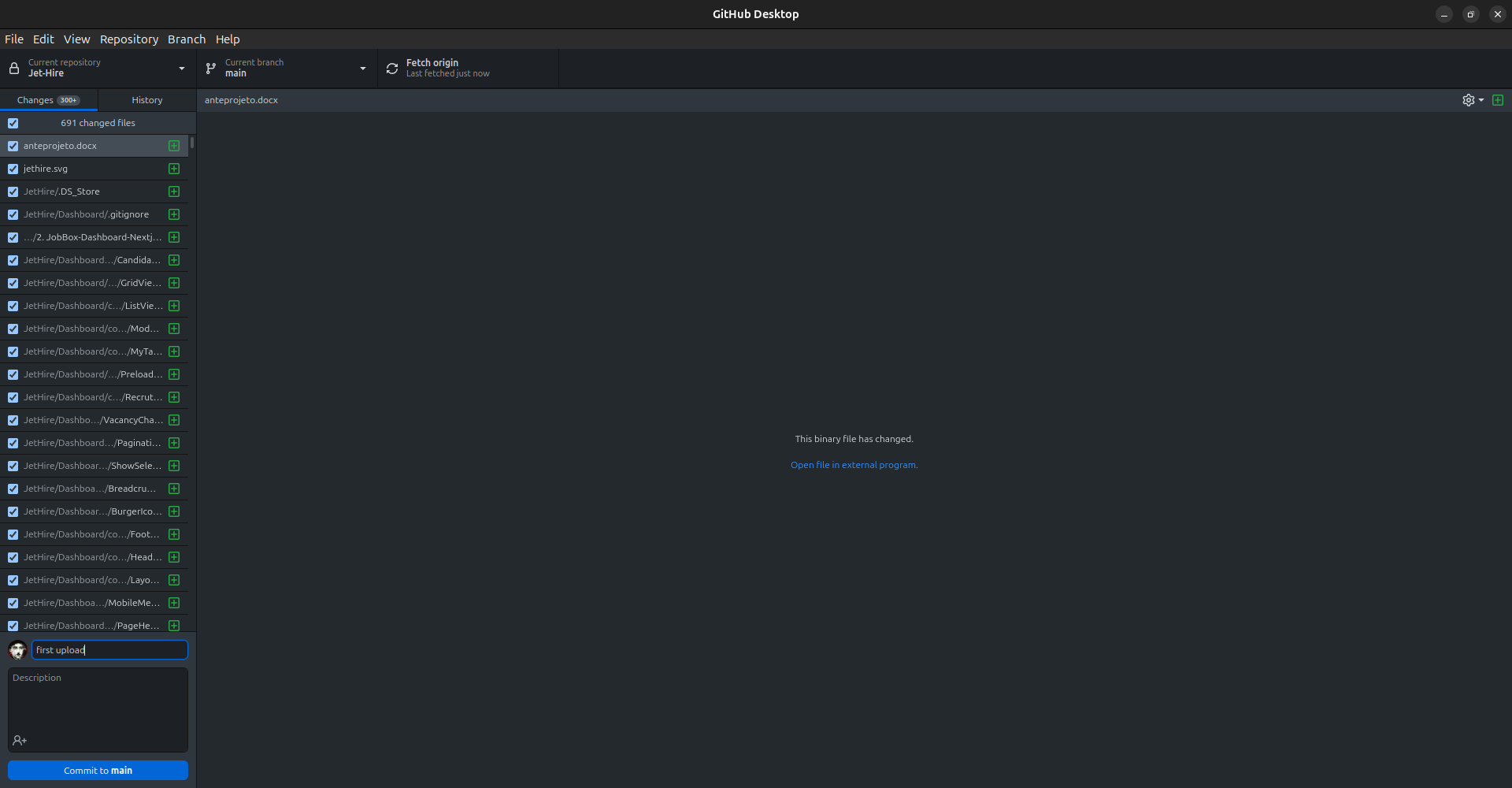


Depois da criação do repositório é preciso publicá-lo:

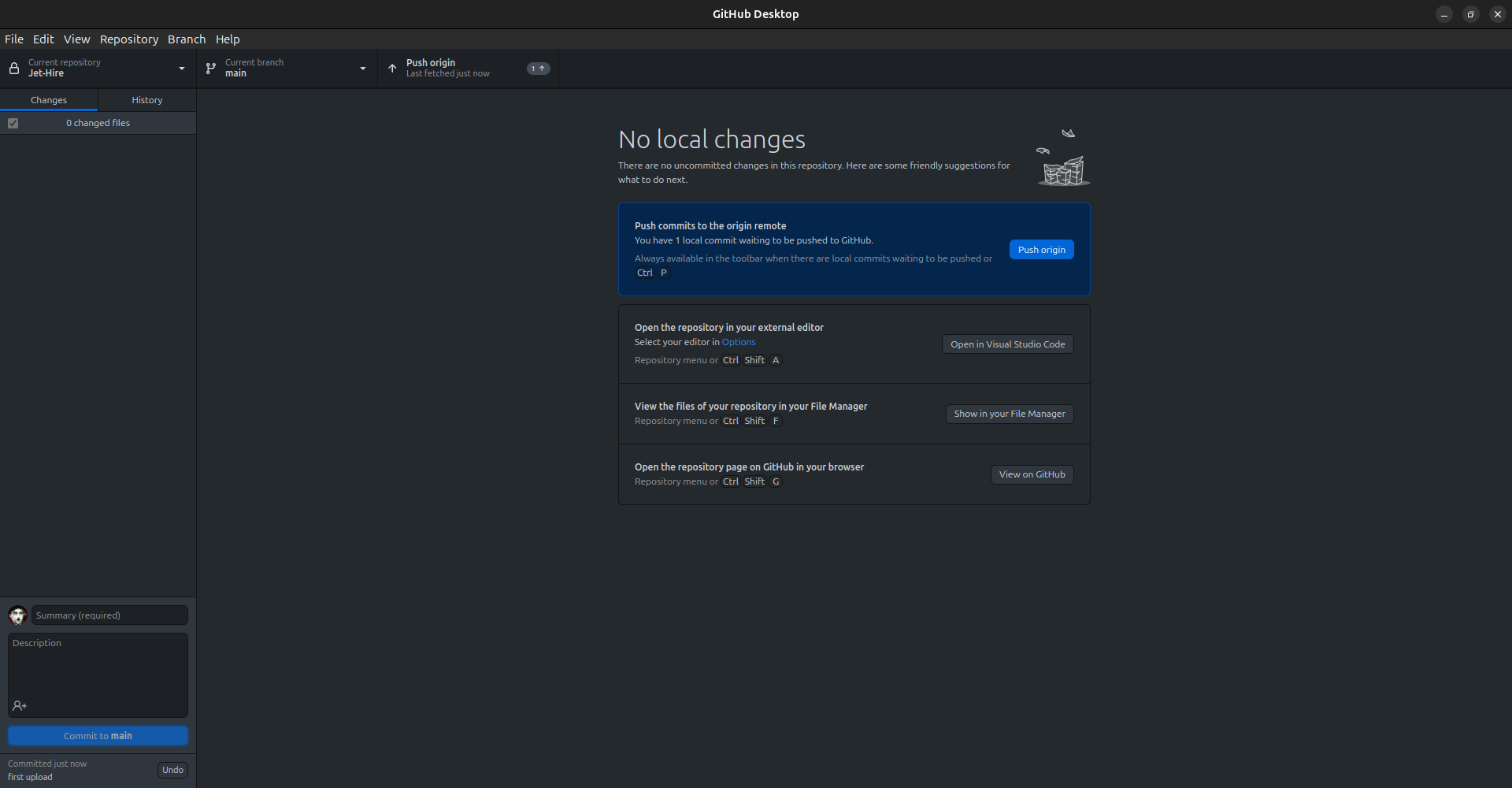


***Push***

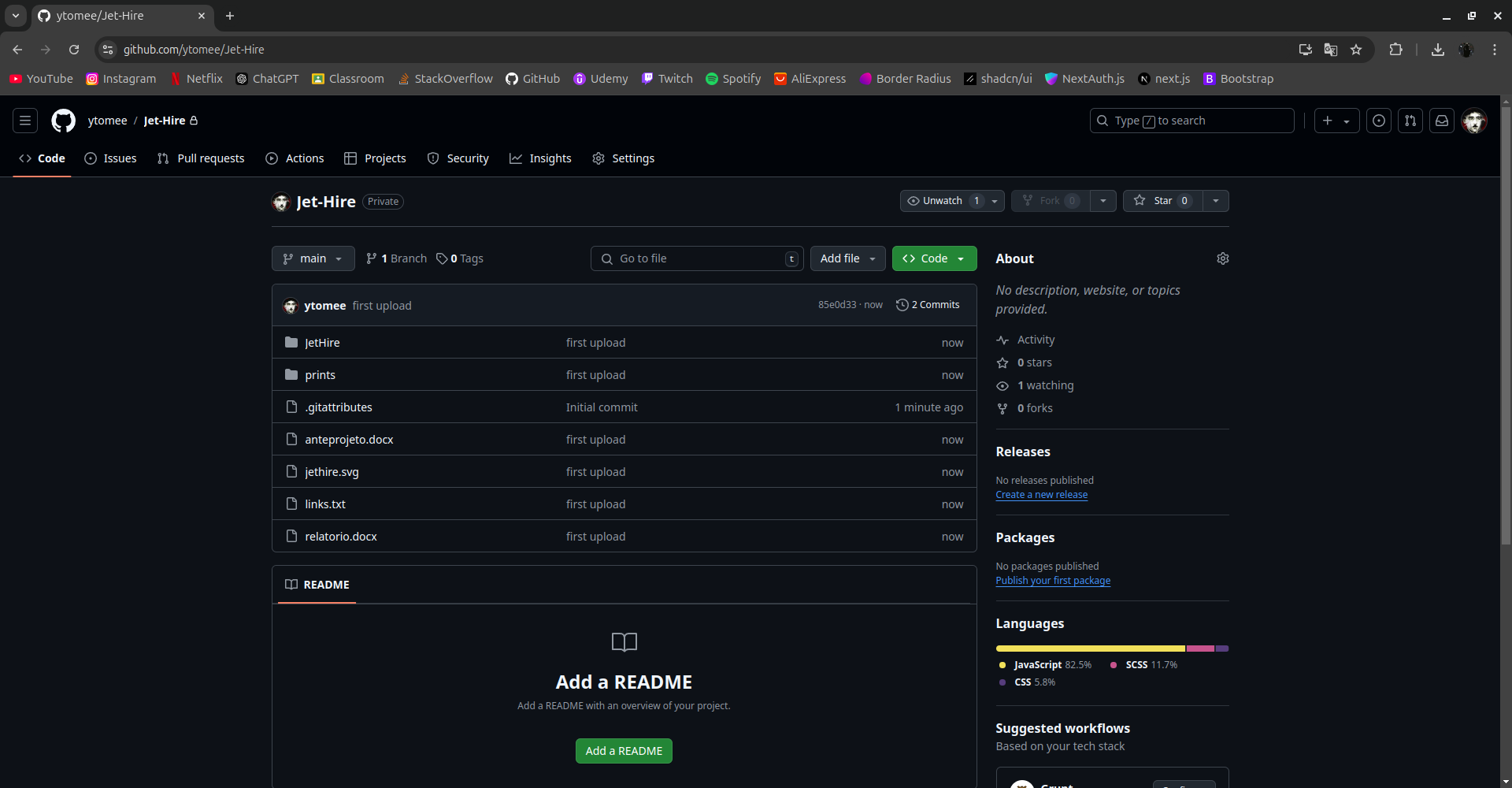
O Github vai criar uma pasta dentro da pasta Documentos do computador e é só preciso passar os ficheiros para a mesma. De seguida, volta-se á aplicação e verifica-se as alterações. No fim, dá-se *commit*, que vai fazer todas as alterações na *cloud.*

****

Depois de dar *commit*, é preciso dar *push*, ou seja, o pedido final para a alteração dos dados na *cloud*:

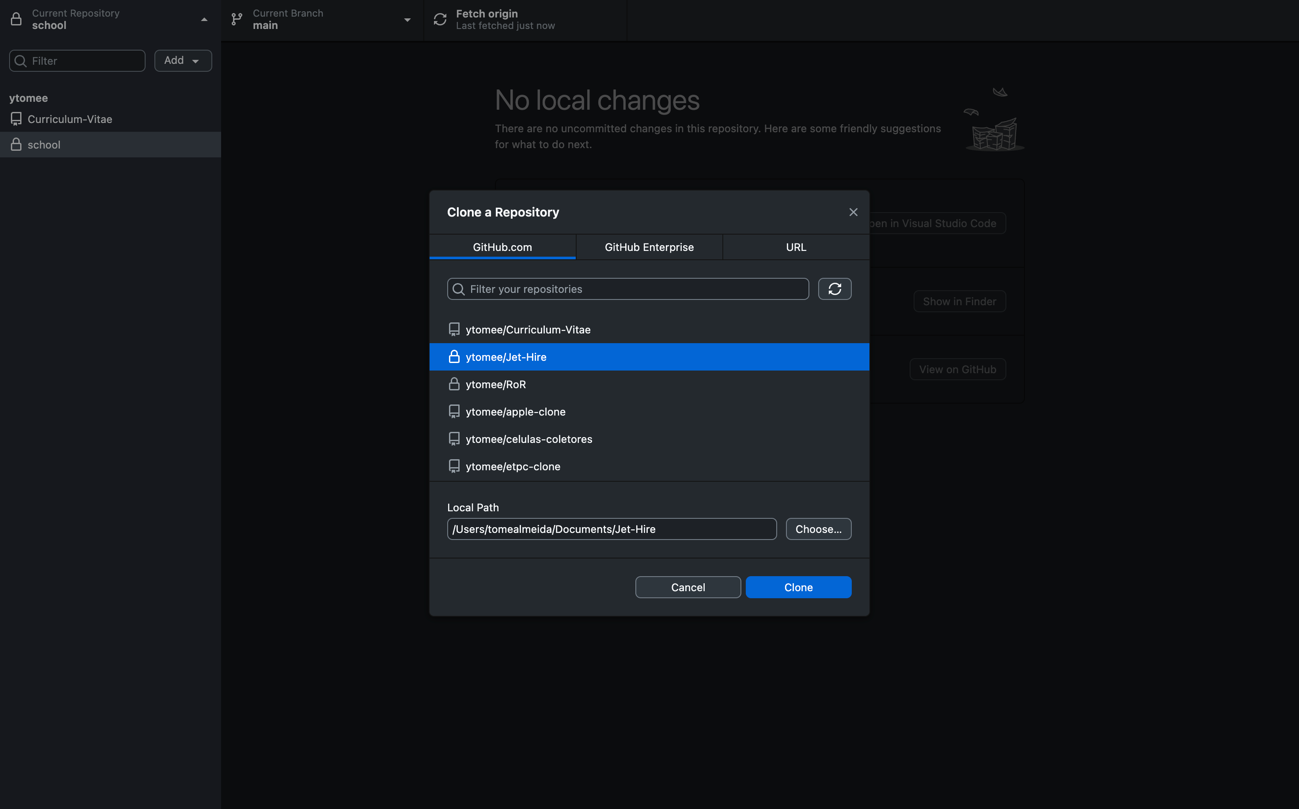


Para confirmar se o repositório foi criado corretamente pode-se confirmar pelo browser:

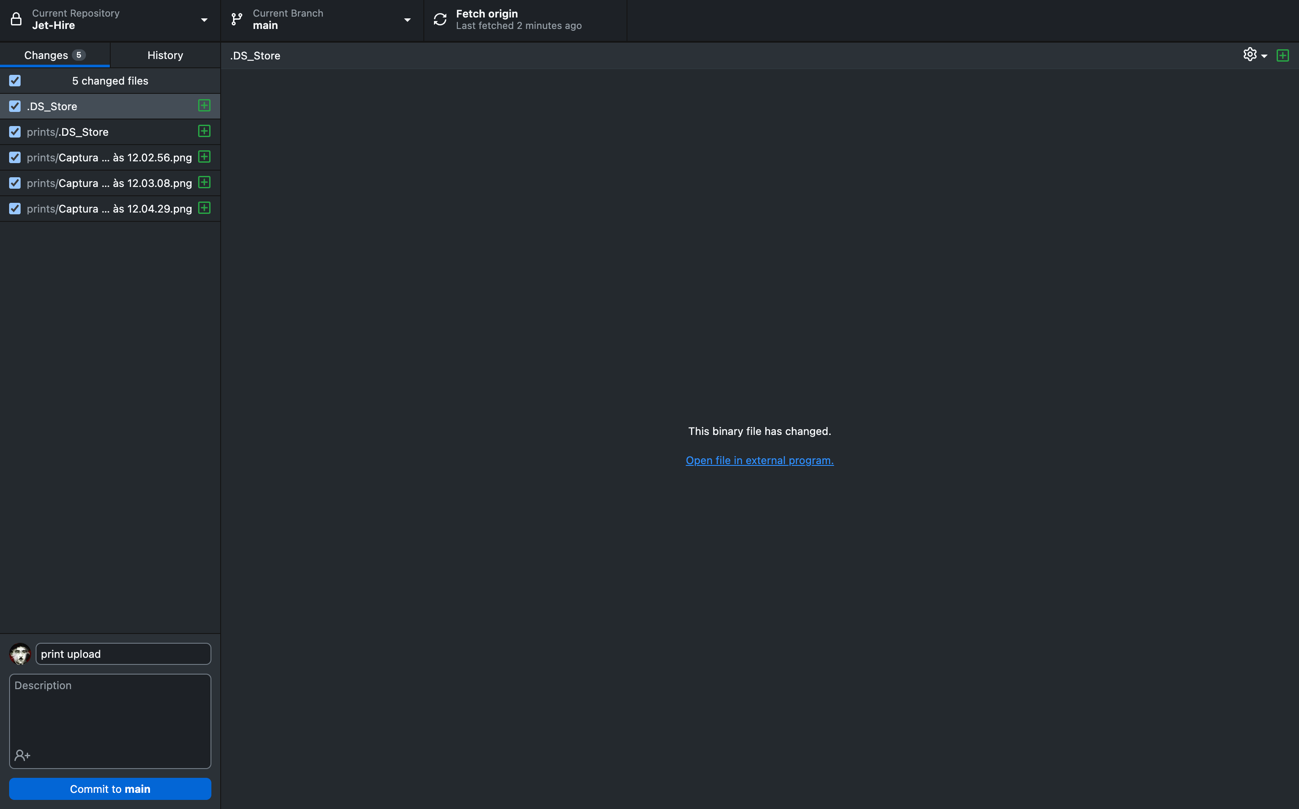


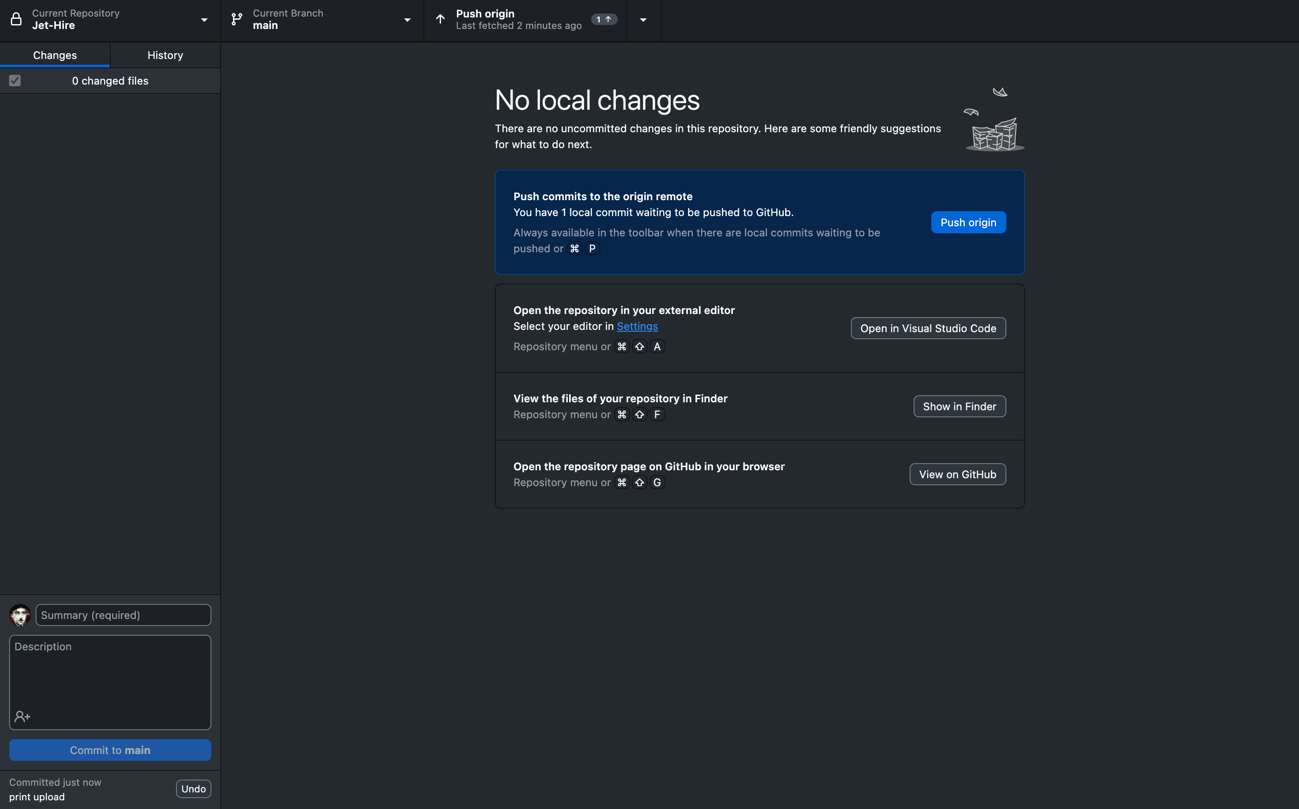
***Pull***

Caso o repositório ainda não exista no computador, tem que se clonar o mesmo.



Se o repositório já existir, dá-se *fetch* para verificar se existem alterações na pasta, caso existam, dá-se *push* para voltar a alterar os dados:





### Fase de Desenvolvimento

► Desenho – nesta atividade faz-se o levantamento dos dados necessários para criação da Base de Dados utilizada no projeto, implementando o Diagrama Entidade Relacionamento e o Modelo de Dados. Faz-se ainda a estruturação da navegação do website e do backoffice.

► Codificação – atividade em que toda a parte de codificação é realizada.

► Testes – ao longo desta atividade realiza-se os testes às funcionalidades implementadas por forma a garantir que nesta fase as mesmas estão a funcionar de acordo com o que foi planeado.

### Fase de Manutenção

► Correção – é nesta atividade em que irei realizar correções ao projeto. Estas correções tanto podem ser à interface gráfica ou ao código realizado até ao momento. Se for necessário também se poderá fazer alterações estruturais à base de dados.

► Adaptação – depois de executado todos os testes e correções, é nesta atividade que vão ser realizadas adaptações ao projeto. Estas adaptações podem ser ao nível da interface gráfica ou ao nível da otimização de código. Estas adaptações surgem da necessidade do projeto ser testado por outras pessoas.

► Evolução – depois de ter os objetivos iniciais concluídos, é nesta fase, havendo tempo que irei tentar implementar novas funcionalidades que foram surgindo ao longo do desenvolvimento do projeto e que inicialmente não estavam previstas.

## Cronograma



Figura 2: Cronograma

# Desenvolvimento

# Conclusão