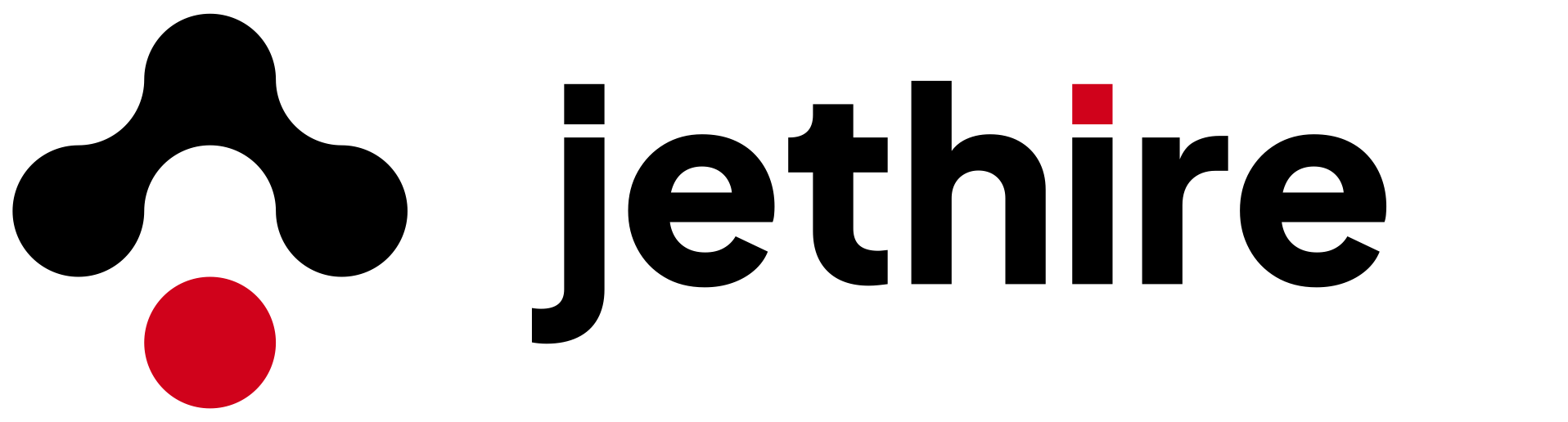


Técnico de Gestão e Programação de Sistemas Informáticos

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**

[Introdução (Português) 1](#_Toc182998010)

[Introduction (Inglês) 2](#_Toc182998011)

[Planeamento 3](#_Toc182998012)

[Fundamentação 3](#_Toc182998013)

[Recursos 4](#_Toc182998014)

[Hardware 4](#_Toc182998015)

[Software 4](#_Toc182998016)

[Metodologia 6](#_Toc182998017)

[Atividades 7](#_Toc182998018)

[Fase de Definição 7](#_Toc182998019)

[Fase de Desenvolvimento 20](#_Toc182998020)

[Fase de Manutenção 21](#_Toc182998021)

[Cronograma 21](#_Toc182998022)

[Projeto 22](#_Toc182998023)

[Instalação do *template* 22](#_Toc182998024)

[Logótipo 24](#_Toc182998025)

[Escolha da palete de cores 24](#_Toc182998026)

[Cor principal 24](#_Toc182998027)

[Cores secundárias 26](#_Toc182998028)

[Cores neutras 27](#_Toc182998029)

[Criação da palete no projeto 27](#_Toc182998030)

[Conclusão 29](#_Toc182998031)

**Índice de Figuras**

[Figura 1 - Metodologia Kanban 6](#_Toc182998090)

[Figura 2 - Instalação Visual Studio Code: Passo 1 8](#_Toc182998091)

[Figura 3 - Instalação Visual Studio Code: Passo 2 8](#_Toc182998092)

[Figura 4 - Extensões 9](#_Toc182998093)

[Figura 5 - Instalação MongoDB Compass: Passo 1 9](#_Toc182998094)

[Figura 6 - Instalação MongoDB Compass: Passo 2 10](#_Toc182998095)

[Figura 7 - sudo apt install nodejs 10](#_Toc182998096)

[Figura 8 - sudo apt install npm 10](#_Toc182998097)

[Figura 9 - Verificação do npm 11](#_Toc182998098)

[Figura 10 - Instalação GitHub Desktop: Passo 1 12](#_Toc182998099)

[Figura 11 - Instalação GitHub Desktop: Passo 2 12](#_Toc182998100)

[Figura 12 - Instalação GitHub Desktop: Passo 3 13](#_Toc182998101)

[Figura 13 - Instalação do Inkscape: Passo 1 13](#_Toc182998102)

[Figura 14 - Instalação do Inkscape: Passo 2 14](#_Toc182998103)

[Figura 15 - Instalação do Krita: Passo 1 15](#_Toc182998104)

[Figura 16 - Instalação do Krita: Passo 2 15](#_Toc182998105)

[Figura 17 - Sincronização entre dois computadores: Passo 1 16](#_Toc182998106)

[Figura 18 - Sincronização entre dois computadores: Passo 2 16](#_Toc182998107)

[Figura 19 - Sincronização entre dois computadores: Passo 3 17](#_Toc182998108)

[Figura 20 - Sincronização entre dois computadores: Passo 4 17](#_Toc182998109)

[Figura 21 - Sincronização entre dois computadores: Passo 5 18](#_Toc182998110)

[Figura 22 - Sincronização entre dois computadores: Passo 6 18](#_Toc182998111)

[Figura 23 - Sincronização entre dois computadores: Passo 7 19](#_Toc182998112)

[Figura 24 - Sincronização entre dois computadores: Passo 8 19](#_Toc182998113)

[Figura 25 – Sitemap principal 20](#_Toc182998114)

[Figura 26 - Cronograma 21](#_Toc182998115)

[Figura 27 - Download do template 22](#_Toc182998116)

[Figura 28 - Instalação dos node\_modules 22](#_Toc182998117)

[Figura 29 - npm run dev 23](#_Toc182998118)

[Figura 30 - Projeto executado 23](#_Toc182998119)

[Figura 31 - Logótipo 24](#_Toc182998120)

[Figura 32 - Escolha da cor principal 24](#_Toc182998121)

[Figura 33 - Escolha do tom claro 25](#_Toc182998122)

[Figura 34 - Escolha do tom escuro 25](#_Toc182998123)

[Figura 35 - Criação de um gradiente claro 25](#_Toc182998124)

[Figura 36 - Criação de um gradiente escuro 26](#_Toc182998125)

[Figura 37 - Vermelho 26](#_Toc182998126)

[Figura 38 - Amarelo 27](#_Toc182998127)

[Figura 39 - Azul 27](#_Toc182998128)

[Figura 40 - Verde 27](#_Toc182998129)

[Figura 41 - Cores neutras 27](#_Toc182998130)

[Figura 42 - Exemplo de variáveis globais 28](#_Toc182998131)

**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.

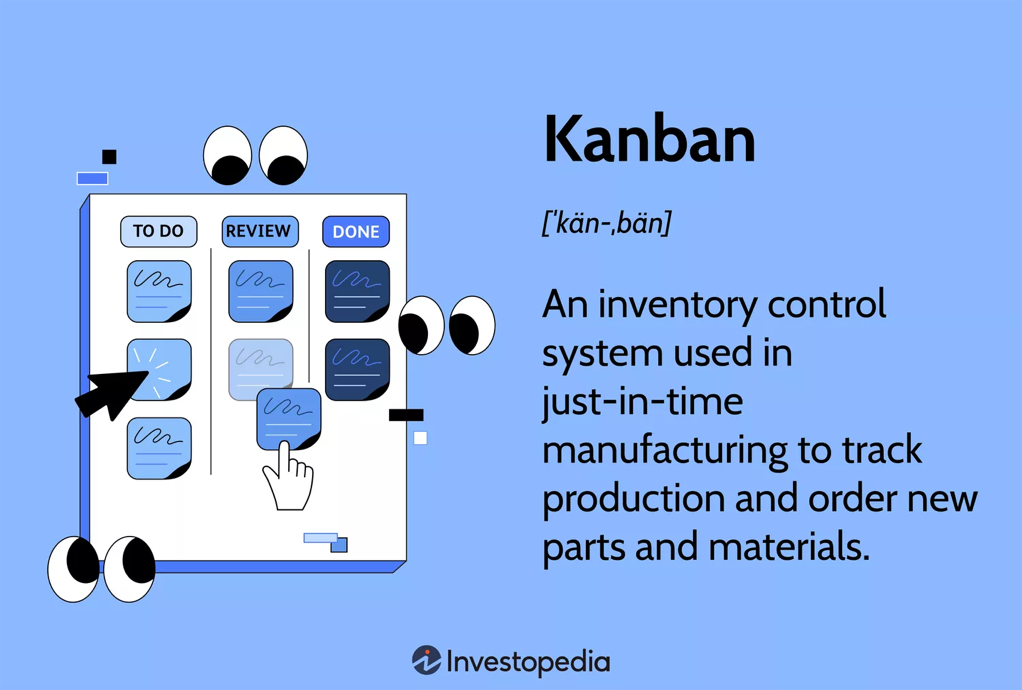


Figura 1 - Metodologia Kanban

*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.

### 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 26 - Cronograma

**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/).

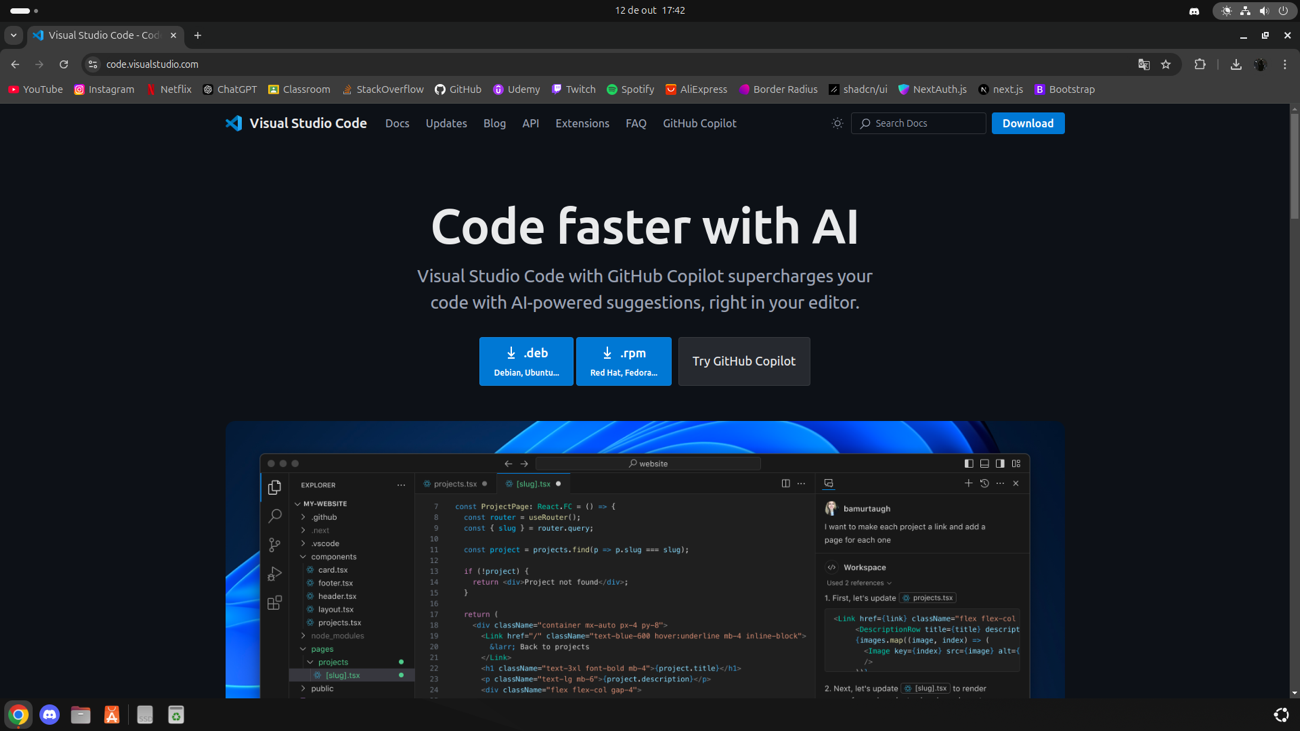


Figura 2 - Instalação Visual Studio Code: Passo 1

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

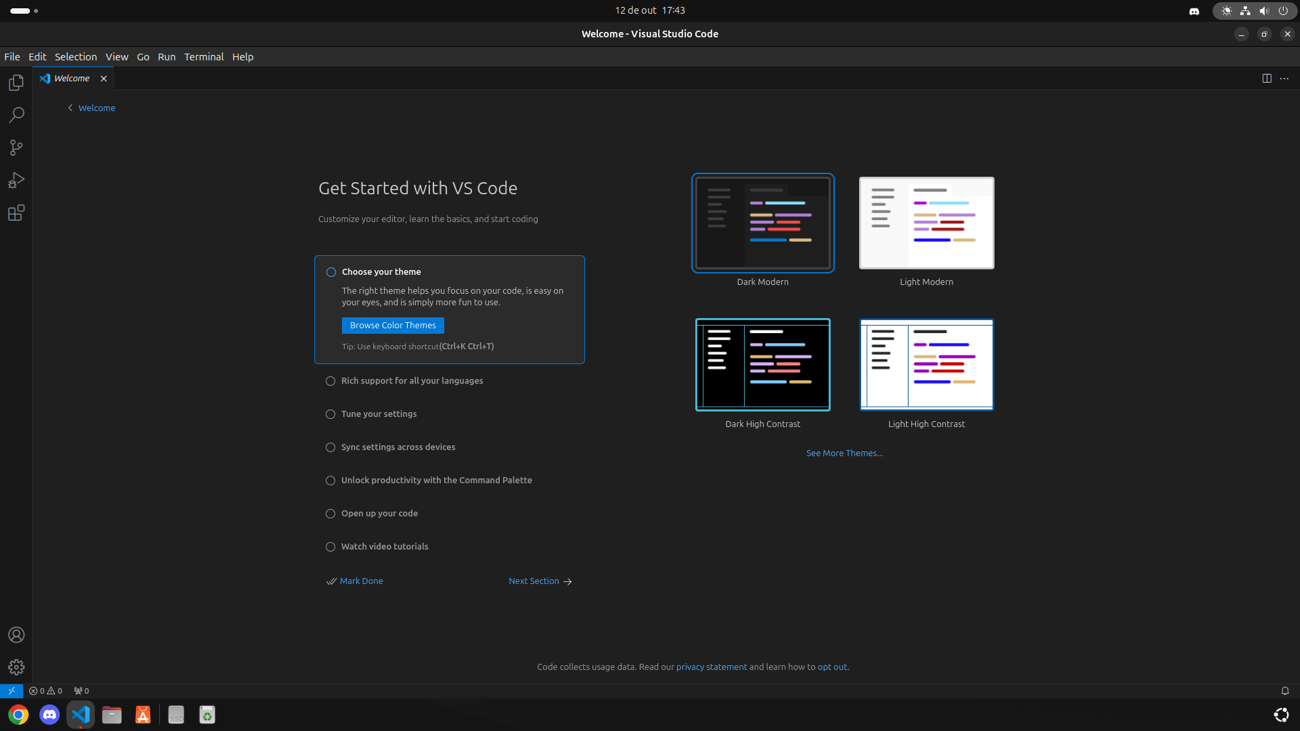


Figura 3 - Instalação Visual Studio Code: Passo 2

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** | Figura 4 - Extensões |

MongoDB Compass

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

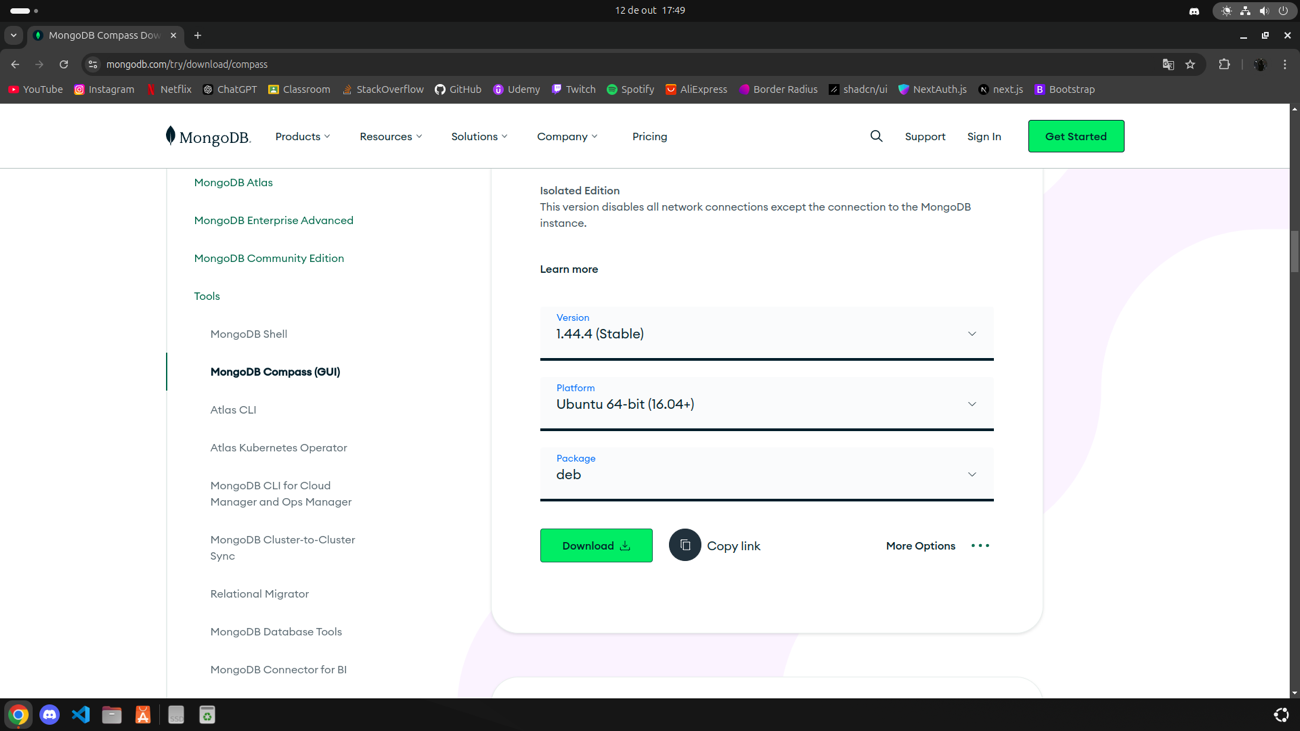


Figura 5 - Instalação MongoDB Compass: Passo 1

De seguida, é só executar o instalador.

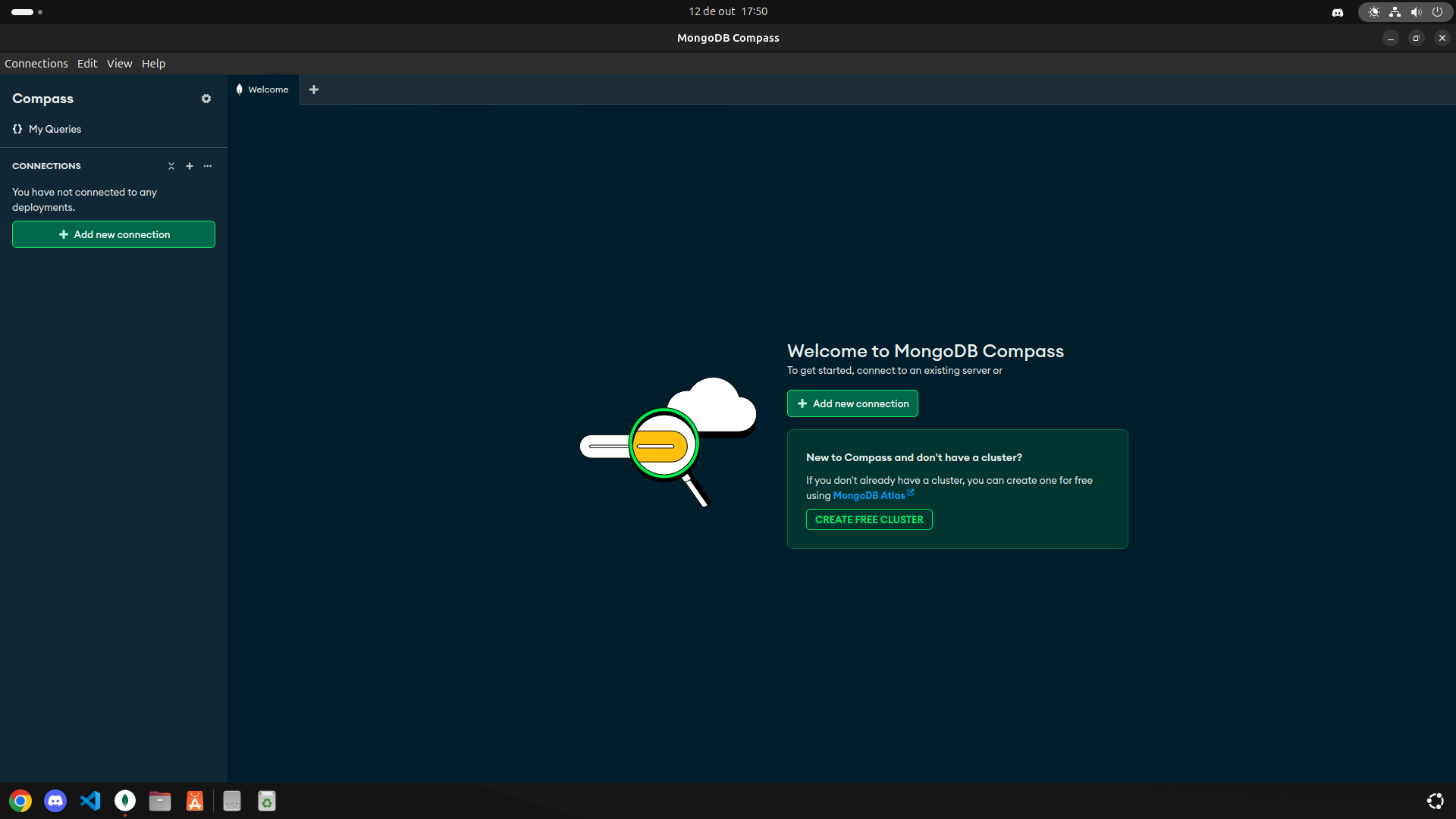


Figura 6 - Instalação MongoDB Compass: Passo 2

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*** |
| Figura 7 - sudo apt install nodejs | Figura 8 - sudo apt install npm |

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

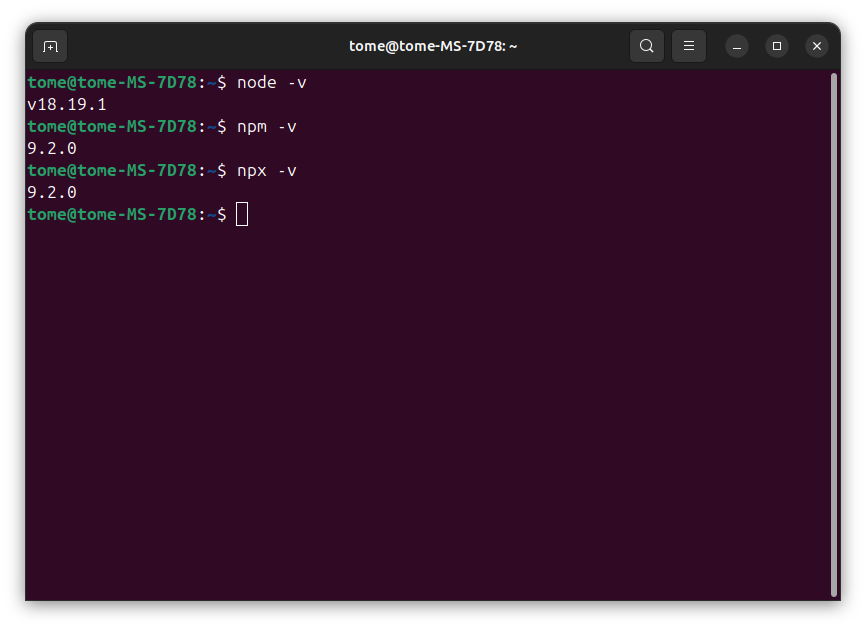


Figura 9 - Verificação do npm

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*

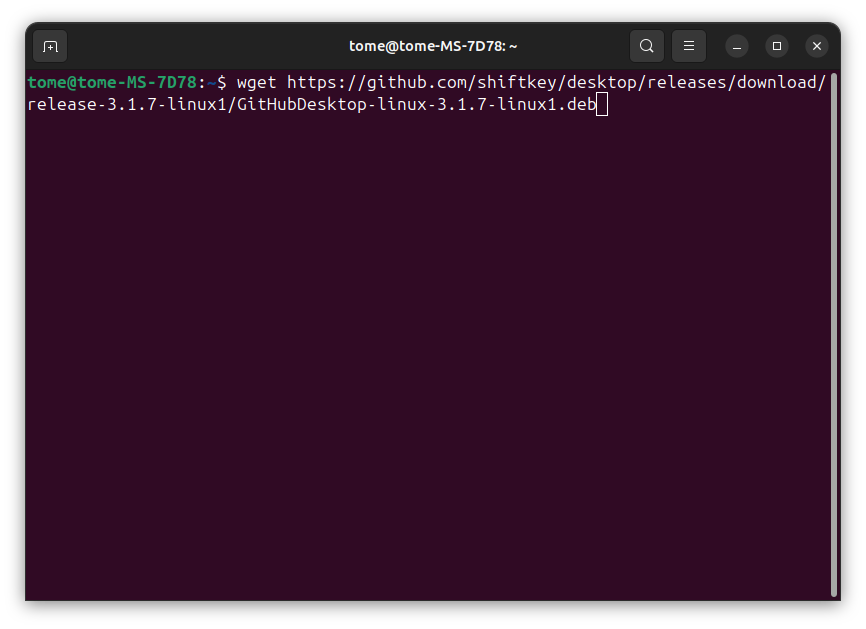


Figura 10 - Instalação GitHub Desktop: Passo 1

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

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

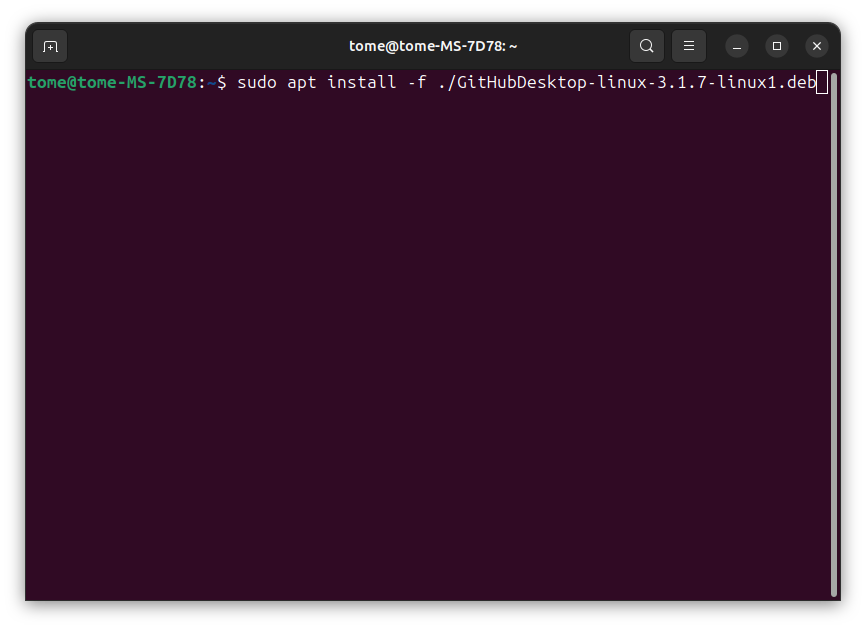


Figura 11 - Instalação GitHub Desktop: Passo 2

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

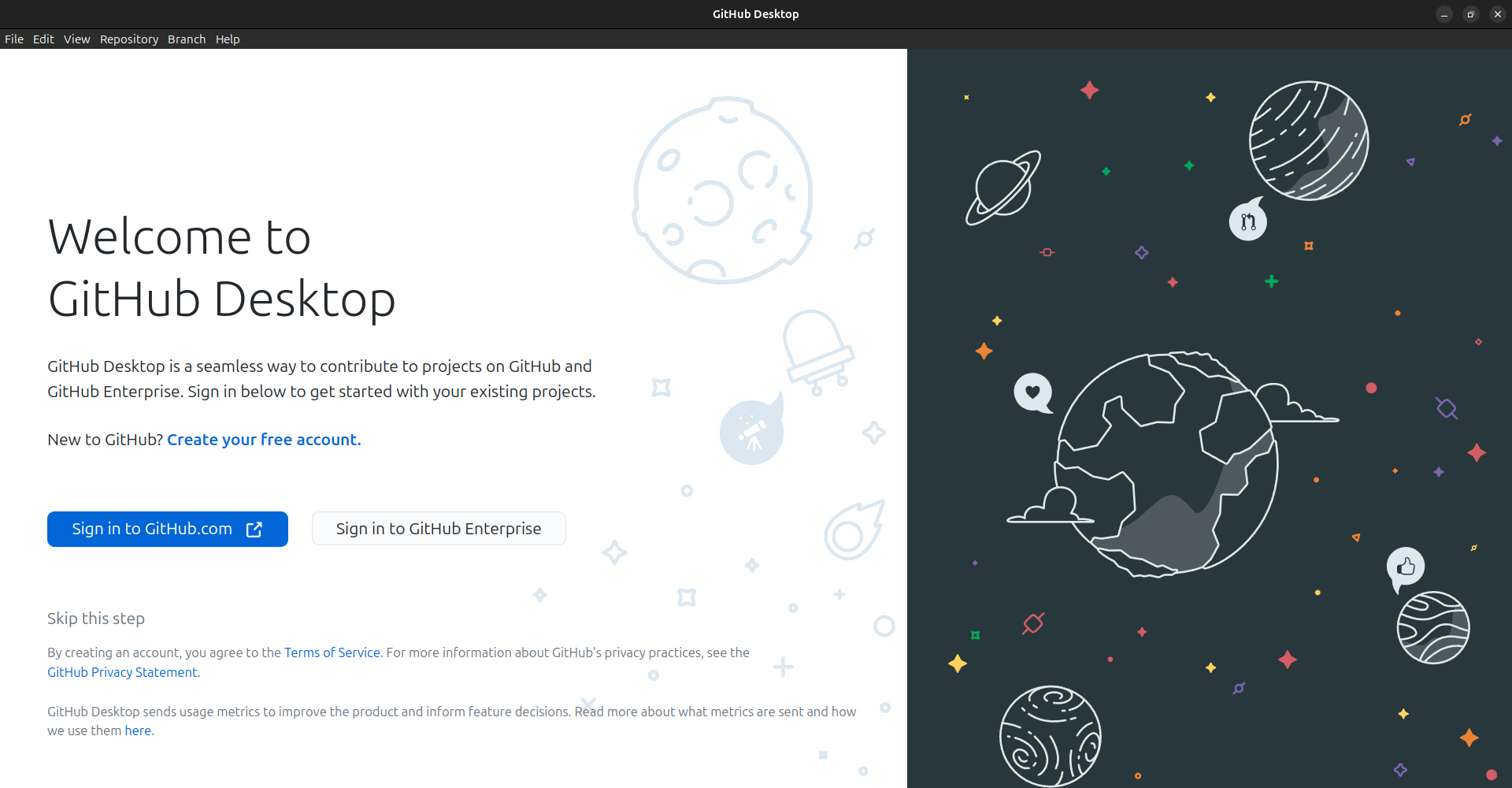


Figura 12 - Instalação GitHub Desktop: Passo 3

Inkscape

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

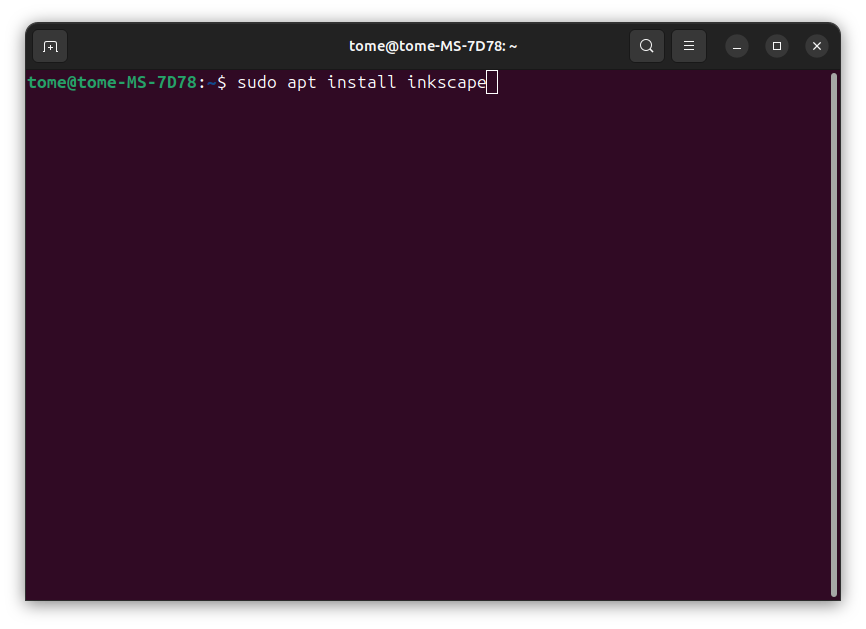


Figura 13 - Instalação do Inkscape: Passo 1

A instalação está concluída e pronta a ser utilizada.

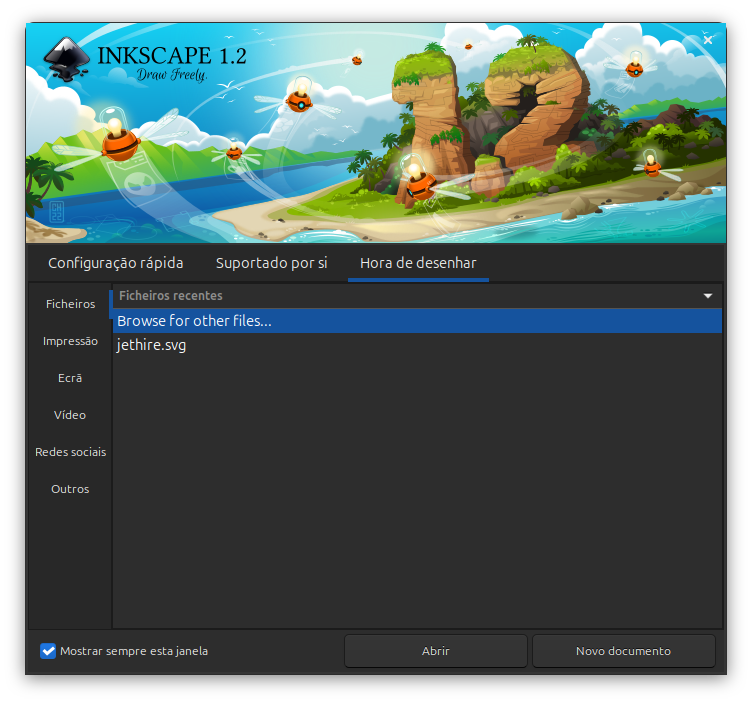


Figura 14 - Instalação do Inkscape: Passo 2

Krita

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

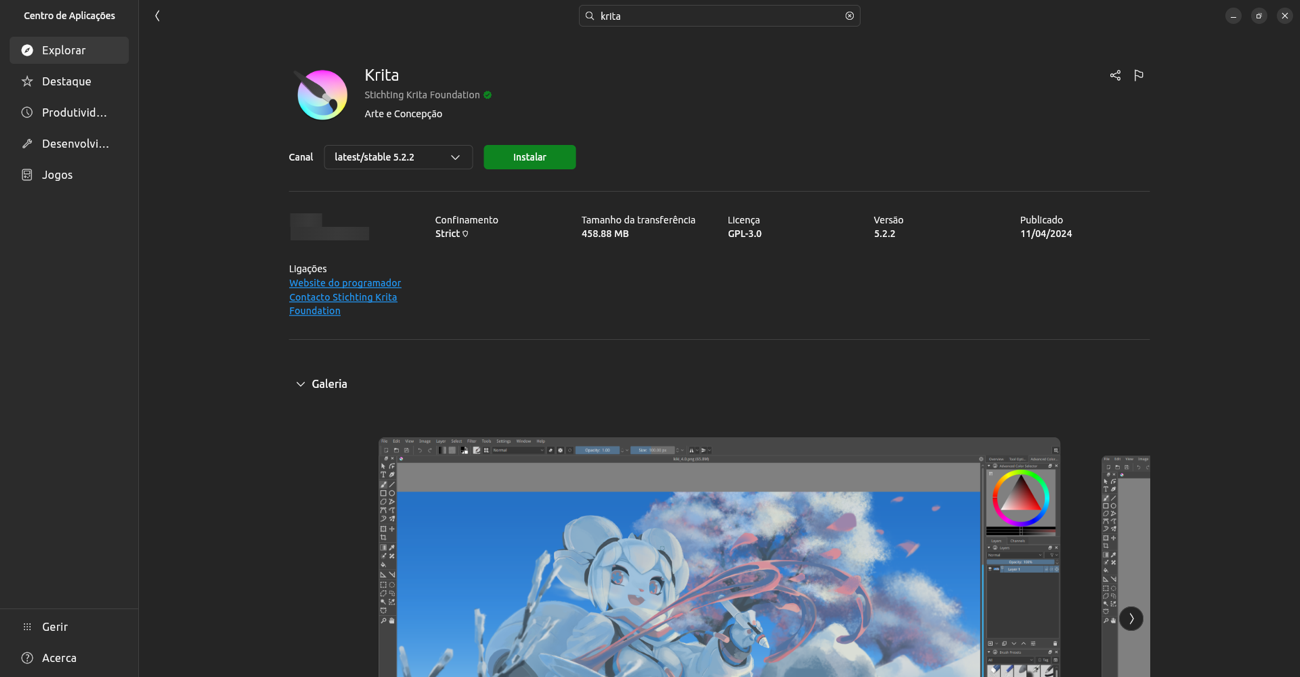


Figura 15 - Instalação do Krita: Passo 1

A aplicação está pronta a ser utilizada.

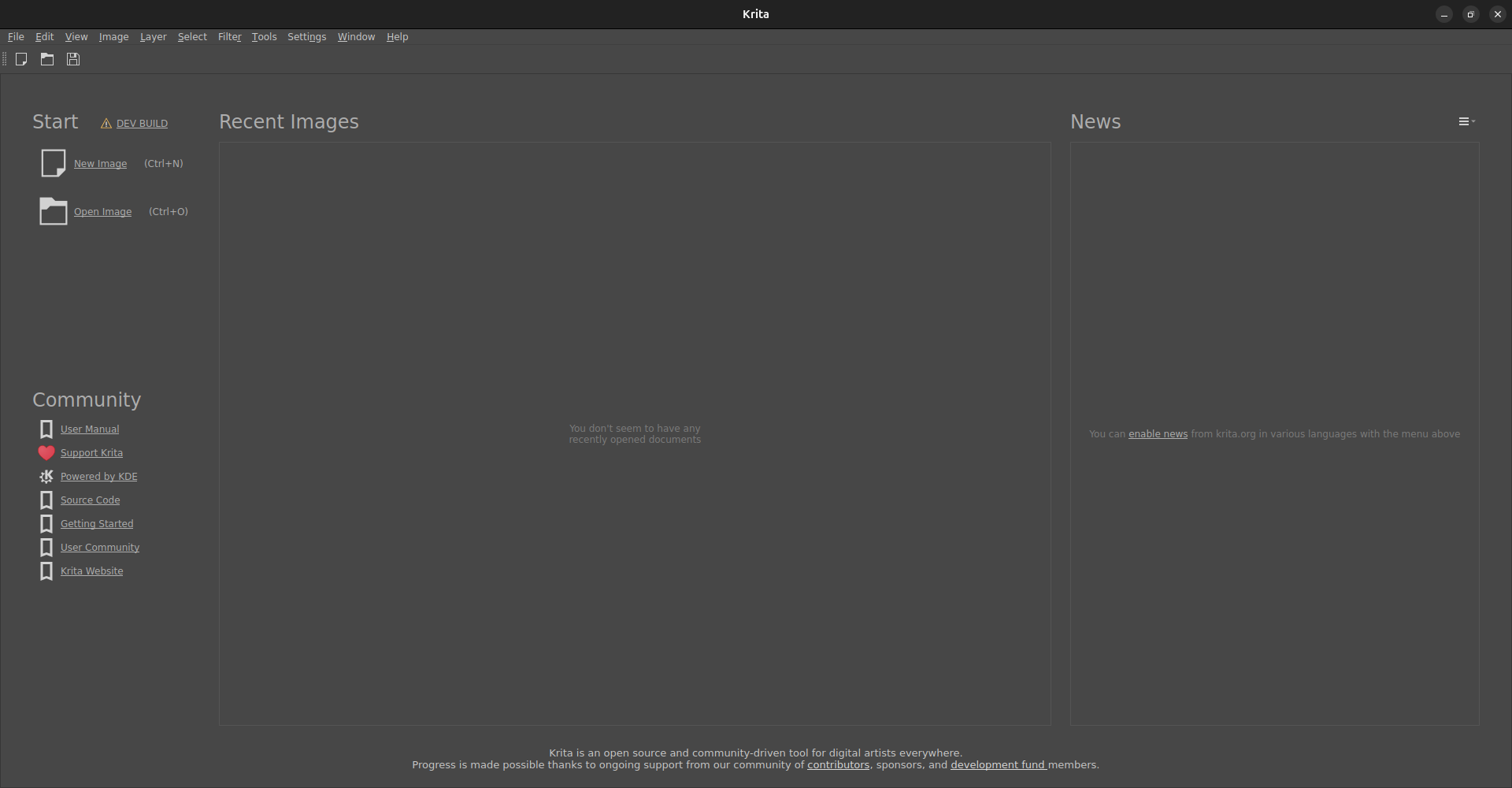


Figura 16 - Instalação do Krita: Passo 2

**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:

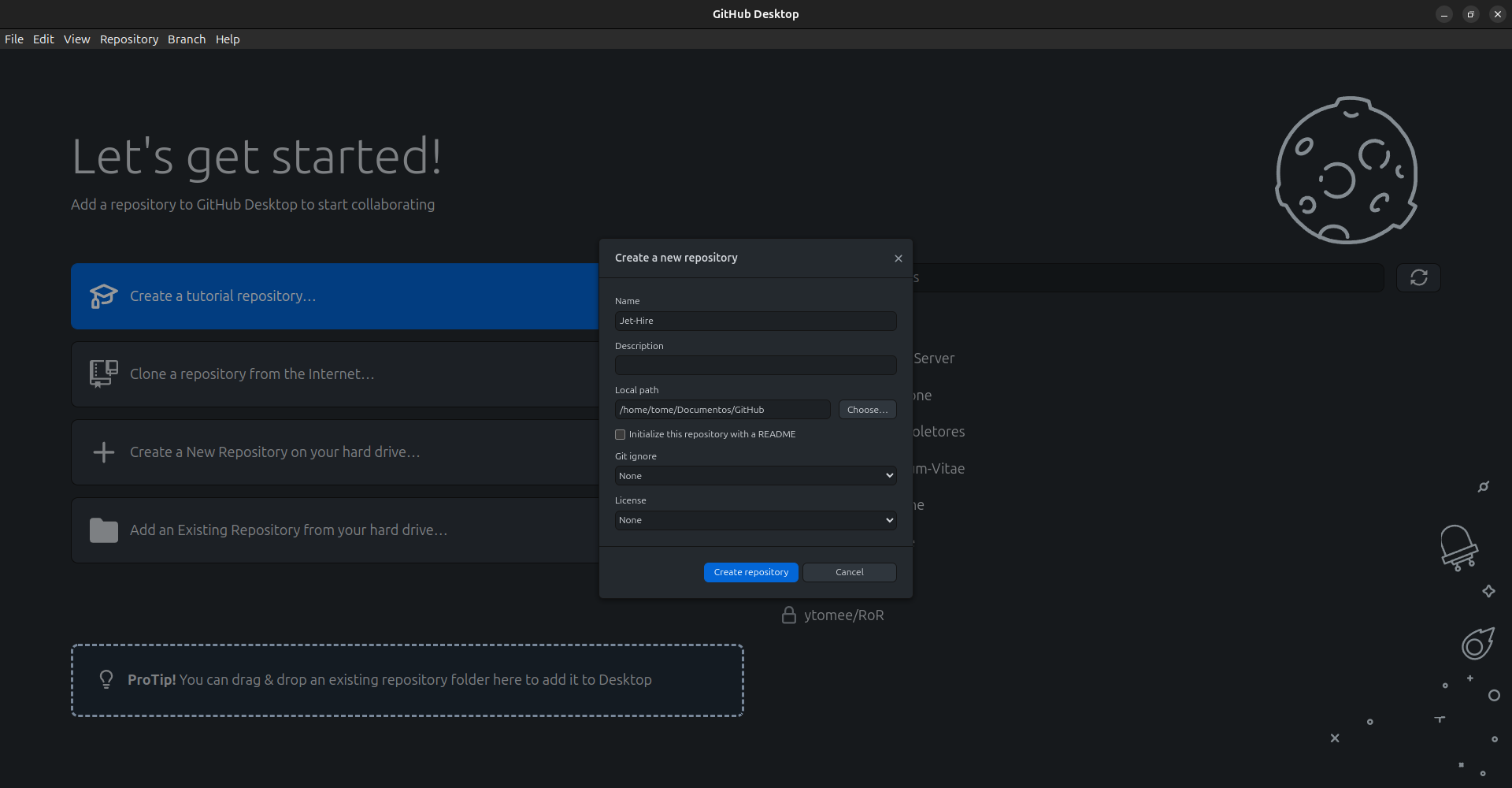


Figura 17 - Sincronização entre dois computadores: Passo 1

Depois da criação do repositório é preciso publicá-lo:

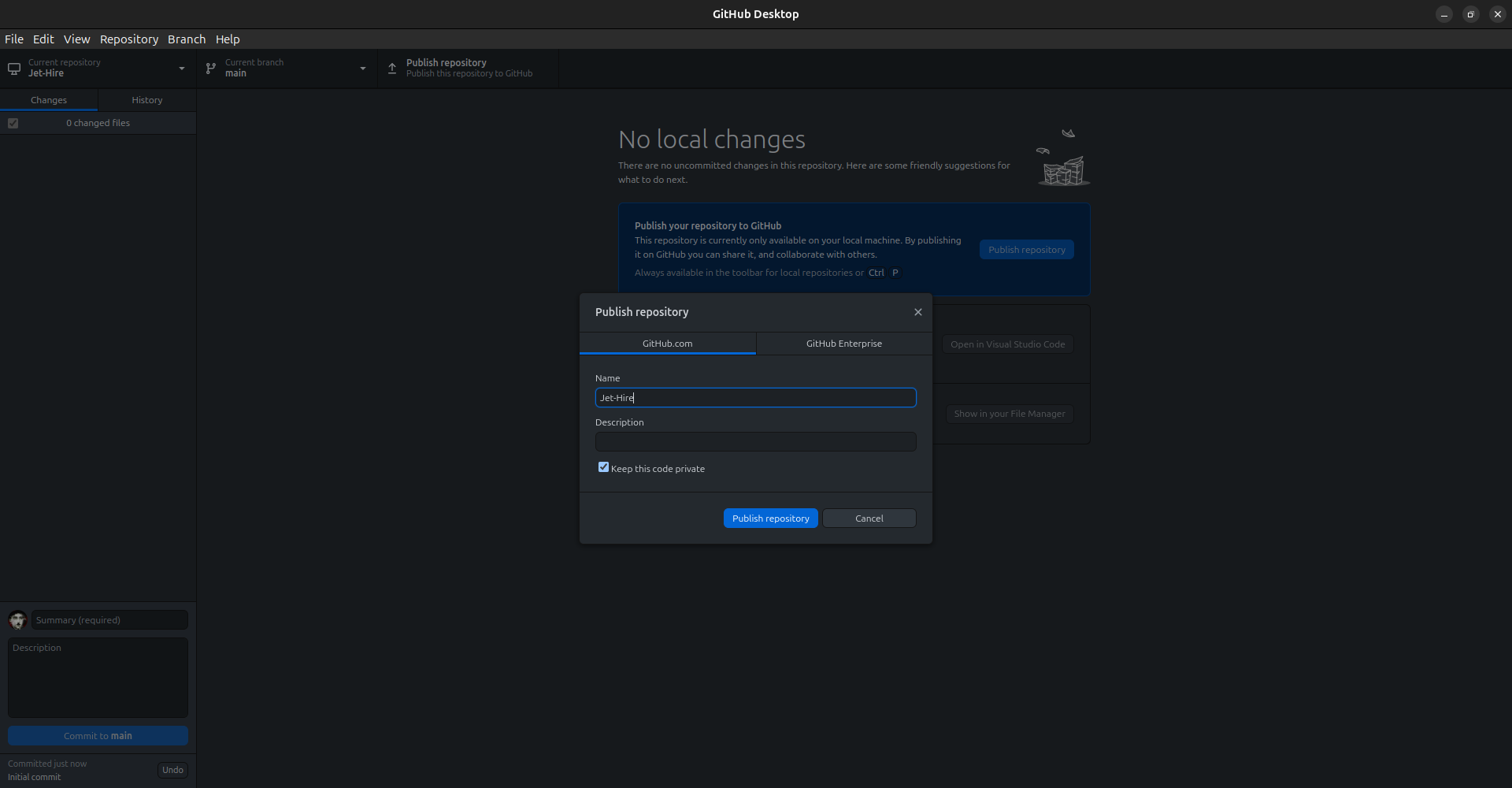


Figura 18 - Sincronização entre dois computadores: Passo 2

***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.*

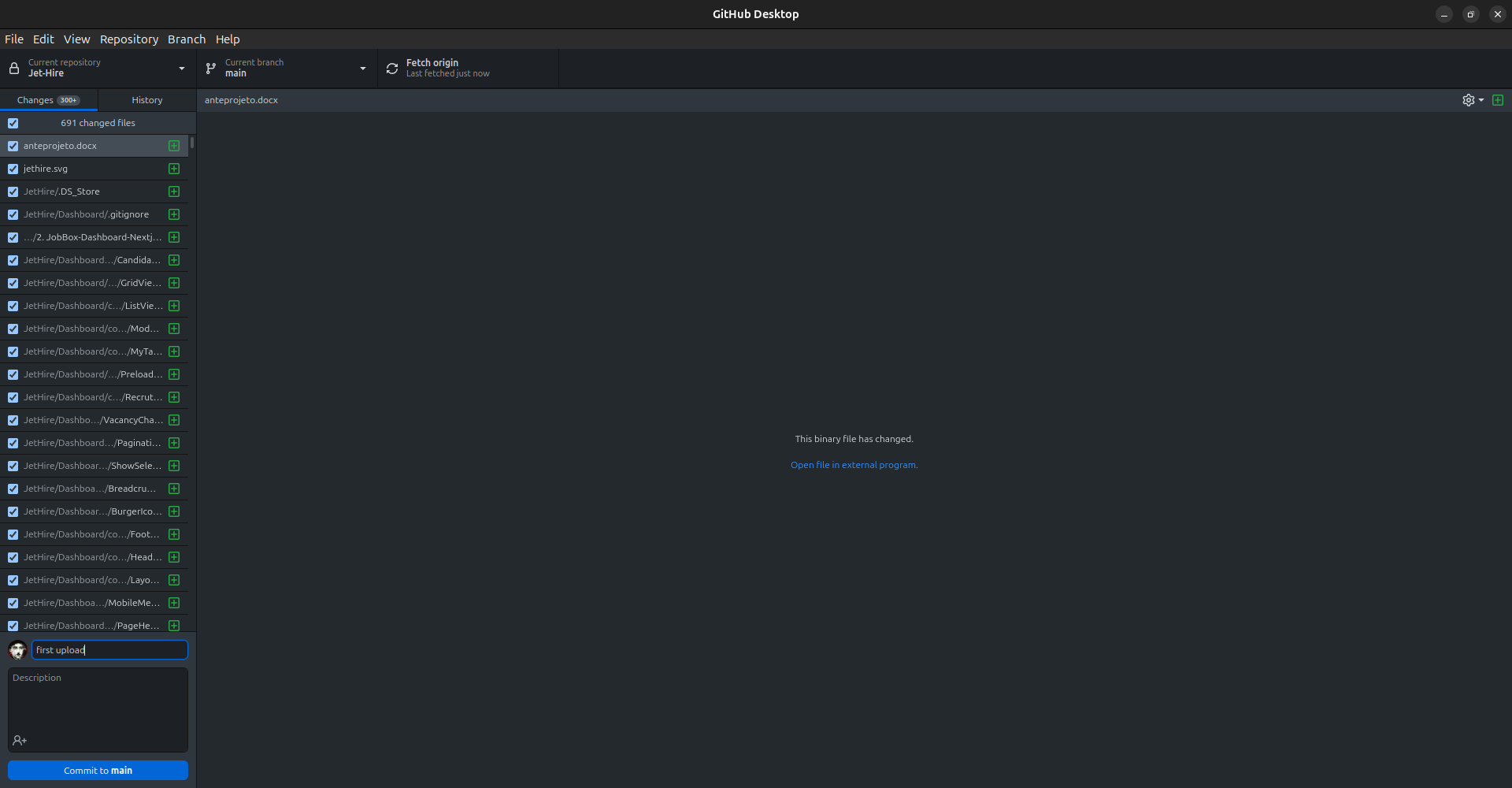
****

Figura 19 - Sincronização entre dois computadores: Passo 3

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

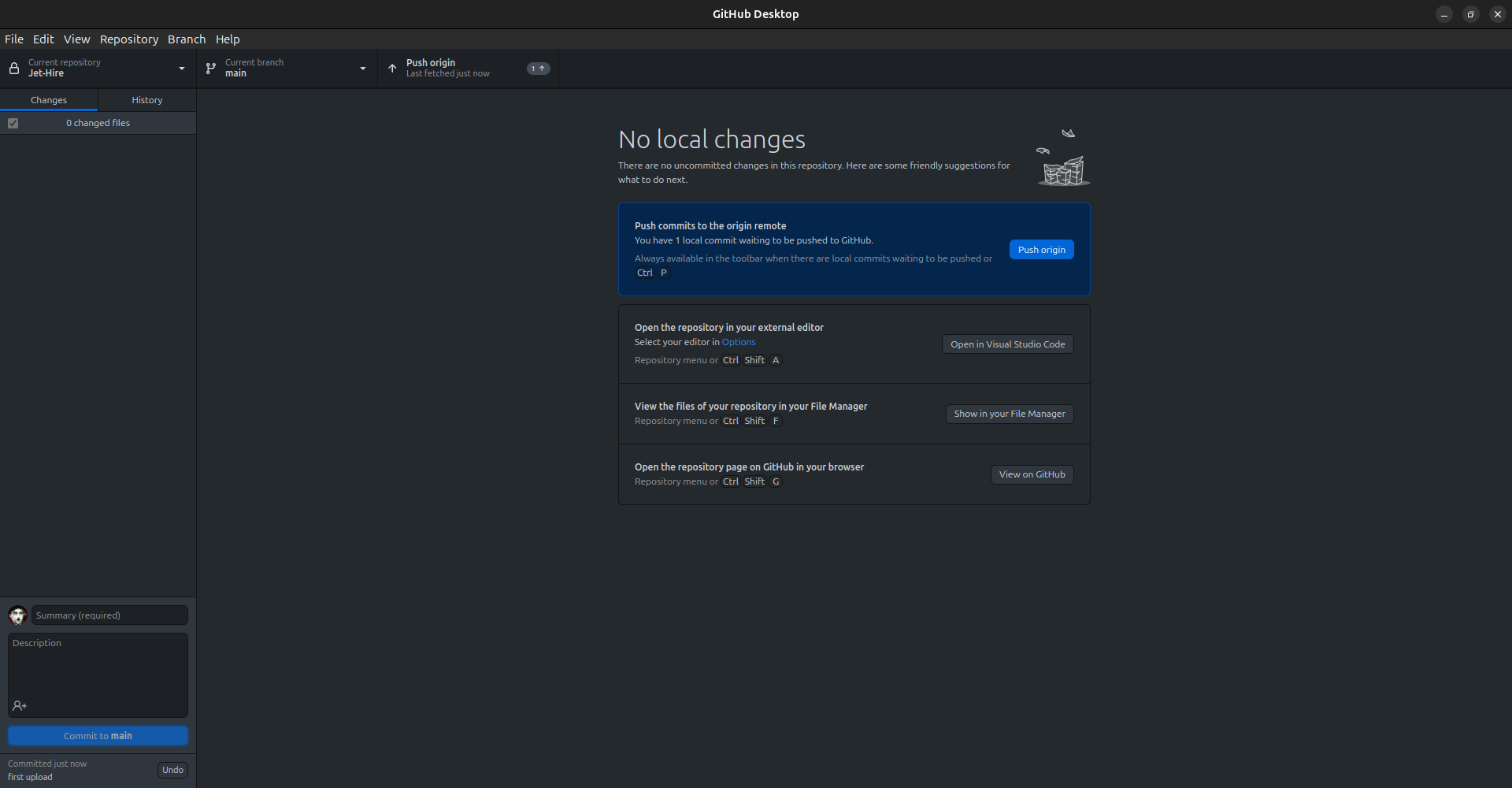


Figura 20 - Sincronização entre dois computadores: Passo 4

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

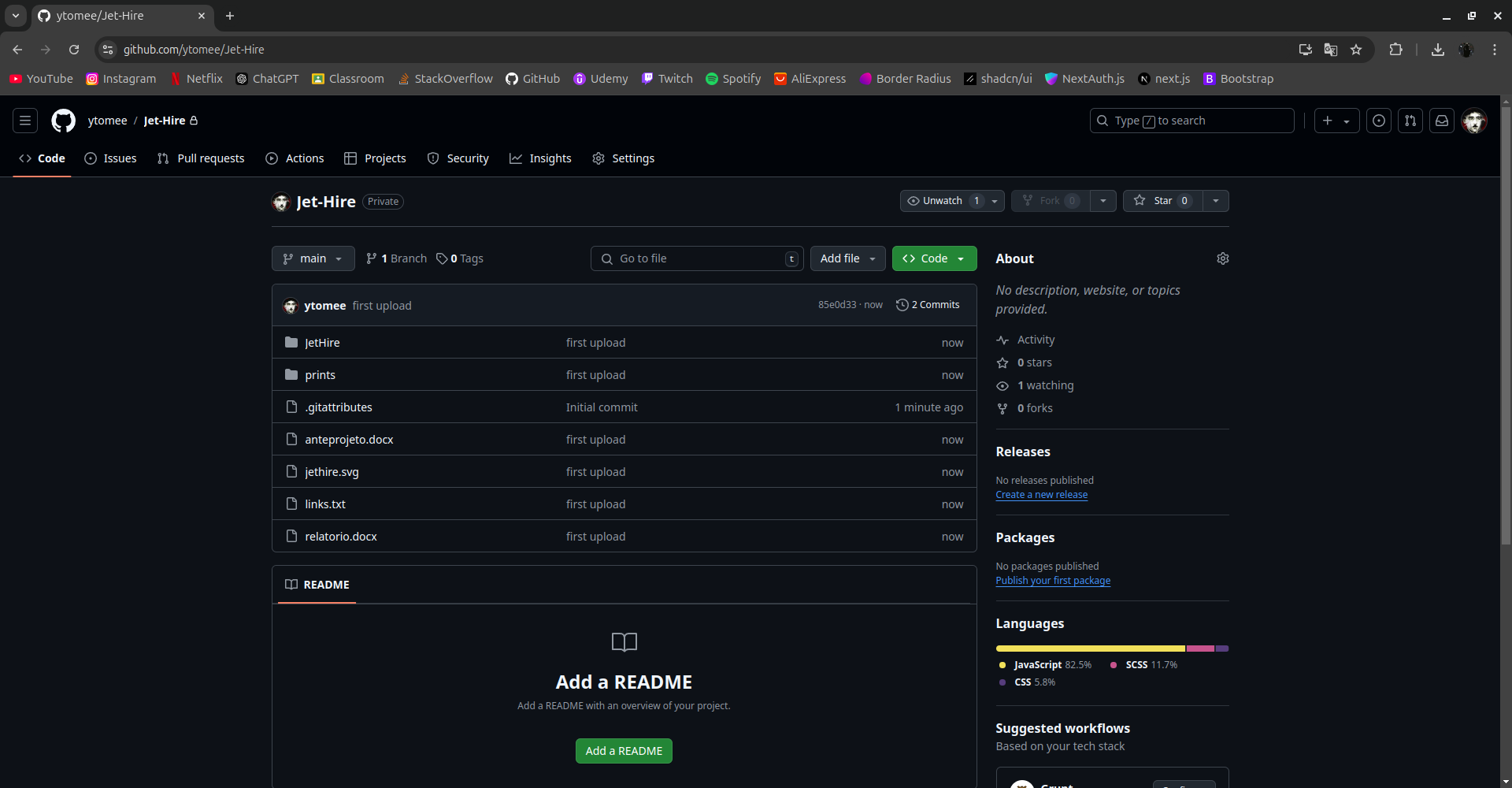


Figura 21 - Sincronização entre dois computadores: Passo 5

***Pull***

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

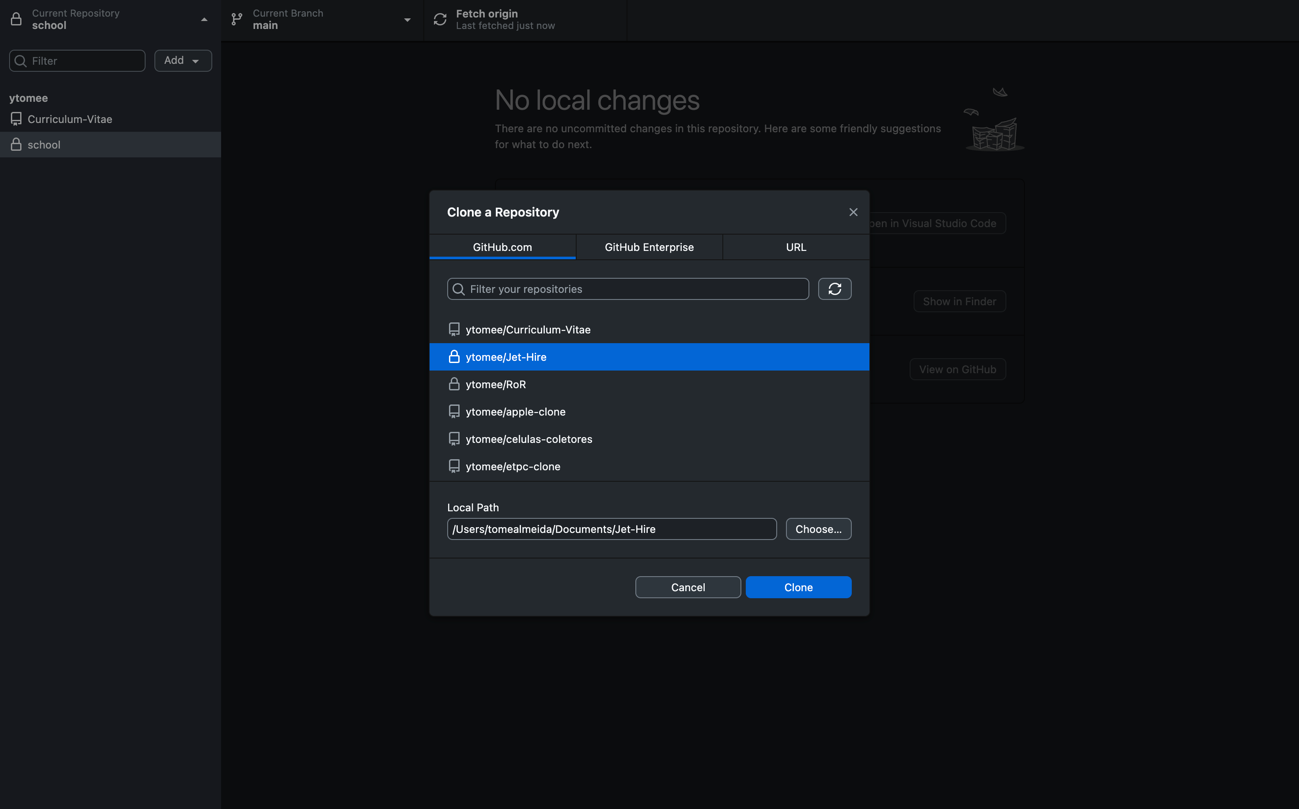


Figura 22 - Sincronização entre dois computadores: Passo 6

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:

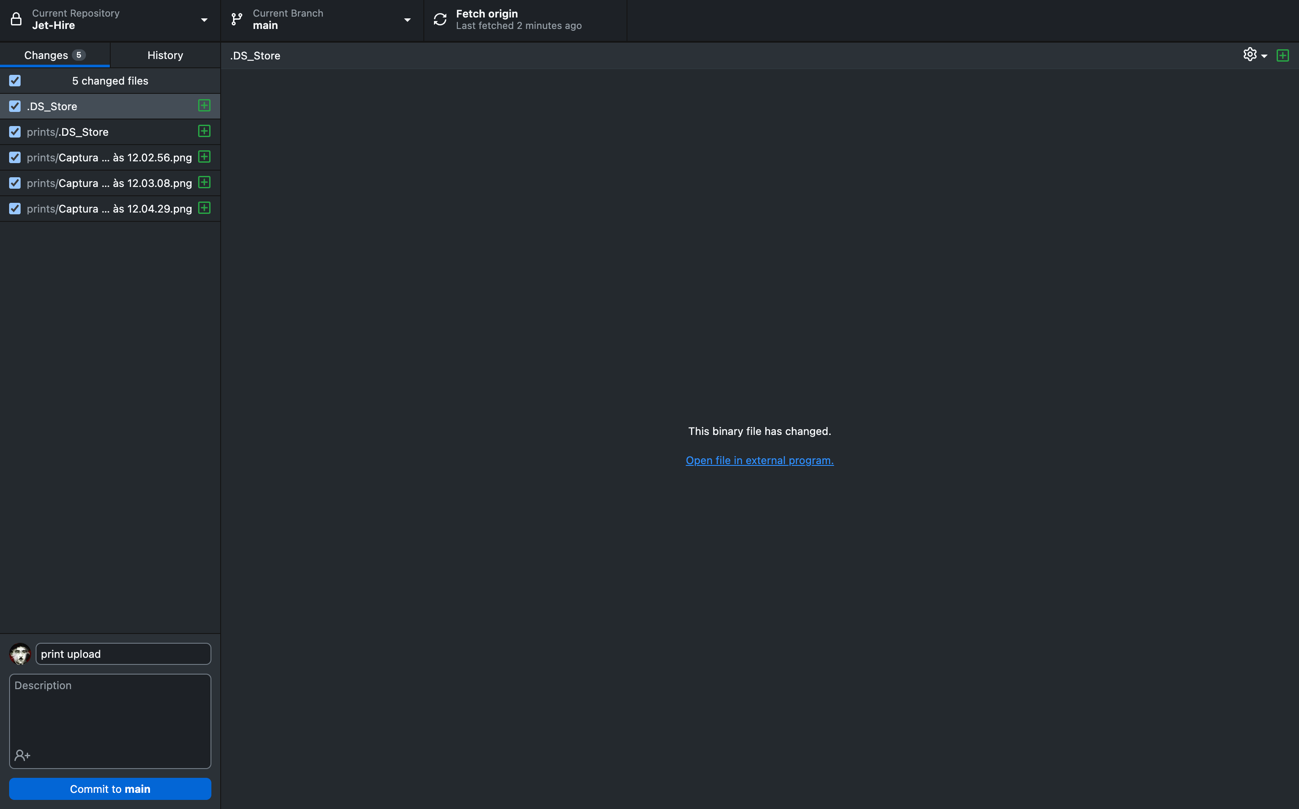


Figura 23 - Sincronização entre dois computadores: Passo 7

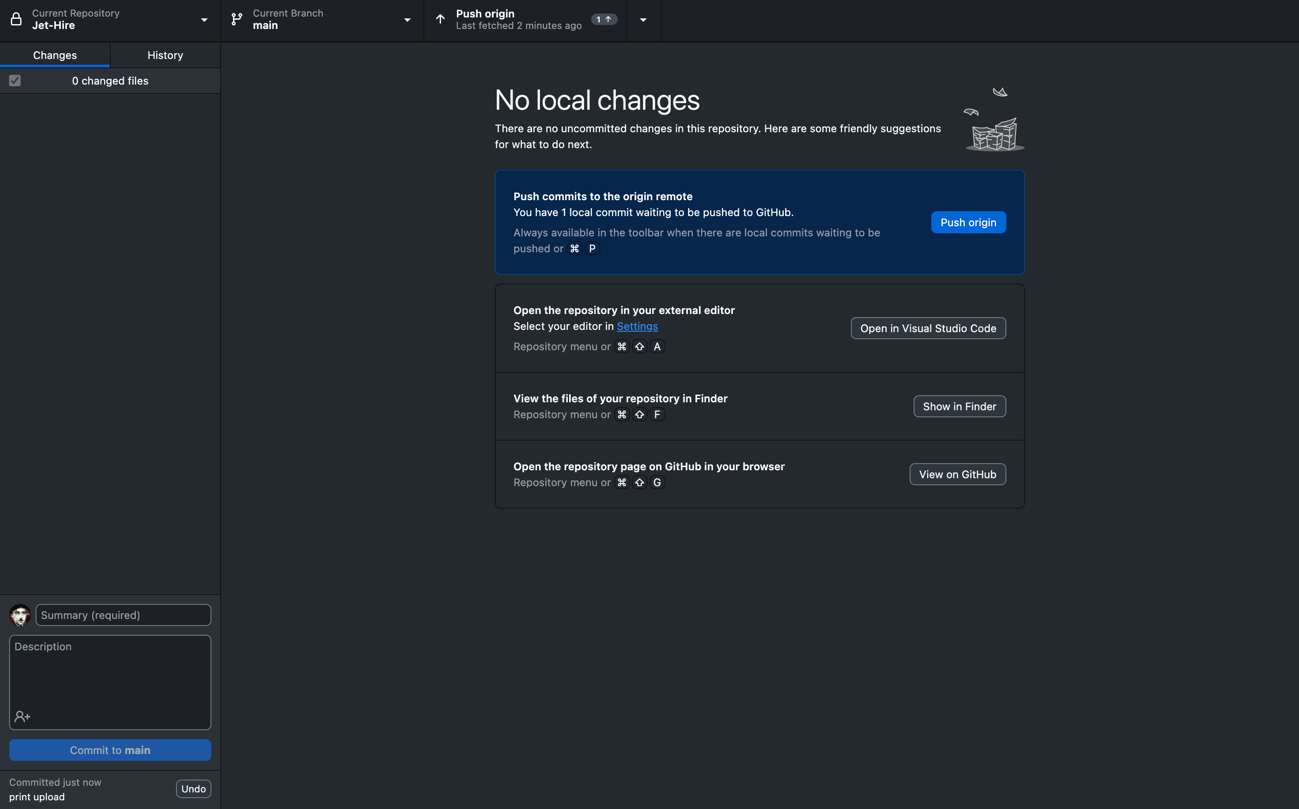


Figura 24 - Sincronização entre dois computadores: Passo 8

***Sitemap***

Para o desenho do projeto decidi criar um *sitemap* na aplicação web FlowMapp. Optei por não criar um modelo DER visto que não vou utilizar uma base de dados relacional. Para uma melhor organização, optei por criar dois *sitemaps* separados, um para as páginas principais e outro para a *dashboard* geral.

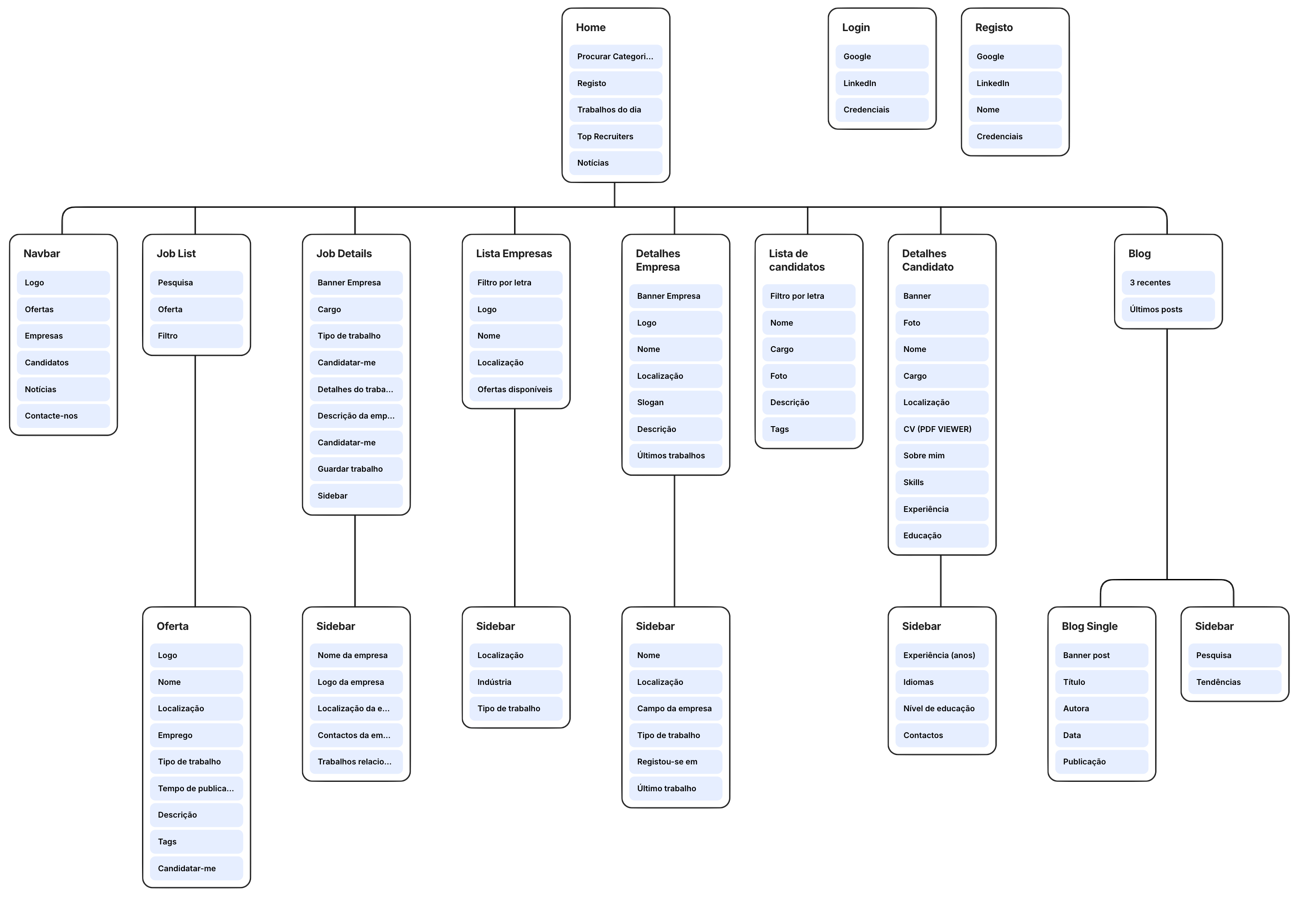


Figura 25 – Sitemap principal

# Projeto

## Instalação do *template*

Após ter escolhido o template, o próximo passo é o *download.*

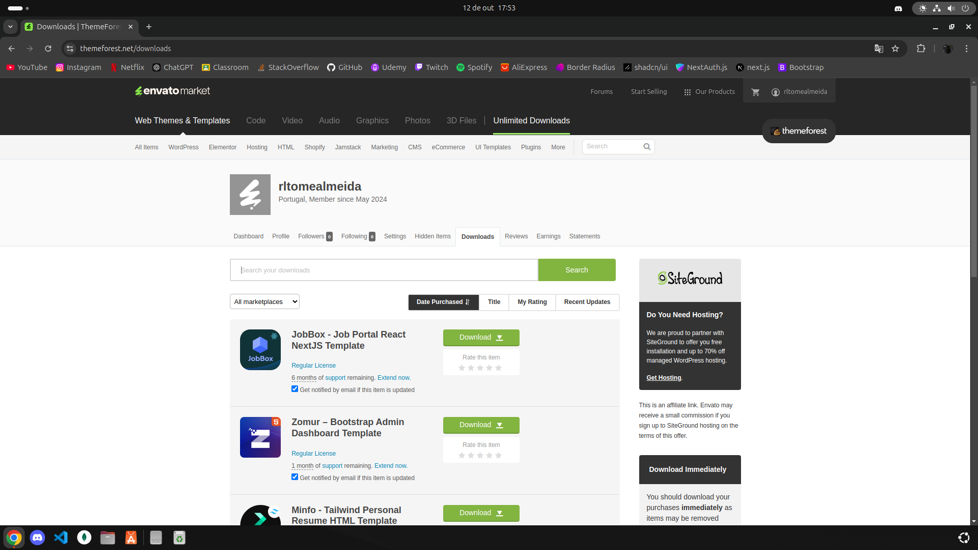
**

Figura 27 - Download do template

De seguida, abre-se a pasta anteriormente transferida através do Visual Studio Code. Cria-se um novo terminal e acede-se á pasta específica através do comando “cd [Nome da pasta]”. Quanto estivermos dentro da pasta passamos á instalação dos *node\_modules* e do ficheiro *package-lock.json* através do comando “npm install”.

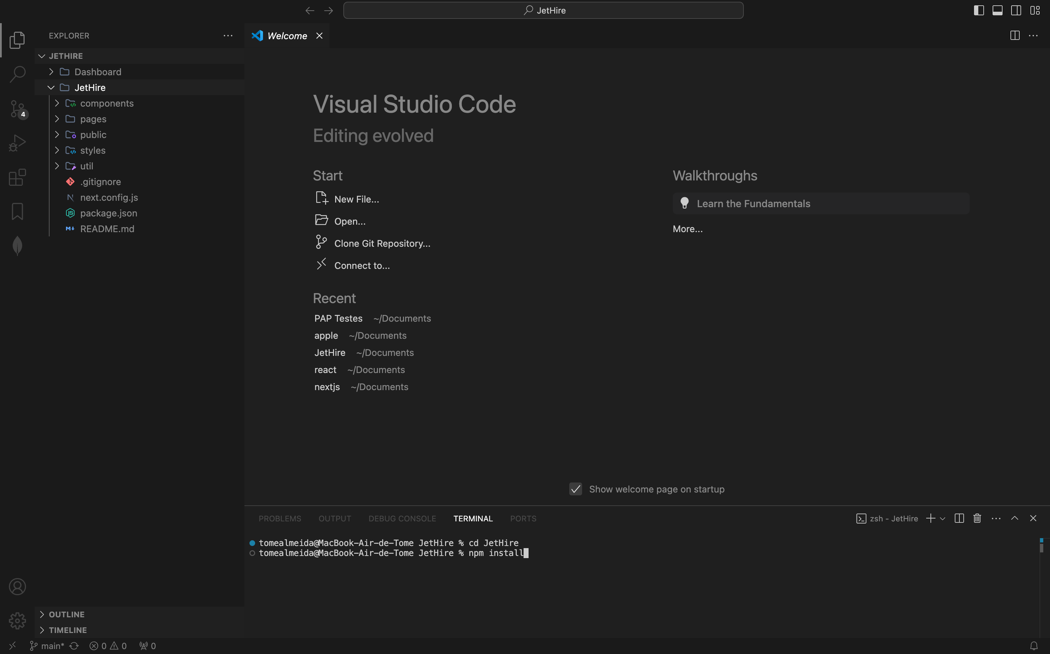


Figura 28 - Instalação dos node\_modules

Quando a instalação tiver terminado, o projeto está pronto a ser executado através do comando “npm run dev”.

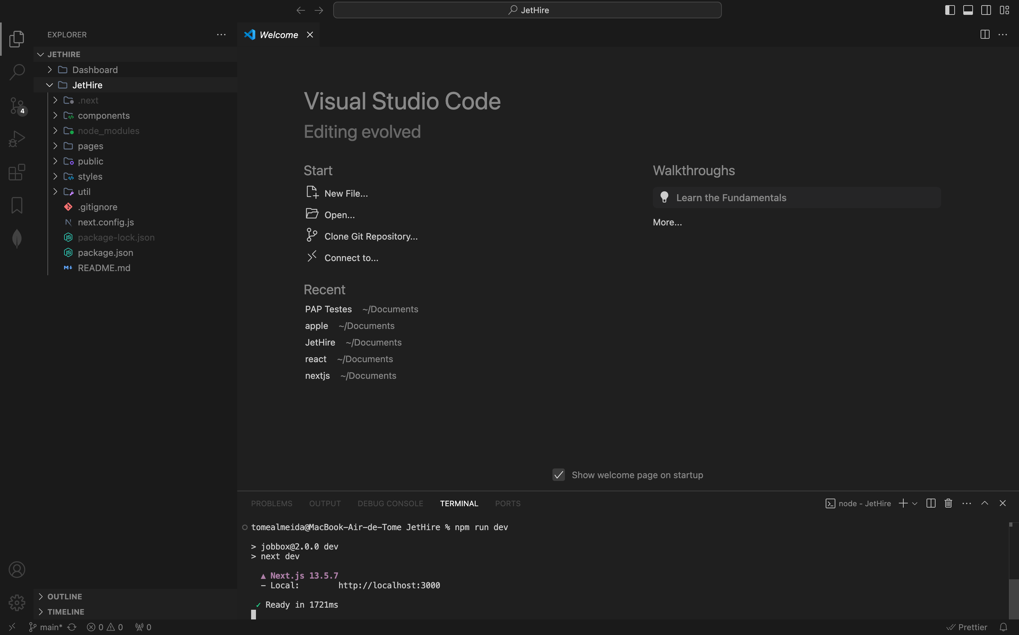


Figura 29 - npm run dev

Ao abrir o link apresentado no terminal, temos acesso ao projeto em tempo real.

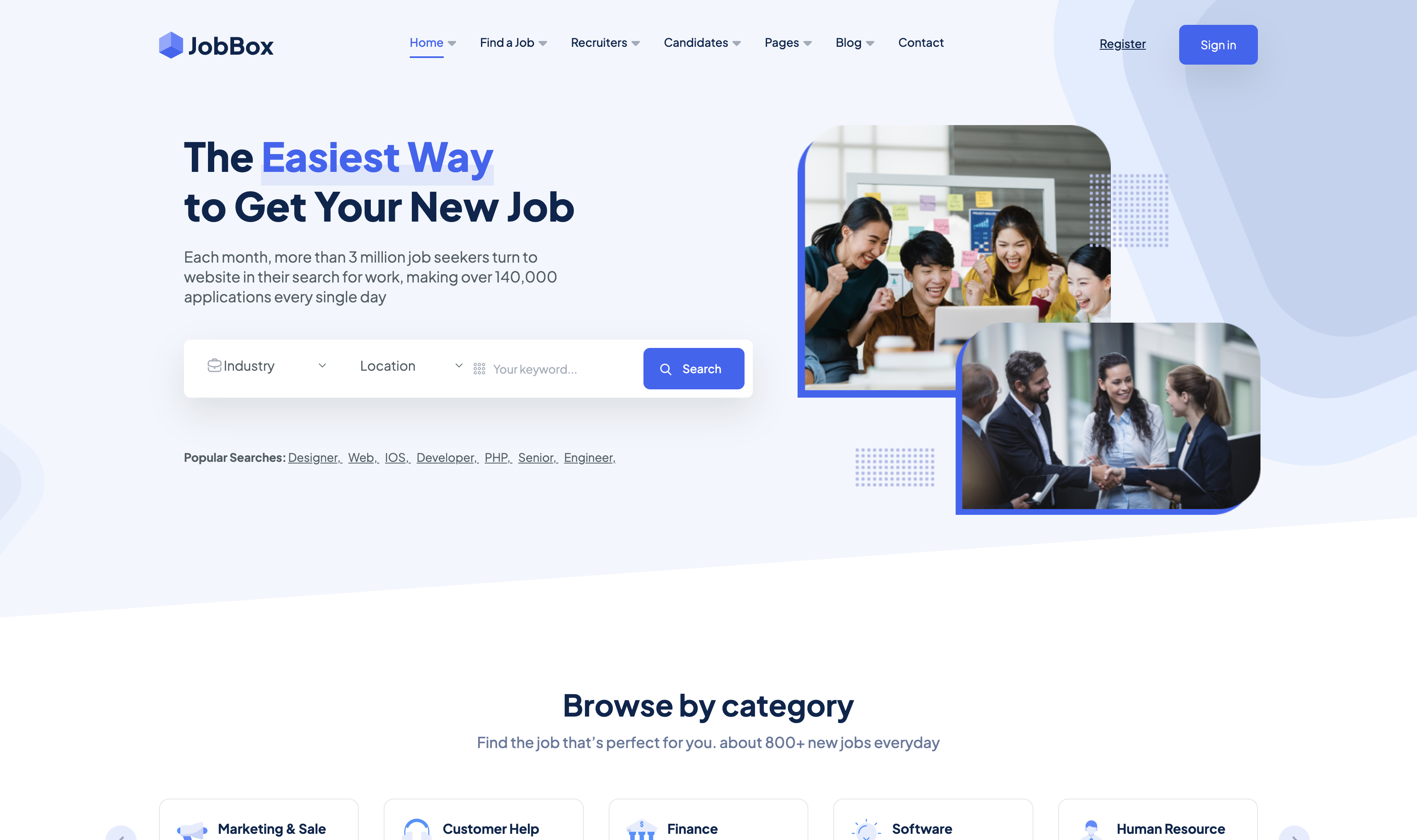


Figura 30 - Projeto executado

### Título

Em Next.js existe uma espécie de “ficheiro-mãe” que é o app.js que é responsável pela junção de todos os elementos da página. Para editar o título da mesma, apenas é preciso importar o módulo *Head* e aplicá-lo dentro da função, obtendo este resultado:

import Head from "next/head";

function MyApp({ Component, pageProps }) {

return (

<>

<Head>

<title>Jet Hire</title>

</Head>

<Component {...pageProps} />

</>

);

}

## 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. | Figura 31 - Logótipo |

## Escolha da palete de cores

### Cor principal

A escolha da palete de cores do meu projeto foi feita através de um guião sobre a estrutura de cores da interface de websites. Utilizei o espaço de cor HSB (*Hue, Saturation, Brightness*) para conseguir criar os tons mais escuros e mais claros. Comecei por escolher a minha cor principal que foi o vermelho.

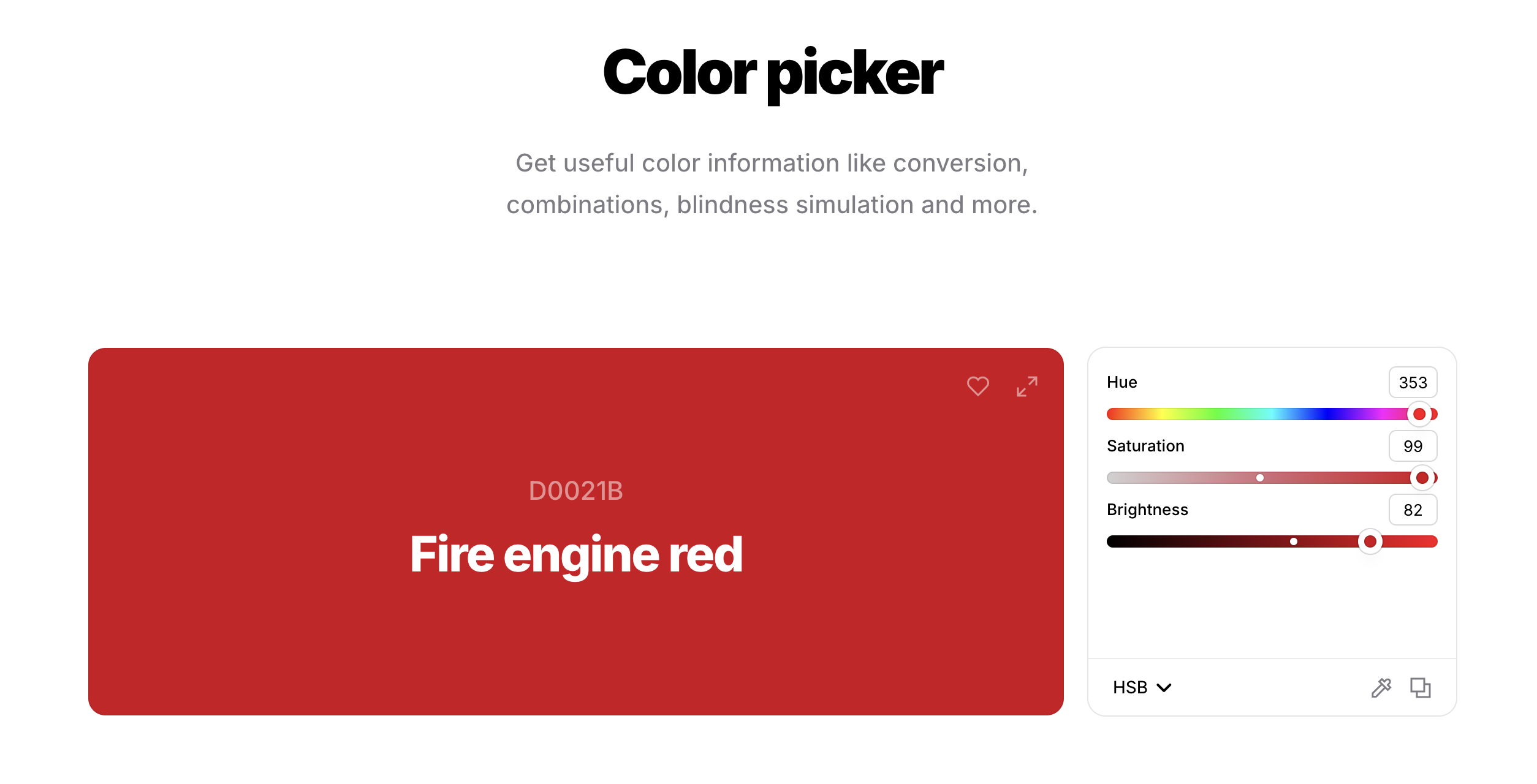


Figura 32 - Escolha da cor principal

Depois da escolha da cor é preciso definir os diferentes tons da cor. O método que utilizei foi escolher o tom mais escuro e mais claro antes dos intermediários.

Para o tom mais claro aumentei a luminosidade e diminuí a saturação.

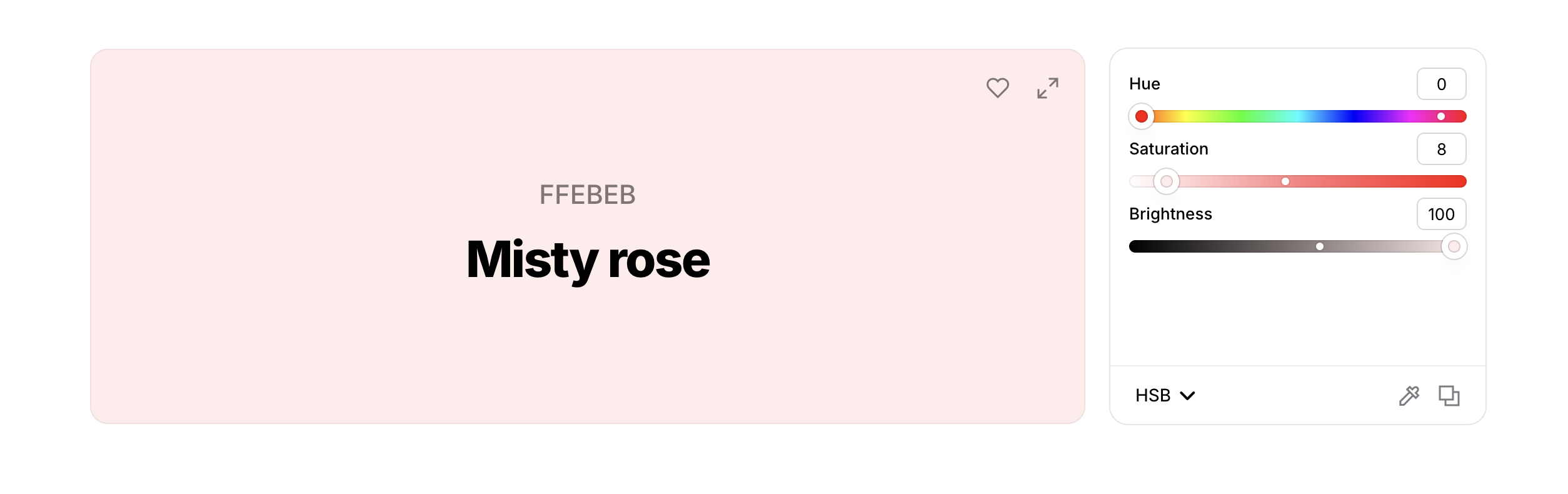


Figura 33 - Escolha do tom claro

Para o tom mais escuro, o processo é o inverso. Diminui-se a luminosidade e aumenta-se a saturação.

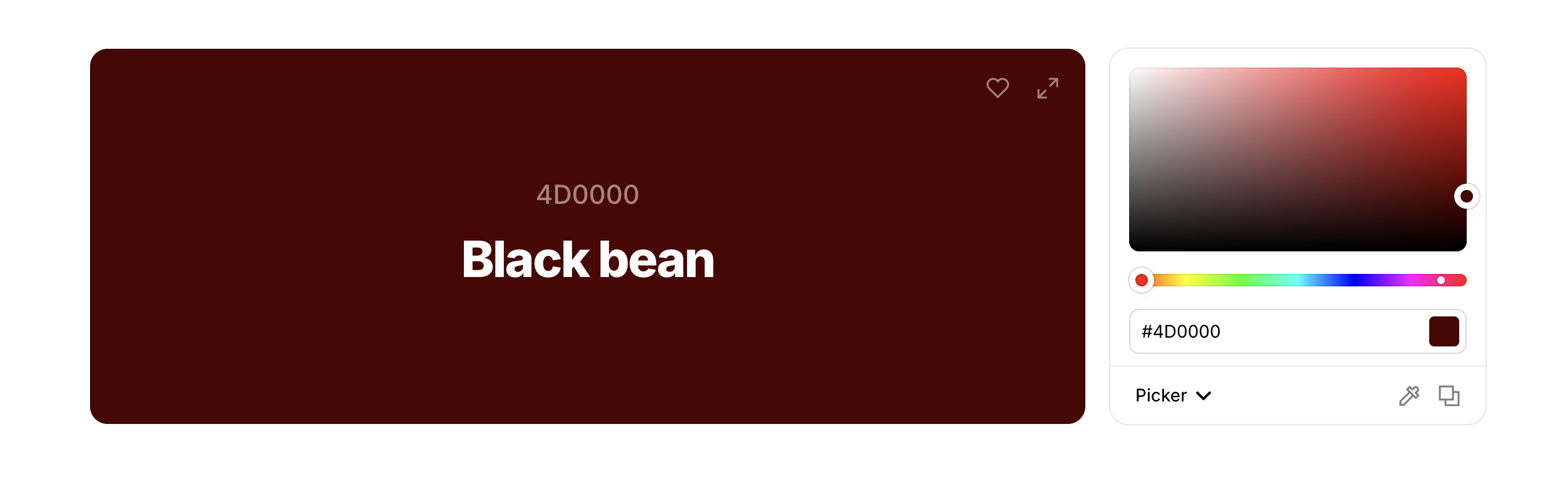


Figura 34 - Escolha do tom escuro

Depois de ter estes três tons, utilizei uma ferramenta que gera gradientes entre duas cores. Primeiro do tom mais claro até à cor principal.

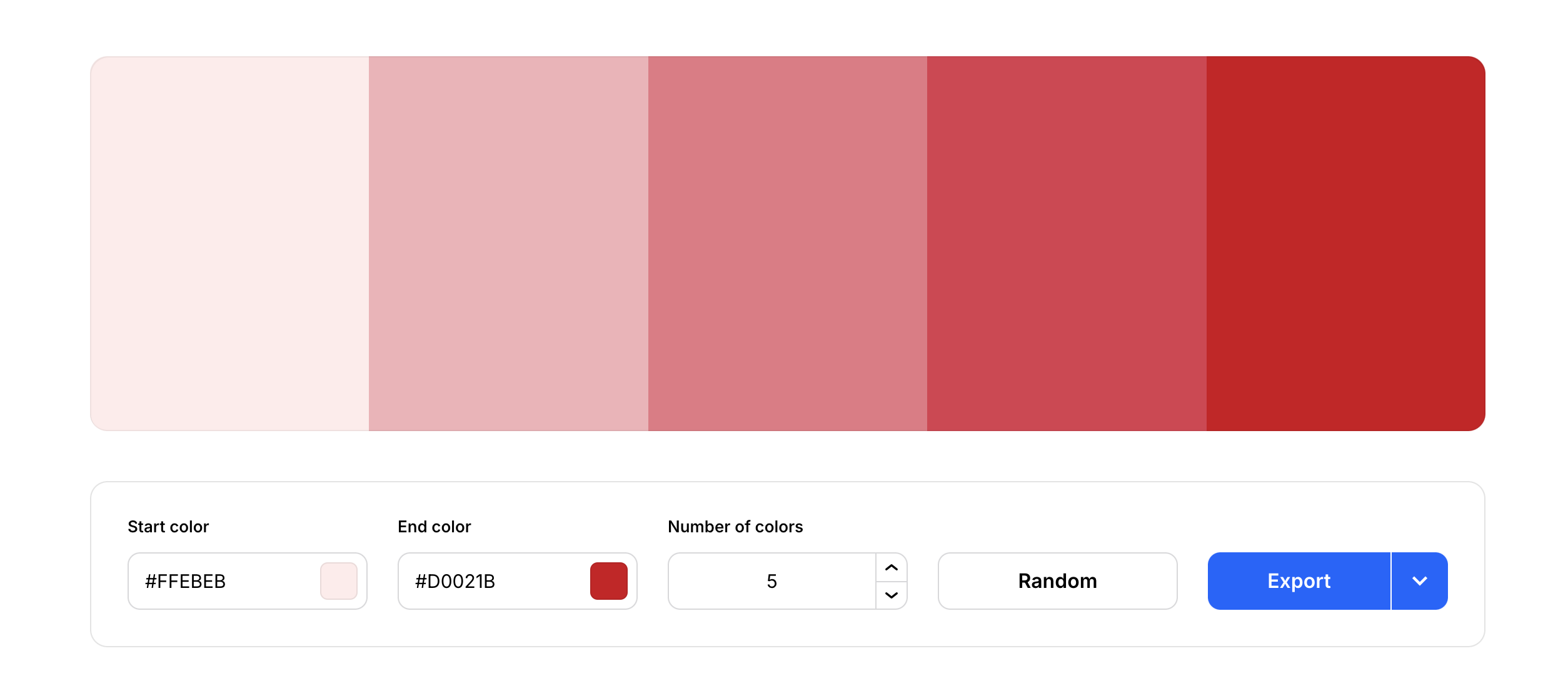


Figura 35 - Criação de um gradiente claro

De seguida, da cor principal até ao tom mais escuro.

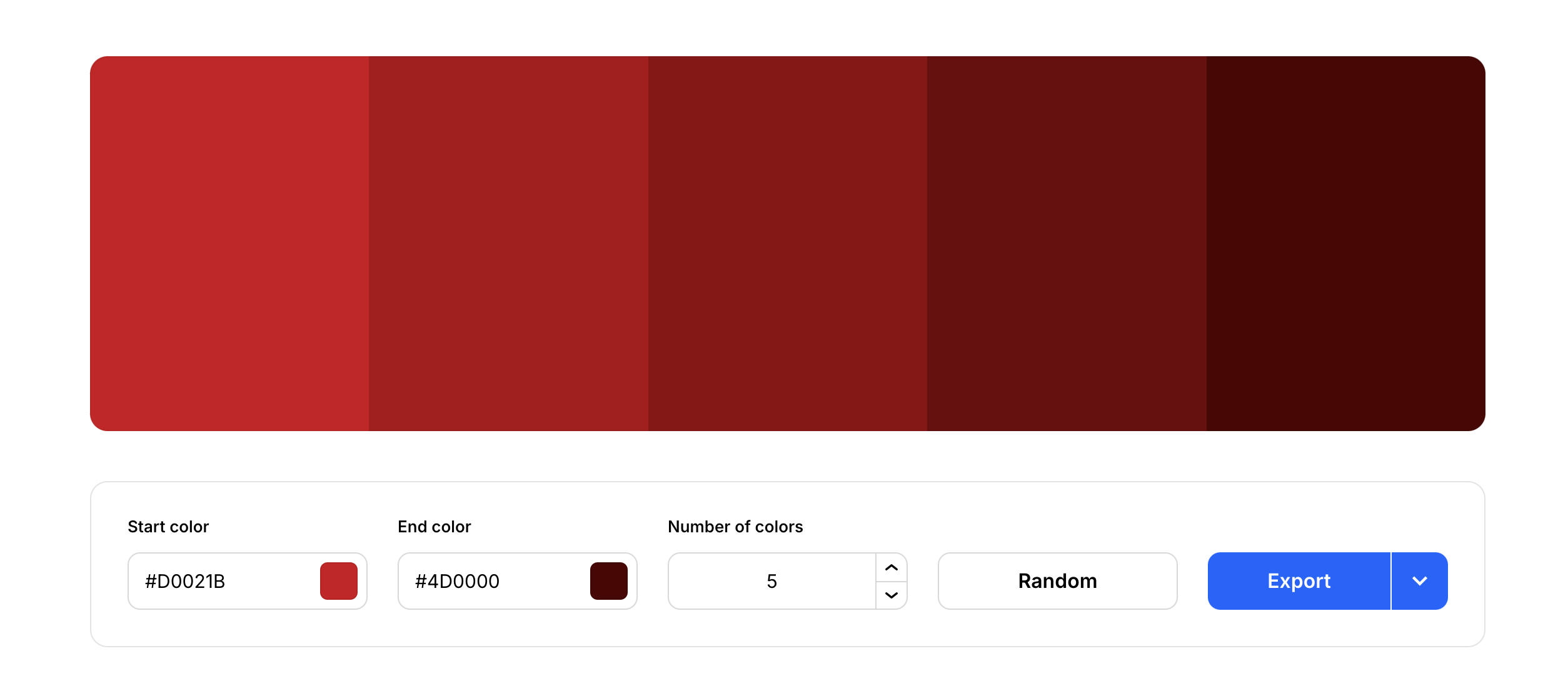


Figura 36 - Criação de um gradiente escuro

E assim, tenho a palete de cores principais do projeto.

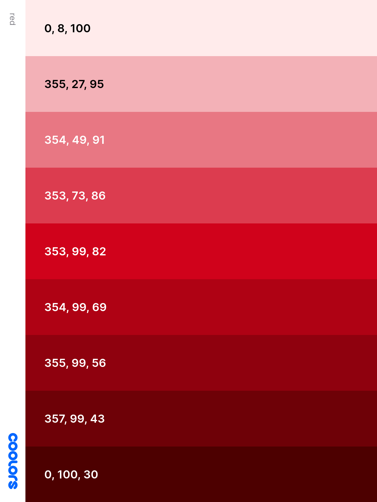


Figura 37 - Vermelho

### Cores secundárias

As cores secundárias vão ser utilizadas em avisos, *pop-ups*, informações, etc. Para estas cores optei pelo amarelo para avisos, azul para informações, verde para confirmações e o vermelho novamente para erros. Repete-se todo o processo da cor principal.

|  |  |  |
| --- | --- | --- |
| Figura 38 - Amarelo | Figura 39 - Azul | Figura 40 - Verde |

### Cores neutras

As cores neutras são as cores que vão acompanhar a cor principal ao longo do site. Estas cores vão desde o branco até ao preto. Novamente o mesmo processo para esta palete.



Figura 41 - Cores neutras

### Criação da palete no projeto

Para adicionar todas as cores no projeto é preciso adicioná-las como variáveis globais. É necessário criar um *root* dentro do ficheiro CSS e adicionar todas as cores.

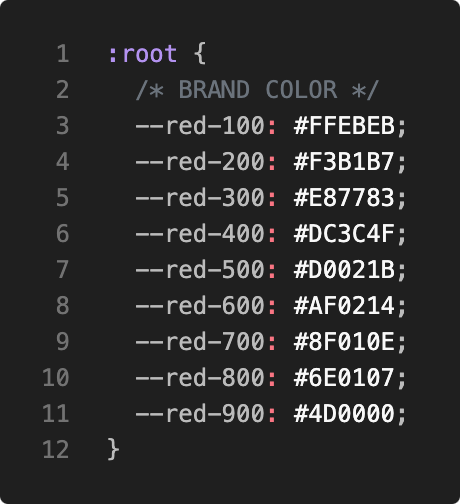


Figura 42 - Exemplo de variáveis globais

# Conclusão