



Aplicações para Internet

Prof. Daniel Brandão

PROGRAMAÇÃO DE HOJE



INTRODUÇÃO

Conceitos de
Versionamento de
Código



DESENVOLVER

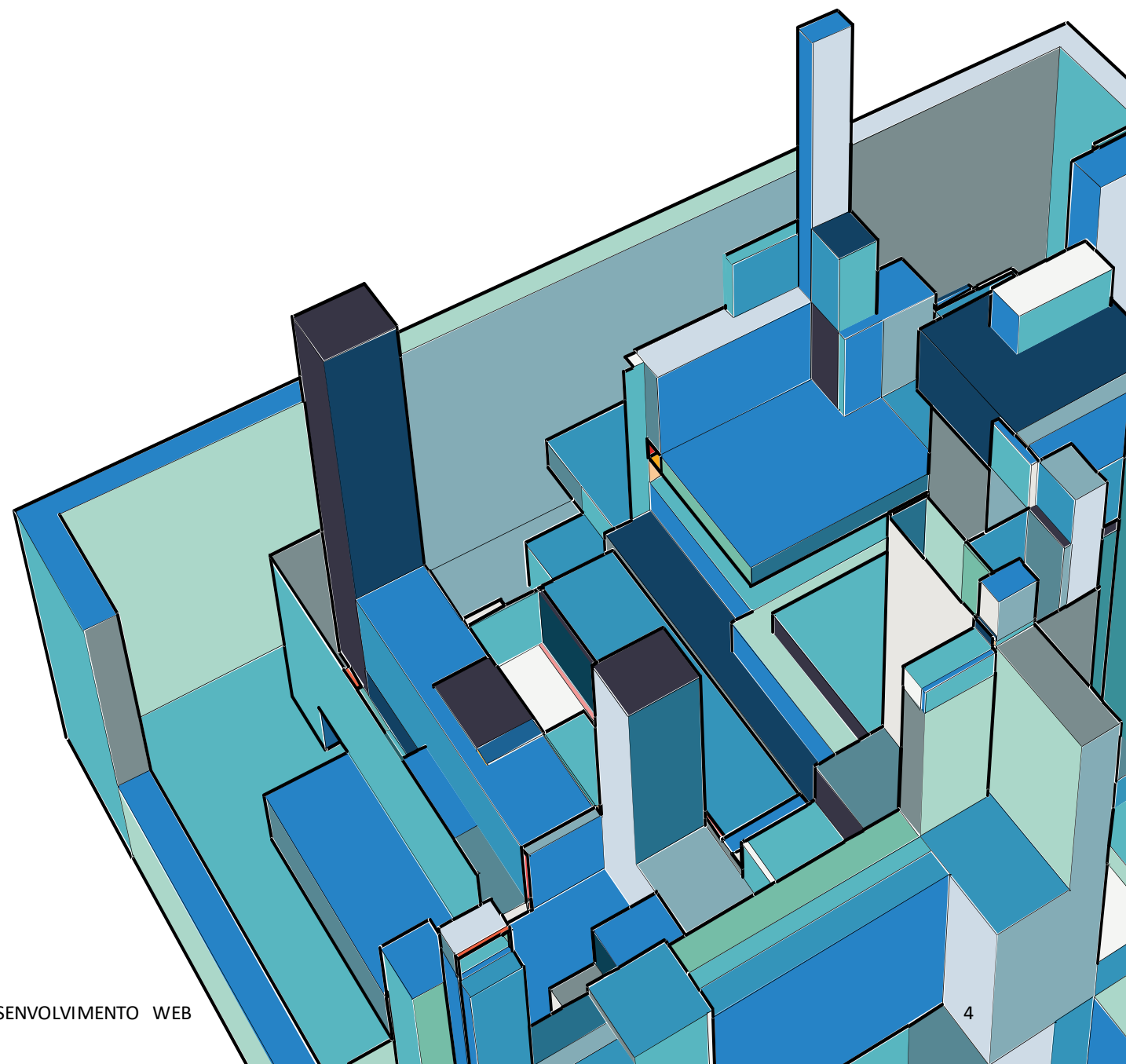
Pesquisa e
Desenvolvimento do
projeto. Envio do tema e
equipe.



PRÁTICA

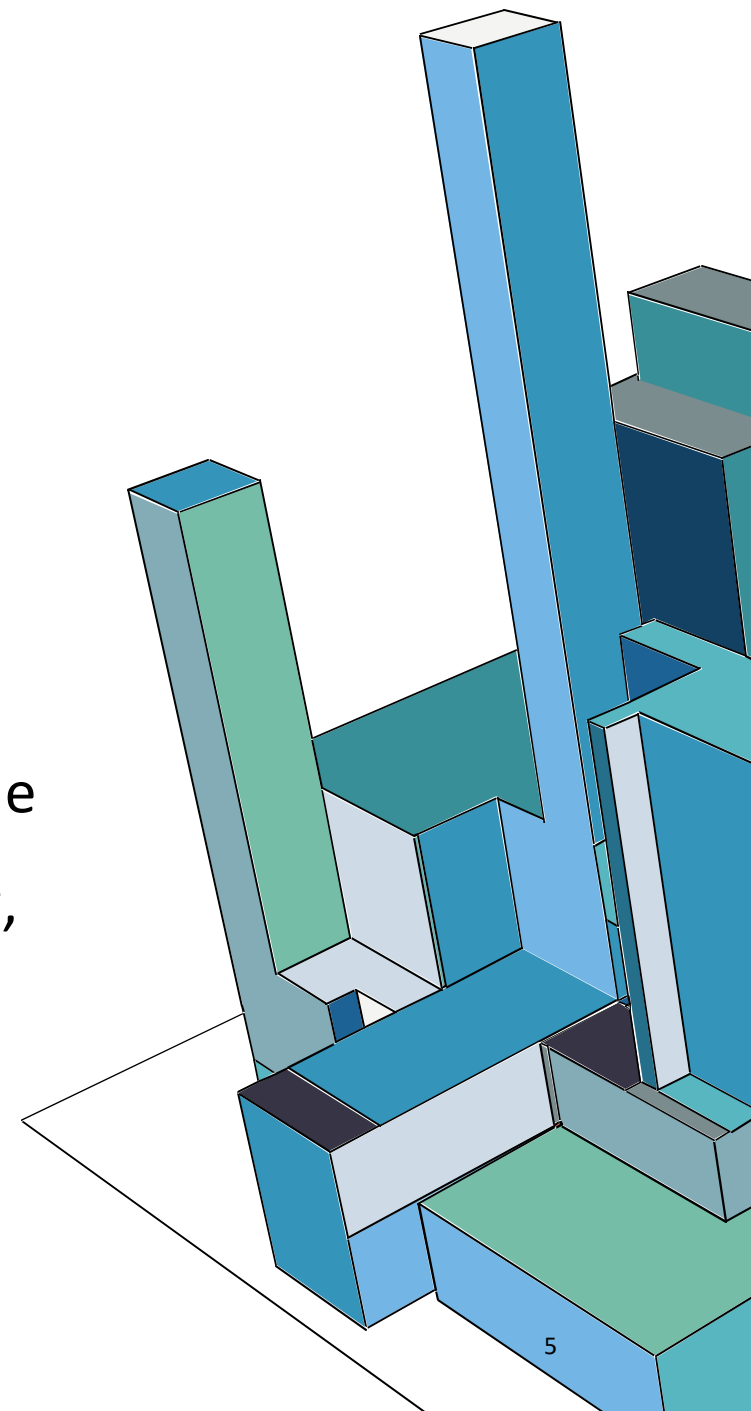
Criação do
Repositório,
Versionamento de
Código e envio.

CONTROLE DE VERSÃO



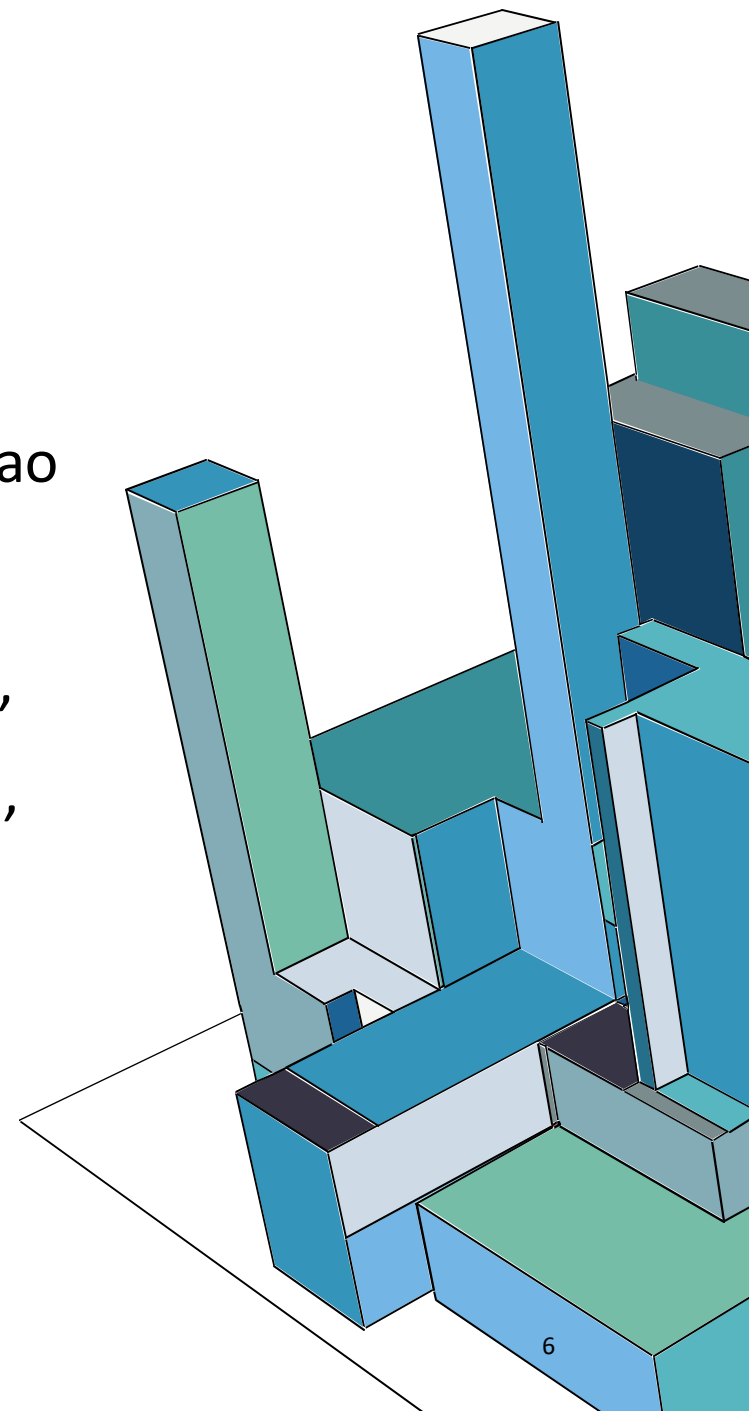
O que é controle de versão?

- O **controle de versão(CV)** é um sistema usado para ter controle sobre todas (isso se for usada corretamente) as mudanças feitas em um determinado arquivo.
- Como exemplo, CV permite você reverter sua aplicação que se encontra em um estado que está apresentando um bug, para um estado anterior onde o bug não havia se manifestado.



O que é controle de versão?

- CV permite você descobrir quem introduziu um problema ao repositório, quando e onde foi introduzido.
- Além disso, quando estiver usando um repositório remoto, não correrá o risco de perder seu arquivos e, melhor ainda, você também não perderá o controle sobre as mudanças feitas localmente.





GIT

- O **Git** é a ferramenta que utilizamos para fazer todo o controle de versão.
- Surgiu quando Linus Torvalds, o criador do Linux, começou a enfrentar problemas com as ferramentas de versionamento da época, quando desenvolvia o kernel do Linux (projeto open-source que ele trabalhava com apoio de uma comunidade) sentindo a necessidade da criação de uma nova ferramenta.



GIT

O Git é, portanto, um projeto de código aberto maduro e com manutenção ativa desenvolvido em 2005, cuja proposta era apresentar algumas features que sistemas antigos não ofereciam, como:

1. Velocidade;
2. Simplicidade;
3. Forte suporte para desenvolvimento não-linear (milhares de ramos paralelos);
4. Completamente distribuído;
5. Capaz de lidar com grandes projetos com velocidade e grande tamanho dos dados.

CLIENTES PARA WINDOWS



GitLab



GitHub



Tortoise Git

PARA LINUX | PARA Mac OS

- Git Force
- Qgit
- GitG

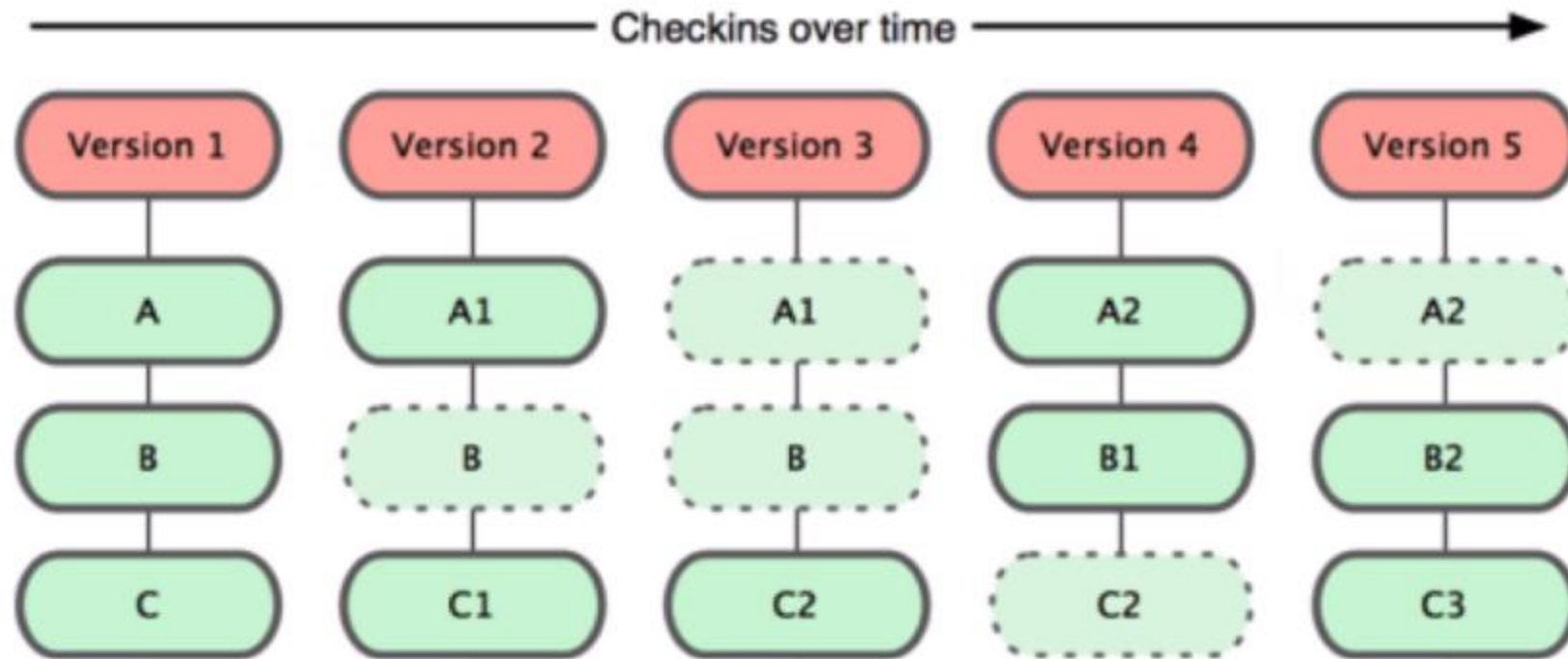
- GitUp
- GitBox
- GitX-Dev



GITHUB

- Mais famosos repositório online, o GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git.
- Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo.

VERSIONAMENTO NO SISTEMA GIT



VERSIONAMENTO e RAMIFICAÇÃO



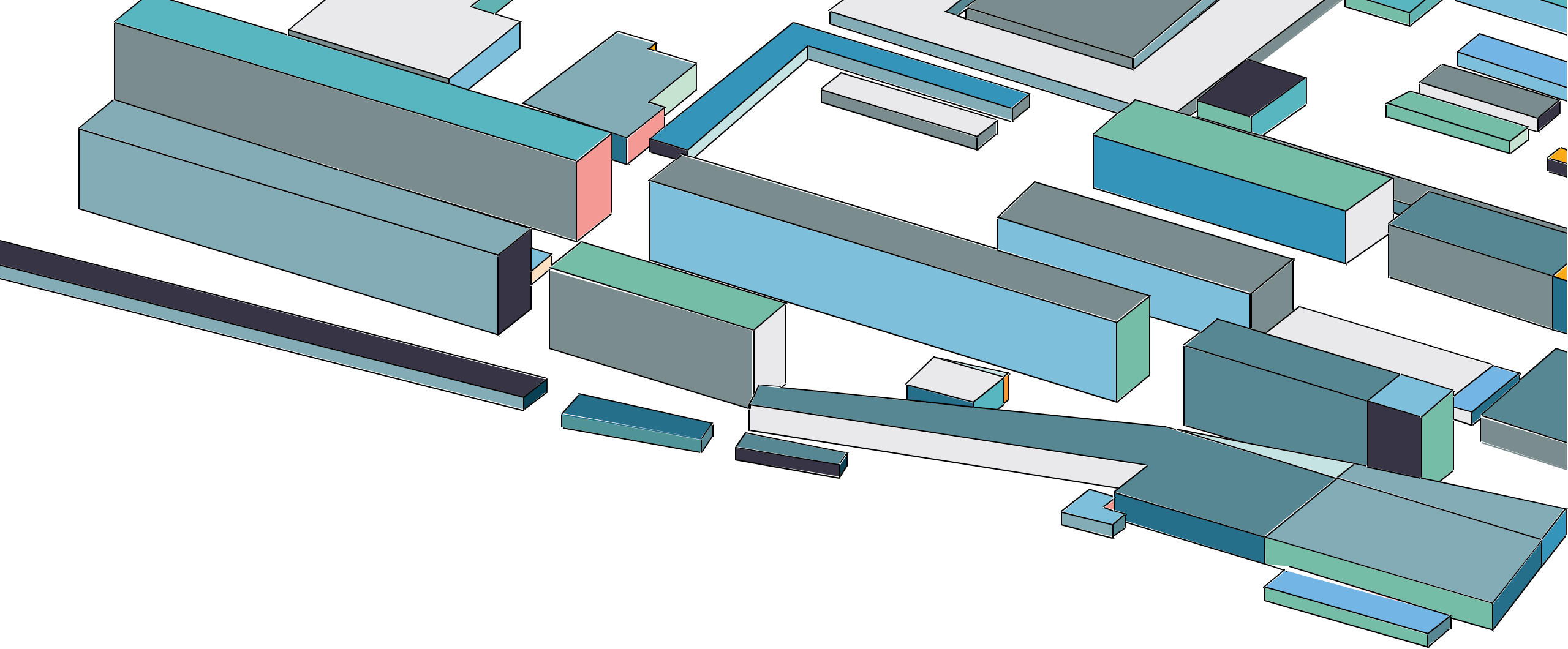
VERSIONAMENTO e RAMIFICAÇÃO



MASTER: MANTER A LINHA DO TEMPO “SAGRADA”

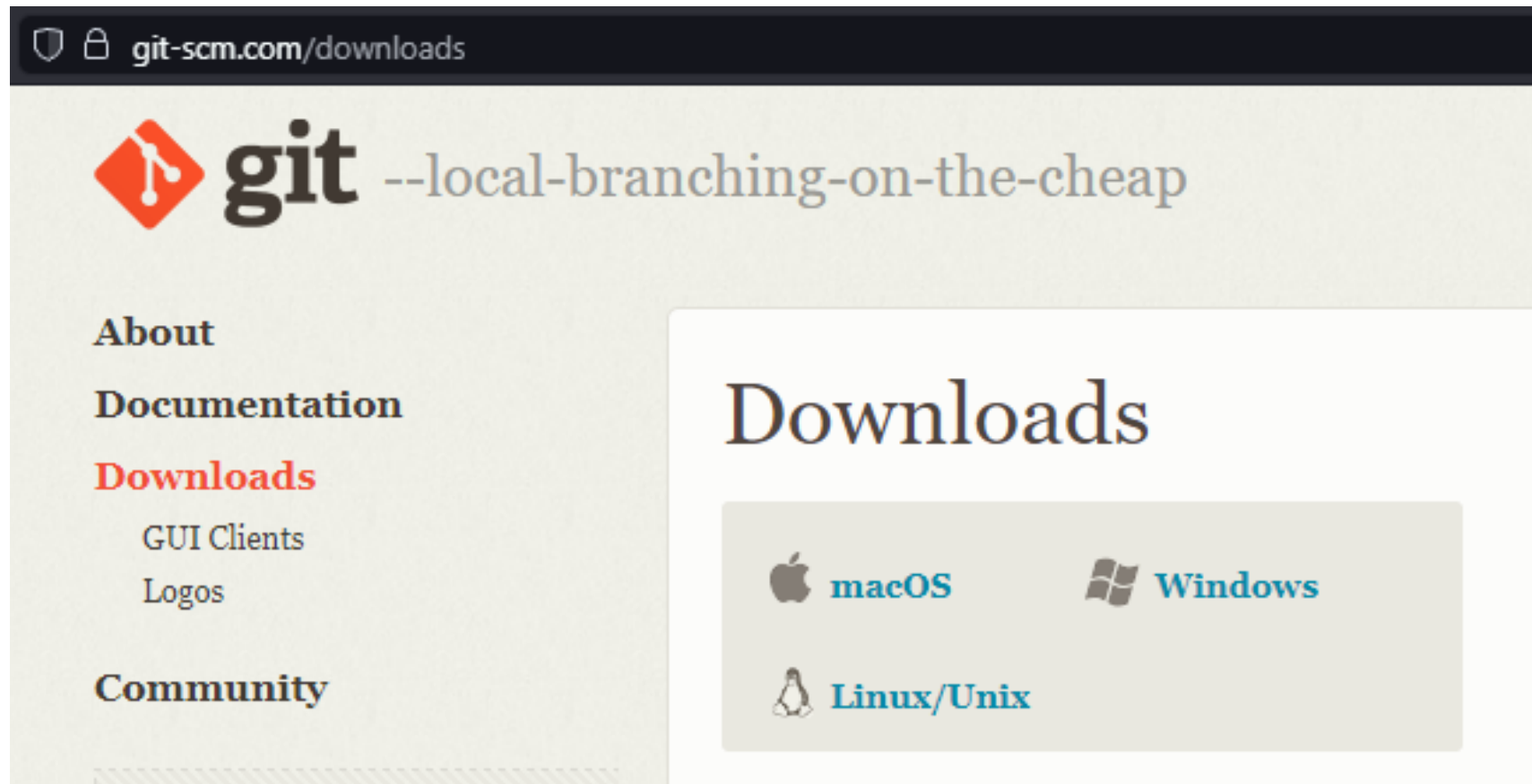


VERSIONAMENTO e RAMIFICAÇÃO



COMEÇANDO NO GIT

COMEÇANDO NO GIT



COMEÇANDO NO GIT



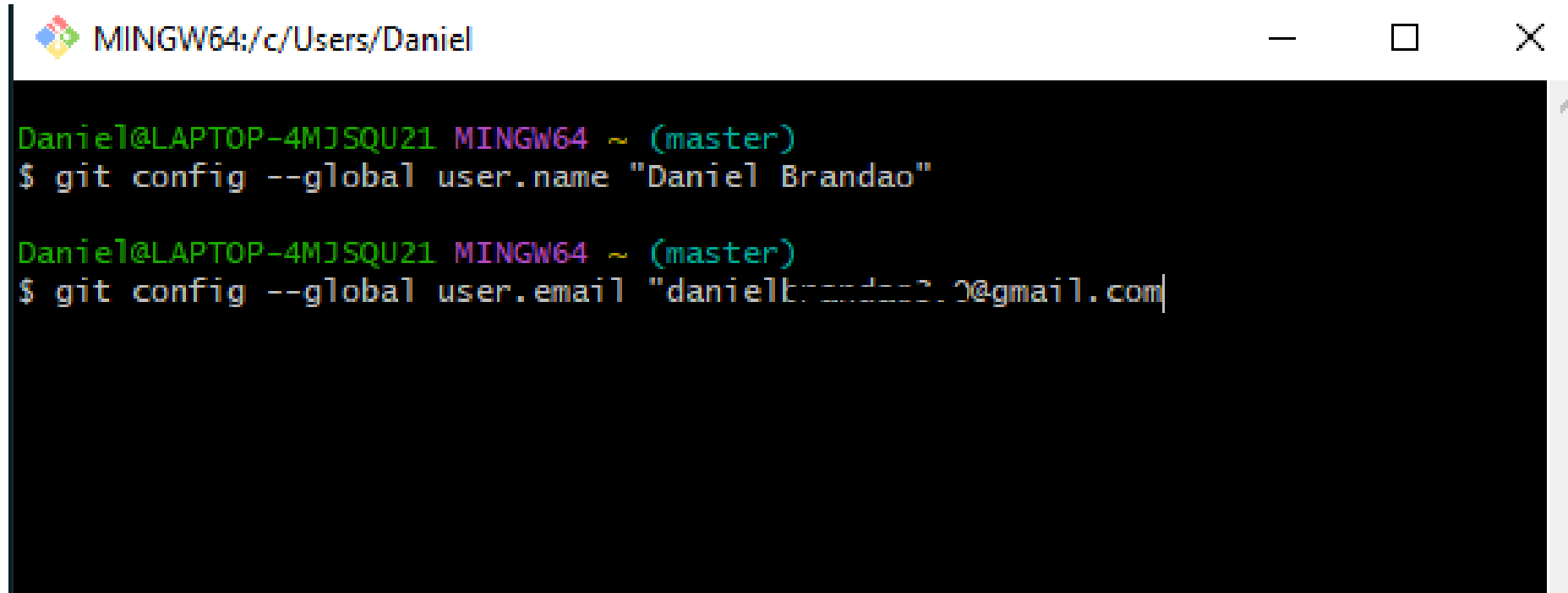
- O Git Bash é o aplicativo para ambientes do Microsoft Windows que oferece a camada de emulação para a experiência de linha de comando Git.
- Bash é acrônimo para "Bourne Again Shell". Shells são aplicativos terminais usados como interface em sistemas operacionais por meio de comandos gravados.

COMEÇANDO NO GIT



```
MINGW64:/c/Users/Daniel  
Daniel@LAPTOP-4MJSQU21 MINGW64 ~ (master)  
$
```

CONFIGURANDO O GIT



A screenshot of a Windows command prompt window. The title bar shows the Windows logo, the text 'MINGW64:/c/Users/Daniel', and standard window controls (minimize, maximize, close). The command prompt has a black background with green text. The prompt shows the user 'Daniel' at 'LAPTOP-4MJSQU21' in a 'MINGW64' environment, currently in the '~ (master)' directory. The first command entered is '\$ git config --global user.name "Daniel Brandao"'. The second command is '\$ git config --global user.email "danielbrandao2.0@gmail.com"', with the cursor at the end of the email address.

```
Daniel@LAPTOP-4MJSQU21 MINGW64 ~ (master)
$ git config --global user.name "Daniel Brandao"

Daniel@LAPTOP-4MJSQU21 MINGW64 ~ (master)
$ git config --global user.email "danielbrandao2.0@gmail.com|
```

VENDO AS CONFIGURAÇÕES

```
Daniel@LAPTOP-4MJSQU21 MINGW64 ~ (master)
$ git config user.name
Daniel Brandao
```

```
Daniel@LAPTOP-4MJSQU21 MINGW64 ~ (master)
$ git config user.email
danielbrandao@gmail.com
```

CRIANDO PASTA E ACESSANDO

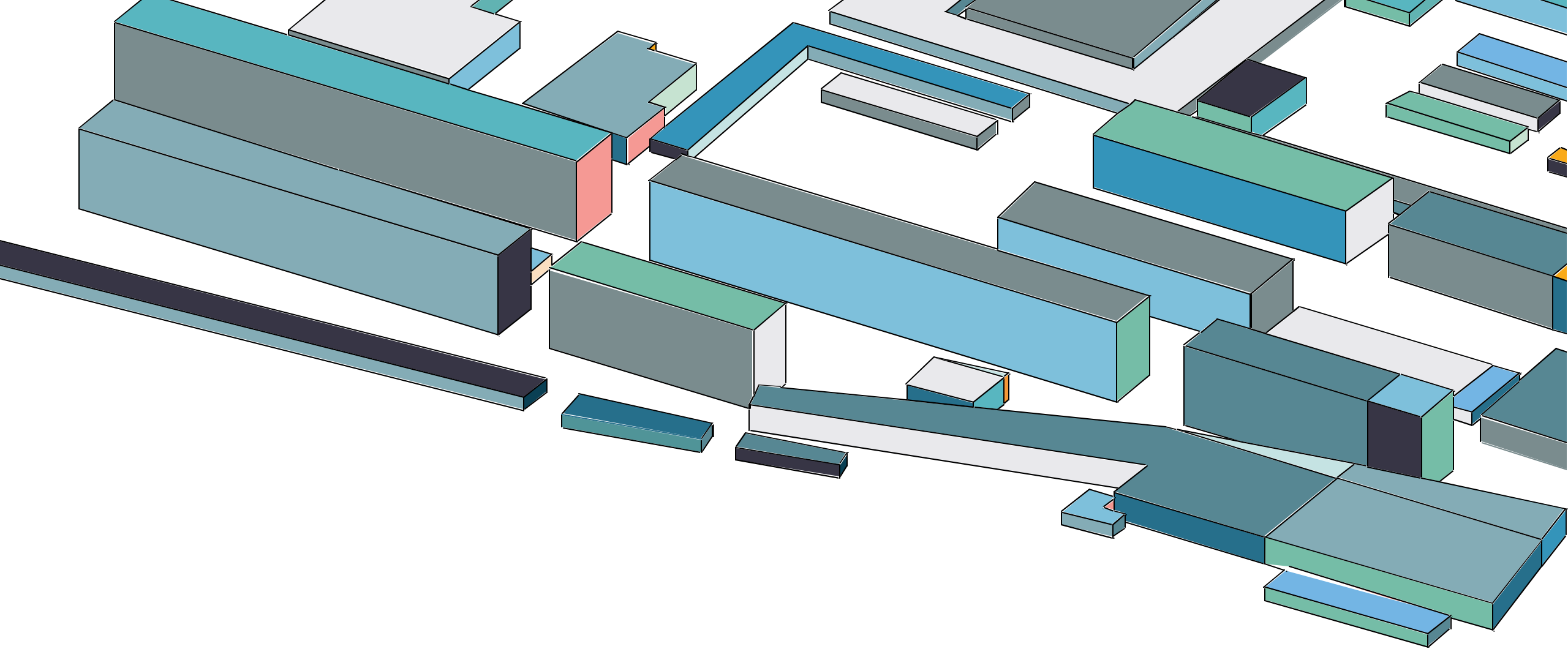
```
Daniel@LAPTOP-4MJSQU21 MINGW64 /e/Users/Daniel/Documents
$ mkdir pos-web

Daniel@LAPTOP-4MJSQU21 MINGW64 /e/Users/Daniel/Documents
$ cd po
|                               pos-web/

Daniel@LAPTOP-4MJSQU21 MINGW64 /e/Users/Daniel/Documents
$ cd pos-web/
```

INICIANDO REPOSITÓRIO LOCAL

```
Daniel@LAPTOP-4MJSQU21 MINGW64 /e/Users/Daniel/Documents/pos-web  
$ git init  
Initialized empty Git repository in E:/Users/Daniel/Documents/pos-web/.git/
```



PRIMEIROS COMANDOS

PRIMEIROS COMANDOS

Git status – Lista o status atual dos arquivos no repositório atual

```
Daniel@LAPTOP-4MJSQU21 MINGW64 /e/Users/Daniel/Documents/pos-web (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

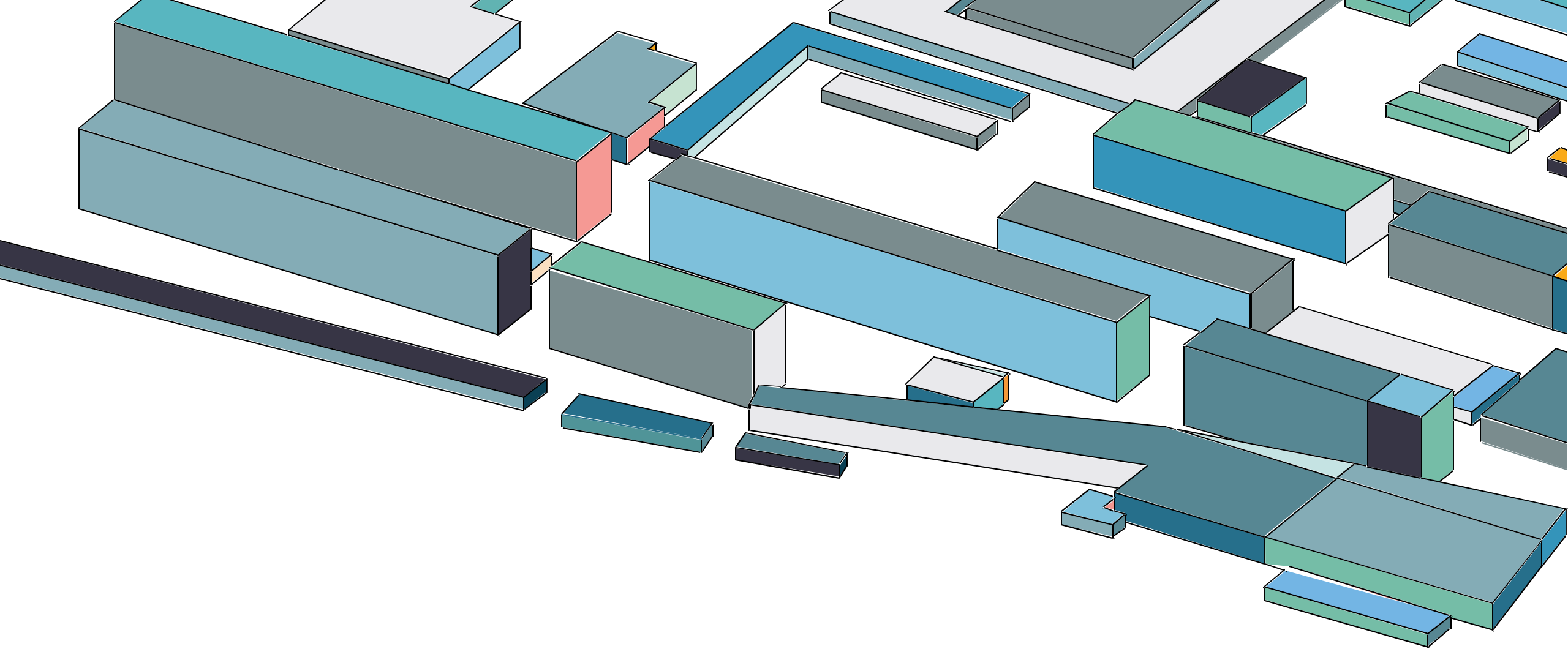
PRIMEIROS COMANDOS

Git add arquivo – adiciona um ou mais arquivos ao status de pronto para dar **commit***

*No contexto de gerenciamento de dados e controle de versão, commit refere-se ao processo de tornar permanente um conjunto de alterações, ou seja, de efetivar as alterações.

PRIMEIROS COMANDOS

Git commit -m “arquivos adicionados” – limpa a lista de arquivos alterados e os prepara para subir ao repositório remoto



CRIANDO REPOSITÓRIO REMOTO



Register for GitHub Universe

Get early bird passes for 20% off



Criar ou logar na conta Github.com

The complete developer platform to build,
scale, and deliver secure software.

Email address

Sign up for GitHub

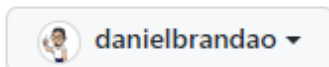
CRIANDO REPOSITÓRIO REMOTO

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 *

NOME DO
REPOSITÓRIO

Great repository names are short and memorable. Need inspiration? How about [curly-computing-machine?](#)

Description (optional)



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.

DEPOIS CLIQUE
AQUI

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.](#)

PÓS EM DESENVOLVIMENTO WEB FULL STACK

Create repository

CRIANDO REPOSITÓRIO REMOTO

AQUI ESTÁ A URL DO NOVO REPOSITÓRIO

 projeto1 Public

 Pin  Unwatch 1  Fork 0  Star 0



Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

[Get started with GitHub Copilot](#)



Add collaborators to this repository

Search for people using their GitHub username or email address.

[Invite collaborators](#)

Quick setup — if you've done this kind of thing before



Set up in Desktop

or

HTTPS

SSH

<https://github.com/dbrandaoford/projeto1.git>



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

CONECTANDO AO REPOSITÓRIO REMOTO

Usando o Gitbash, use o comando:

Git remote add origin

<https://github.com/nomeusuario/projeto1.git>

– conectando ao repositório remoto do endereço copiado na criação do repositório

CONECTANDO AO REPOSITÓRIO REMOTO

git init

git add README.md

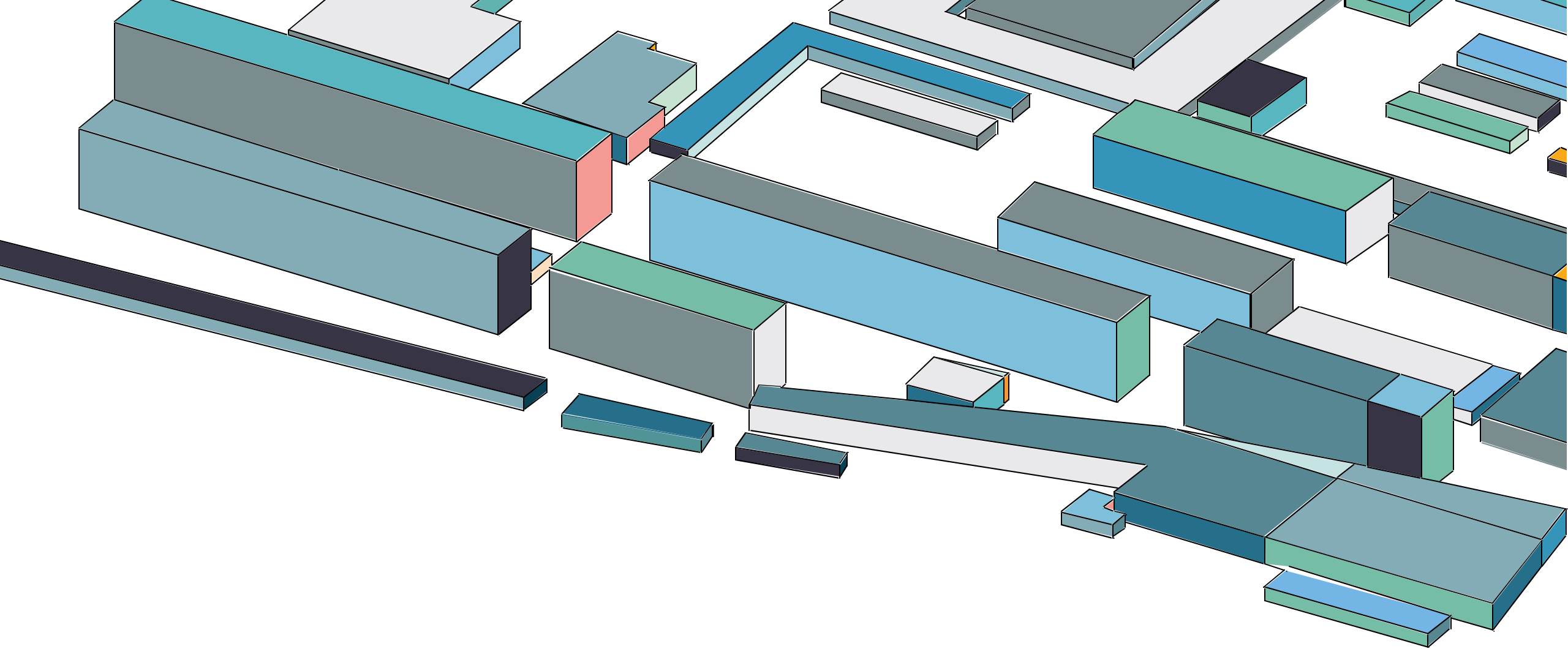
git commit -m "first commit"

git branch -M main

git remote add origin

<https://github.com/nomeusuario/projeto1.git>

git push -u origin main



ATUALIZANDO O REPOSITÓRIO

ATUALIZANDO O REPOSITÓRIO

git status

git add .

git commit -am "primeiro commit"

git push origin branch

*** No lugar de BRANCH coloca-se o nome da branch**

CRIANDO BRANCH NOVA

git status

git checkout main

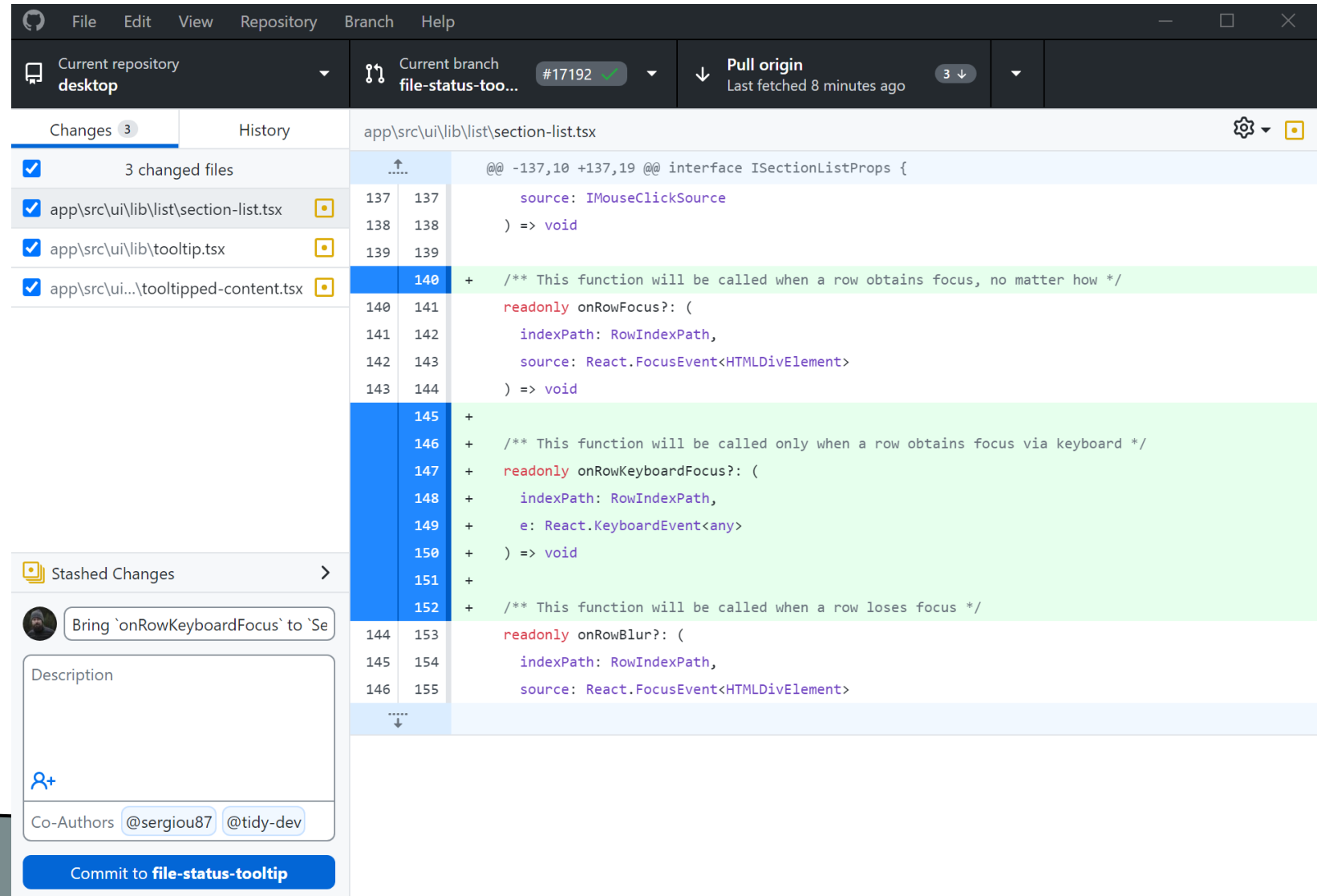
git checkout -b novabranch

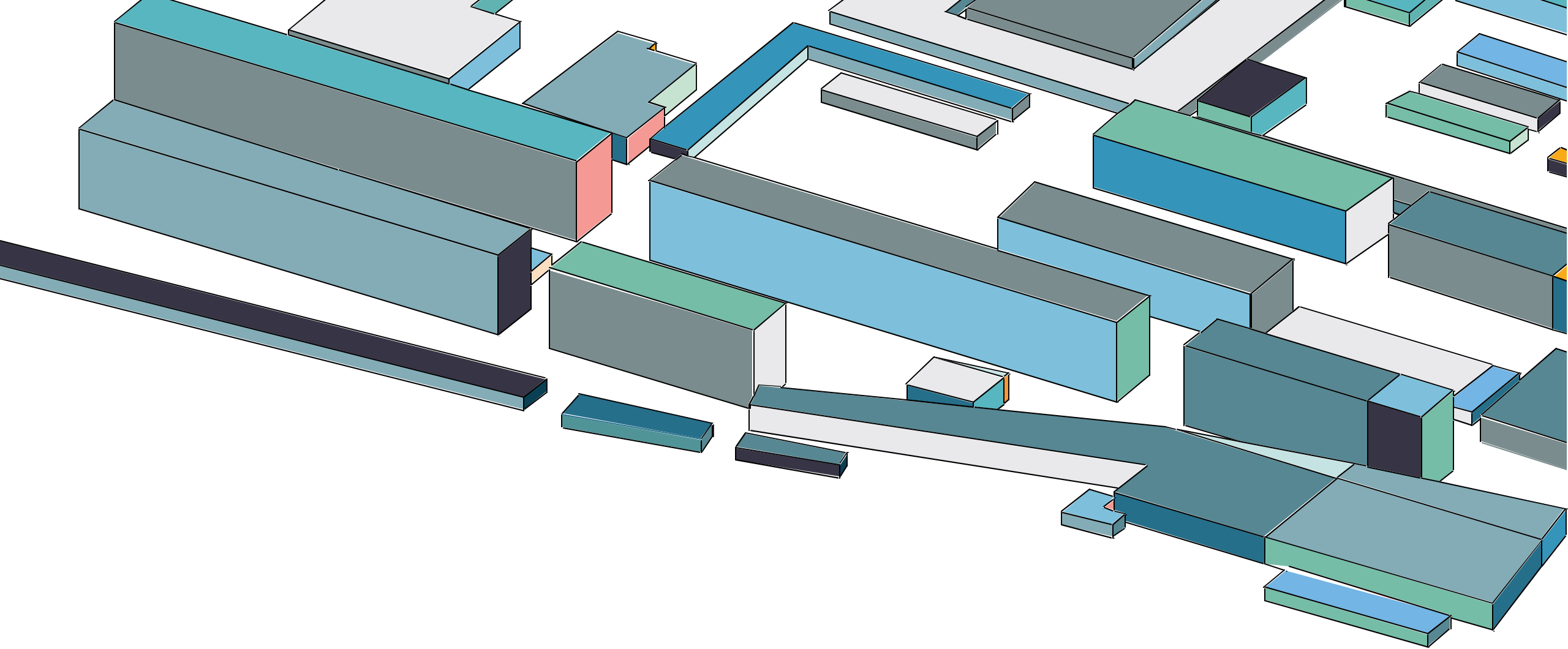
*** No lugar de NOVABRANCH coloca-se o nome da branch a ser criada como cópia da MAIN**

ATUALIZANDO O REPOSITÓRIO

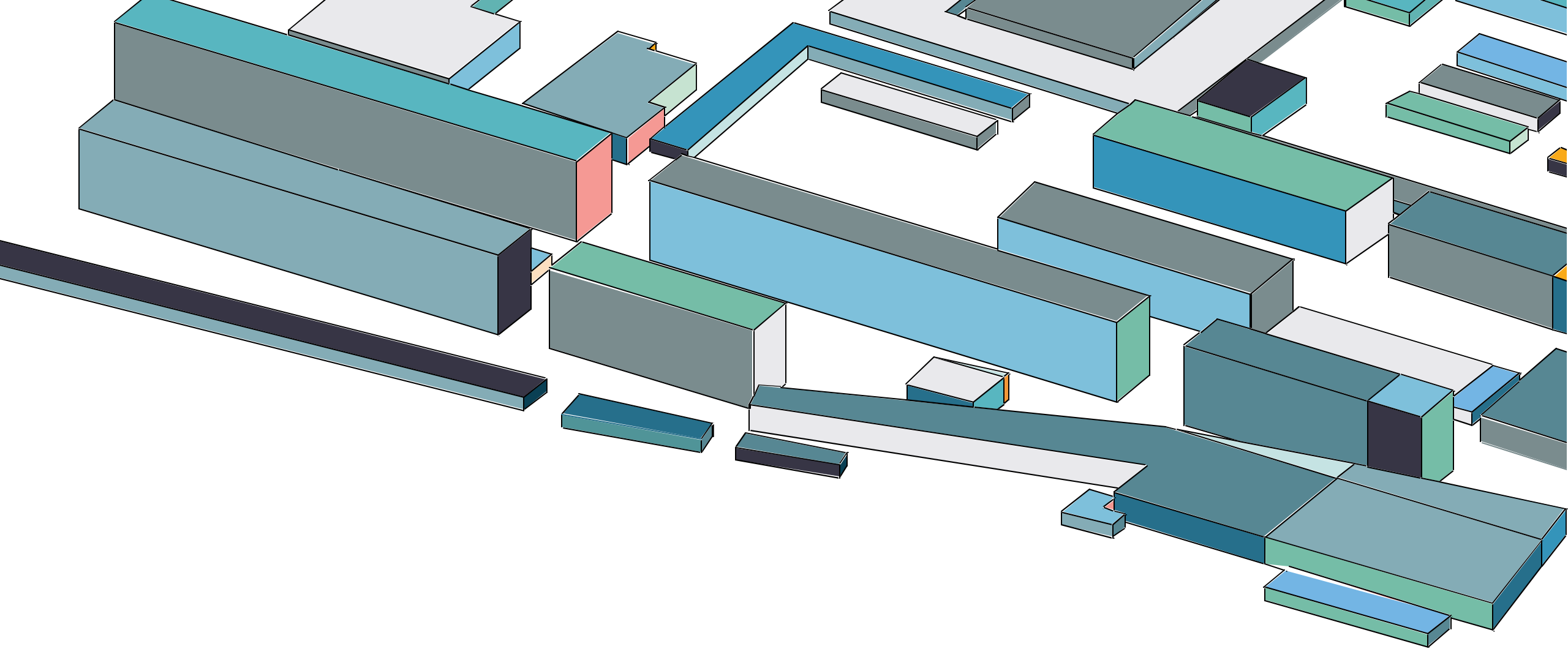
Github Desktop

- Baixe o programa
- conecte seu usuário
- Crie sua branch como Cópia da branch main





CRIANDO REPOSITÓRIO REMOTO



PROJETO PRÁTICO

O QUE O PROJETO PRECISA TER



#1

CRIAÇÃO **USUÁRIOS**



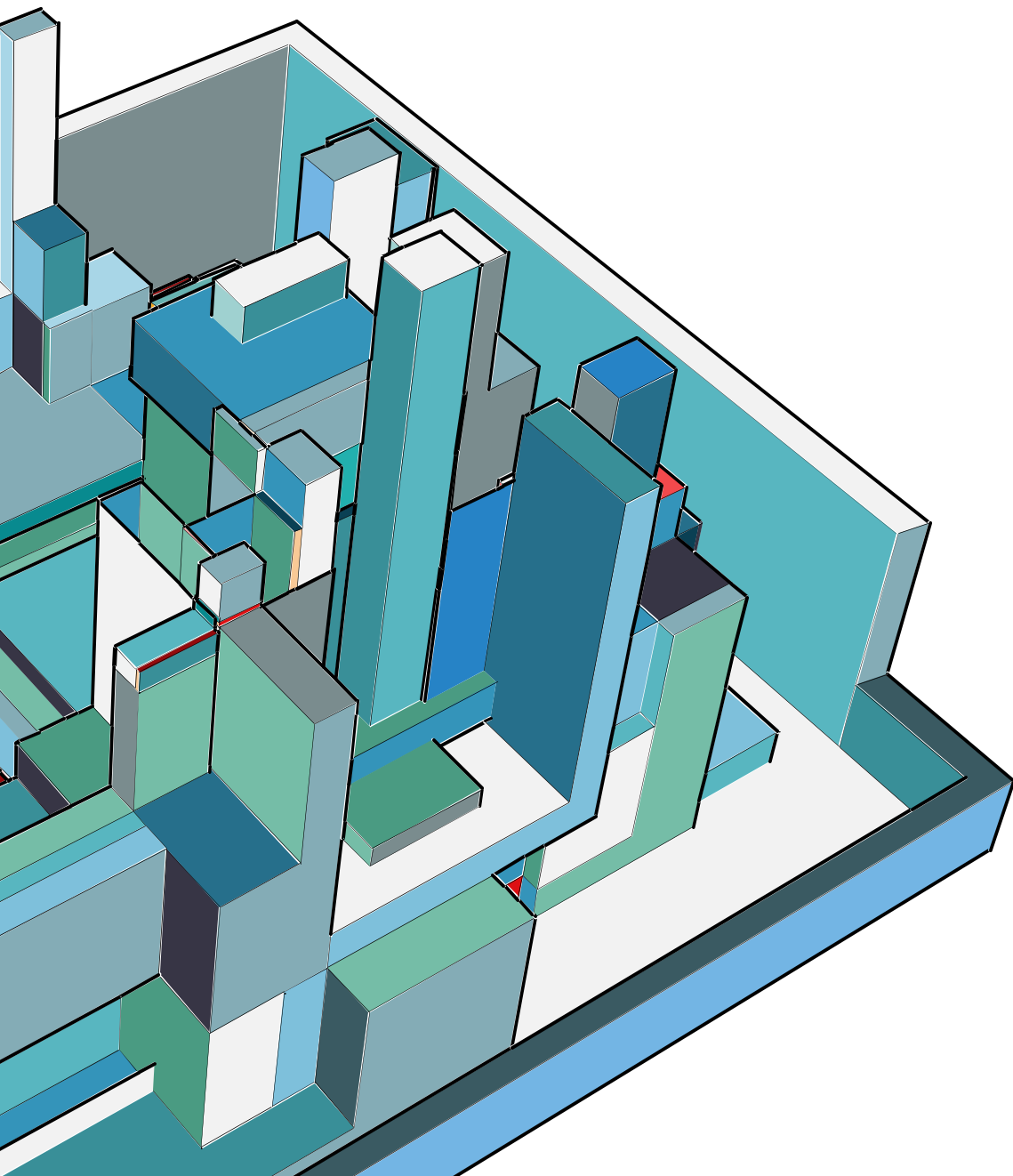
#2

CRIAÇÃO DO **REPOSITÓRIO**



#3

ADICIONAR **README** E
COLABORADORES AO
REPOSITÓRIO



**ALGUMA
DÚVIDA?**

CONTATOS

Blackboard

professordanielbrandao@Gmail.com

@ProfDanielBrandao