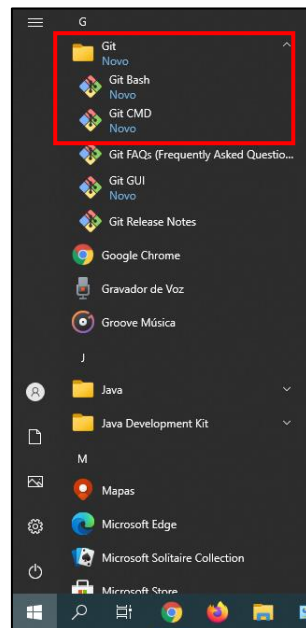


# TUTORIAL PARA ENVIO DOS PROJETOS DE IW-II E DDM-I PARA O GITHUB

PROF. ANDERSON VANIN

1) É necessário ter instalado em seu computador o git.



<https://git-scm.com/>

2) Abra seu github e crie um repositório vazio com o nome do projeto.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Owner <sup>\*</sup> ProfAndersonVanin / Repository name <sup>\*</sup> garconapp ✓

Great repository names are short and memorable. Need inspiration? How about [reimagined-meme?](#)

Description (optional)  
Aplicativo do Garçom

☒ 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.

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.](#)

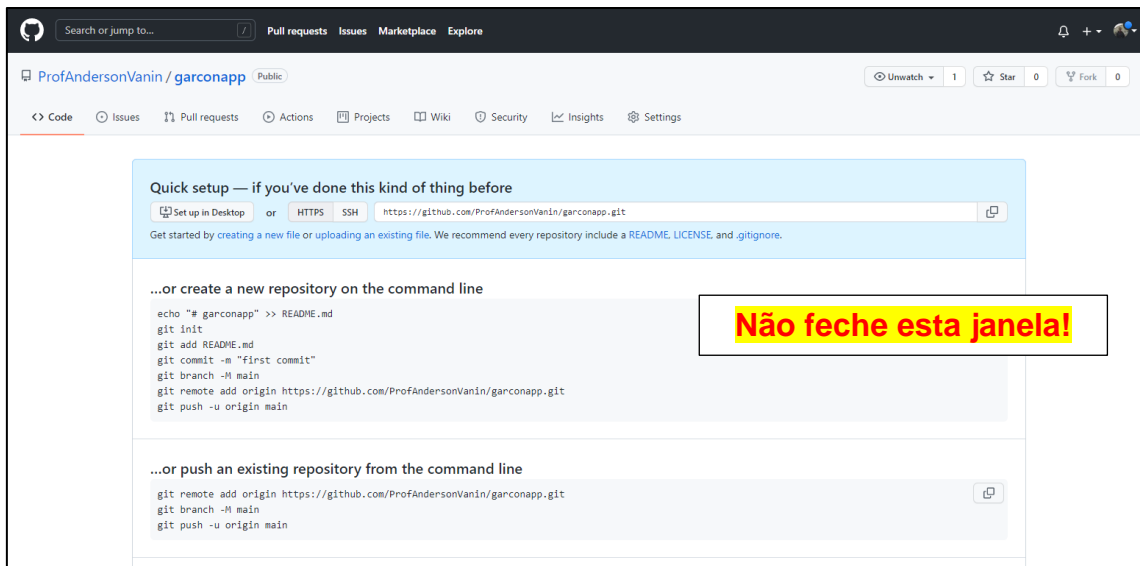
☐ Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

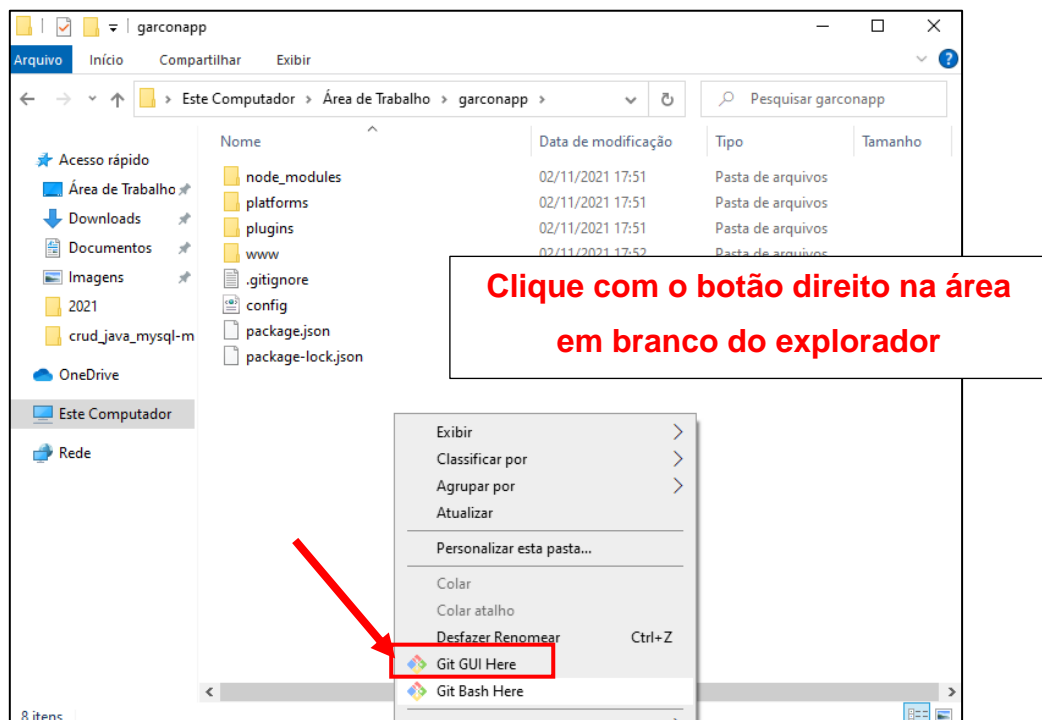
Create repository

Não marcar que est as opções

3) Após clicar em CREATE REPOSITORY, deixe a próxima janela aberta no navegador para utilizar nos as informações exibidas na linha de comando do git bash.



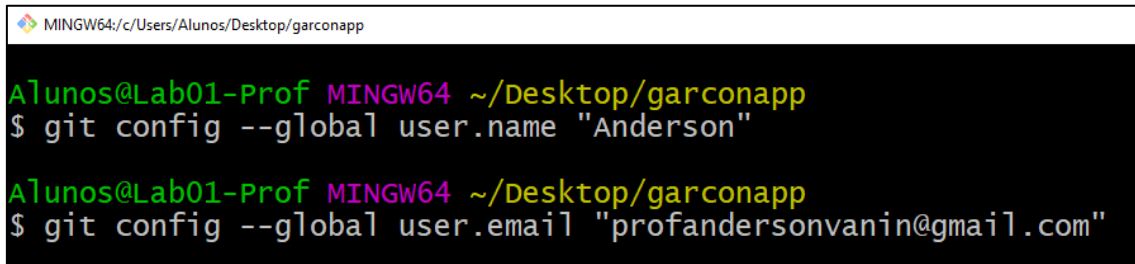
4) Abra a pasta de seu projeto no explorador de arquivos e clique com o botão direito na área vazia e selecione GIT BASH HERE.



5) Na janela de comandos digite os comandos:

```
git config --global user.name "Seu nome"
```

```
git config --global user.email "seu-email@example.com"
```

A terminal window with a black background and white text. The title bar at the top reads 'MINGW64; c:/Users/Alunos/Desktop/garconapp'. The prompt is 'Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp'. The first command entered is '\$ git config --global user.name "Anderson"', followed by a blank line. The second command entered is '\$ git config --global user.email "profandersonvanin@gmail.com"', followed by a blank line.

```
Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp
$ git config --global user.name "Anderson"

Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp
$ git config --global user.email "profandersonvanin@gmail.com"
```

6) Inicialize um repositório local vazão digitando o comando:

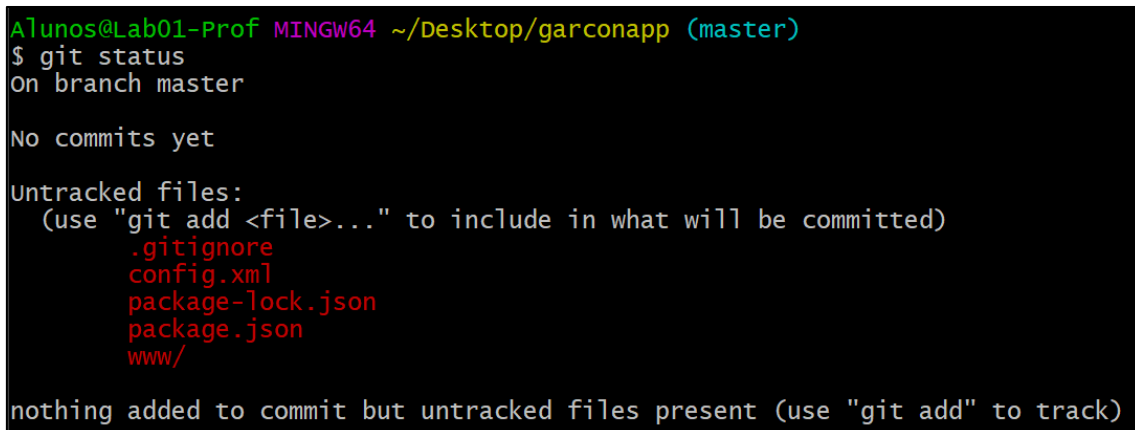
```
git init
```

A terminal window with a black background and white text. The title bar at the top reads 'MINGW64; c:/Users/Alunos/Desktop/garconapp'. The prompt is 'Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp'. The command entered is '\$ git init', followed by a blank line. The output is 'Initialized empty Git repository in c:/Users/Alunos/Desktop/garconapp/.git/'.

```
Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp
$ git init
Initialized empty Git repository in c:/Users/Alunos/Desktop/garconapp/.git/
```

7) Digite o comando:

```
git status
```

A terminal window with a black background and white text. The title bar at the top reads 'MINGW64; c:/Users/Alunos/Desktop/garconapp'. The prompt is 'Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp (master)'. The command entered is '\$ git status', followed by a blank line. The output is 'On branch master', a blank line, 'No commits yet', a blank line, 'Untracked files:', a blank line, '(use "git add <file>..." to include in what will be committed)', a blank line, and a list of untracked files: '.gitignore', 'config.xml', 'package-lock.json', 'package.json', and 'www/'. The final line of output is 'nothing added to commit but untracked files present (use "git add" to track)'.

```
Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    config.xml
    package-lock.json
    package.json
    www/

nothing added to commit but untracked files present (use "git add" to track)
```

8) Digite o comando:

```
git add .
```

```

Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp (master)
$ git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in config.xml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package-lock.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in www/css/estilo.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in www/css/materialize.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in www/css/materialize.min.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in www/font/material-design-icons/LICENSE.txt.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in www/font/material-design-icons/Material-Design-Icons.svg.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in www/icons/material.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in www/index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in www/js/jquery.min.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in www/js/materialize.js.

```

9) Digite o comando:

```
git commit -m "Primeira versão do projeto"
```

```

Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp (master)
$ git commit -m "Primeira versão do projeto garconapp"
[master (root-commit) c95600b] Primeira versão do projeto garconapp
35 files changed, 16718 insertions(+)
create mode 100644 .gitignore
create mode 100644 config.xml
create mode 100644 package-lock.json
create mode 100644 package.json
create mode 100644 www/css/estilo.css
create mode 100644 www/css/materialize.css
create mode 100644 www/css/materialize.min.css
create mode 100644 www/font/material-design-icons/LICENSE.txt
create mode 100644 www/font/material-design-icons/Material-Design-Icons.eot
create mode 100644 www/font/material-design-icons/Material-Design-Icons.svg
create mode 100644 www/font/material-design-icons/Material-Design-Icons.ttf
create mode 100644 www/font/material-design-icons/Material-Design-Icons.woff
create mode 100644 www/font/material-design-icons/Material-Design-Icons.woff2
create mode 100644 www/font/roboto/Roboto-Bold.ttf
create mode 100644 www/font/roboto/Roboto-Bold.woff
create mode 100644 www/font/roboto/Roboto-Bold.woff2
create mode 100644 www/font/roboto/Roboto-Light.ttf
create mode 100644 www/font/roboto/Roboto-Light.woff
create mode 100644 www/font/roboto/Roboto-Light.woff2
create mode 100644 www/font/roboto/Roboto-Medium.ttf

```

10) Agora na janela aberta do repositório criado no git hub, vamos utilizar a opção para enviar arquivos em um repositório criado.

...or push an existing repository from the command line

```
git remote add origin https://github.com/ProfAndersonVanin/garconapp.git  
git branch -M main  
git push -u origin main
```

**Utilize um comando de cada vez.  
Não faça os 3 ao mesmo tempo!**

11) Digite o comando:

```
git remote add origin
```

```
https://github.com/ProfAndersonVanin/garconapp.git
```

```
Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp (master)  
$ git remote add origin https://github.com/ProfAndersonVanin/garconapp.git
```

12) Digite o comando:

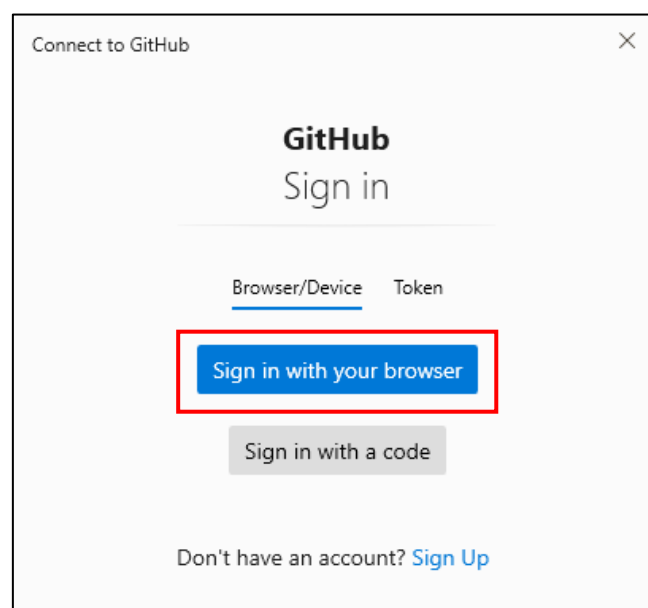
```
git branch -M main
```

```
Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp (master)  
$ git branch -M main
```

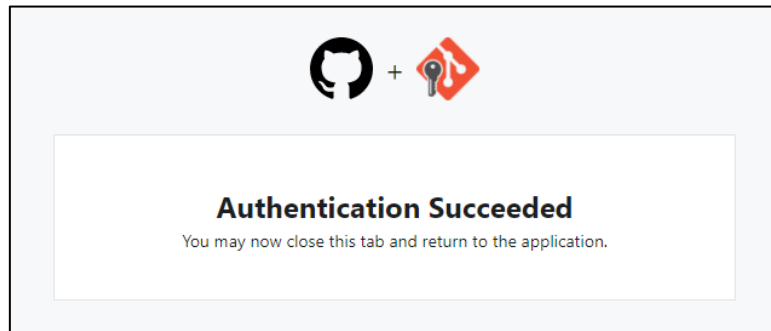
13) Digite o comando:

```
git push -u origin main
```

Ao digitar este comando, será exibida uma janela para confirmação de login. Escolha a opção para logar como browser.



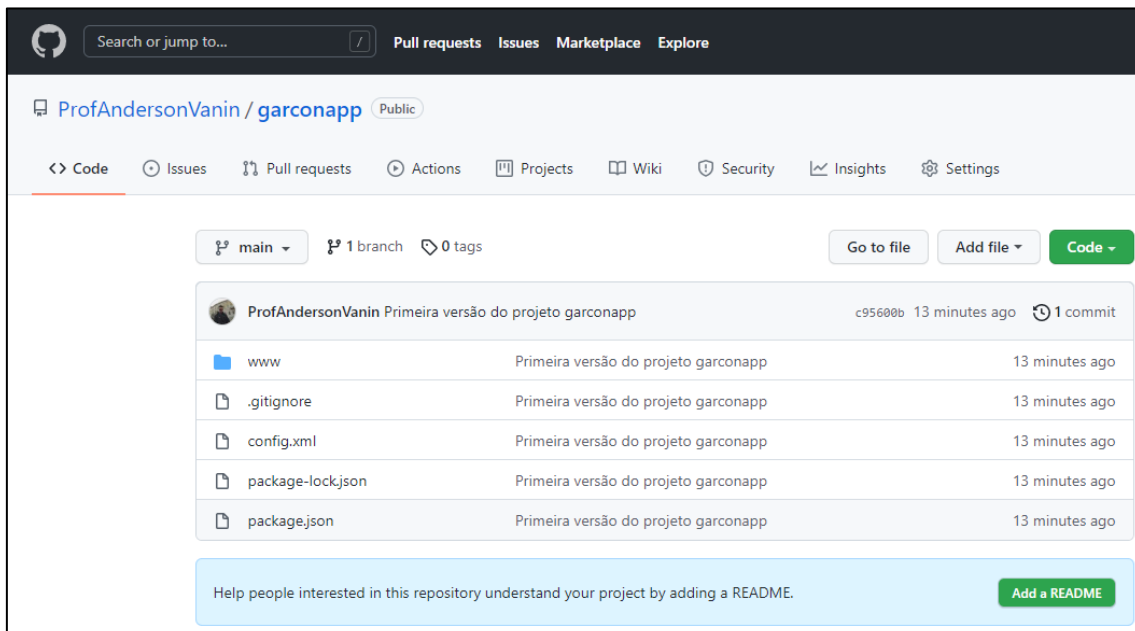
14) Verifique o status de login



15) Arquivos enviados para o repositório.

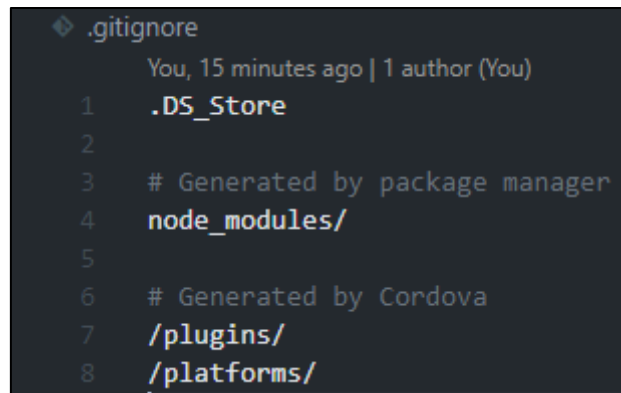
```
Alunos@Lab01-Prof MINGW64 ~/Desktop/garconapp (main)
$ git push -u origin main
Enumerating objects: 44, done.
Counting objects: 100% (44/44), done.
Delta compression using up to 4 threads
Compressing objects: 100% (44/44), done.
Writing objects: 100% (44/44), 1.31 MiB | 3.65 MiB/s, done.
Total 44 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ProfAndersonVanin/garconapp.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

16) Atualize a janela do navegador em seu repositório do GitHub.



**OBSERVAÇÃO**

Especificamente neste projeto do **garconapp**, na estrutura de pastas e arquivos, temos um arquivo chamado **.gitignore**. Este arquivo faz com que certas pastas ou arquivos nele listados, não sejam enviados ao git hub.

A screenshot of a code editor showing the content of a .gitignore file. The file is named ".gitignore" and was edited 15 minutes ago by the user. The content lists directories to be ignored: ".DS\_Store", "node\_modules/", "/plugins/", and "/platforms/". Comments indicate that "node\_modules/" was generated by a package manager and the others by Cordova.

```
.gitignore
You, 15 minutes ago | 1 author (You)
1  .DS_Store
2
3  # Generated by package manager
4  node_modules/
5
6  # Generated by Cordova
7  /plugins/
8  /platforms/
```

Neste caso as pastas: **node\_modules, plugins e platforms**, não serão enviadas.

Caso queira que estas pastas sejam enviadas também, altere este arquivo apagando estas informações antes do envio.