

# GestSio

---

## Documentation Technique GitHub et Git

Hassani Saïd Chazal

MANUEL Gérald

HIDJAZOU Attoumani

Etablissement : NELSON MANDELA Polyvalent

# GestSio

---

## Sommaire

---

1- <a href="#">Présentation de GitHub et Git</a> .....	P3-4
1.1- <a href="#">Définition GitHub</a> .....	P3
1.2- <a href="#">Définition contrôle de version</a> .....	P3
1.3- <a href="#">Définition Git</a> .....	P3-4
2- <a href="#">GitHub</a> .....	P5-9
2.1- <a href="#">Créer un compte GitHub</a> .....	P5-6
2.2- <a href="#">Créer votre propre dépôt</a> .....	P6-7
2.3- <a href="#">Créer un nouvel répertoire</a> .....	P8-9
3- <a href="#">Git</a> .....	P9-23
3.1- <a href="#">Installez Git sur votre ordinateur</a> .....	P9-13
3.2- <a href="#">Explication de git de l'envoi du projet à distance</a> .....	P14-23
4- <a href="#">Résultat de Git à distance</a> .....	P23-29

# GestSio

---

## 1- Présentation de GitHub et Git

---

### 1.1- Définition GitHub :

GitHub est une plateforme de développement inspirée par votre manière de travailler. De l'open source à l'entreprise, vous pouvez héberger et revoir du code, gérer des projets et construire des logiciels aux côtés de millions d'autres développeurs.

A un haut niveau, GitHub est un site web et un service de cloud qui aide les développeurs à stocker et à gérer leur code, ainsi qu'à suivre et contrôler les modifications qui lui sont apportées. Pour comprendre exactement ce qu'est GitHub, vous devez connaître deux principes liés :

- ❖ Contrôle de version
- ❖ Git

### 1.2- Définition contrôle version :

Les systèmes de contrôle de version sont une catégorie d'outils de développement qui aident une équipe de développement à gérer les changements apportés au code source au fil du temps. Les logiciels de contrôle de version gardent une trace de chaque changement apporté au code dans un type spécial de base de données. Si une erreur est commise, les développeurs peuvent revenir en arrière et comparer les versions antérieures du code, ce qui leur permet de corriger l'erreur tout en minimisant les perturbations pour tous les membres de l'équipe.

### 1.3- Définition Git :

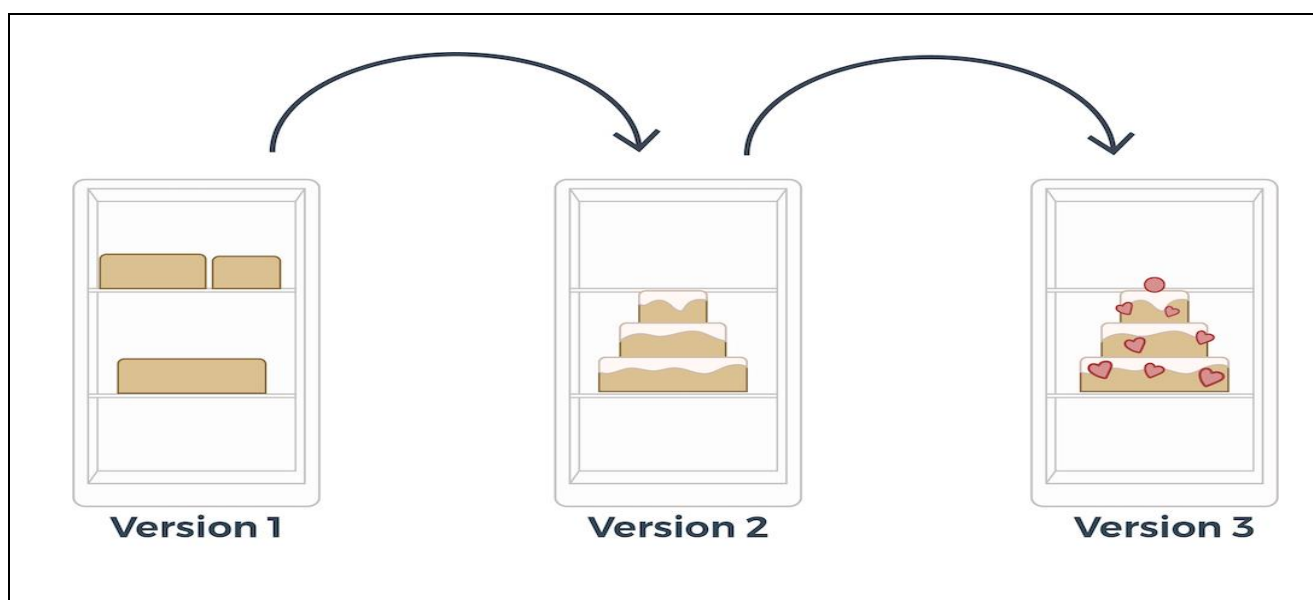
Git est de loin le système de contrôle de version le plus largement utilisé aujourd'hui. Git est un projet open source avancé, qui est activement maintenu.

Concrètement, Git est un système de contrôle de version distribué, ce qui signifie que l'ensemble de la base du code et de l'historique est disponible sur l'ordinateur de chaque développeur, ce qui permet des branchements et une fusion faciles.

# GestSio

---

Par exemple: Prenons l'image de la réalisation d'un gâteau, on va dire que le réfrigérateur est le dépôt local. C'est l'endroit où l'on va stocker nos préparations au fur et à mesure. Dans un premier temps, on réalise la pâte, on la stocke au réfrigérateur, puis on réalise la crème, on l'assemble avec la pâte et on stocke l'ensemble au réfrigérateur. Finalement, on réalise la décoration du gâteau, on finalise notre gâteau en y ajoutant les décorations et on le remet au réfrigérateur.



Voilà le fonctionnement de notre dépôt local ! On réalise une version, que l'on va petit à petit venir améliorer en stockant toutes ces versions.

Le dépôt distant est un peu différent. Il permet de stocker certaines versions qu'on lui aura envoyées, afin de garder un historique délocalisé. Ben oui, imaginez que votre PC rende l'âme demain, vous aurez toujours vos super programmes sur GitHub. Mais en plus de les stocker, vous pouvez aussi les rendre publics, et chacun pourra alors venir y ajouter ses évolutions.

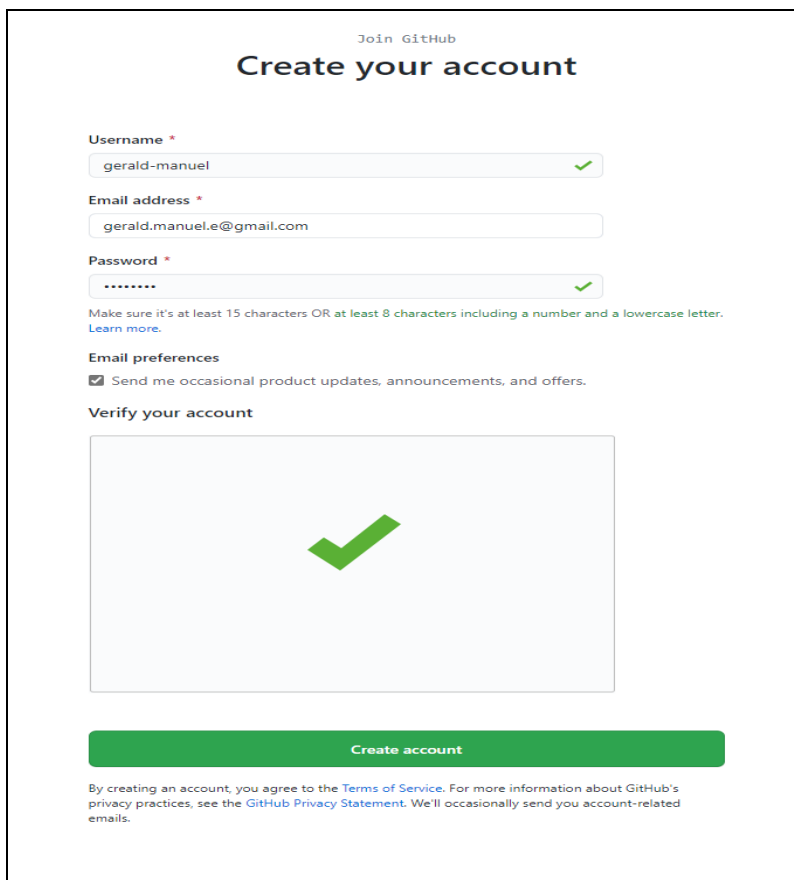
# GestSio

## 2- GitHub

### 2.1- Créer un compte GitHub :

Comme nous l'avons vu dans le chapitre précédent, GitHub est un service en ligne permettant d'héberger ses dépôts distants.

Pour créer votre compte GitHub, rendez-vous sur la page d'accueil, cliquez sur [Sign up \(S'inscrire\)](#). On vous demandera alors de renseigner un nom d'utilisateur, un email, un mot de passe et un petit verification avant **Create account (Créer un compte)**.



Join GitHub

### Create your account

Username \*  
gerald-manuel ✓

Email address \*  
gerald.manuel.e@gmail.com

Password \*  
..... ✓

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.  
[Learn more.](#)

Email preferences  
☒ Send me occasional product updates, announcements, and offers.

Verify your account

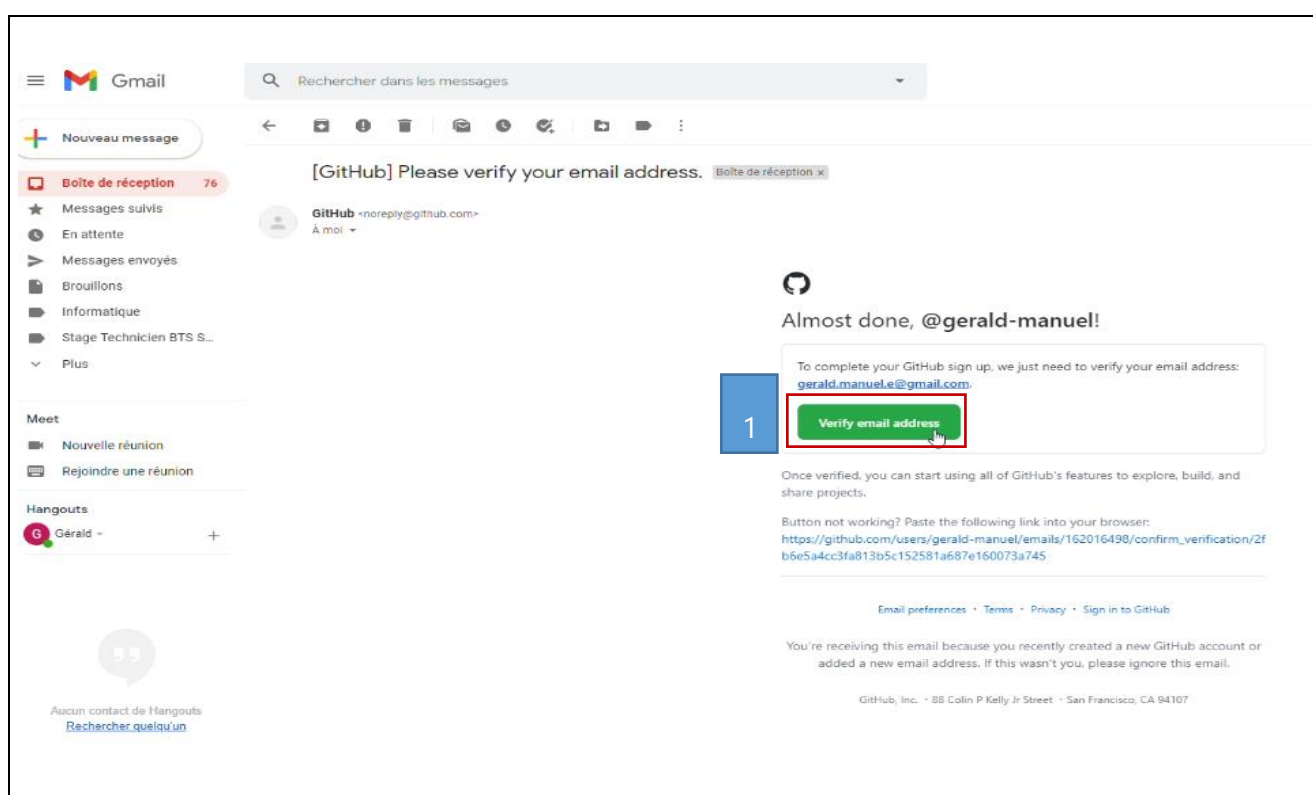
✓

Create account

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

# GestSio

Qu'on vous avait fini allez sur votre email, vous avez reçu un email de confirmation et cliquez dessus pour bien confirmer que ce votre email.

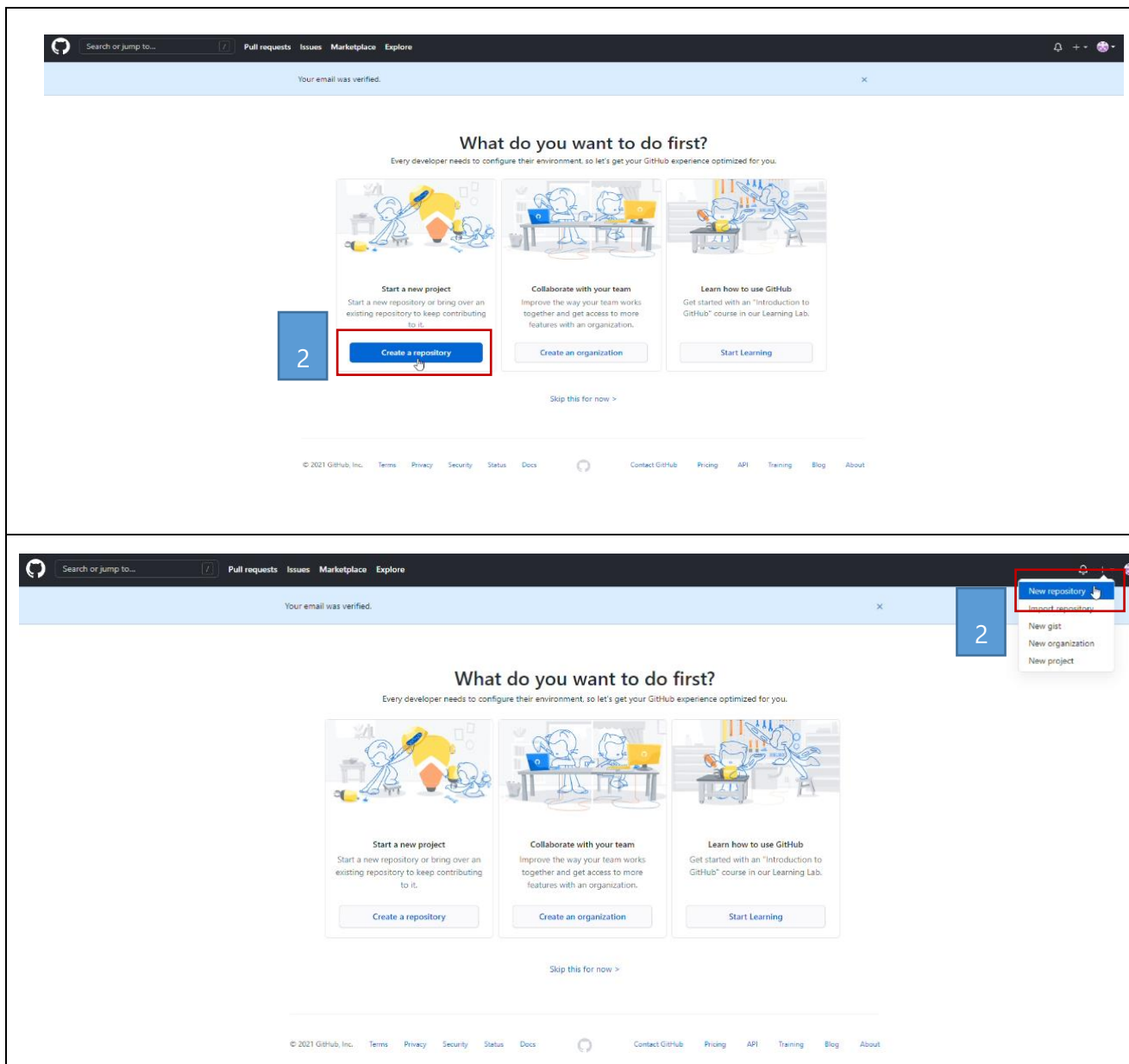


## 2.2- Créer votre propre dépôt :

Pour mettre votre projet sur GitHub, vous devez créer un référentiel dans lequel il pourra être installé.

Comme vos clics sur **Verify email address (Vérifier l'adresse e-mail)** soit il vous montre un page crée votre **New repository (dépôt)** ou Cliquez sur le "+" dans le coin supérieur droit, pour faire apparaître l'option **New repository (dépôt)**.

# GestSio



The image displays two screenshots of the GitHub homepage, illustrating the initial setup process for a new user. Both screenshots show the 'What do you want to do first?' section, which offers three main options: 'Start a new project', 'Collaborate with your team', and 'Learn how to use GitHub'.

**First Screenshot:** A blue box with the number '2' highlights the 'Create a repository' button under the 'Start a new project' section. The button is highlighted with a red rectangle.

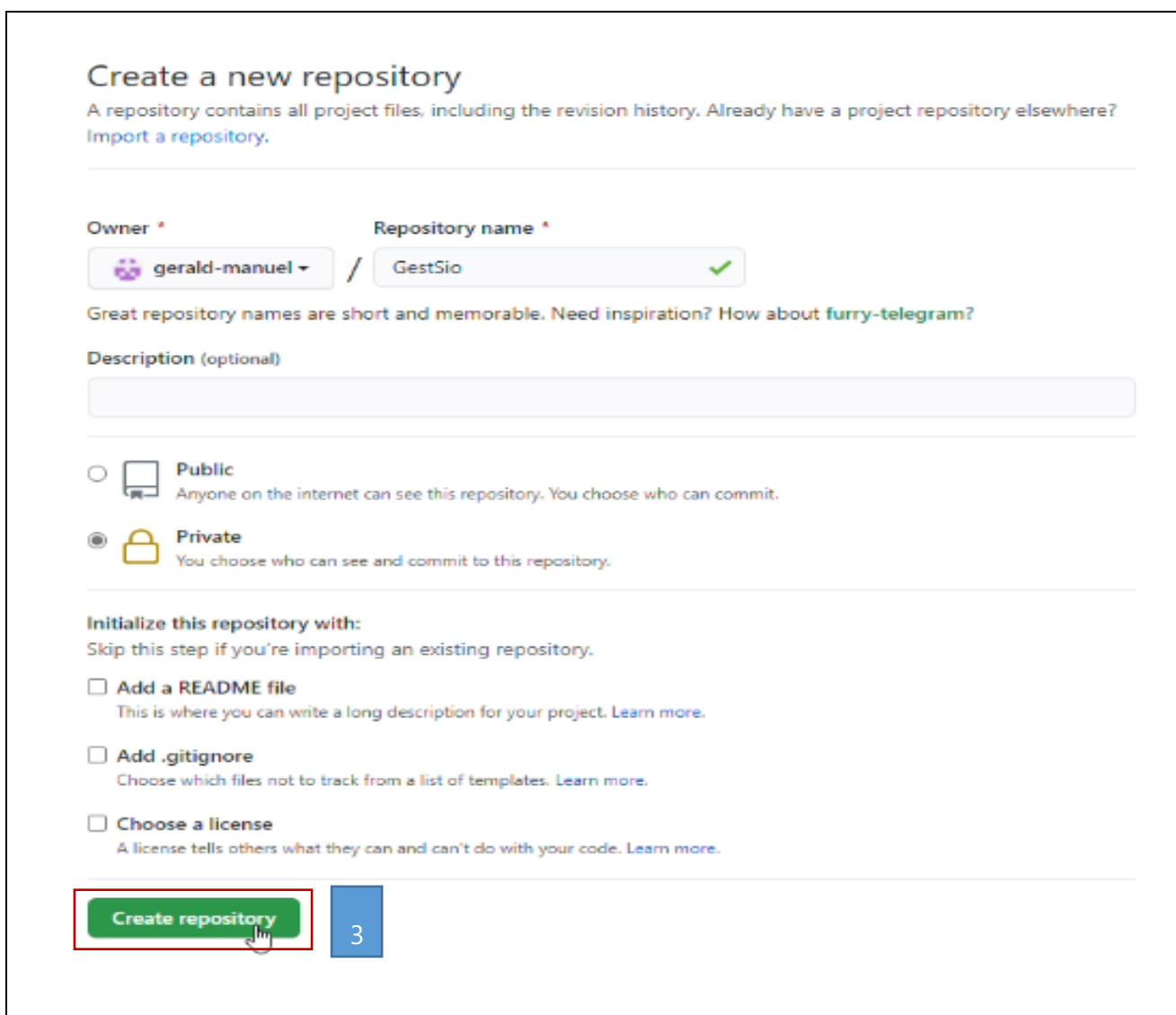
**Second Screenshot:** A blue box with the number '2' highlights the user profile dropdown menu in the top right corner. The 'New repository' option is highlighted with a red rectangle.

The GitHub homepage includes a search bar, navigation links (Pull requests, Issues, Marketplace, Explore), a notification bar ('Your email was verified.'), and a footer with copyright information and various links.

# GestSio

## 2.3- Créer un nouvel répertoire :



Choisissez un nom simple pour votre dépôt, choisissez si vous souhaitez créer un dépôt public ou privé et le reste à votre choix.



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 \*

 gerald-manuel / GestSio 

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

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.

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.](#)

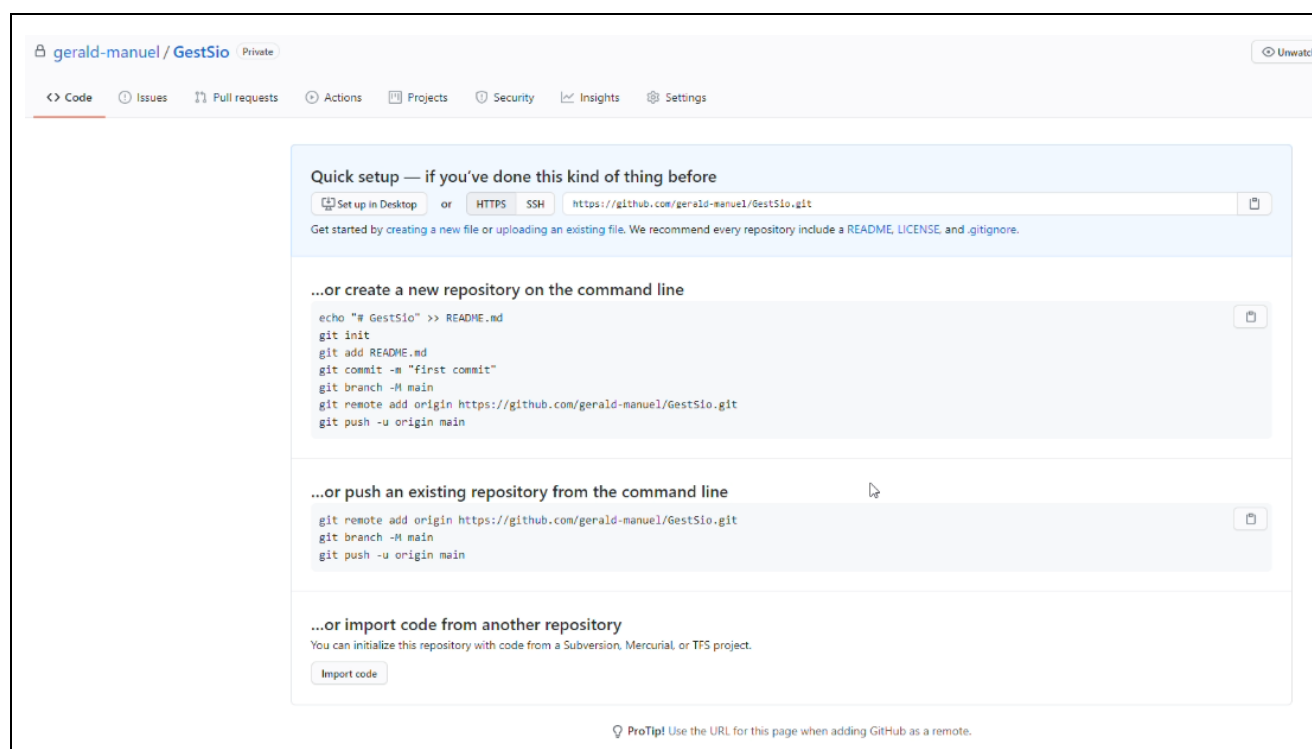
Create repository 3

Cliquez sur Create repository pour créer un nouveau répertoire qui nomme GestSio.



# GestSio

Nouvelle **repository (répertoire)** apparaître :



## 3- Git

### 3.1- Installez Git sur votre ordinateur :

Par la suite, en installe Git.

Afin d'installer Git, il faut dans un premier temps le télécharger. Vous pourrez trouver le [le lien de téléchargement ici](#). Choisissez la version qui correspond à votre installation et téléchargez-la.


1

Et comme vous avez fini d'**ouvrir le fichier** et cliquez sur **Oui**.

2

3

# GestSio



**git** --local-branching-on-the-cheap

Search entire site...

About

Documentation

**Downloads**




GUI Clients

Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on [Amazon.com](#).

## Downloads

 **macOS**
 **Windows** 1
  **Linux/Unix**

Older releases are available and the Git source repository is on [GitHub](#).

### GUI Clients


Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)


### Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)



Latest source Release  
**2.31.1**  
[Release Notes \(2021-03-26\)](#)  
[Download 2.31.1 for Windows](#)



**git** --distributed-even-if-your-workflow-isnt

Search entire site...

About

Documentation

**Downloads**


GUI Clients

Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on [Amazon.com](#).

## Downloading Git



**Your download is starting...**

You are downloading the latest (**2.31.1**) 64-bit version of Git for Windows. This is the most recent [maintained build](#). It was released **about 2 months ago**, on 2021-03-27.

[Click here to download manually](#), if your download hasn't started.

**Other Git for Windows downloads**

[Git for Windows Setup](#)

[32-bit Git for Windows Setup.](#)

[64-bit Git for Windows Setup.](#)


[Git for Windows Portable \("thumbdrive edition"\)](#)

[32-bit Git for Windows Portable.](#)

[64-bit Git for Windows Portable.](#)

The current source code release is version 2.31.1. If you want the newer version, you can build it from the [source code](#).

Téléchargements

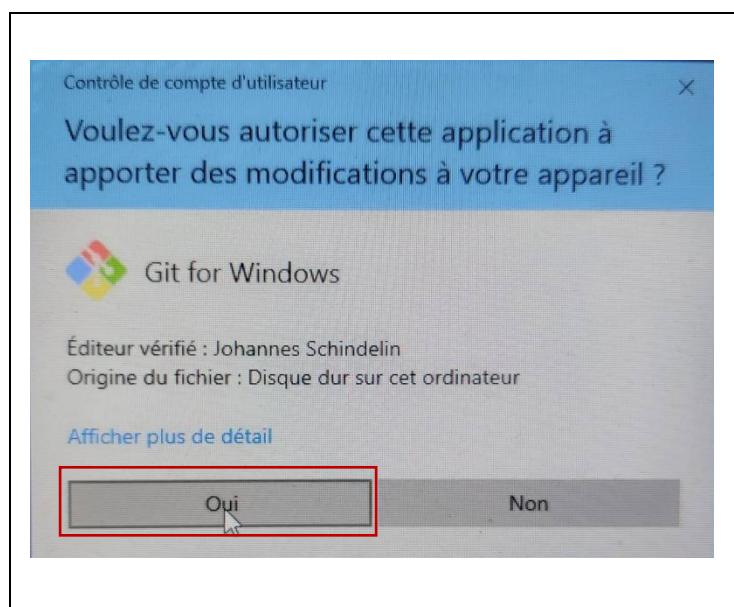


Git-2.31.1-64-bit.exe

Ouvrir un fichier

[Afficher plus](#)

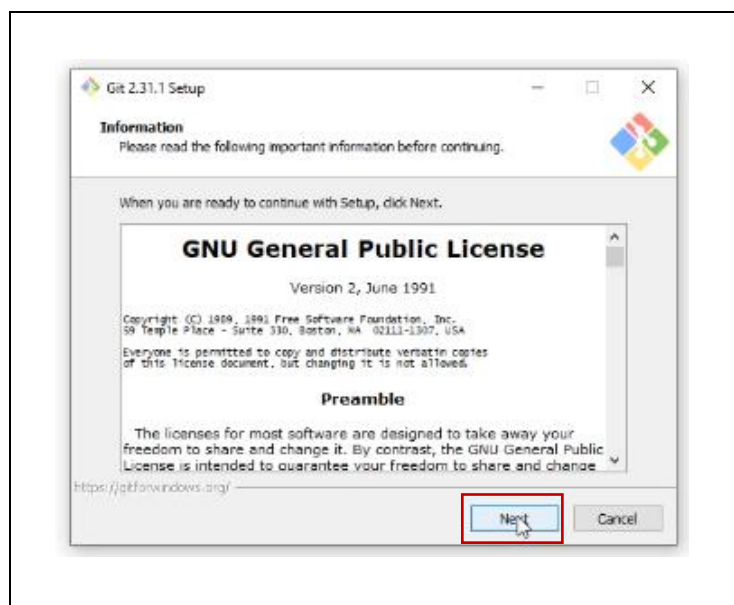
# GestSio



3

Appuyez sur **Next (Suivant).**

4

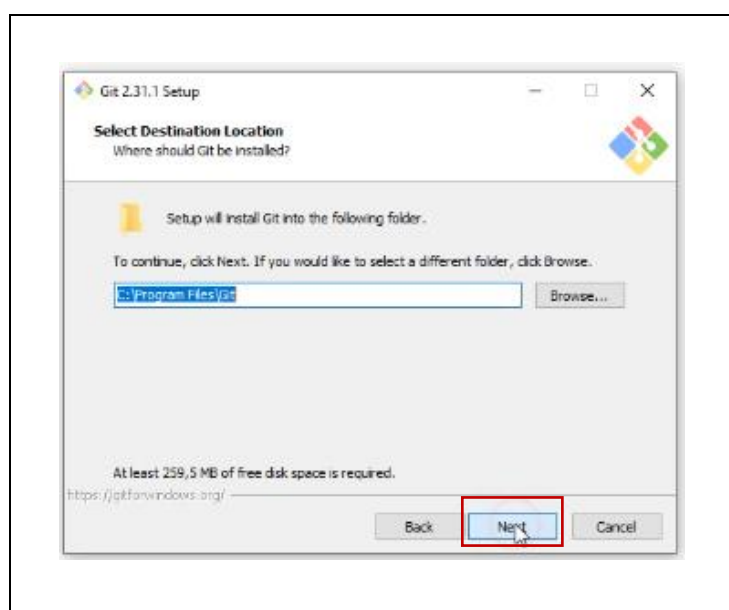


4

# GestSio

Choisissez votre l'emplacement du dossier de git et cliquez sur **Next (Suivant)**

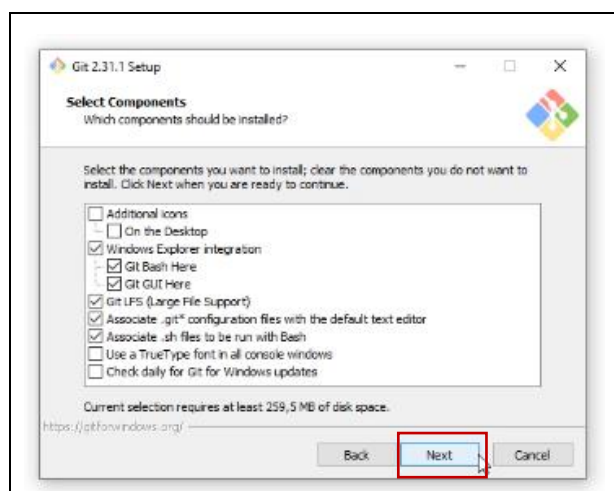
5



5

6

Appuyez sur **Next (Suivant)** à chaque fenêtre puis sur Installer. Lors de l'installation, laissez toutes les options par défaut, elles conviennent bien.

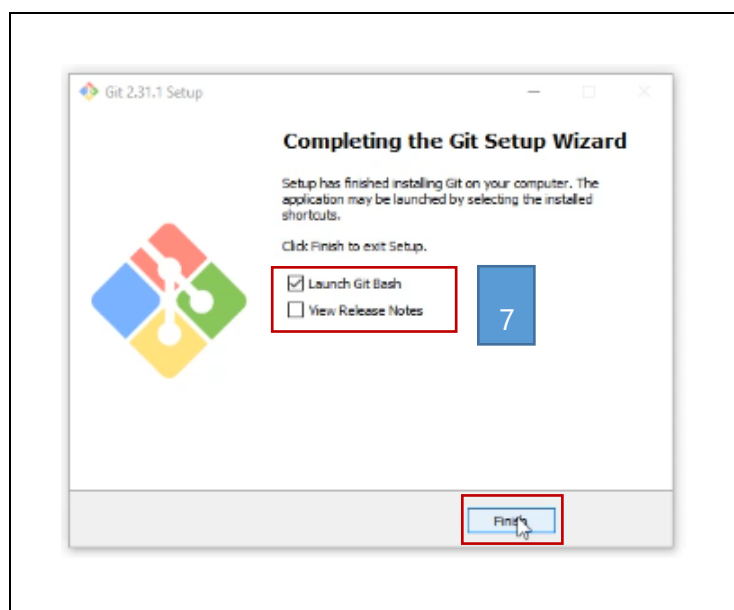


6

# GestSio

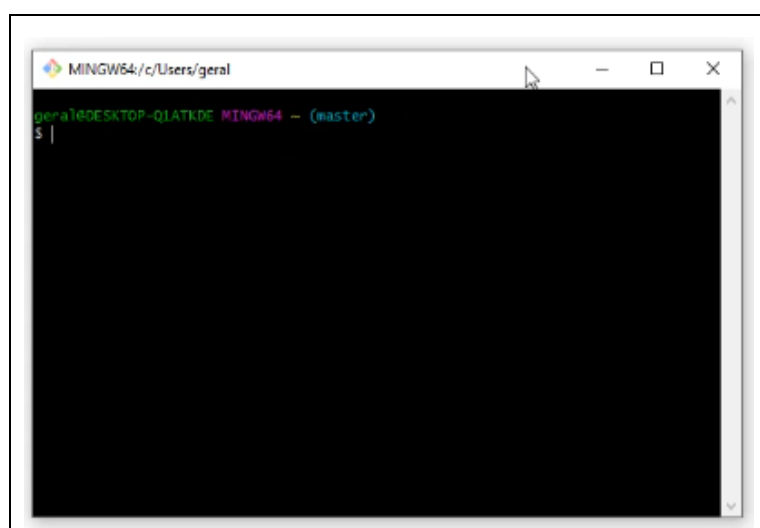
Décoche la case **View Release Notes**, cochez ensuite la case **Launch Git Bash** et cliquez **Finish**.

7



7

**Git Bash** se lance et s'ouvre cette affiche :



# GestSio

---

## 3.2- Explication de git de l'envoi du projet à distance :

On va d'abord sur le disque **d** dans répertoire du **Desktop (Bureau)** qui se trouve mon dossier du projet. En tapant cette commande :

```
$ cd /d/Utilisateur\ geral/Desktop/
```

Tout sa dépend de racine que vous allez installer votre git.

Puis, on va configurer son identité. Pour cela, nous allons entrer dans le monde des lignes de commande !

Nous allons commencer par renseigner votre nom et votre adresse e-mail. C'est une information importante car toutes les validations dans Git utilisent cette information et elle est indélébile dans toutes les validations que vous pourrez réaliser :

```
$ git config --global user.name "gerald-manuel"  
$ git config --global user.email "gerald.manuel.e@gmail.com"
```

Grâce à l'option `--global`, vous n'aurez besoin de le faire qu'une fois.

Si vous souhaitez par contre, pour **un projet spécifique**, changer votre nom d'utilisateur, vous devrez repasser cette ligne mais sans le `--global`.

# GestSio

---

Afin de vérifier que vos paramètres aient bien été pris en compte, et vérifier les autres paramètres, il suffit de passer la commande **git config --list**.

```
geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
core.editor=nano.exe
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=gerald.manuel.e@gmail.com
user.name=gerald-manuel
core.editor=subl
merge.tool=filemerge
```

Il est recommandé d'activer les couleurs afin d'améliorer la lisibilité des différentes branches. Pour cela, passez ces trois lignes dans Git Bash :

```
$ git config --global color.diff auto
$ git config --global color.status auto
$ git config --global color.branch auto
```

Par défaut, Git utilisera Vim comme éditeur et Vimdiff comme outil de merge. Vous pouvez les modifier en utilisant :

```
$ git config --global core.editor notepad++
$ git config --global merge.tool vimdiff
```

# GestSio

Maintenant que nous avons paramétré les paramètres de base, nous allons créer notre fameux dépôt local. Pour ce faire, deux solutions possibles :

- ❖ Créer un dépôt local vide pour accueillir un nouveau projet
- ❖ Cloner un dépôt distant, c'est-à-dire rapatrier tout l'historique d'un dépôt distant en local, afin de pouvoir travailler par dessus.

Dans un logiciel de gestion de versions comme Git, un dépôt représente une copie du projet.

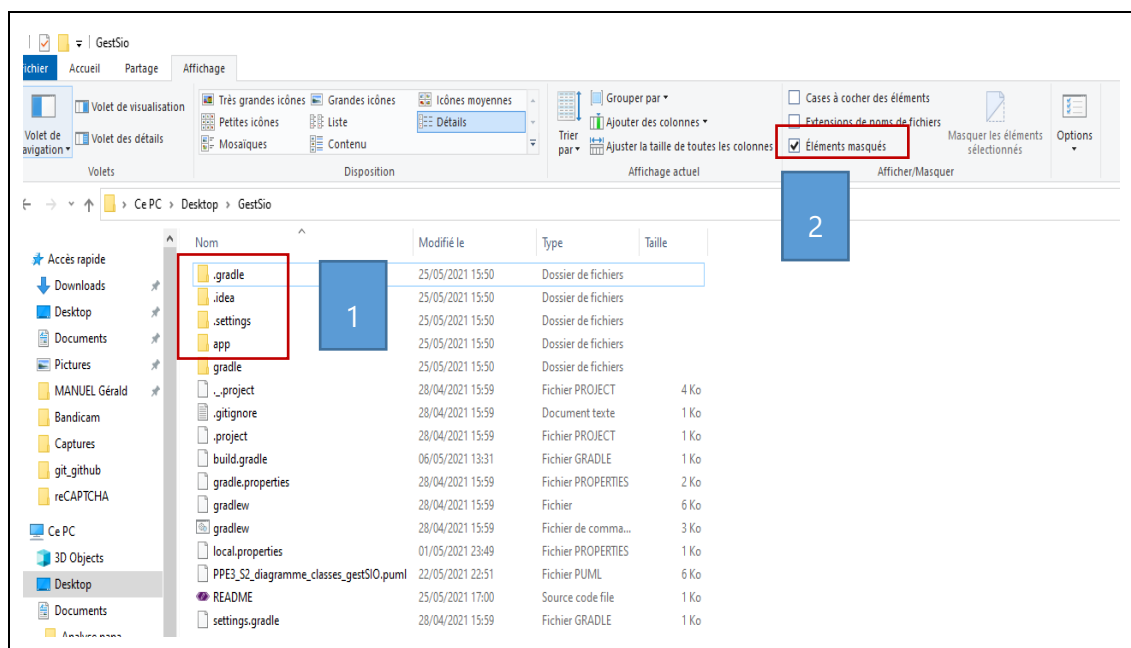
Chaque ordinateur d'un développeur qui travaille sur le projet **possède donc une copie du dépôt**.

Dans chaque dépôt, on trouve les fichiers du projet ainsi que leur historique.

Accédez à votre dossier, et lancez la ligne ci-dessous dans Git Bash en ciblant ce dossier :

```
$ cd GestSio/
```

Premier affiche :



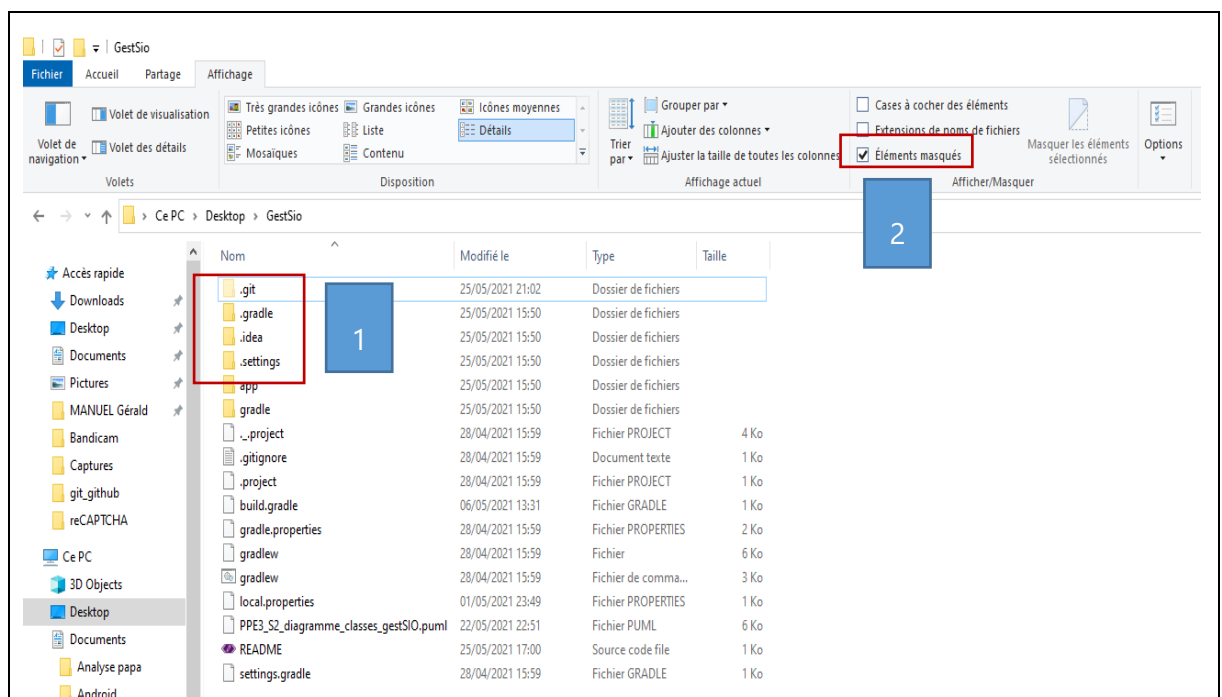


# GestSio

Créer un nouveau dépôt local (dans le dossier courant), il suffit d'y faire un **git init** avec l'option **--bare**.  
Initialiser **git init** et **--bare** pour initialiser à nouveau.

\$ git init --bare //À exécuter sur le serveur.

Initialized empty Git repository in D:/Utilisateur geral/Desktop/GestSio/.git/



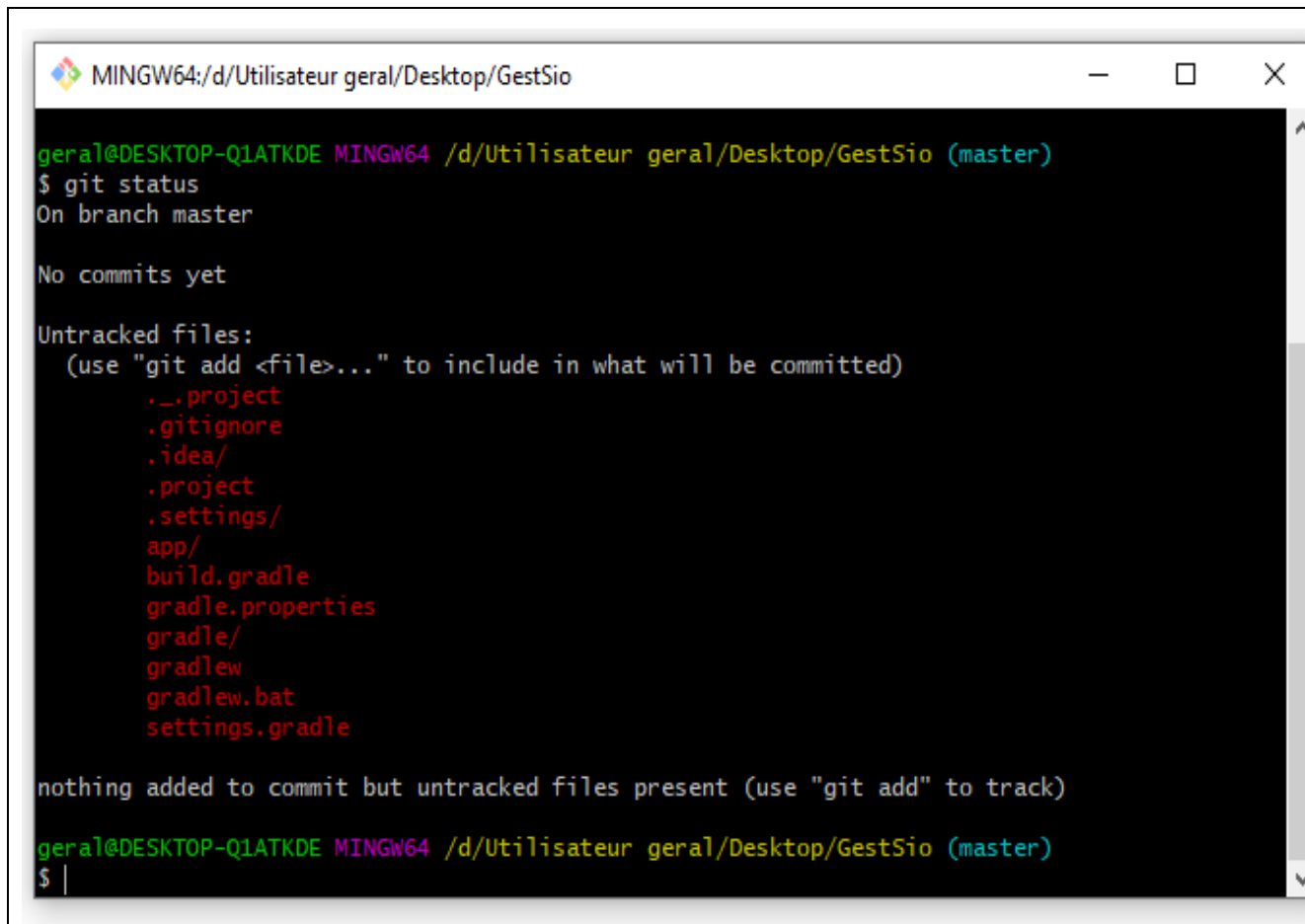
Vous avez bien vu dans **cadre couleur rouge** que git a été bien initialiser.

Pour afficher, vous pouvez l'afficher en allant dans **Affichage>Éléments masqués**.



# GestSio

Par la suite, on va vérifier l'état des **modification locales** en tapant **git status** :



```
MINGW64:/d/Utilisateur geral/Desktop/GestSio
geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ._.project
    .gitignore
    .idea/
    .project
    .settings/
    app/
    build.gradle
    gradle.properties
    gradle/
    gradlew
    gradlew.bat
    settings.gradle

nothing added to commit but untracked files present (use "git add" to track)
geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ |
```

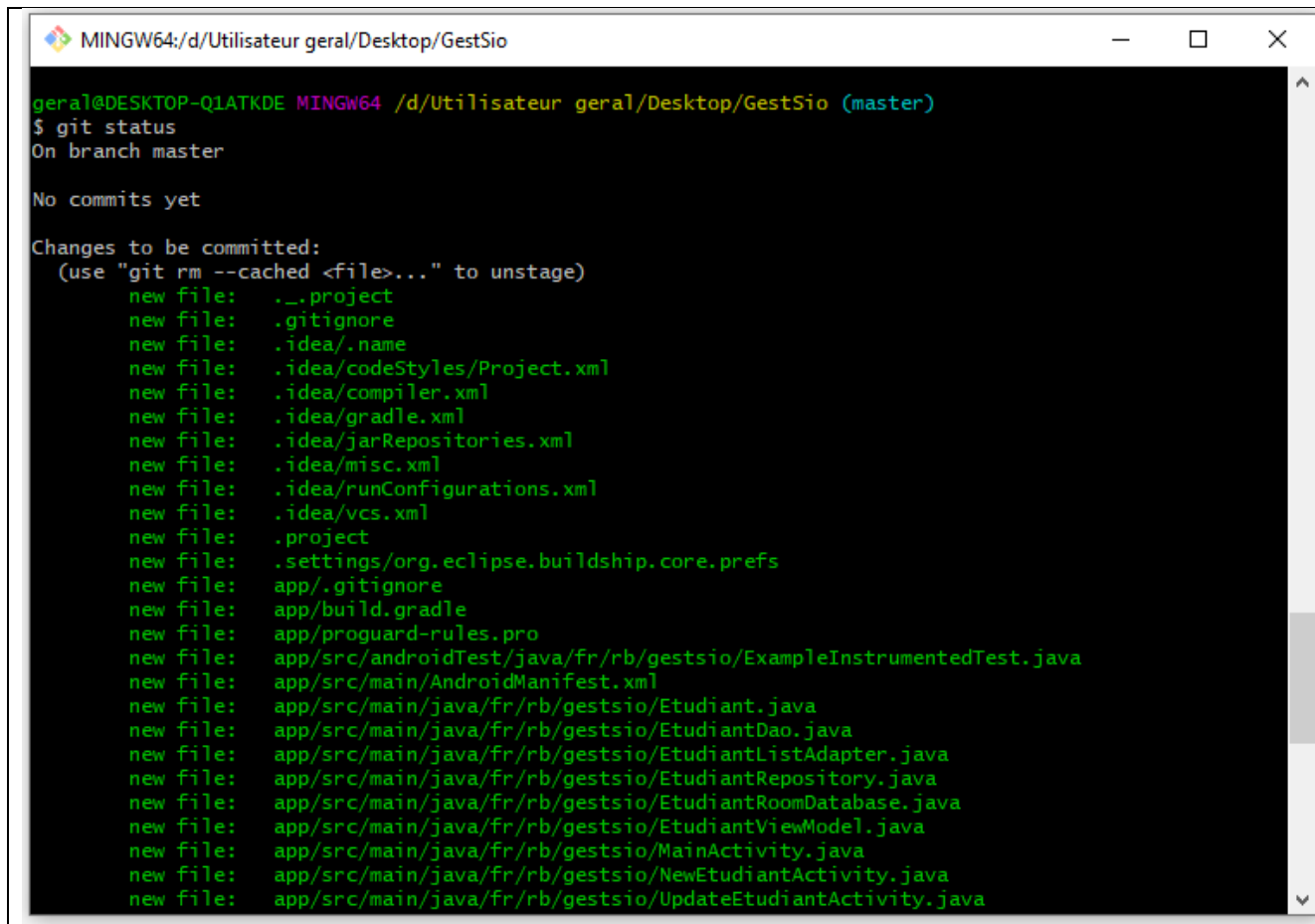
Donc, on voit que l'état n' a pas encore ajouté dans dépôt

Pour ajouter les modifications à l'index de Git en tape **git add** avec l'option **--all**. Ajouter les modifications **git add** et **--all** tous les répertoire.

```
$ git add --all
```

# GestSio

Ré-vérifier encore en tapant **git status**.



```
MINGW64:/d/Utilisateur geral/Desktop/GestSio

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   ./.project
    new file:   .gitignore
    new file:   .idea/.name
    new file:   .idea/codeStyles/Project.xml
    new file:   .idea/compiler.xml
    new file:   .idea/gradle.xml
    new file:   .idea/jarRepositories.xml
    new file:   .idea/misc.xml
    new file:   .idea/runConfigurations.xml
    new file:   .idea/vcs.xml
    new file:   .project
    new file:   .settings/org.eclipse.buildship.core.prefs
    new file:   app/.gitignore
    new file:   app/build.gradle
    new file:   app/proguard-rules.pro
    new file:   app/src/androidTest/java/fr/rb/gestsio/ExampleInstrumentedTest.java
    new file:   app/src/main/AndroidManifest.xml
    new file:   app/src/main/java/fr/rb/gestsio/Etudiant.java
    new file:   app/src/main/java/fr/rb/gestsio/EtudiantDao.java
    new file:   app/src/main/java/fr/rb/gestsio/EtudiantListAdapter.java
    new file:   app/src/main/java/fr/rb/gestsio/EtudiantRepository.java
    new file:   app/src/main/java/fr/rb/gestsio/EtudiantRoomDatabase.java
    new file:   app/src/main/java/fr/rb/gestsio/EtudiantViewModel.java
    new file:   app/src/main/java/fr/rb/gestsio/MainActivity.java
    new file:   app/src/main/java/fr/rb/gestsio/NewEtudiantActivity.java
    new file:   app/src/main/java/fr/rb/gestsio/UpdateEtudiantActivity.java
```

Donc, on voit que l'état n' a pas encore enregistré (c'est-à-dire faire un commit) dans dépôt

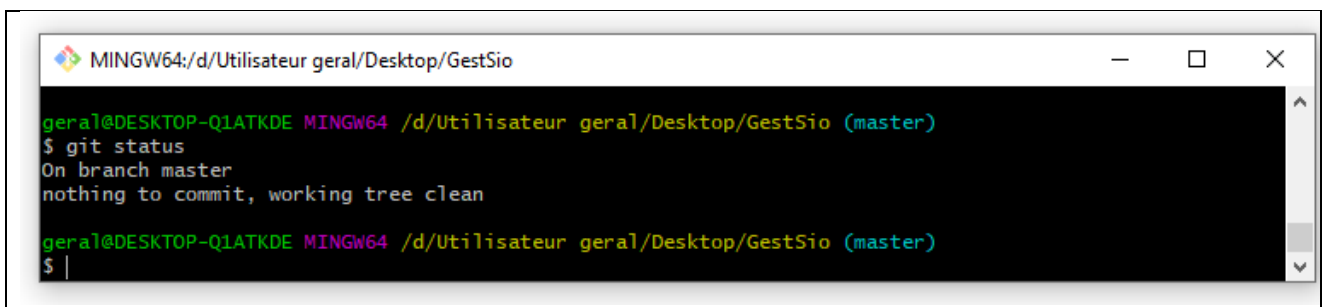
Pour faire un commit on tape **git commit** avec l'option **-am**. Faire un commit **git commit, -a** commiter tous les changements locaux des fichiers suivis, **-m** commiter les modifications en attente et il est possible de combiner les deux commandes : **-am**.

```
$ git commit -am "First commit"
```

# GestSio

---

Ré-vérifier une dernière fois l'état en tapant **git status**.



```
MINGW64:/d/Utilisateur geral/Desktop/GestSio
geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git status
On branch master
nothing to commit, working tree clean

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ |
```

Donc, on voit que l'état est dans le dépôt

On peut voir tous les commits en tapant **git log** :



```
MINGW64:/d/Utilisateur geral/Desktop/GestSio
geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git log
commit 3334f2e689810a337bddcdc712fd2e2943026867 (HEAD -> master)
Author: gerald-manuel <gerald.manuel.e@gmail.com>
Date: Tue May 25 22:24:37 2021 +0400

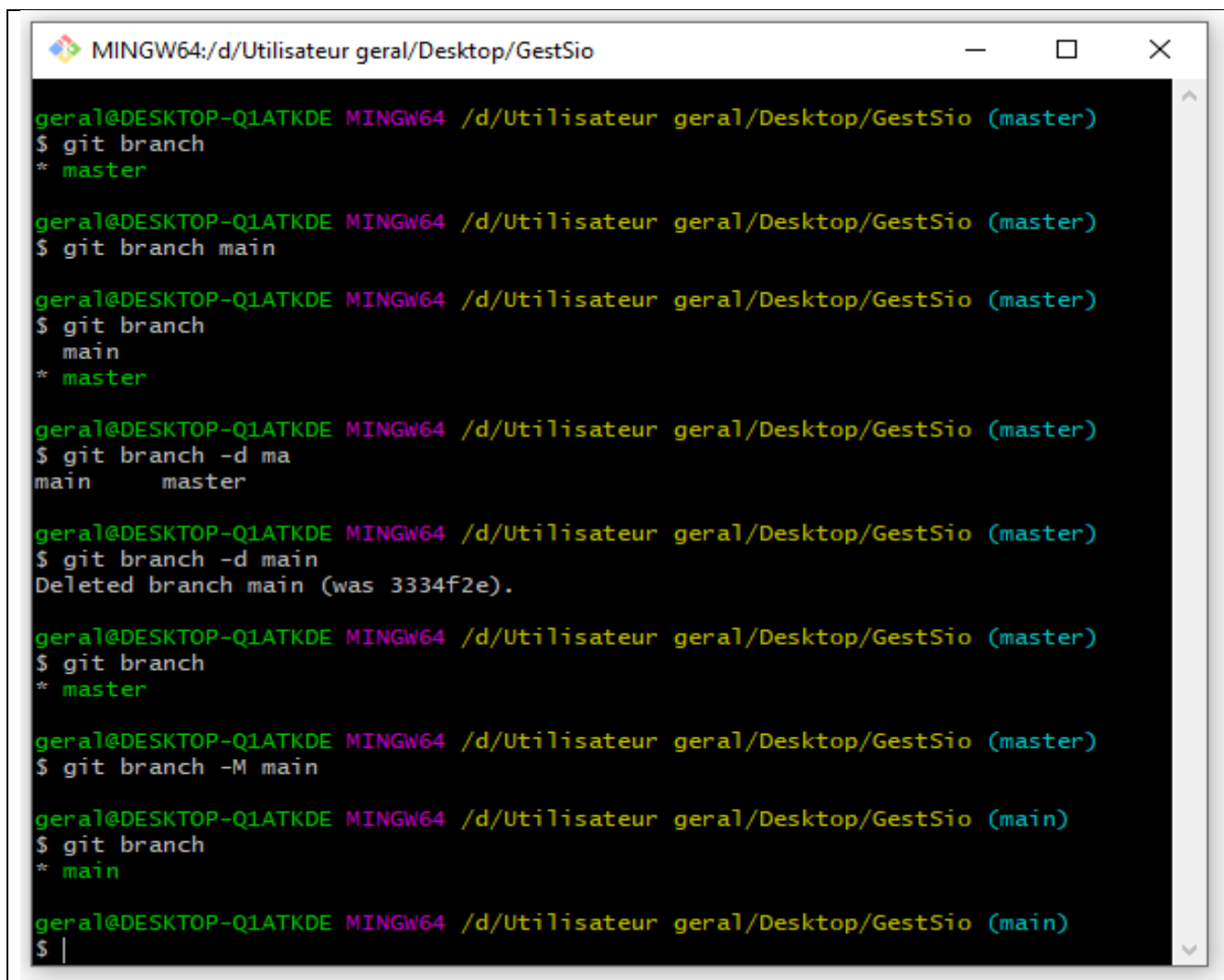
    first commit

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ |
```

# GestSio

On peut faire un **git branch** pour ajouter, supprimer et remplacer de branche :

```
$ git branch // voir le branch
$ git branch main //ajouter le branch
$ git branch -d //supprimer le branch
$ git branch -M main //remplacer le branch
```



```
MINGW64:/d/Utilisateur geral/Desktop/GestSio
geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git branch
* master

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git branch main

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git branch
  main
* master

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git branch -d ma
main      master

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git branch -d main
Deleted branch main (was 3334f2e).

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git branch
* master

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git branch -M main

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (main)
$ git branch
* main

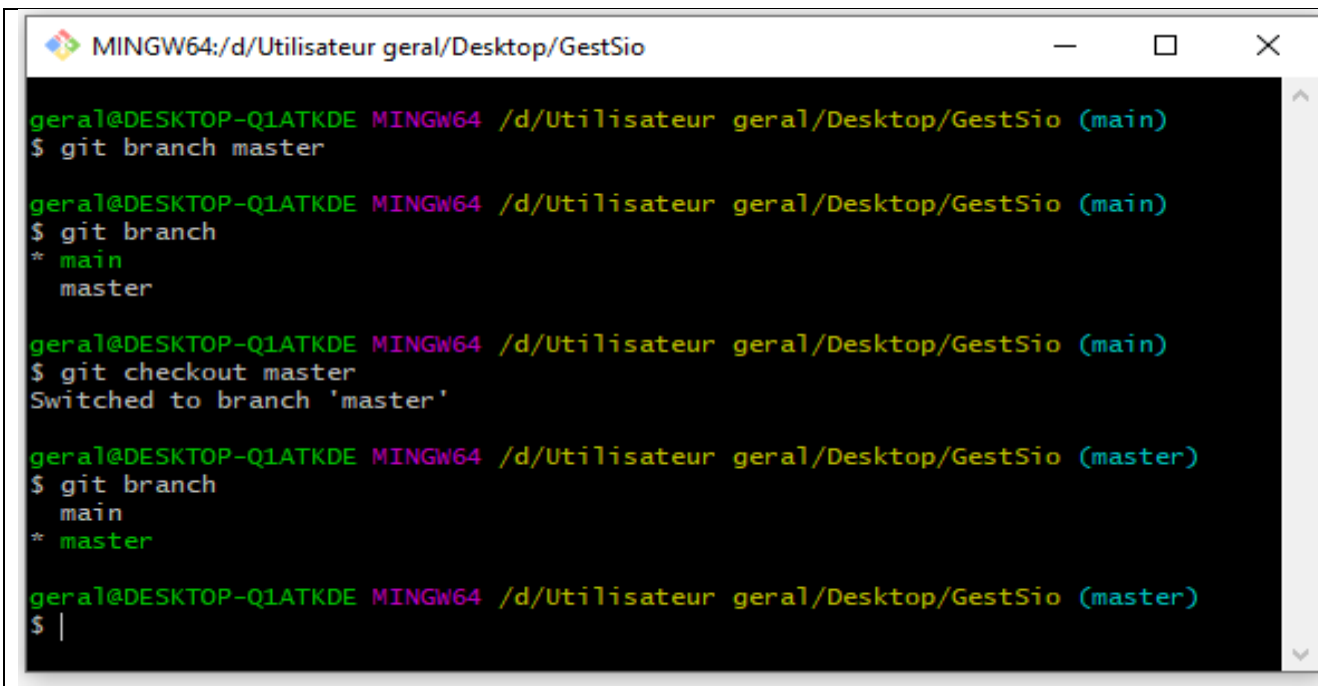
geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (main)
$ |
```

# GestSio

On peut aussi changer et fusionner le branch.

```
$ git checkout master //changer de branch
```

```
$ git merge <votre-branch> //fusionner la branche avec ne autre branch
```



```
MINGW64:/d/Utilisateur geral/Desktop/GestSio

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (main)
$ git branch master

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (main)
$ git branch
* main
  master

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (main)
$ git checkout master
Switched to branch 'master'

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ git branch
  main
* master

geral@DESKTOP-Q1ATKDE MINGW64 /d/Utilisateur geral/Desktop/GestSio (master)
$ |
```

Par la suite, on remplace le branch en tapant :

```
$ git branch -M main //remplacer le branch
```

Ensuite, on ajoute un nouveau dépôt distant, nommé origin :

```
$ git remote add <remote> <url> //exemple
```

```
$ git remote add origin https://github.com/gerald-manuel/GestSio.git //ajoute un nouveau dépôt distant, nommé origin
```

# GestSio

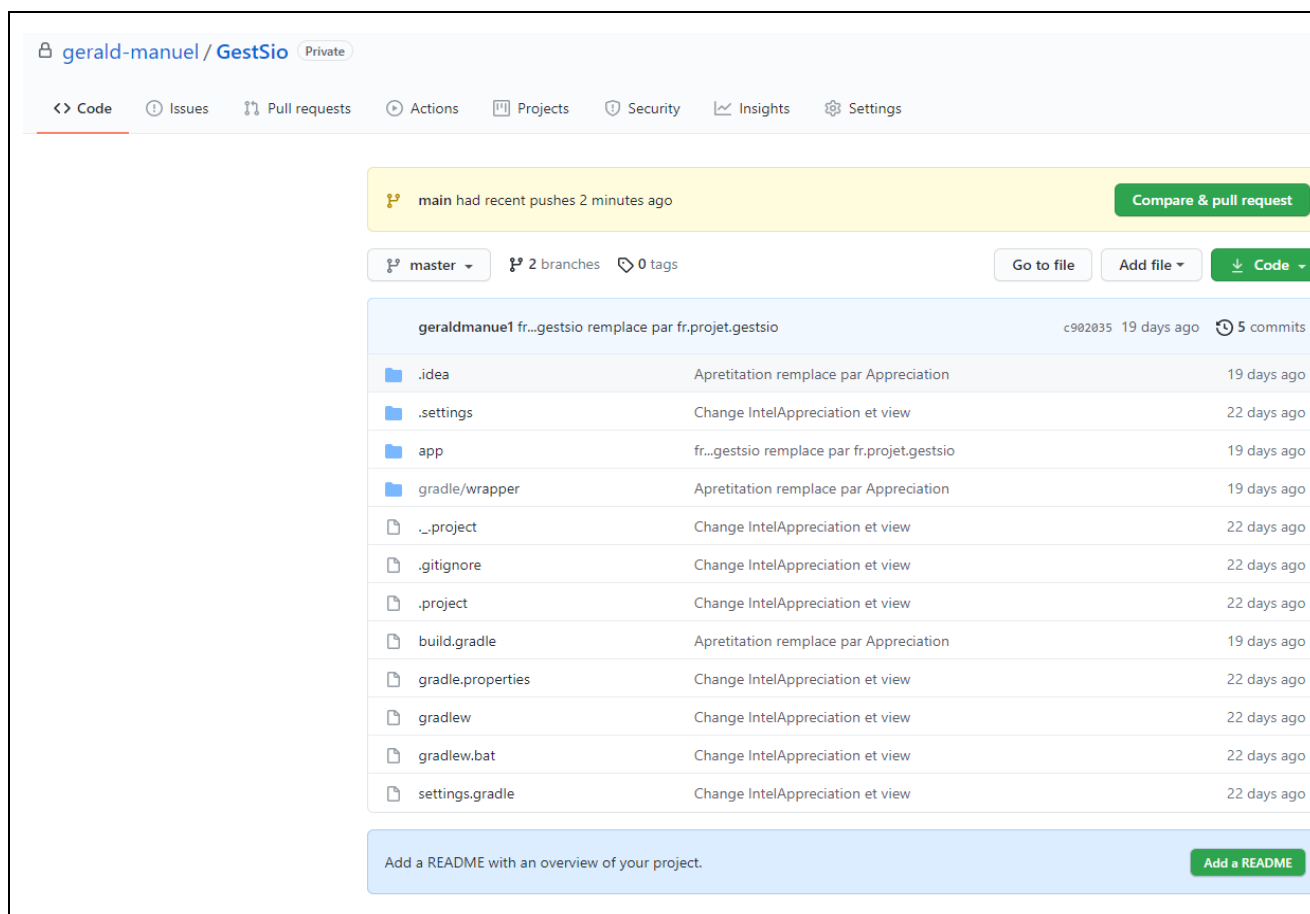
Enfin, pour publier les modifications locales sur un dépôt distant en tape **git push** :

```
$ git push -u <remote> <branch> //exemple
```

```
$ git push -u origin main //publier les modifications locales sur un dépôt distant
```

## 4- Résultat de Git à distance

Pour finir en voit le dépôt à distant a été publier :



The screenshot shows the GitHub interface for a repository named 'GestSio' by user 'gerald-manuel'. The repository is private. The main branch is 'main', which has recent pushes 2 minutes ago. The repository contains 2 branches and 0 tags. The file list shows the following files and their commit history:

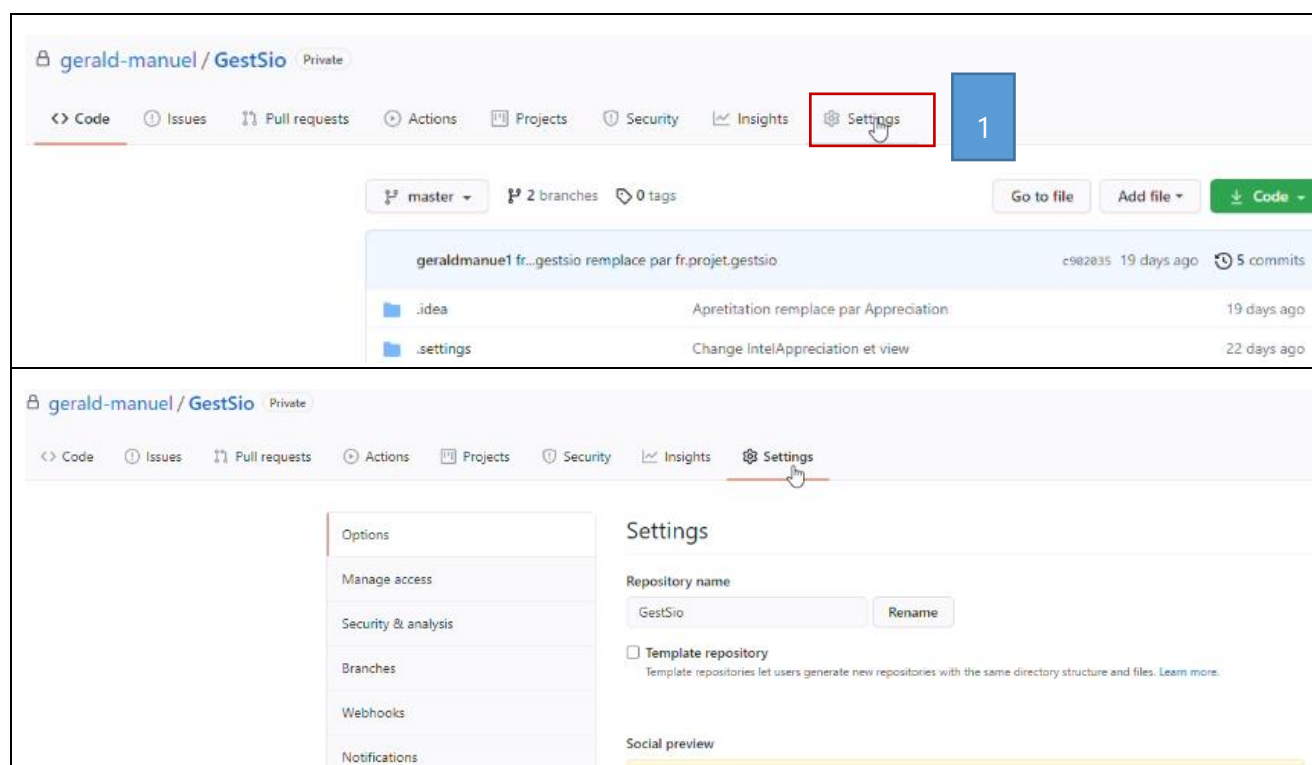
File	Commit Message	Commit Hash	Time Ago	Commits
.idea	Apretitation remplace par Appreciation	c902835	19 days ago	5
.settings	Change IntelAppreciation et view		22 days ago	
app	fr...gestsio remplace par fr.projet.gestsio		19 days ago	
gradle/wrapper	Apretitation remplace par Appreciation		19 days ago	
..project	Change IntelAppreciation et view		22 days ago	
.gitignore	Change IntelAppreciation et view		22 days ago	
.project	Change IntelAppreciation et view		22 days ago	
build.gradle	Apretitation remplace par Appreciation		19 days ago	
gradle.properties	Change IntelAppreciation et view		22 days ago	
gradlew	Change IntelAppreciation et view		22 days ago	
gradlew.bat	Change IntelAppreciation et view		22 days ago	
settings.gradle	Change IntelAppreciation et view		22 days ago	

At the bottom, there is a prompt to 'Add a README with an overview of your project.' and a button to 'Add a README'.

# GestSio

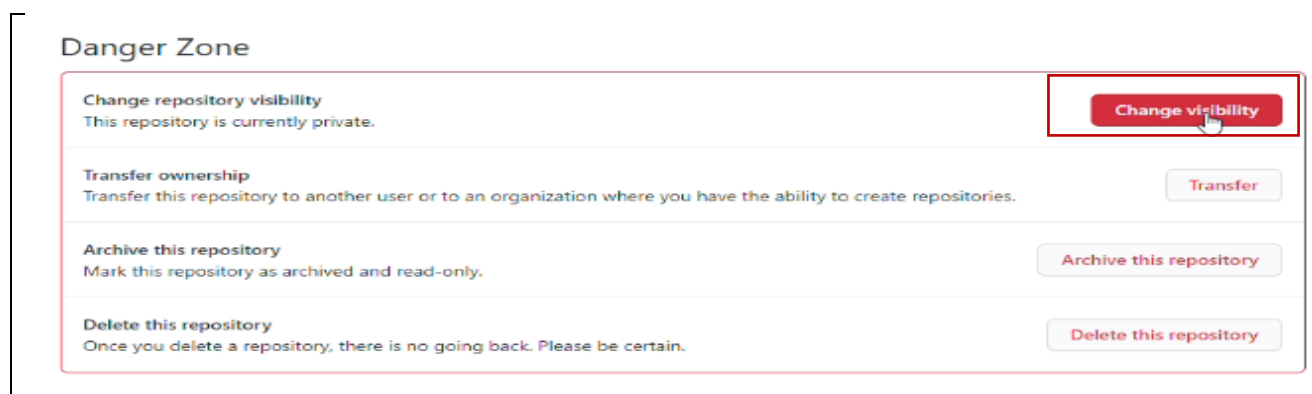
Pour rendre public, va sur **Settings (Paramètre)** :

1



Descent dernier page, vous voyez **Danger Zone** et cliquez sur **Change visibility (Change la visibilité)**

2



2

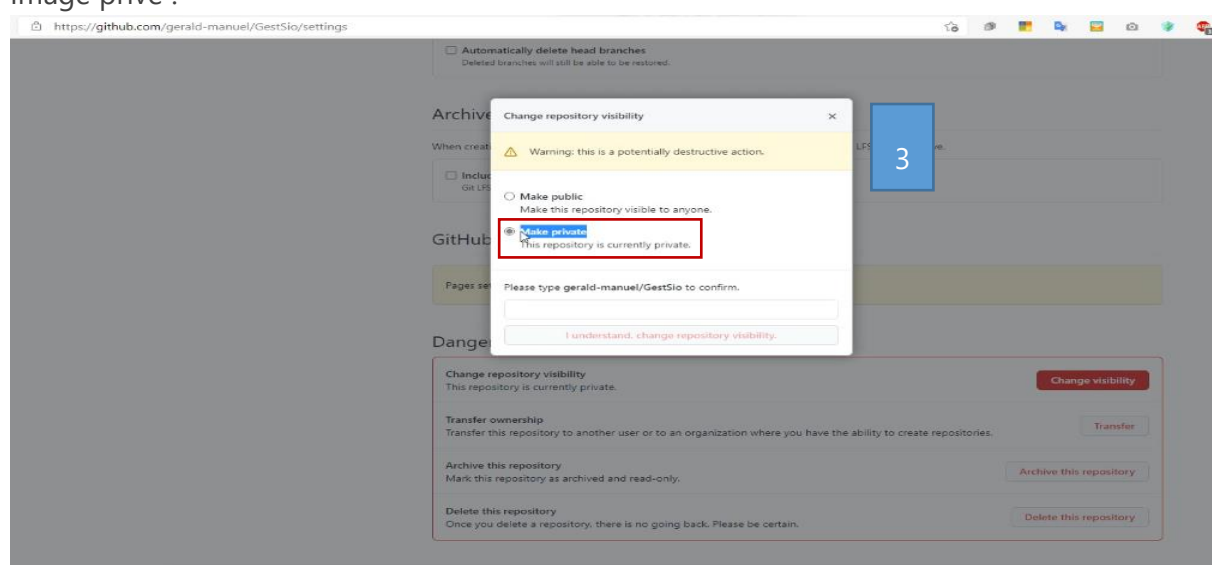


# GestSio

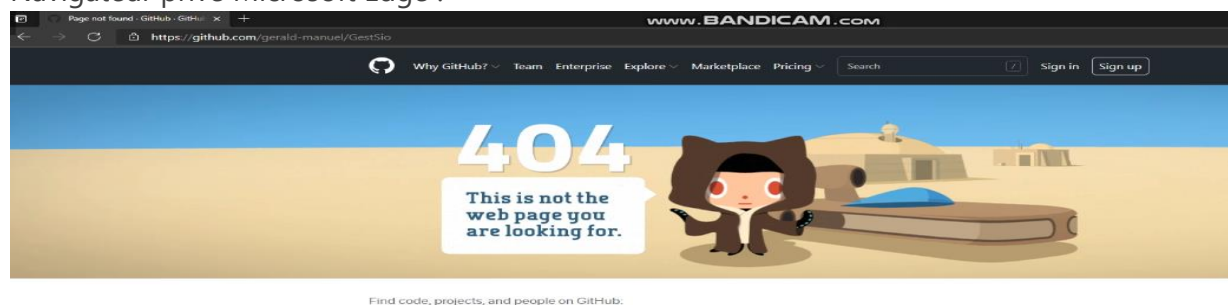
C'est en **Make private (Rendre privée)** en va sur la navigateur privé Microsoft Edge et Chrome pour voir si la visibilité peut accès à tout le monde.

3

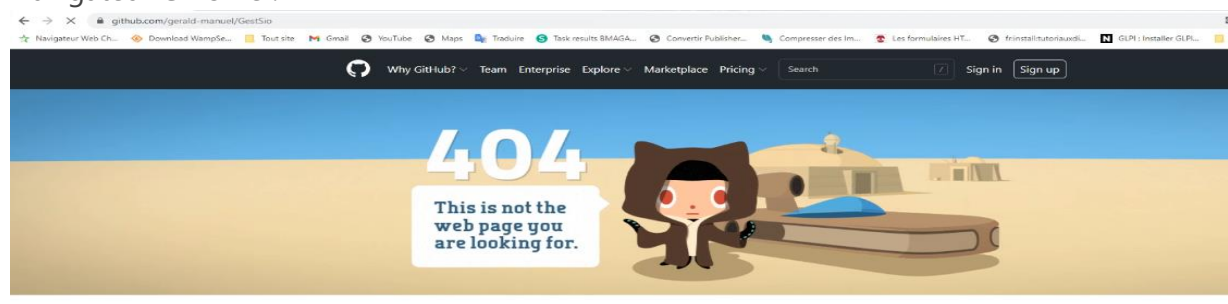
Image privé :



Navigateur privé Microsoft Edge :



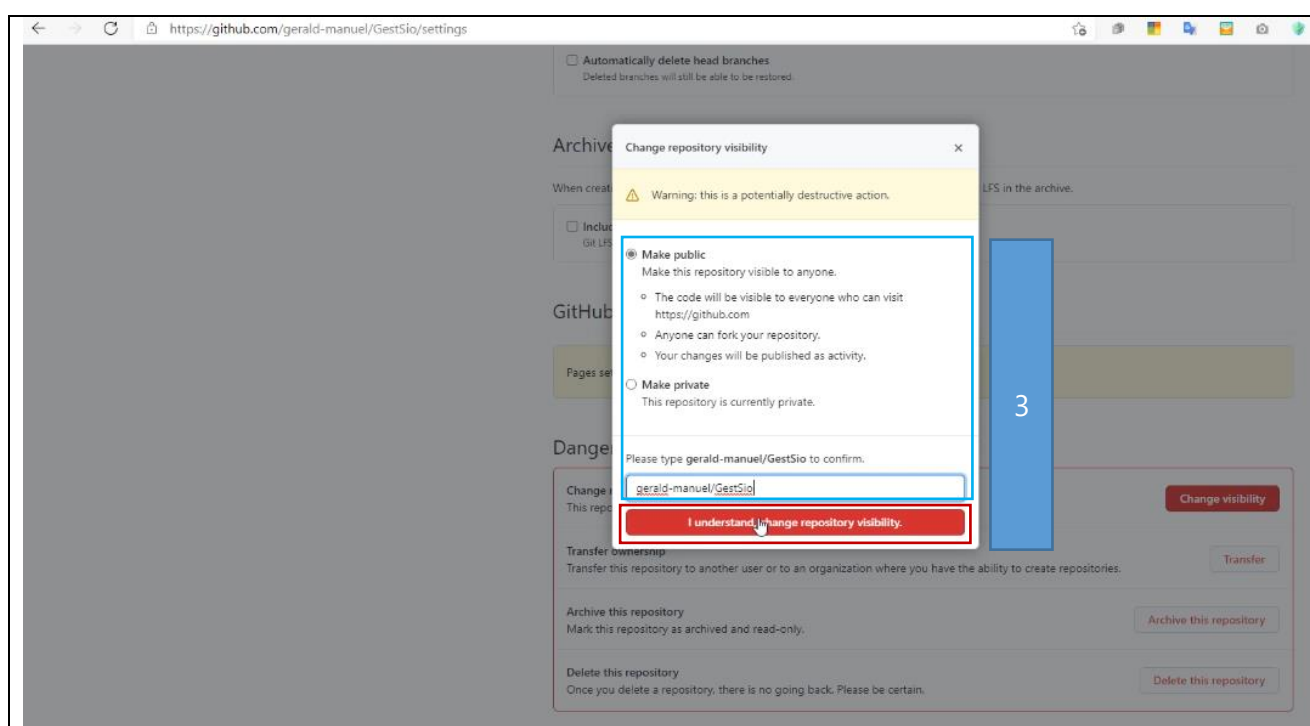
Navigateur Chrome :



# GestSio

Pour rendre public, il faut cliquer sur **Make public (Rendre publique)** copier/coller **gerald-manuel/GestSio** ci-dessous et appuyer sur **I understand change repository visibility pour confirmer** (**Je comprends la visibilité du référentiel de modification pour confirmer**).

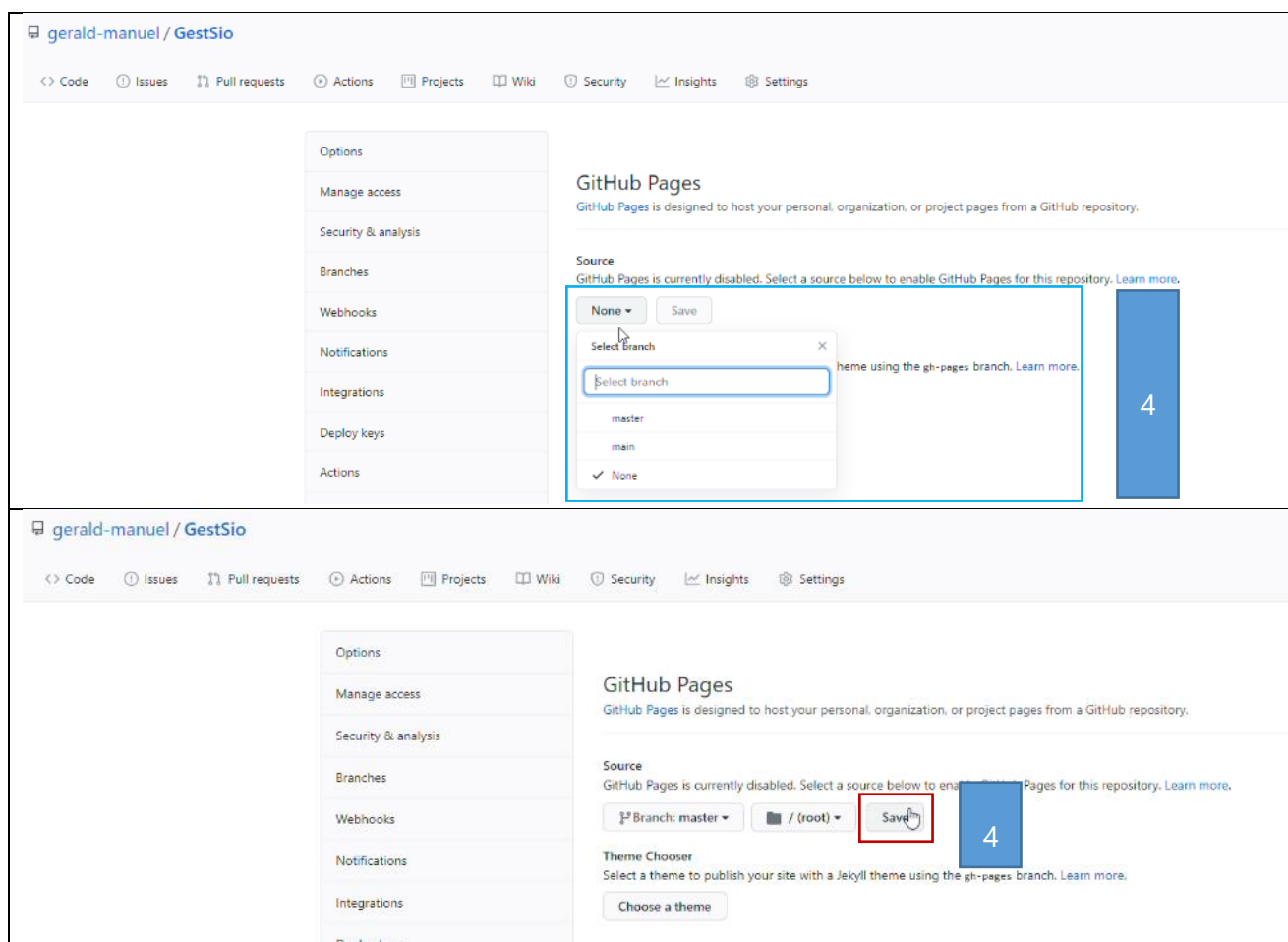
3



# GestSio

Cliquez sur None pour choisir le branch et ensuite **save (sauvegarde)** :

4

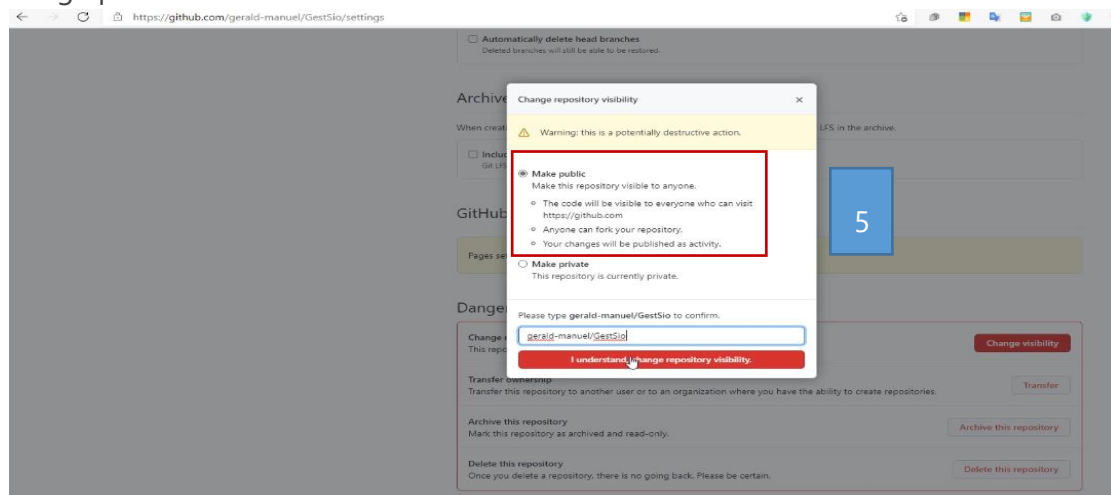


C'est en **Make public (Rendre publique)** en va sur la navigateur privé Microsoft Edge et Chrome pour voir si la visibilité peut être accès à tout le monde.

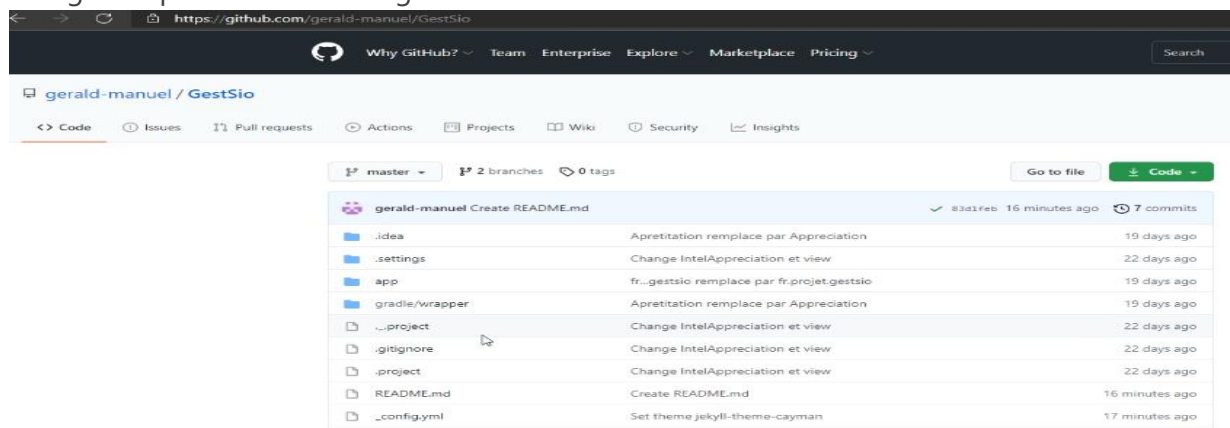
5

# GestSio

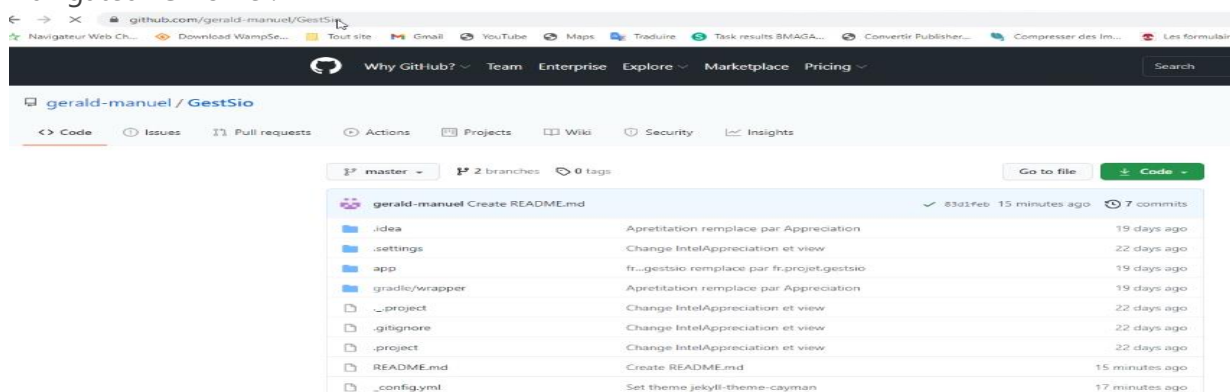
Image privé :



Navigateur privé Microsoft Edge :



Navigateur Chrome :



# GestSio

---

Donc, tous le monde peut accéder sur le projet GestSio

Pour cloner le projet en tape **git clone** :

```
$ git clone <url> //exemple  
$ git clone https://github.com/gerald-manuel/GestSio.git
```

Pour fusionner les modifications de la master distante sur la branche courante :

```
$ git pull <remote> <branch> //exemple  
$ git pull origin main //fusionner les modifications de la master distante sur la branche courante
```

Quelques pistes :

[Tirez le maximum de ce cours ! - Utilisez Git et GitHub pour vos projets de développement - OpenClassrooms](#)

[Gérez vos codes source avec Git - OpenClassrooms](#)

[Git - Book \(git-scm.com\)](#)

[Débuter avec Git et Github en 30 min - YouTube](#)

[geraldmanue1.github.io/Commandes\\_principales\\_de\\_Git.pdf](https://geraldmanue1.github.io/Commandes_principales_de_Git.pdf)