**COMPTE RENDU TP1**

Fiche de TP n°1 : Utilisation de Git

**Partie I : Commandes du Git**

Dans cette partie, nous avons utilisé les commandes de base de Git pour la gestion des dépôts locaux et distants.

1. Installation du Git :

Sur Linux - Ubuntu, l'installation s'est déroulée avec succès, et aucune erreur n'a été rencontrée.

2. Configuration du Git :

- Les configurations du nom d'utilisateur et de l'email ont été effectuées avec succès avec les commande :

git config --global user.name "medis"

git config --global user.email "medisadido@gmail.com"

pour s’assurer que cela a bien fonctionné, nous avons exécuté la commande ; git config –list et nous avons obtenu le resulta suivant :

user.email=medisadido@gamil.com

user.name=medis

core.repositoryformatversion=0

core.filemode=true

core.bare=false

core.logallrefupdates=true

3. Création d'un nouveau répertoire :

- Le dossier de travail "TP1" a été créé avec succès.

- L'initialisation du dépôt local avec git init s'est déroulée sans erreur. Dans la console on voit des information sur l’tat initial du dépôt avec pour message principal :

« Dépôt Git vide initialisé dans /home/medis/Documents/INSA/L2/Genie logiciel/TP1/.git/ »

4. Vérification de l'état des fichiers :

- L'exécution de git status a affiché les fichiers non suivis et confirmé que le dépôt était propre.

Résultat :

Sur la branche master

Aucun commit

Modifications qui seront validées :

(utilisez "git rm --cached <fichier>..." pour désindexer)

nouveau fichier : test.txt

Fichiers non suivis:

(utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)

Gr1\_ADIDO\_Juste\_Medis.txt

5. Ajout des fichiers au transit :

- Le fichier "test.txt" a été ajouté au transit avec succès avec la commande git add « test.txt », comme confirmé par git status.

6. Ajout d'un commit :

- Le commit avec le message descriptif a été effectué avec succès, sans erreur aveec la commande

git commit -m "premier commit"

7. Création d'une branche :

- La nouvelle branche "mon\_fichier" a été créée avec succès : git branch stonely

- Le changement vers la nouvelle branche avec git checkout mon\_fichier s'est déroulé sans problème : git checkout stonely

Résultats : Basculement sur la branche 'stonely'

- L'ajout et le commit du fichier "Gr1\_ADIDO\_Juste\_Medis.txt" dans la branche "stonely" se sont déroulés avec succès.

[stonely 4e1b0a6] premier commit

1 file changed, 1 insertion(+)

create mode 100644 Gr1\_ADIDO\_Juste\_Medis.txt

effectivement à l’issu des manipulations nous avons navigué de nouveau vers la branche master, et nous n’avons plus vu apparaître notre le fichier contenant nos noms et prénoms le dossier.

Cela veut dire que nous avons réussir à versionner notre dossier grâce à deux branches(master et stonely ) qui contiennent des fichiers bien distincts et bien versionné grâce au commits .

- A la fin nous avons exécuté la commande « git log --oneline --graph --name-status » pour voir le journal de nos actions .

Résultat :

\* 4c66fe2 (HEAD -> stonely) activite utilisateur

| M Gr1\_ADIDO\_Juste\_Medis.txt

\* 4e1b0a6 premier commit

| A Gr1\_ADIDO\_Juste\_Medis.txt

\* 4a345be (master) premier commit

A test.txt

**Partie II : Utilisation avancée de Git**

1. Récupération du répertoire distant :

- L'ajout du lien du répertoire distant avec

git remote add origin https://github.com/TPGLINSA/TP1GIT

s'est déroulé sans erreur.

- En voulant télécharger le contenu actuel du répertoire distant grâce à la commande

git pull origin master

Nous avons obtenu une erreur d’incompatibilité des projets :

fatal: Besoin de spécifier comment réconcilier des branches divergentes.

Pour forcer le tirage , nous avons exécuté la commande :

git pull origin master –allow-unrelated-histories

et avons obtenu une sortie de succès :

Depuis https://github.com/TPGLINSA/TP1GIT

\* branch master -> FETCH\_HEAD

Merge made by the 'ort' strategy.

System\_hospitalier.java | 46 ++++++++++++++++++++++++++++++++++++++++++++++

1 file changed, 46 insertions(+)

create mode 100644 System\_hospitalier.java

Le téléchargement du contenu du répertoire distant avec git pull origin master --allow-unrelated-histories a été effectué avec succès, malgré le message d'erreur initial.

2. Création d'une nouvelle branche :

- La création de la nouvelle branche avec le nommage approprié s'est déroulée sans problème nous avons ensuite basculé vers cette branche

git branch Gr1\_ADIDO\_Juste\_Medis

git branch checkout Gr1\_ADIDO\_Juste\_Medis

3. Modification de fichiers et commit :

- Les modifications apportées au fichier "System\_hospitalier.java" ont été réalisées avec succès.

- Le commit des modifications dans la nouvelle branche s'est déroulé sans erreur.

4. Création d'un nouveau répertoire sur GitHub :

addresse : git remote add origin https://github.com/Juste-medis/tp1\_glogiciel.git

- Le nouveau répertoire public a été créé sur GitHub avec succès.

5. Push de la nouvelle branche :

- Le push de la nouvelle branche vers le répertoire distant a été effectué avec succès.

6. Compte rendu :

- Le compte rendu détaillé a été préparé avec succès, et le lien du répertoire public a été ajouté conformément aux instructions.

Ceci conclut notre travail sur la fiche de TP n°1.