PWA - GIT GUIDE

0. Cloner le projet (COTE ORIGIN vers COTE LOCAL)

```
git clone https://github.com/moundjis/MediSoft-Pil.git
```

1. Récupérer les dernières mises a jour de la branche principale MAIN

```
git pull origin main
```

2. Créer une branche secondaire (DEPOT LOCAL) et se placer dessus

```
git switch -c branche-secondaire
```

3. Vérifier sur quelle branche on se situe

```
git branch
```

4. Créer la branche secondaire sur le dépôt distant (ORIGIN) et établir la liaison : LOCAL-ORIGIN

```
git push -u origin branche-secondaire
```

5. Copier et lier une branche secondaire distante (ORIGIN) déjà créée par un collègue vers ton LOCAL

```
git switch --track origin/branche-secondaire
```

6. Ajouter les fichiers modifies

```
git add .
```

7. Enregistrer le travail effectue avec un titre

```
git commit -m "Docs: Resume de mon travail"
```

8. Envoyer le travail effectue sur la branche secondaire (DEPOT LOCAL) –vers–> (ORIGIN)

```
git push origin branche-secondaire
```

9. Nettoyer et mettre a jour la liste des branches secondaires dans le depot distant (ORIGIN)

```
git fetch --prune   : Nettoie les branches secondaires distantes supprimees
git branch -r       : Affiche toutes les branches distantes
git branch -a       : Affiche toutes les branches
```

10. Supprimer une branche-secondaire (DEPOT LOCAL)

```
git branch -d branche-secondaire
```

11. Supprimer une branche-secondaire (ORIGIN)

```
git push origin --delete branche-secondaire
```

12. Création : Commits & Branche secondaires

CREATION - COMMIT

```
Fix             : pour des corrections de bugs.
Feat            : pour les nouvelles fonctionnalites.
Chore           : pour les taches techniques non visibles pour l'utilisateur
(comme la mise à jour des dependances).
Docs            : pour les modifications liees a la documentation.
Refactor        : pour les refontes de code sans ajout de
                    nouvelles fonctionnalites.
Style           : pour les corrections de style (formatage, indentation).
```

Exemple :

```
git commit -m "Feat: Ajoute une fonctionnalite de recherche par categorie
```

Permet à l'utilisateur de filtrer les résultats par catégorie. La recherche est effectuee sur la base de donnees en temps reel."

CREATION - BRANCHE SECONDAIRE

```
feature/        : Pour ajouter une nouvelle fonctionnalite
bugfix/         : Pour corriger un bug detecte en developpement
hotfix/         : Pour une correction urgente en production
refactor/       : Pour reorganiser ou ameliorer du code sans ajouter
                    de nouvelles fonctionnalites
docs/           : Pour modifier ou ajouter de la documentation
```

Exemple :

```
git branch feature/ajout-authentification
git branch bugfix/correction-dashboard
git branch docs/mise-a-jour-readme
```

⚠ FUSION DES BRANCHES

Cette commande sera effectuée uniquement par la personne responsable des fusions des branches.

```
git checkout main
git pull origin main
git merge branche-secondaire
git push origin main
```

ⓘ INFOS SUR LES BRANCHES SECONDAIRES

Cette commande permet de connaitre qui a cree ou modifie chaque branche secondaire.

```
git for-each-ref --format='%(committername) -> %(refname:short)'
refs/remotes/
```