



Cégep de Saint-Hyacinthe
Département d'informatique

Programmation orientée objet

420-2DP-HY

(3-3-3)

Initiation à un gestionnaire de versions (GIT) #2

(Version 1.2)

0.5 à 1 heure

Préparé par

Martin Lalancette

Comprendre les éléments suivants:

- Étiqueter une branche
- Créer une branche à partir d'une autre branche
- Fusionner deux branches

Table des matières

Introduction.....	3
Les étiquettes (<i>tag</i>) ou balises.....	3
Deux types d'étiquettes	3
Étiqueter une branche via VS	4
Les branches	6
Créer une branche à partir d'une autre branche	7
Fusionner une branche à partir d'une autre branche	9
Bibliographie.....	11

Introduction

Cette séquence a pour but de vous initier aux notions de base nécessaires à la gestion de versions en utilisant DevOps (Azure Repos – GIT). Nous commencerons par énoncer les éléments théoriques appuyés d'exemples simples et faciles à reproduire. Afin d'axer l'attention sur la compréhension de ces notions, il y aura des exercices à faire tout au long de cette séquence. Cette séquence touche principalement **l'étiquetage** et la gestion de **branches**.

Les étiquettes (*tag*) ou balises

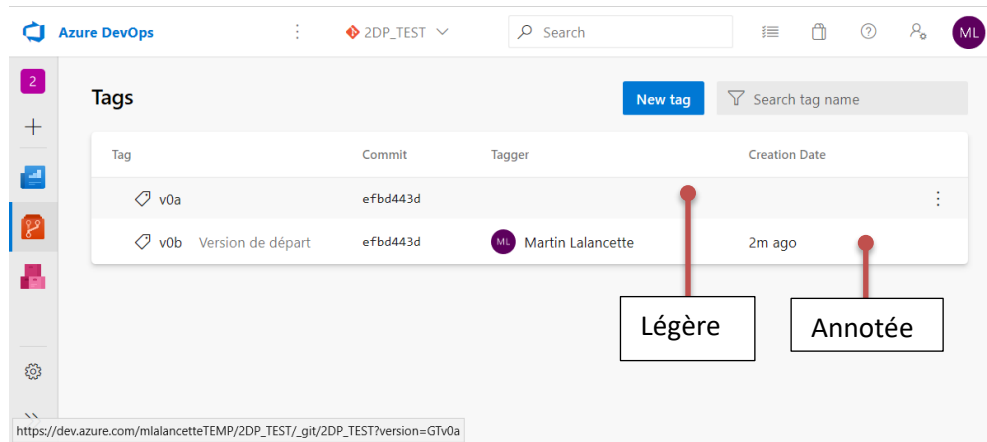
La plupart des gestionnaires de source offrent la possibilité d'étiqueter un état (*commit*) dans l'historique d'une branche comme étant important. **Souvent utiliser pour marquer des états de publication** (ex. : version1.0, v1.0, etc.).

Deux types d'étiquettes

Avec GIT, il existe deux types de balises: Légères et annotées.

Étiquette légère : Est une étiquette rattachée à un *commit* en particulier et qui ressemble à une branche qu'on ne peut modifier. Elle est la forme la plus simple, car elle comporte uniquement **un nom**. Elle est considérée comme un **pointeur directe** sur un *commit* et contient seulement son nom comme information. Souvent utilisé en mode privé.

Étiquette annotée : Est une étiquette qui est emmagasinée en tant qu'**objet** dans la base de données permettant ainsi de conserver plus d'informations : nom de l'étiquette, nom et courriel du développeur qui a créé l'étiquette, la date, message relié, peut être signée et vérifiée GNU Privacy Guard (GPG). Elle peut être référencée → clé SHA (Secure Hash Algorithm). Souvent utilisé en mode public.

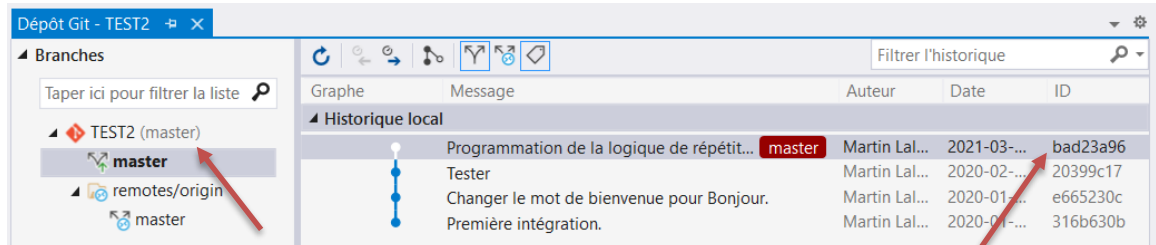


L'avantage principal d'étiqueter une branche (à un *commit* précis) est que, dans le futur, vous pouvez vous créer une nouvelle branche à partir de cette étiquette en tout temps en sachant qu'aucun *commit* ne l'aura affecté.

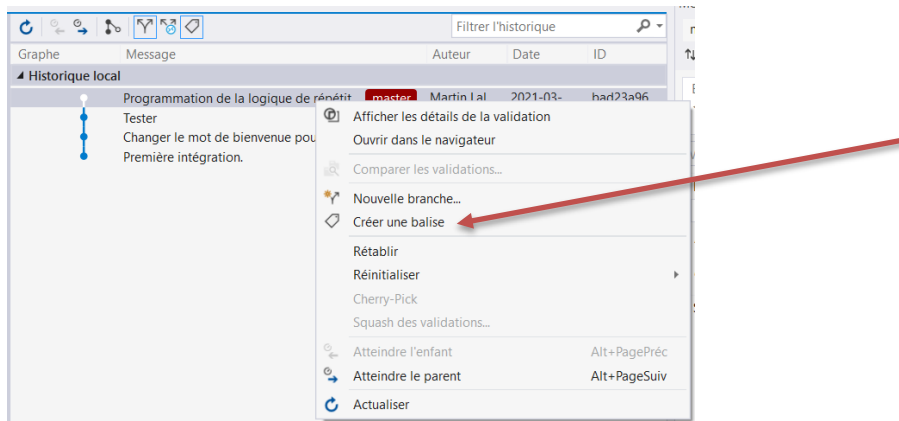
Étiqueter une branche via VS

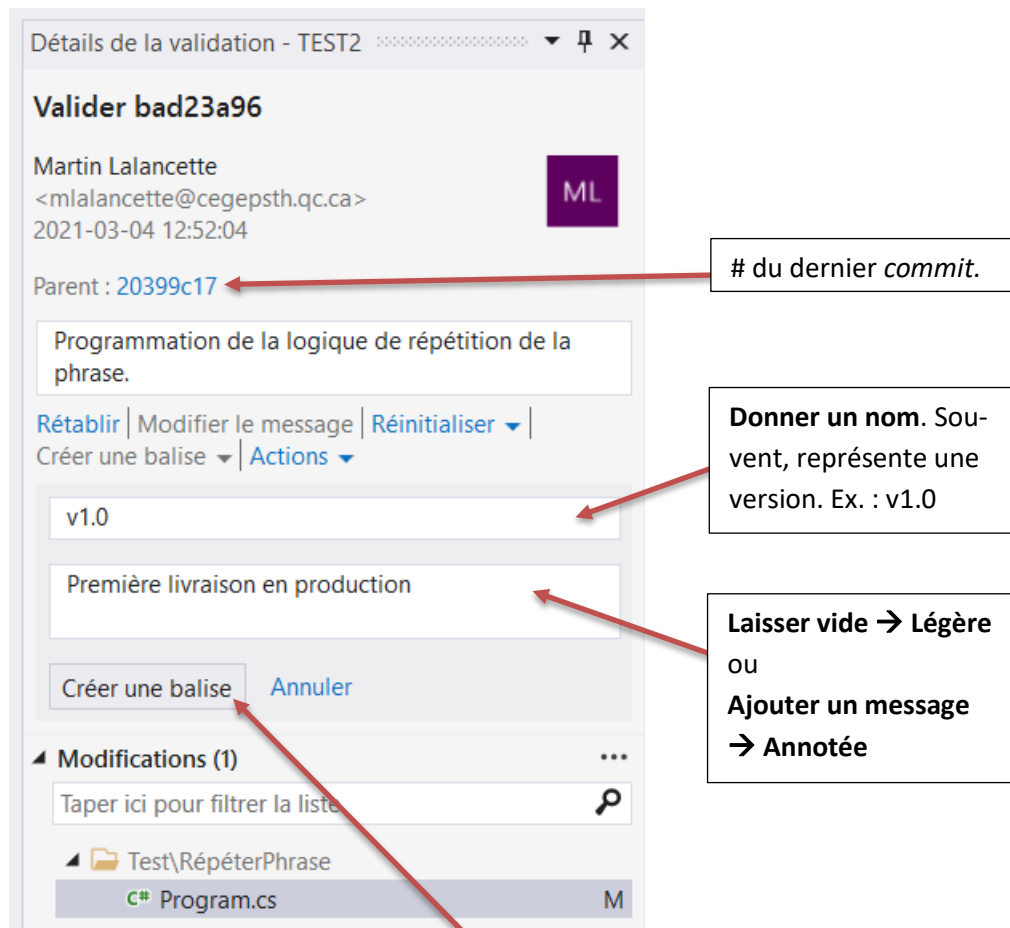
Voici la procédure à suivre pour étiqueter une branche au dernier *commit* :

1. Dans le menu **GIT**, choisir **Gérer les branches**,
2. Sélectionner la branche et le *commit* à étiqueter. Exemple :

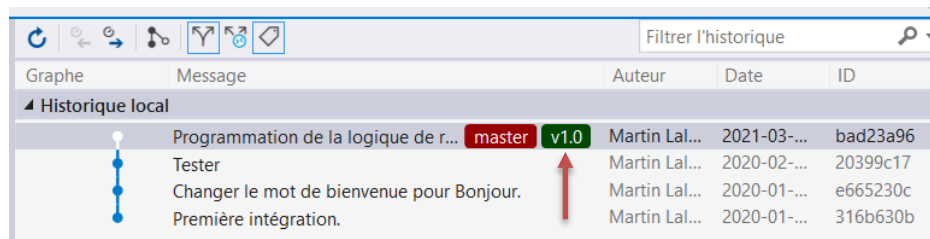


3. Cliquer-droit sur le *commit* et choisir **Créer une balise**. Exemple :

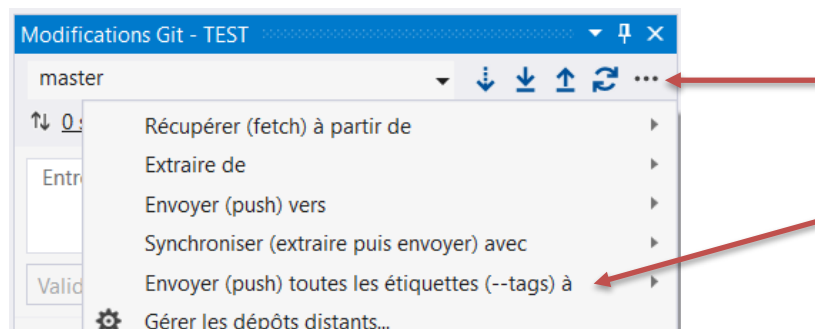




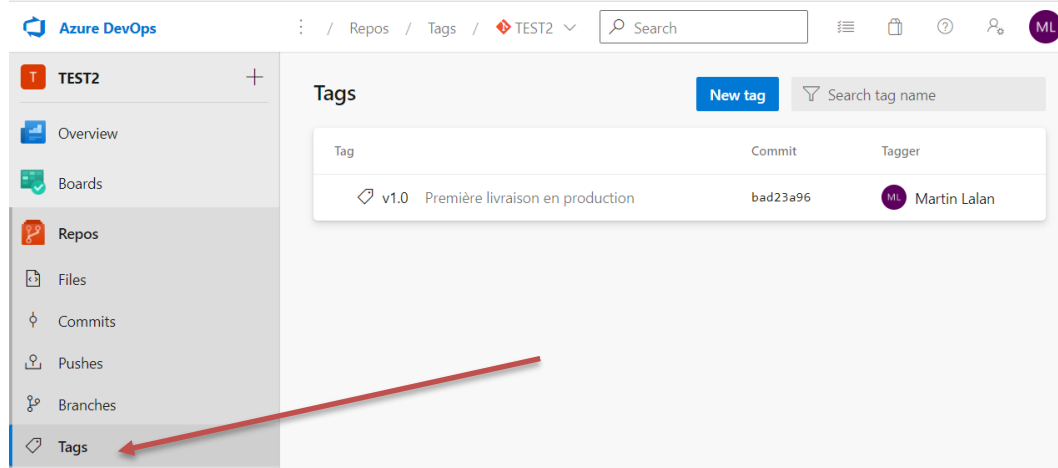
4. Cliquer sur le bouton **Créer une étiquette**. Exemple après:



5. Vous devez pousser l'étiquette dans le référentiel central via **Modifications GIT** et le bouton **Push (Envoyer toutes les étiquettes)**. Exemple :



6. Consulter les étiquettes dans le référentiel central. Exemple :

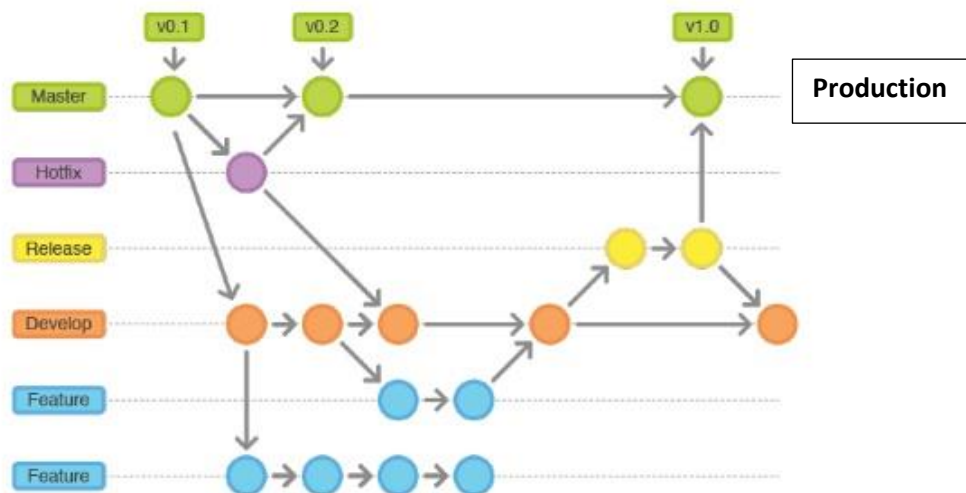


Par la suite, si un jour vous souhaitez créer une nouvelle branche à partir de cette étiquette, vous pourrez la sélectionner (cliquer droit) et choisir **Créer une nouvelle branche locale**.

Exercice 1. : Ouvrir la solution **TestConsole** et le projet **RépéterPhrase**.
Étiqueter la branche master à la version **v1.0**.

Les branches

Maintenance Branches



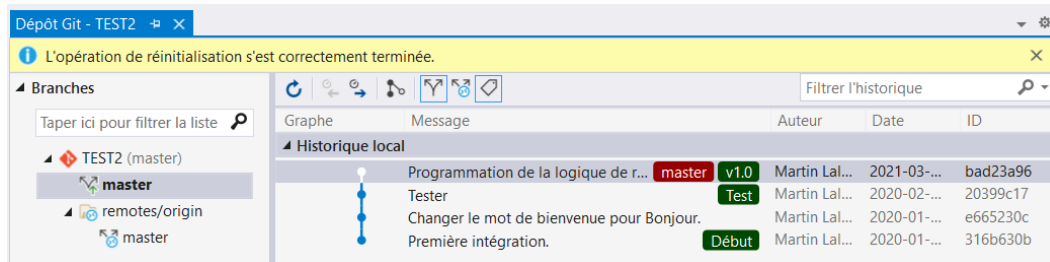
(Milbaum, 2020)

Dans un gestionnaire de sources, les [branches](#) font partie intégrante d'une saine gestion dans la conception d'une application. « **Créer une branche signifie diverger de la ligne principale de développement et continuer à travailler sans impacter cette ligne.** » (GIT-SCM, 2020). GIT encourage la création et la fusion de branches.

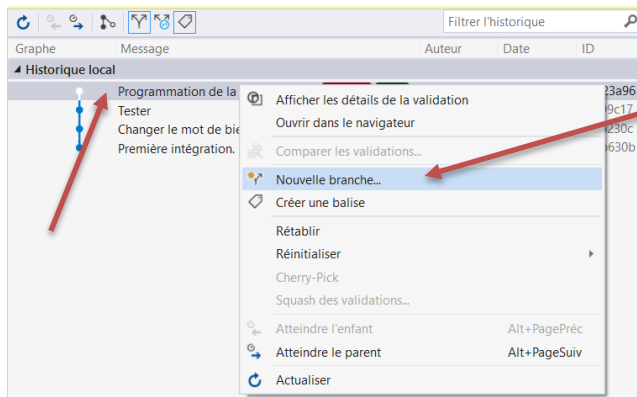
Créer une branche à partir d'une autre branche

Cette section consiste à créer une nouvelle branche à partir d'une branche existante. Voici la procédure :

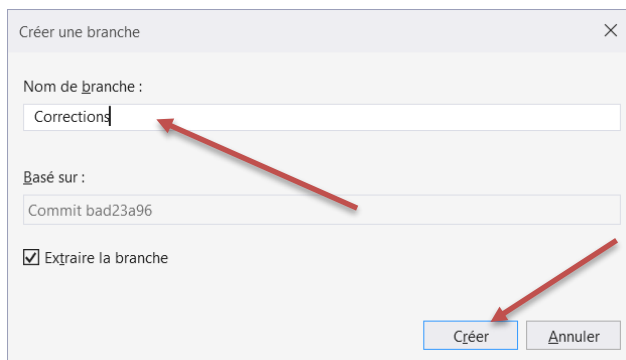
1. Dans **Dépôt GIT**, sélection la **branche de départ** et le *commit*,

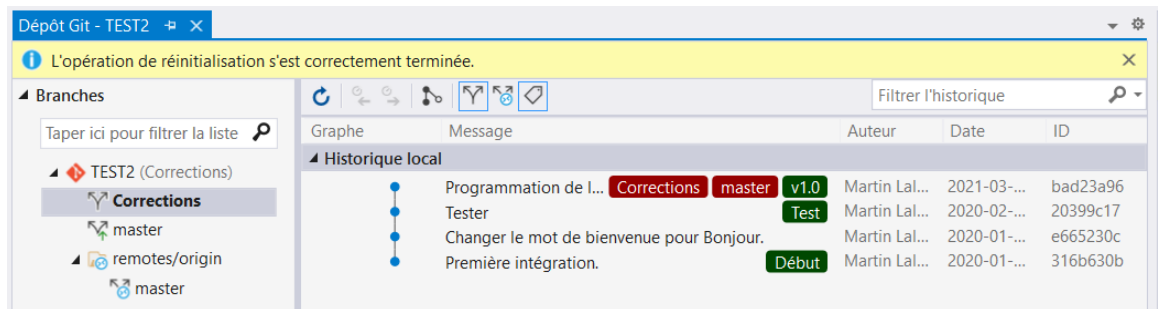


2. Cliquer-droit sur le *commit* et choisir **Nouvelle branche...** Exemple :

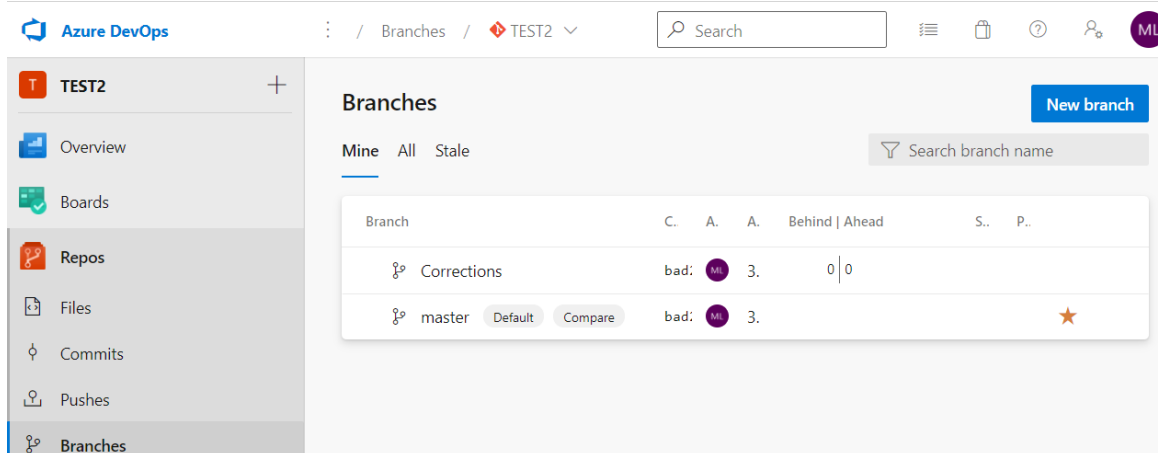


3. Entrer le nom de la nouvelle branche et cliquer sur **Créer**. Exemple :

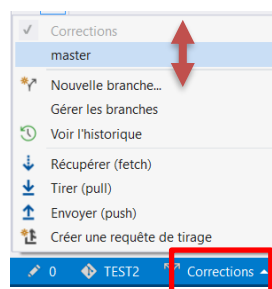




4. Vous devez pousser la branche dans le référentiel central, en cliquant droit sur le nom de la branche et choisir **Envoyer (push)**.
5. Vérifier sur le site DevOps. Exemple :



Porter une attention particulière à VS, lorsque plus d'une branche est créée dans un référentiel. Vous pouvez basculer d'une branche à l'autre via (en bas à droite):



Exercice 2. : Créer une nouvelle branche locale nommée **Corrections** à partir de la branche *master* et la pousser dans le référentiel central.

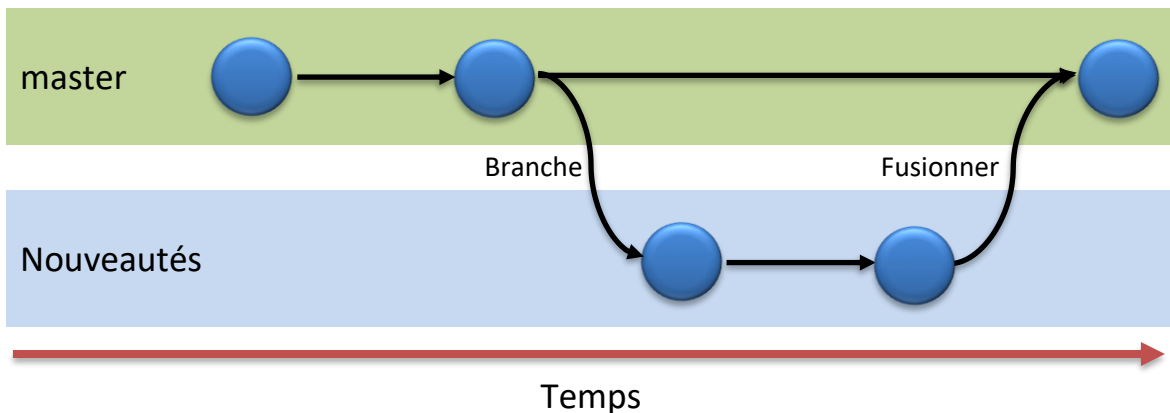
Exercice 3. : Créer une nouvelle branche **Nouveautés** à partir de **master**.

Exercice 4. : Branche **Corrections**, effectuer deux modifications mineures dans **program.cs**, les valider et les pousser dans le référentiel central.

Exercice 5. : Branche **Nouveautés**, ajouter la fonctionnalité suivante: permettre d'afficher la phrase inversée en posant la question à l'utilisateur dans **program.cs**, les valider et les pousser dans le référentiel central.

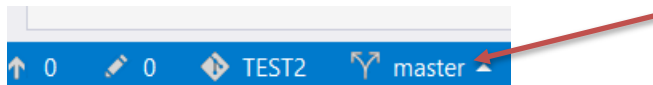
Fusionner une branche à partir d'une autre branche

La prochaine section traite de fusion entre les branches. GIT offre la possibilité de travailler avec plusieurs branches de mener à bien plusieurs versions en parallèle d'un produit.

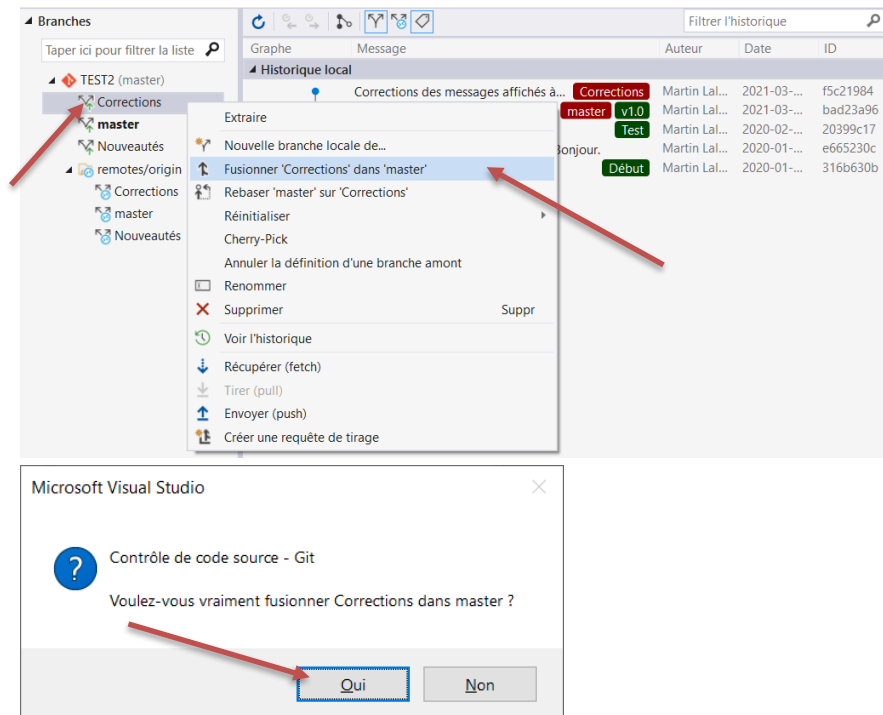


Voici la procédure à suivre afin de fusionner deux branches entre elles :

1. Définir la branche vers laquelle nous souhaitons fusionner une autre branche dans la barre de statut. Exemple:



2. Dans **Dépôt GIT**, sélectionner la branche à fusionner, cliquer-droit et choisir Fusionner.... Exemple:



3. Pousser la fusion vers le dépôt central.

Exercice 6. : Fusionner la branche **Corrections** dans **master**.

Exercice 7. : Fusionner la branche **Corrections** dans **Nouveautés**.

Bibliographie

GIT-SCM. (2020, 02 18). *Les branches avec Git*. Retrieved from Git-Scm: <https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Les-branches-en-bref>

Milbaum, L. (2020, 02 18). *Maintenance Branches*. Retrieved from SlideShare: <https://pt.slideshare.net/>