

Laboratoire introductif (formatif)

- a) Créer un compte GitHub personnel
- b) Faites une équipe avec un collègue du local
- c) Prenez le projet de départ **Conversions.zip** fourni sur Léa
- d) Compléter le Bouton « `btnBinaireDecimal` ». **Ce bouton** lit un nombre binaire et le convertit en nombre décimal. Valider que le nombre binaire soit de 8 bits au maximum et qu'il ne contient que des 1 et des 0. (Une solution proposée de code est fournie au verso de cette page pour ce bouton)
- e) Faites le bouton2; après avoir fait un algo et/ou dessin sur papier.

The screenshot shows a Windows application window titled "Les conversions". The window contains a grid of conversion controls. On the left side, there are four rows of controls, each consisting of a text input field, a button with a right-pointing arrow, and a label. The labels are: "Binaire", "Décimal", "Hexadécimal", and "Décimal". The first row shows the input "101" and the output "5". On the right side, there are four rows of controls, each consisting of a text input field, a button with a right-pointing arrow, and a label. The labels are: "Décimal", "Binaire", "Décimal", and "Hexadécimal". The first row shows the input "5" and the output "101".

From	To
Binaire	Décimal
Décimal	Binaire
Hexadécimal	Décimal
Décimal	Hexadécimal
Binaire	Hexadécimal
Hexadécimal	Binaire

```

private void btnBinaireDecimal_Click(object sender, EventArgs e)
{
    //lecture
    string sBinaire = txtBinaireADec.Text;
    int nbDecimal=0;
    int i = 0;
    int exposant = 0;

    //validation
    if (sBinaire.Length > 8)
    {
        MessageBox.Show("doit être sur 8 bits ou moins");
        return;
    }
    while ((i<sBinaire.Length) && (sBinaire[i]!='0' || sBinaire[i]!='1'))
    {
        i++;
    }
    if(i< sBinaire.Length)
    {
        MessageBox.Show("quand c'est binaire, c'est des 1 ou des 0");
        return;
    }

    //traitement
    for (i = sBinaire.Length - 1; i >= 0; i--)
    {
        if((sBinaire[i] == '1'))
            nbDecimal += (int)Math.Pow(2, exposant);
        exposant++;
    }

    //affichage
    txtDecimalDeBin.Text = nbDecimal.ToString();
}

```

Laboratoire #1 (5%) à venir sera de compléter totalement le projet « Conversion »

Donc de compléter aussi les autres boutons des différentes conversions et de remettre le projet zippé au professeur par **mio** (En équipe de 2 pour pratiquer GitHub entre coéquipiers).

Programmation Système

Laboratoire #1

Conversion de base

420-2C5-LL

Nom(s) : _____

/5

La maîtrise de ces objectifs sera évaluée aux examens:

- Expérimenter les étapes de résolution de problèmes;
- Compléter une application en C#, qui expérimente le conversion entre différentes bases;
- **Respecter les normes de programmation qui ont été apprises en 420-2A5-LL;**
- Utiliser les notions du langage C#;
- Utiliser les bons types de données;
- Concevoir les données de test afin de valider les résultats de l'application;

Produits à remettre:


Projet zipper à remettre sur Mio

Spécifications du travail:

Directives : Faire les algorithmes et les jeux d'essai pour la conversion sous différentes formes (les algorithmes ne sont pas à remettre mais c'est pour vous aider).

Votre application convertira des valeurs en Décimal, Hexadécimal et en Binaire à partir d'une chaîne saisie au clavier.

Notez bien que l'utilisateur doit saisir une valeur à convertir, votre application doit faire une **validation** sur cette valeur selon le choix de conversion choisi.

Après avoir saisi la donnée à convertir, on presse sur un des boutons , la conversion se fera et le résultat sera affiché dans la **zone d'affichage correspondante**. **Vous ne devez pas utiliser de fonctions qui permettent de faire la conversion des nombres automatiquement, vous devez faire vos propres code de conversions.**

Décimal, en entrée : **234** (entre 0 et 255 en int)

Bon travail

Hexadécimal, en entrée : **B7** (string de 2 caractères de '0' à '9' et 'A' à 'F')

Binaire: en entrée : **11101010** (1 à 8 bits en string)

Par la suite, en appuyant sur les boutons de conversion les résultats doivent s'afficher aux endroits appropriés.

Ex :

The screenshot shows a web application titled "Les conversions". It contains five rows of conversion controls, each with a label, an input field, a button with a right-pointing arrow, and an output field.

Label	Input	Button	Output
Binaire	00101111	→	47
Décimal	47	→	Binaire 00101111
Hexadécimal	2F	→	Décimal 47
Décimal	47	→	Hexadécimal 2F
Binaire	00101111	→	Hexadécimal 2F

Barème de correction :

Les 6 boutons fonctionnent et le code est bien structuré (**3 points**).

Une classe **CBase** contenant les codes des traitements utilisés dans les boutons (**1 point**).

`public static class CBase {... } // donc tous ses méthodes membres sont static`

Les méthodes statiques reçoivent une donnée validée et retourne un string contenant un résultat (**1 point**).

Bonus de 1 point pour l'optimisation du code (*enlever des répétition de code*)

Des méthodes du formulaire pourraient être créées pour valider

Date de remise : **Vendredi le 10 septembre.**

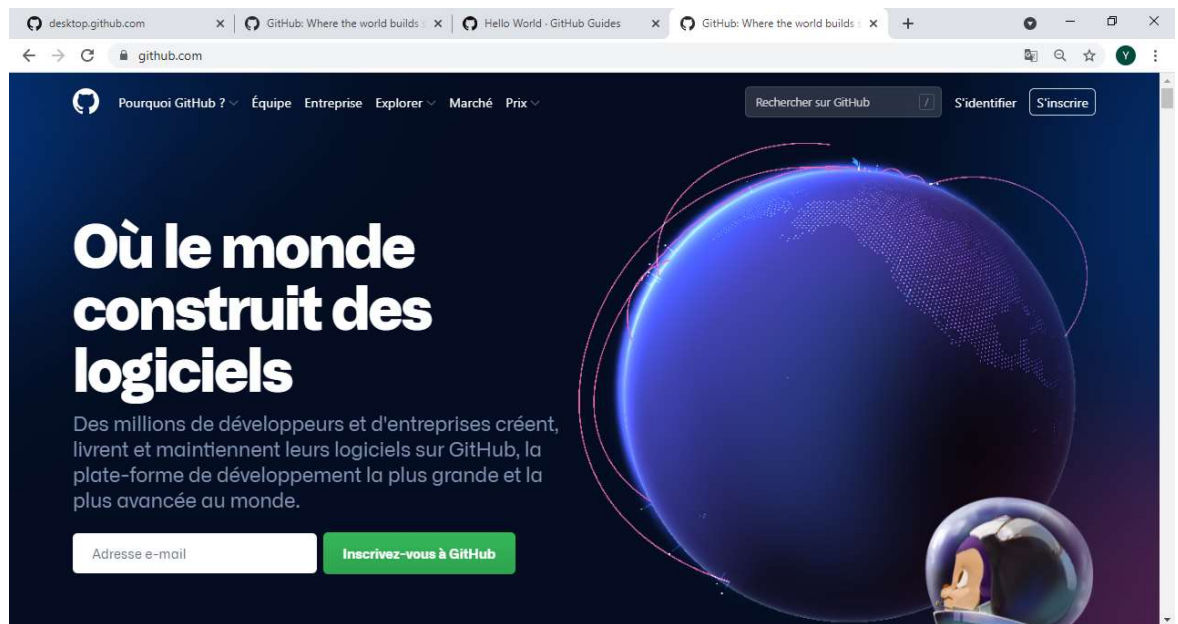
(-10% par jour de semaine en retard)

Bon travail

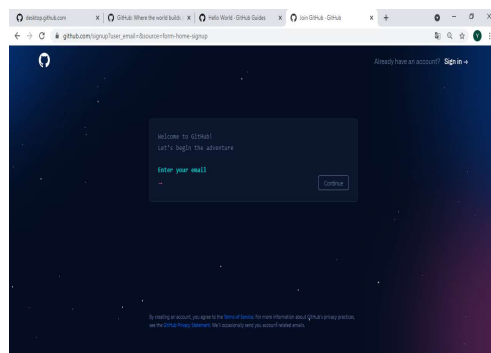
Projet Visual Studio avec Git

(Gestionnaire de version distant)

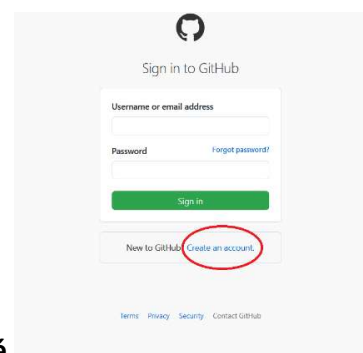
1. Aller sur le lien suivant : <https://github.com/>
2. On devrait avoir la fenêtre suivant : Cliquer sur sign in (*s'inscrire en français*)




3. On retrouve la fenêtre ci-bas.



ou l'an passé



4. On clique sur Create an account (suivre les étapes de création de votre compte)
5. Une fois votre compte créer faire un sign in avec votre login, password



Sign in to GitHub

Username or email address

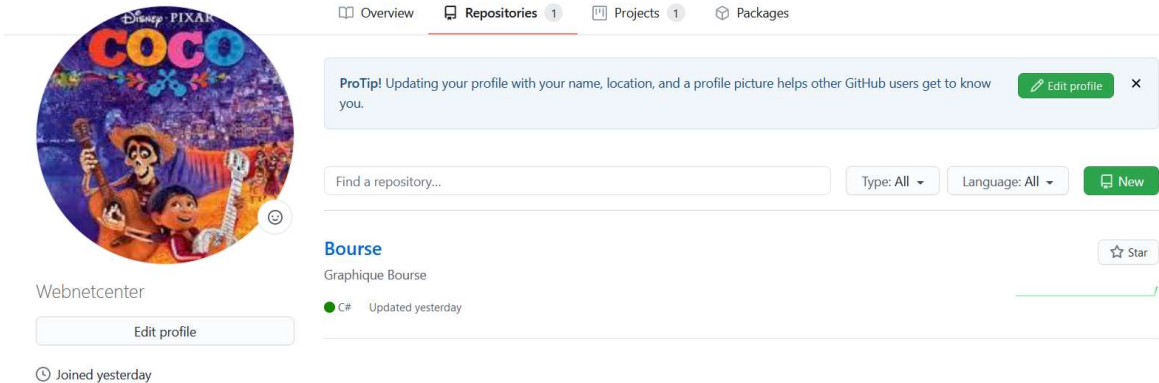
Password [Forgot password?](#)

[Sign in](#)

New to GitHub? [Create an account.](#)

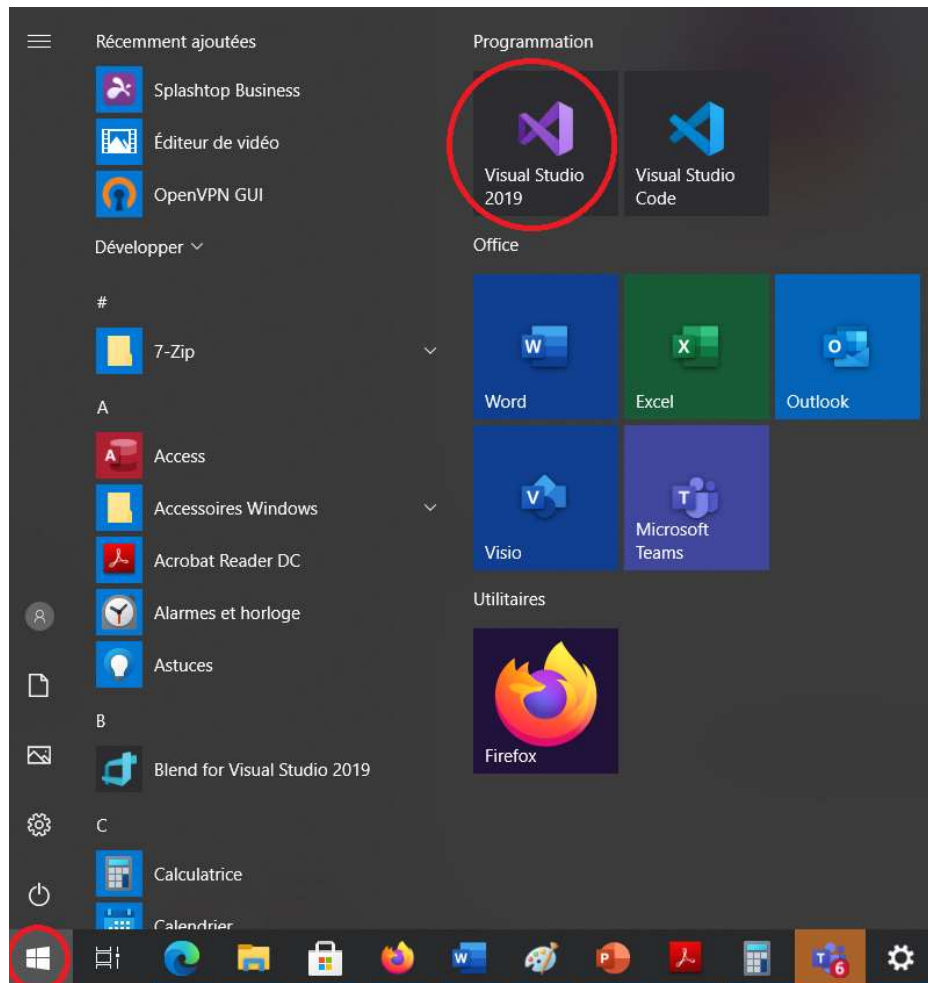
[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)

6. Vous devriez pouvoir entrer dans votre compte GIT(N.B. vous ne devriez pas avoir le projet que j'ai dans mon compte)

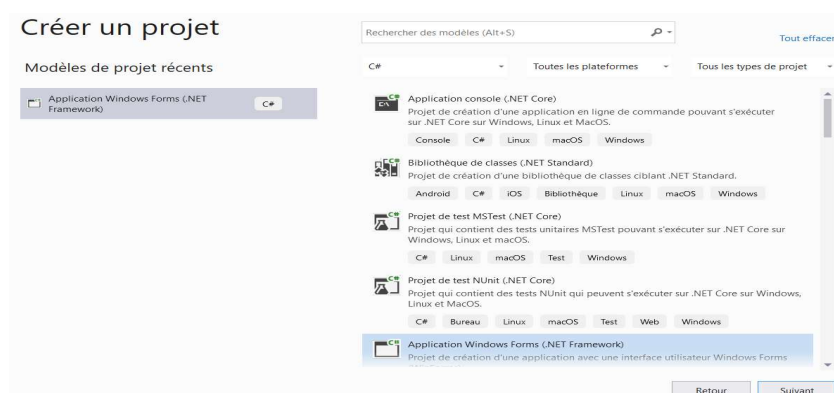


The image shows a GitHub profile page for a user named "Webnetcenter". The profile picture is a circular image featuring characters from the Disney Pixar movie "Coco". The page layout includes a header with navigation tabs: "Overview", "Repositories" (with a count of 1), "Projects" (with a count of 1), and "Packages". Below the header, there is a "ProTip!" banner encouraging profile updates, followed by a search bar for repositories and filters for "Type" and "Language". The main content area displays a repository named "Bourse" with the description "Graphique Bourse", a C# language icon, and a status of "Updated yesterday". A "Star" button is visible next to the repository. The user's name "Webnetcenter" and a bio "Joined yesterday" are shown at the bottom left.

7. Le compte GIT est créer, on va créer un projet via Visual Studio 2019



8. On clique sur démarrer Windows et on sélectionne Visual Studio 2019
9. On devrait avoir cette fenêtre, on sélectionne Application Windows Forms(.net Framework). On clique sur le bouton suivant.



10. Sélectionner un nom de projet, un endroit local de l'emplacement de votre projet, cocher Placer la solution et le projet dans le même répertoire.
11. Cliquer sur le bouton Créer.

Configurer votre nouveau projet

Application Windows Forms (.NET Framework) C# Bureau Windows

Nom du projet
Projet CSharp via GIT

Emplacement
C:\Users\prof\Documents\420-2C5-LL Programmation Systeme\

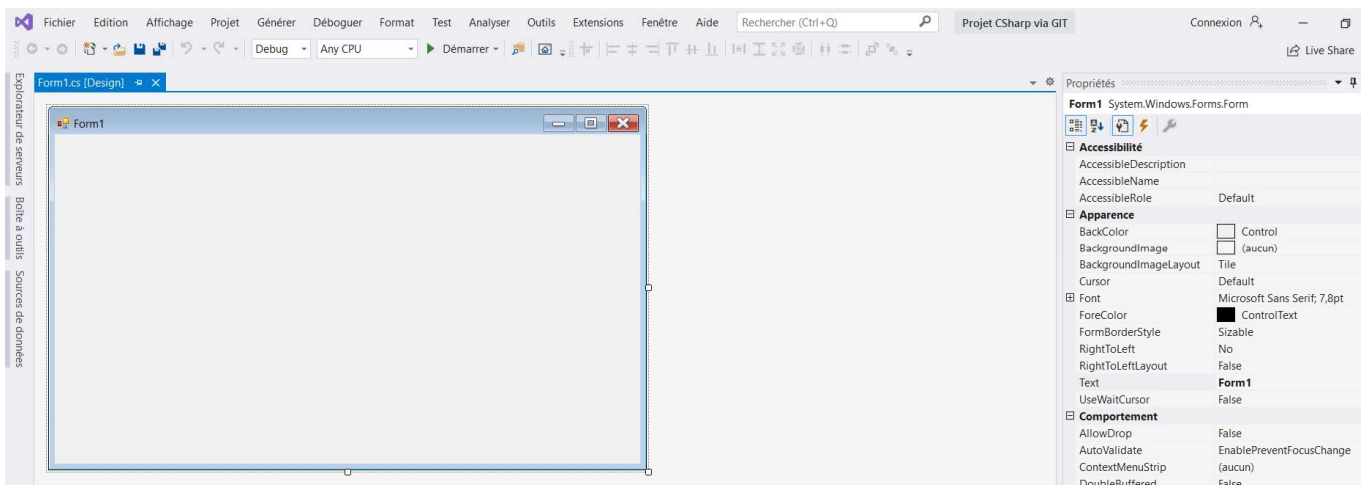
Nom de la solution
Projet CSharp via GIT

☒ Placer la solution et le projet dans le même répertoire

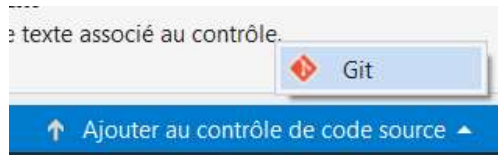
Framework
.NET Framework 4.7.2

Retour Créer

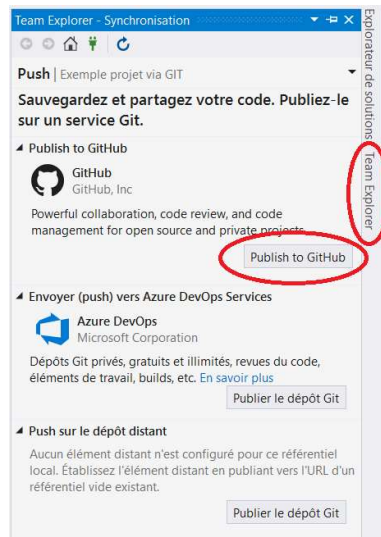
12. Votre projet est créer !



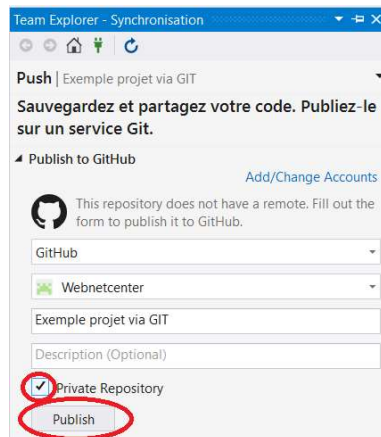
13. On doit ajouter le projet au contrôle de code source, faire Fichier/Ajouter au contrôle de code source
14. Aller dans le bas de l'écran à droite et cliquer sur Ajouter au contrôle de code source et par la suite Git



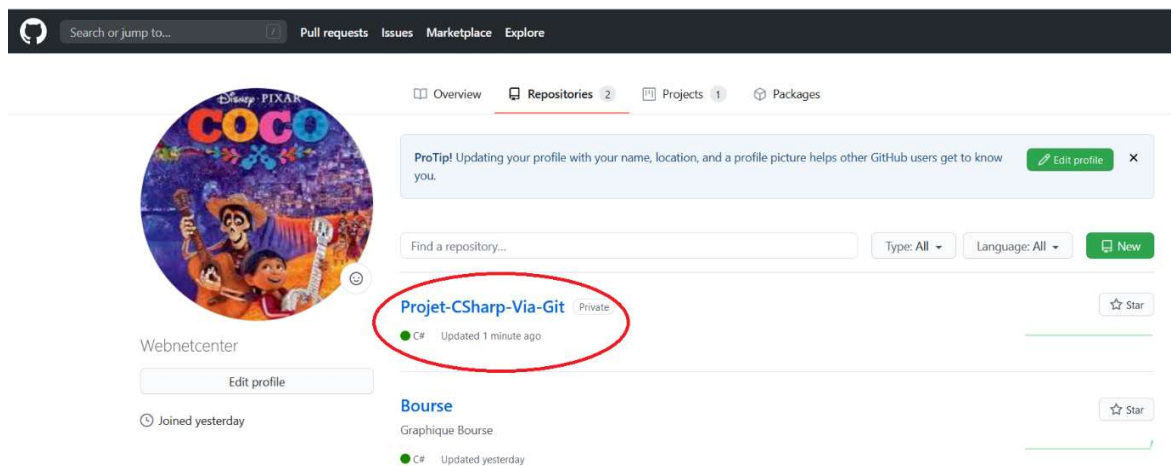
15. Sélectionner Team Exploreur et Publish to GitHub



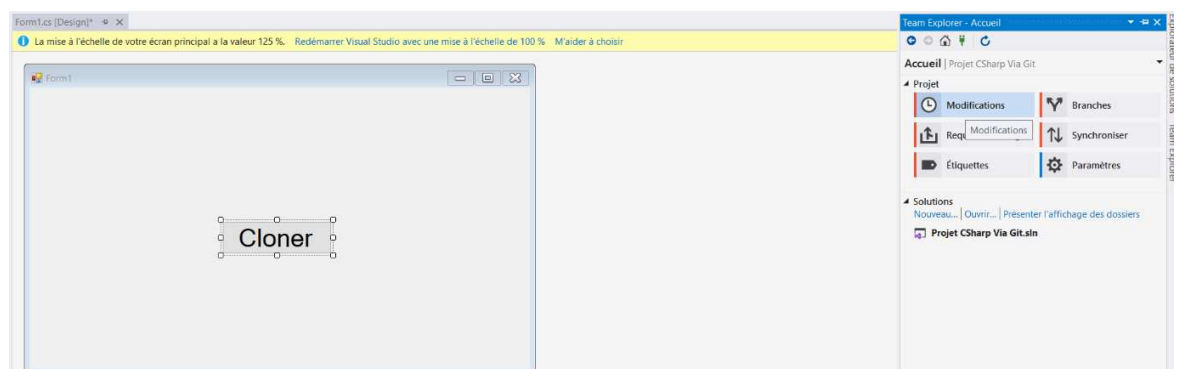
16. On se prépare à publier le projet sur GIT afin qu'il puisse être partagé en plusieurs personnes



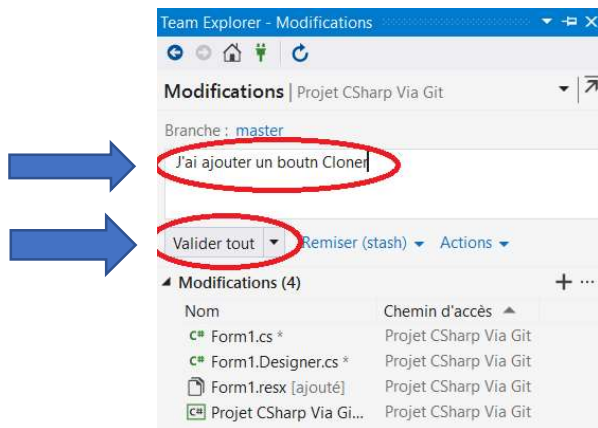
17. On va voir sur votre compte Git et votre nouveau projet a été publié



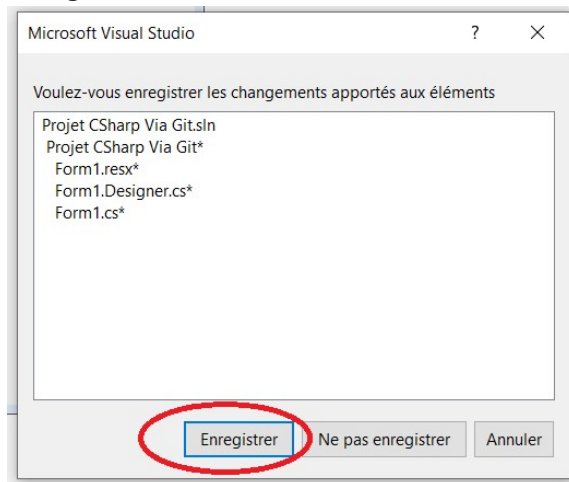
18. On reviens sur votre projet Visual Studio on ajoute un bouton Cloner et par la suite on clique sur Team Solution et l'icône  à votre droite et par la suite on clique sur modification pour enregistrer les modification sur votre Git.



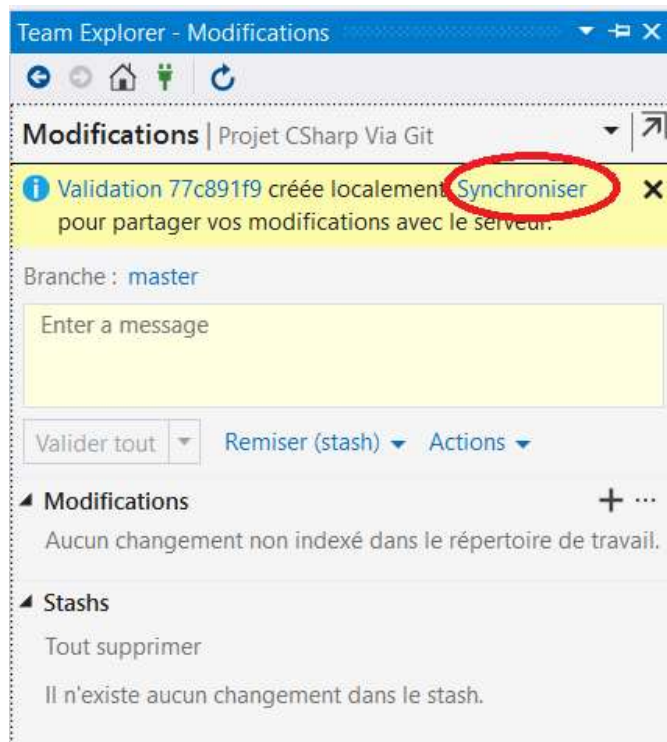
19. On ajoute un message de la modification effectuée et on clique sur Valider tout



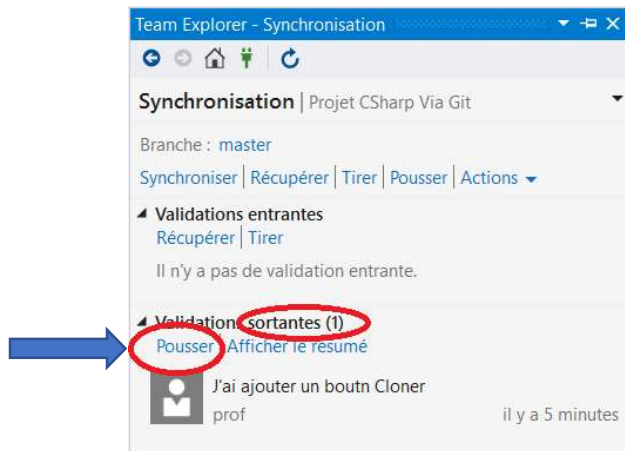
20. On vous demande si on veut enregistrer les modifications, on clique sur enregistrer



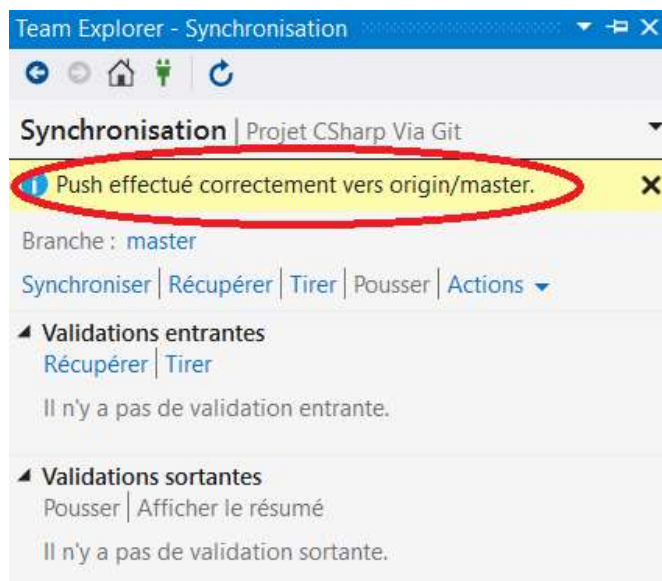
21. On clique sur synchroniser pour synchroniser les modifications via le dépôt Git



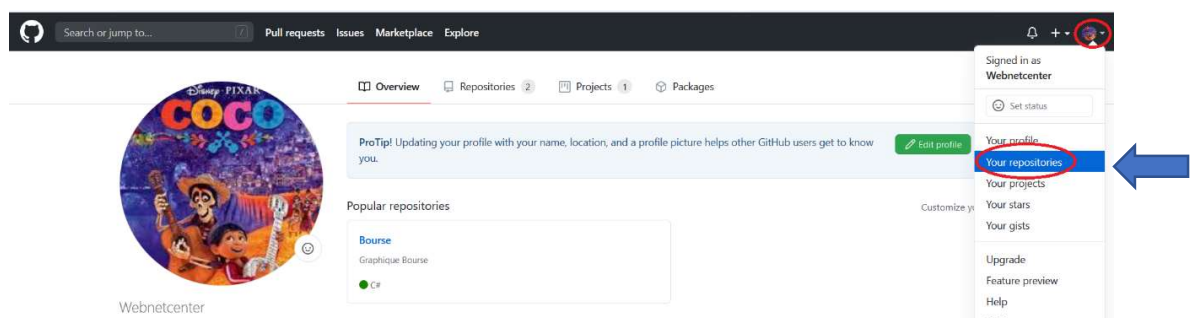
22. Par la suite on pousse les modifications sur le dépôt Git en cliquant sur Pousser



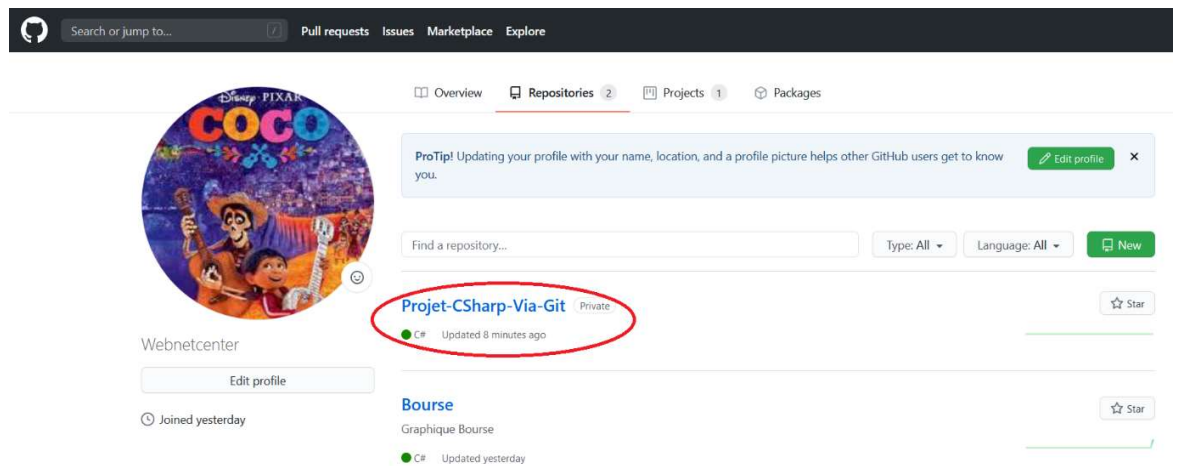
23. Le Push vers le dépôt a bien fonctionné












24. On va voir les modifications effectuées sur votre dépôt Git



25. On clique sur le Projet CSharp Via Git



26. On clique sur le fichier Form1.cs afin de voir la modification effectuée.

 prof and prof Je viens ajouter un commentaire		043e469 11 minutes ago	 4 commits
 Properties	Ajoutez des fichiers projet.	37 minutes ago	
 .gitattributes	Ajoutez les fichiers .gitignore et .gitattributes.	37 minutes ago	
 .gitignore	Ajoutez les fichiers .gitignore et .gitattributes.	37 minutes ago	
 App.config	Ajoutez des fichiers projet.	37 minutes ago	
 Form1.Designer.cs	Je viens ajouter un commentaire	11 minutes ago	
 Form1.cs	Je viens ajouter un commentaire	11 minutes ago	
 Form1.resx	J'ai ajouter un boutn Cloner	20 minutes ago	
 Program.cs	Ajoutez des fichiers projet.	37 minutes ago	

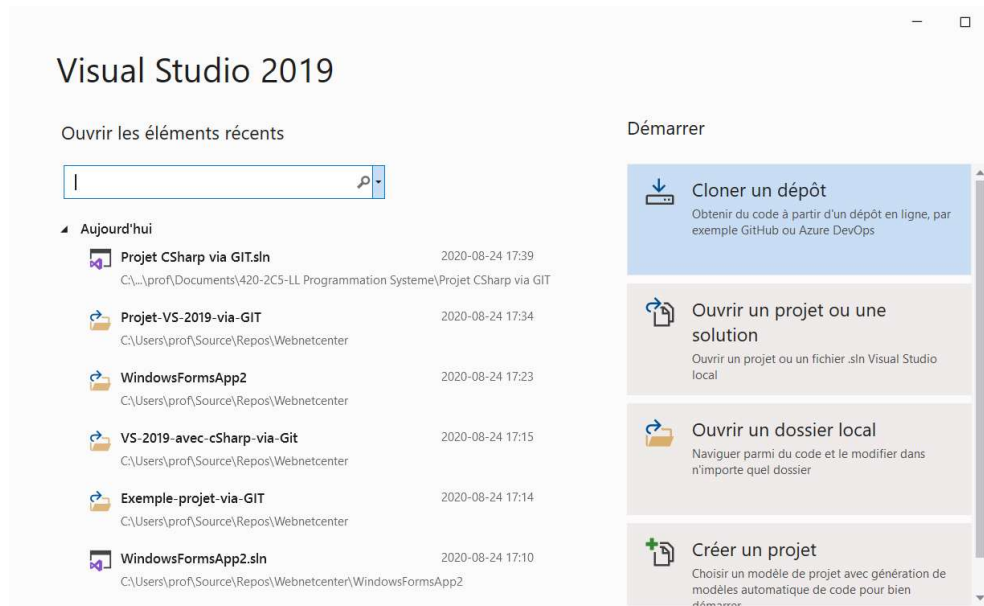
27. Voici la modification effectuée

```
25 lines (23 sloc) 501 Bytes
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Projet_CSharp_Via_Git
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19
20         private void button1_Click(object sender, EventArgs e)
21         {
22             // commentaire ajouté
23         }
24     }
25 }
```

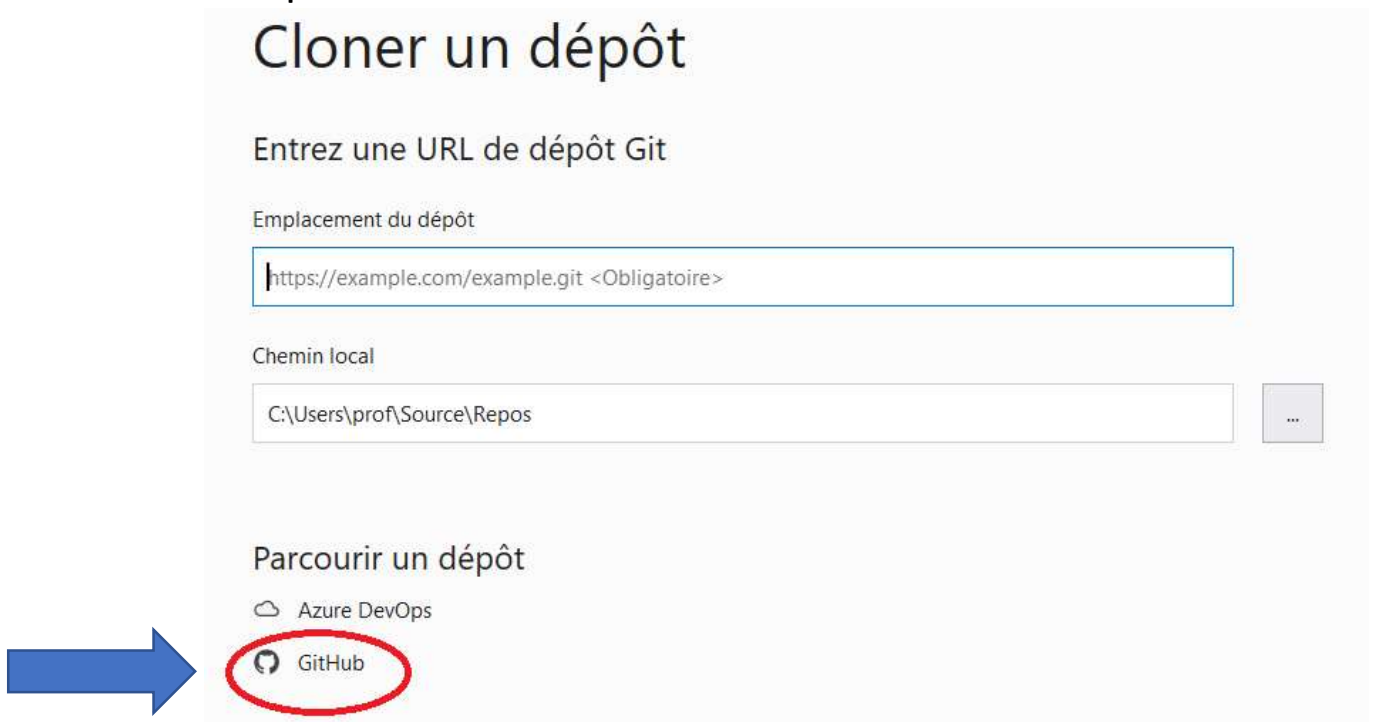
28. On ferme Visual Studio 2019

29. On efface le dossier projet de disque Local

30. On ouvre Visual Studio 2019, et on clique sur Cloner un dépôt



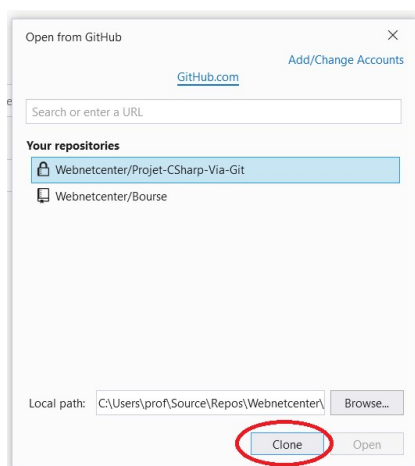
31. On clique sur GitHub



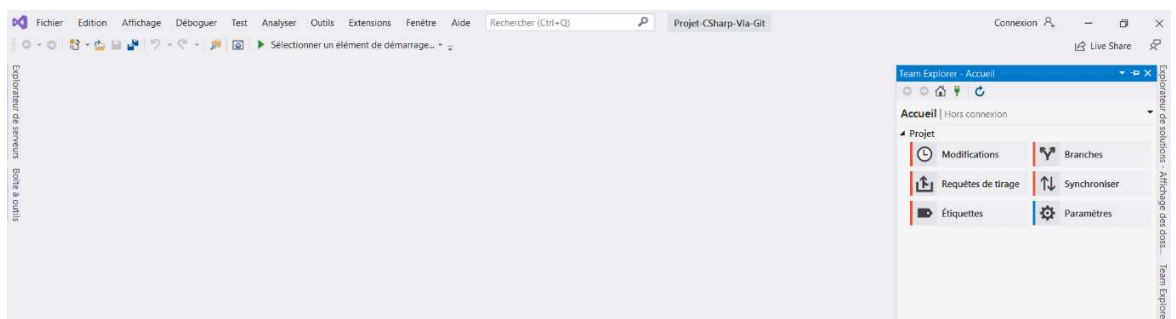
32. Si votre site Git est fermé, la fenêtre suivante va être affichée pour le login password.



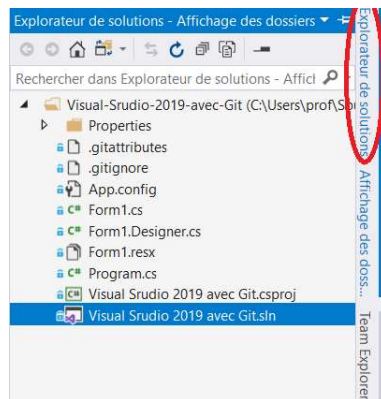
33. Sinon on peut sélectionner le projet qu'on veut ouvrir de notre dépôt Git et on clique sur Clone par la suite



34. On arrive à cette fenêtre de Visual Studio

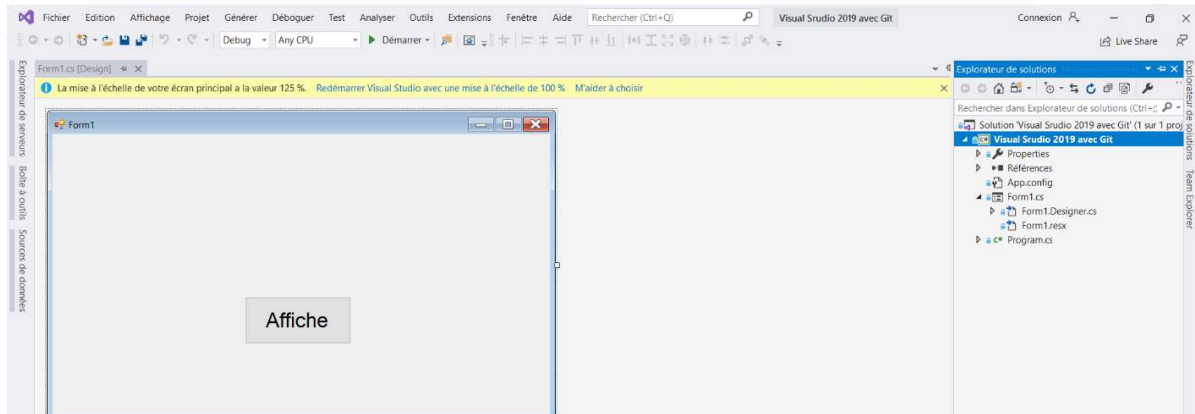


35. On clique sur explorateur de solution et Visual Studio 2019 avec Git.sln



36. Vous avez votre projet pour travailler

37.



Fin