Hands-On Assignment: Basic Git And GitHub Workflow

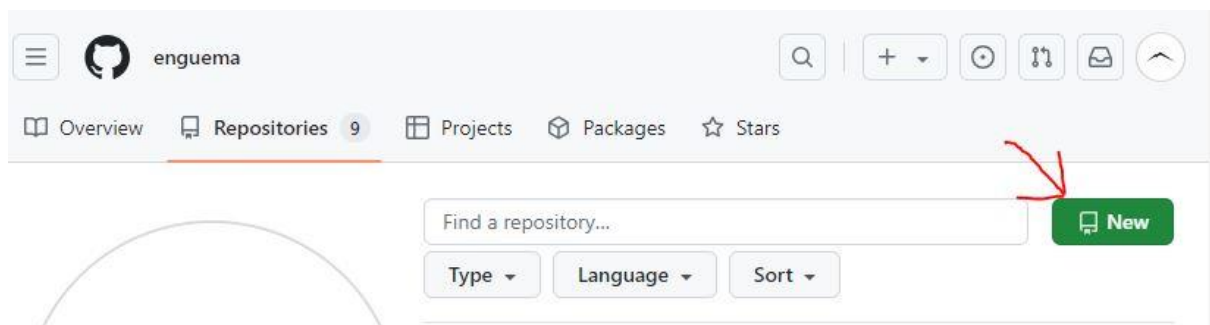**Student: Emanuel Nguema Oyono**

Task 1: Repository Setup

1. GitHub Repository Creation:
   - Log in to your GitHub account.
   - Create a new repository on GitHub (let's call it "PLPBasicGitAssignment").
   - Initialize it with a README file.

Solution:
Para crear un nuevo repositorio nos dirigimos a la pestaña "repositorios" y damos clic en "new".

Seguidamente introducimos el nombre que tendrá el nuevo repositorio y las configuraciones que deseamos:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner *   **Repository name ***

☉ enguema ▾ /   PLPBasicGitAssignment

✅ PLPBasicGitAssignment is available.

Great repository names are short and memorable. Need inspiration? How about **symmetrical-rotary-phone** ?

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:

☑ **Add a README file**
     This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. Learn more about licenses.

Task 2: Local Setup

2. Local Folder Setup:
  - Create a new folder on your local machine (e.g., "PLPBasicGitAssignment").
  - Open a terminal or command prompt and navigate to the created folder.

```
enguema@DESKTOP-ENQLOEK MINGW64 ~
$ cd ~/Desktop/Clases/plp/PLPBasicGitAssignment

enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLPBasicGitAssignment
$ |
```

3. Git Initialization:
  - Initialize a new Git repository in your local folder.

```
enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLPBasicGitAssignment
$ git init
Initialized empty Git repository in C:/Users/enguema/Desktop/Clases/plp/PLPBasic
GitAssignment/.git/
```

```
enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLPBasicGitAssignment (master)
$ |
```

4. Connecting to GitHub:
  - Link your local repository to the GitHub repository you created in Task 1

```
enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLPBasicGitAssignment (master)
$ git remote add origin https://github.com/enguema/PLPBasicGitAssignment.git
```

Task 3: Making Changes

5. Create a File:
  - Inside your local folder, create a new text file (e.g., `hello.txt`).
  - Add a simple text message (e.g., "Hello, Git!").

6. Committing Changes:
  - Stage the changes.

```
enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLPBasicGitAssignment (master)
$ git add Hello.txt

enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLPBasicGitAssignment (master)
$ git commit -m "Hello.txt with a greeting"
[master (root-commit) 6544f66] Hello.txt with a greeting
 1 file changed, 1 insertion(+)
 create mode 100644 Hello.txt

enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLPBasicGitAssignment (master)
$ |
```

Task 4: Pushing to GitHub

7. Pushing to GitHub:
  - Push the committed changes to your GitHub repository.

```
enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLPBasicGitAssignment (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 245 bytes | 30.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:        https://github.com/enguema/PLPBasicGitAssignment/pull/new/master
remote:
To https://github.com/enguema/PLPBasicGitAssignment.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Task 5: Verification

8. Verify on GitHub:

- Visit your GitHub repository in a web browser and confirm that the `hello.txt` file and commit message are visible.