

SE-Assignment-4

Student: Emanuel Nguema Oyono

Task:

GitHub and Visual Studio Instructions: Answer the following questions based on your understanding of GitHub and Visual Studio. Provide detailed explanations and examples where appropriate.

Questions:

Introduction to GitHub:

What is GitHub and what are its main functions and features? Explain how you support collaborative software development.

Solution:

GitHub is a platform where you can store, share and work together with other users to write code.

Main functions and features:

- **Version control:** GitHub uses Git to allow tracking of source code changes over time
- **Repositories-** These are containers for your code and other related files. Projects on GitHub are organized into repositories that can be; public (accessible to anyone) or private (only accessible to specific people).
- **Collaboration:** GitHub makes it easy to collaborate on projects. Users can clone repositories, make changes, and then push them.
- **Branches:** Branches allow developers to work on different features or bug fixes independently. You can create a branch for a new feature, work on it, and then merge it into the main branch once it's complete and reviewed.
- **Actions:** GitHub Actions is a continuous integration and continuous deployment (CI/CD) tool that allows you to automate tasks such as testing, builds, and deployments.
- **GitHub Pages:** This feature allows you to host static websites directly from a GitHub repository. It is useful for creating documentation, blogs or personal websites.
- **A week:** Each repository can have its own wiki, which is a space for additional documentation. This is useful for maintaining manuals, guides and other relevant information related to the project.
- **Security:** GitHub offers various security tools such as dependency scanning, vulnerability alerts, and security policies to protect code and data.

- **Integrations-** GitHub integrates with a wide variety of third-party tools and services, allowing developers to customize their workflow and improve productivity

Repositories on GitHub:

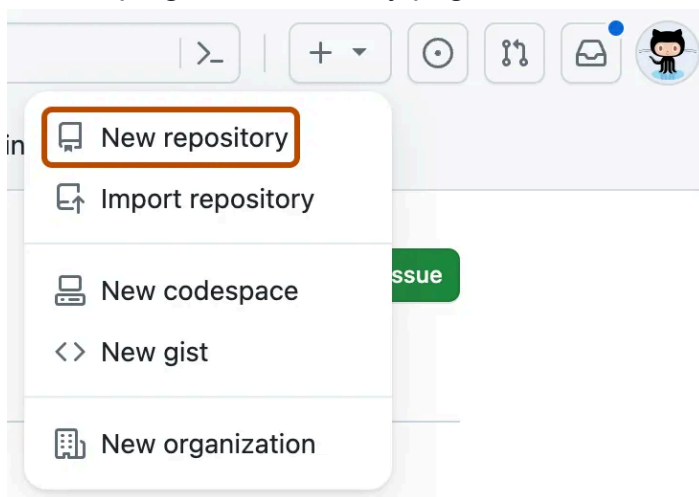
What is a GitHub repository? Describe how to create a new repository and the essential elements that should be included in it.

Solution:

A repository is the most basic element of GitHub. It is a place where you can store the code, files, and revision history of each file. Repositories can have multiple collaborators and can be public or private.

Below we show the process of creating a repository:

1. In the top right corner of any page, select and then click New Repository.



2. Enter a short, easy-to-remember name for the repository.

Owner *

Repository name *

⌵

enguema



▼

 /

Great repository names are short and memorable. Need inspiration? How about [redesigned-enigma](#) ?

Description (optional)

3. Optionally, you can add a description of the repository. For example, "My first GitHub repository."
4. Choose the visibility of the repository.

-
- ☒  **Público**
Cualquier persona en Internet puede ver este repositorio. Tú eliges quién puede contribuir.
- ☐  **Privado**
Tú eliges quién puede ver y comprometerse con este repositorio.
-

5. Select Initialize this repository with a README. And finally click on create repository.

Inicialice este repositorio con:

☒ **Agregar un archivo README**

Aquí puedes escribir una descripción detallada de tu proyecto. [Obtén más información sobre los archivos README.](#)

Agregar .gitignore


Plantilla .gitignore : Ninguna ▼

Seleccione los archivos que no desea rastrear de una lista de plantillas. [Obtenga más información sobre cómo ignorar archivos.](#)

Elija una licencia

Licencia : Ninguna ▼

Una licencia indica a los demás lo que pueden y no pueden hacer con su código. [Obtenga más información sobre las licencias.](#)

Esto establecerá  principal como rama predeterminada. Cambie el nombre predeterminado en su [configuración](#) .

 Estás creando un repositorio público en tu cuenta personal.

Crear repositorio

As for the essential elements that must be included when creating a repository, we have:

- **README.md**- Provide clear and useful information about the repository so that users and collaborators can understand and use it effectively.
- **License**- Defines the terms under which your code may be used, modified, and distributed.
- **.gitignore**: Tells Git which files and directories to ignore when a commit is made.

Version control with Git:

Explain the concept of version control in the context of Git. How does GitHub improve version control for developers?

Solution:

Version control is a system that records changes made to a file or set of files over time so that you can recover specific versions later. And, we can talk about three models of version control system; local, centralized and distributed.

GitHub improves version control through a series of features and tools (described in the first question) that make source code management more efficient, collaborative, and organized.

Branching and merging on GitHub:

What are branches in GitHub and why are they important? Describes the process of creating a branch, making changes, and merging it back into the master branch.

Solution:

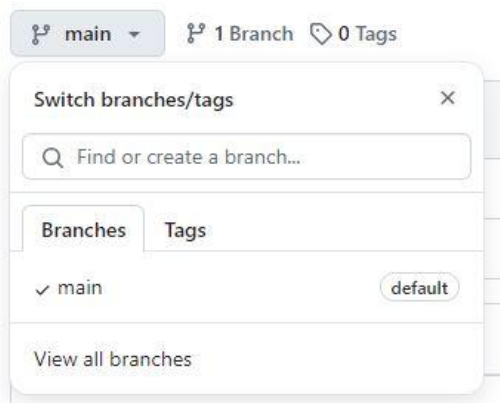
The ramifications They are central to collaboration on GitHub.

The ramifications are important for the following aspects:

- Parallel development: allows multiple teams to work on different functionalities at the same time without interfering with each other.
- Experimentation: Allows testing without affecting the main code.
- Organization: Helps keep a large, complex project organized by breaking work into smaller, more manageable branches.

Branch creation process:

1. We go to the repository.
2. In the file tree view on the left, select the branches drop-down menu, then click view all branches.



3. Click new branch.

Branches

[New branch](#)

Overview Yours Active Stale All

Q Search branches...

Default

Branch	Updated	Check status	Behind	Ahead	Pull request
main	9 minutes ago			Default	...

4. We write a name for the new branch and click create new branch.

Create a branch

New branch name

MyFirstBranch

Source

main ▼

Cancel

Create new branch

Branches

[New branch](#)

Overview Yours Active Stale All

Q Search branches...

Default

Branch	Updated	Check status	Behind	Ahead	Pull request
main	20 minutes ago			Default	...

Your branches

Branch	Updated	Check status	Behind	Ahead	Pull request
MyFirstBranch	2 minutes ago		0	0	...

Active branches

Branch	Updated	Check status	Behind	Ahead	Pull request
MyFirstBranch	2 minutes ago		0	0	...

From the CLI it would be:

git branch rama2: With this command we create a new branch named “branch2”.

Making changes to the created branch and merging it with the main branch:

1. To make changes to the new branch we must first access it as follows:

```
enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLP-My-First-Repository (main)
● $ git checkout rama2
Switched to branch 'rama2'
Your branch is up to date with 'origin/rama2'.

enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLP-My-First-Repository (rama2)
○ $
```

2. The change I am going to make is to add an "index.html" file

```
enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLP-My-First-Repository (rama2)
○ $ vim index.html
```

3. Next we add the file, commit and finally upload the changes.

```
enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLP-My-First-Repository (rama2)
● $ git add index.html
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory

enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLP-My-First-Repository (rama2)
● $ git commit -m "My web page from other branch"
[rama2 74f8236] My web page from other branch
1 file changed, 9 insertions(+)
create mode 100644 index.html

enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLP-My-First-Repository (rama2)
● $ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 450 bytes | 450.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/EnguemaDev/PLP-My-First-Repository.git
99c6352..74f8236 rama2 -> rama2
```

For fusion:

First we move to the main branch and then merge "branch2" with it.

```

enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLP-My-First-Repository (rama2)
$ git checkout main
Switched to branch 'main'
<html>
Your branch is up to date with 'origin/main'.

enguema@DESKTOP-ENQLOEK MINGW64 ~/Desktop/Clases/plp/PLP-My-First-Repository (main)
$ git merge rama2
Updating 99c6352..74f8236
Fast-forward
 index.html | 9 ++++++++
 1 file changed, 9 insertions(+)
 create mode 100644 index.html

```

Pull Requests and Code Reviews:

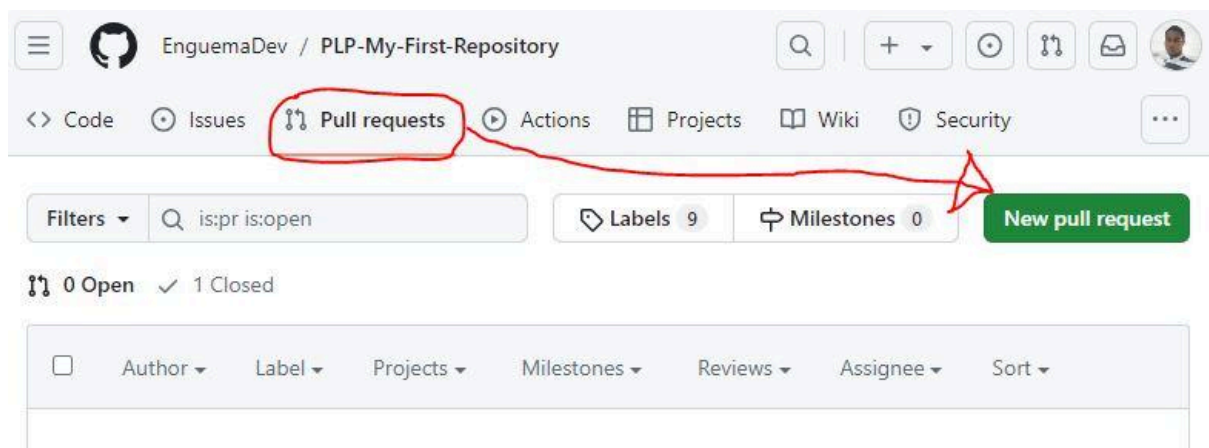
What is a pull request on GitHub and how does it facilitate code review and collaboration? Describes the steps to create and review a pull request.

Solution:

A pull request on GitHub is a proposal to merge a set of changes from one branch to another. In a pull request, collaborators can review and discuss the set of proposed changes before integrating the changes into the main code base.


Steps to create:

From our repository we go to the pull requests tab and create a new request “New pull request”



We add a title and a description. Then we click on “create pull request”

base: main ← compare: rama2 ✓ **Able to merge.** These branches can be automatically merged.

 **Add a title**

Se ha agregado estilos a la pagina web

Add a description

Write Preview H B I ≡ <> 🔗 | ≡ ≡ ≡ ...

Add your description here...

Markdown is supported Paste, drop, or click to add files

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Reviewers No reviews

Assignees No one—[assign yourself](#)

Labels None yet

Projects None yet

Milestone No milestone

Development Use [Closing keywords](#) in the description to automatically close issues

Helpful resources [GitHub Community Guidelines](#)

1 commit 1 file changed 1 contributor

Commits on Aug 5, 2024

Se ha agregado estilos a la pagina web a66798a <>

enguema committed 3 hours ago

Review of a join request:

From the “pull request” tab we can see the number of available requests.

EnguemaDev / PLP-My-First-Repository

<> Code Issues **Pull requests 1** Actions Projects Wiki Security

PLP-My-First-Repository Public Pin Unwatch 1 Fork 0 Star 0

To review the incorporation requests, we go to the tab indicated above where all the requests made to the main branch will be shown.

🔗 1 Open ✓ 1 Closed

Author ▾ Label ▾ Assignee ▾ Sort ▾

🔗 Se ha agregado estilos a la pagina web
#2 opened 19 minutes ago by EnguemaDev

Where we select the only request made.

Se ha agregado estilos a la pagina web #2

🔗 Open EnguemaDev wants to merge 1 commit into `main` from `rama2`

Conversation 0 Commits 1 Checks 0 Files changed 1

EnguemaDev commented 22 minutes ago Owner ...
My first pull request

🔗 ^ Se ha agregado estilos a la pagina web a66798a

Require approval from specific reviewers before merging
[Rulesets](#) ensure specific people approve pull requests before they're merged. Add rule ×

Continuous integration has not been set up
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request ▾ You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

GitHub Actions:

Explain what GitHub Actions are and how they can be used to automate workflows. Provide an example of a simple CI/CD pipeline using GitHub Actions.

Solution:

GitHub Actions is a continuous integration and deployment (CI/DC) platform that allows you to automate your build, test, and deployment roadmap.

In GitHub Action you can configure workflows so that they are triggered when an event occurs in the repository or manually, these workflows are defined using a YAML file that is verified in your repository.

Example of a simple CI/CD pipeline using GitHub Action:

Introduction to Visual Studio:

What is Visual Studio and what are its key features? How is it different from Visual Studio Code?

Solution:

Visual Studio is a complete integrated development environment (IDE) that can be used to write, edit, debug, and compile code; It is an effective development tool that allows you to complete the entire development cycle in one place.

The main difference between vs code and visual studio is that; vs code is a code editor while Visual Studio is an integrated development environment (IDE).

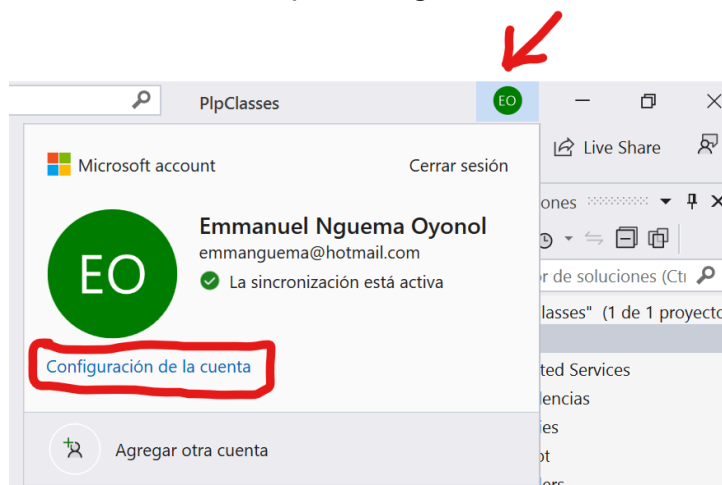
By integrating GitHub with Visual Studio:

Describe the steps to integrate a GitHub repository with Visual Studio. How does this integration improve your development workflow?

Solution:


To integrate a GitHub repository to Visual Studio we simply must link our GitHub account to Visual Studio as I show you below:

1. With Visual Studio open, we go to:



The “account settings” option takes us to the next screen where we will add our GitHub account. To do this, if it is the first time, we will be redirected to GitHub to authenticate. After successful authentication our account will be added.

Cuenta de personalización



Emmanuel Nguema Oyonol
emmanguema@hotmail.com

[Administrar perfil de Visual Studio](#)

[Opciones de la cuenta](#)

[Cerrar sesión](#)


La cuenta de personalización se usa para aplicar la itinerancia de la configuración entre dispositivos y para proporcionar otros servicios personalizados.

Visual Studio

Community 2022


Licencia: Clave de producto aplicada

Todas las cuentas




Microsoft account
emmanguema@hotmail.com

+ Agregar ▾



Microsoft

Cuenta profesional, educativa o personal




GitHub

Cuenta de GitHub

17.6.5

Cerrar

Cuenta de personalización



Emmanuel Nguema Oyonol
emmanguema@hotmail.com

[Administrar perfil de Visual Studio](#)

[Opciones de la cuenta](#)

[Cerrar sesión](#)


La cuenta de personalización se usa para aplicar la itinerancia de la configuración entre dispositivos y para proporcionar otros servicios personalizados.

Visual Studio


Community 2022

Licencia: Clave de producto aplicada

Todas las cuentas



GitHub
enguema



Microsoft account
emmanguema@hotmail.com

+ Agregar ▾

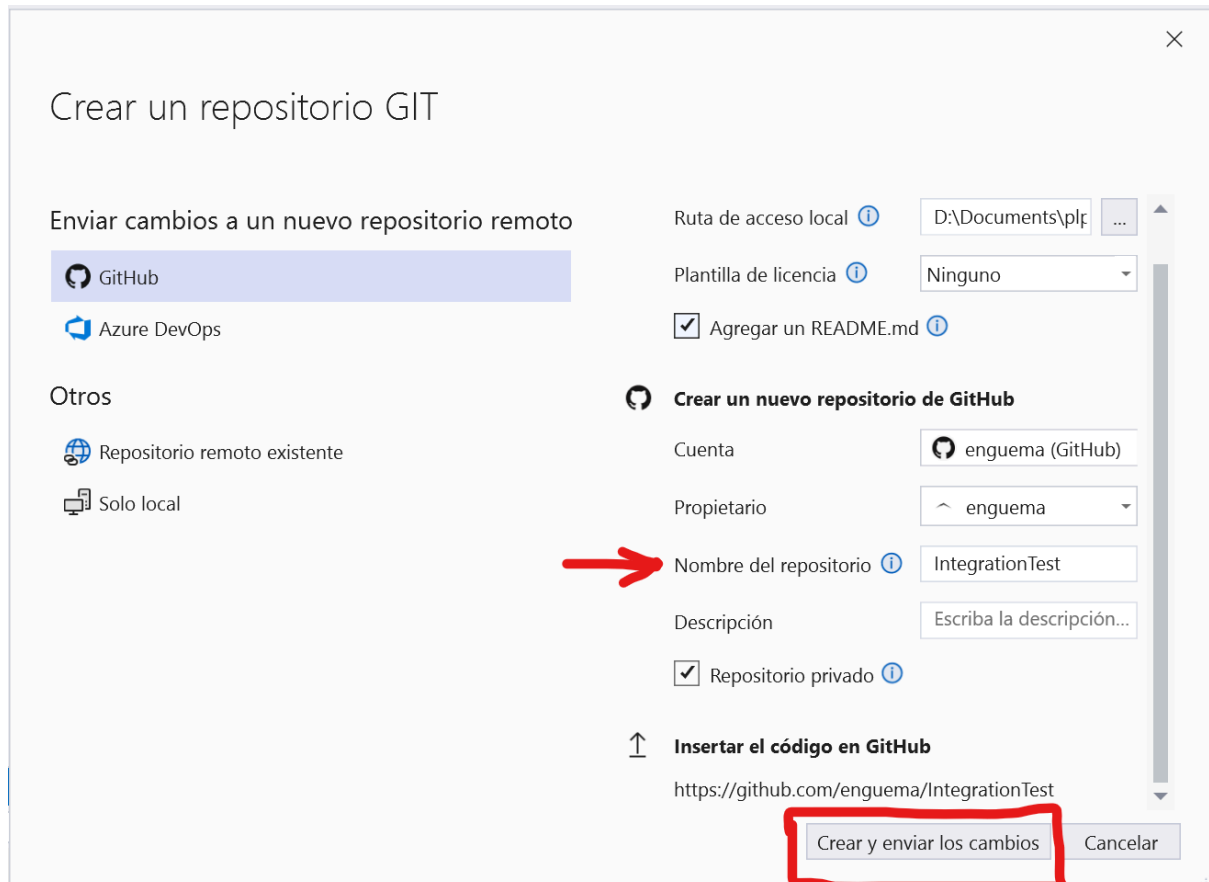
Quitar

Quitar

17.6.5

Cerrar

After linking our GitHub account to Visual Studio we can now integrate our repository that we are going to create next.



This integration helps in efficiency because all operations regarding the repository can now be done from Visual Studio without having to go to another external tool.

Debugging in Visual Studio:

Explain the debugging tools available in Visual Studio. How can developers use these tools to identify and fix problems in their code?

Solution:

Among the main debugging tools in Visual Studio are:

- Breakpoints.
- Execution window
- Local variables window: Displays the current value of the variables in the current scope.
- Call stack window

sources used:

<https://docs.github.com/es>

<https://git-scm.com/book/en/v2>

<https://visualstudio.microsoft.com/es/vs/github/>

