

**E-ACTIVIDAD 4.1 USO DEL REPOSITORIO GITHUB PARA CONTROL DE
VERSIONES**

Universidad Valle del Momboy (UVM)

Facultad de ingeniería (Computación)

T.S.U. María Molina Zambrano

El Vigía, estado Mérida, Venezuela

Noviembre, 2025

Contenido

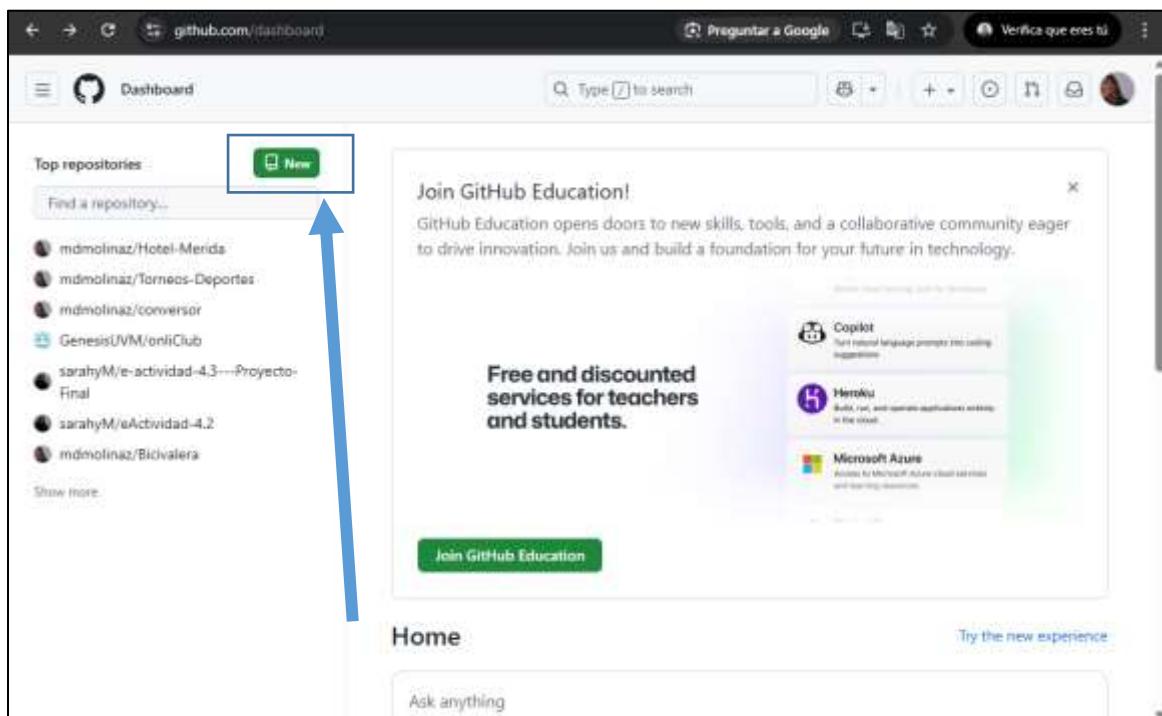
E-ACTIVIDAD 4.1 USO DEL REPOSITORIO GITHUB PARA CONTROL DE VERSIONES

.....	3
Paso 1: Crear el repositorio	3
Paso 2: Clonar el repositorio localmente	5
Paso 3: Crear la estructura	5
Paso 4: crear README.md	6
Paso 5: Crear CHANGELOG.md	7
Paso 6: Crear Issues	8
Paso 7: Configurar Project Board.....	9
Paso 8: Crear el archivo principal del manual	11
Paso 9: Primer commit - Estructura inicial.....	12
Paso 10: Trabajar en la primera sección	12
Paso 11: Crear Pull Request.....	13
Paso 12: Actualizar Project Board e Issues.....	13
Paso 13: Crear primera versión	13
Paso 14: Actualizar CHANGELOG.md.....	14
Paso 15: Generar PDF.....	14

E-ACTIVIDAD 4.1 USO DEL REPOSITORIO GITHUB PARA CONTROL DE VERSIONES

Paso 1: Crear el repositorio

- Ve a GitHub.com
- Haz clic en "New "



- Llena los datos:

Repository name: manual-buenas-practicas-universitarias

Description: "Manual individual de buenas prácticas para trabajos universitarios"

Public (o private según requiera tu profesor)

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#)

Required fields are marked with an asterisk (*).

1 General

Owner * mdmolinaz / Repository name * manual-buenas-practicas-universitarias is available.

Great repository names are short and memorable. How about [fictional-tribble](#)?

Description 66 / 350 characters

2 Configuration

Choose visibility * Public

Choose who can see and commit to this repository

Add README Off

READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore

No .gitignore

- Haz clic en "Create repository"

Description 66 / 350 characters

2 Configuration

Choose visibility * Public

Choose who can see and commit to this repository

Add README Off

READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore

No .gitignore

Add license

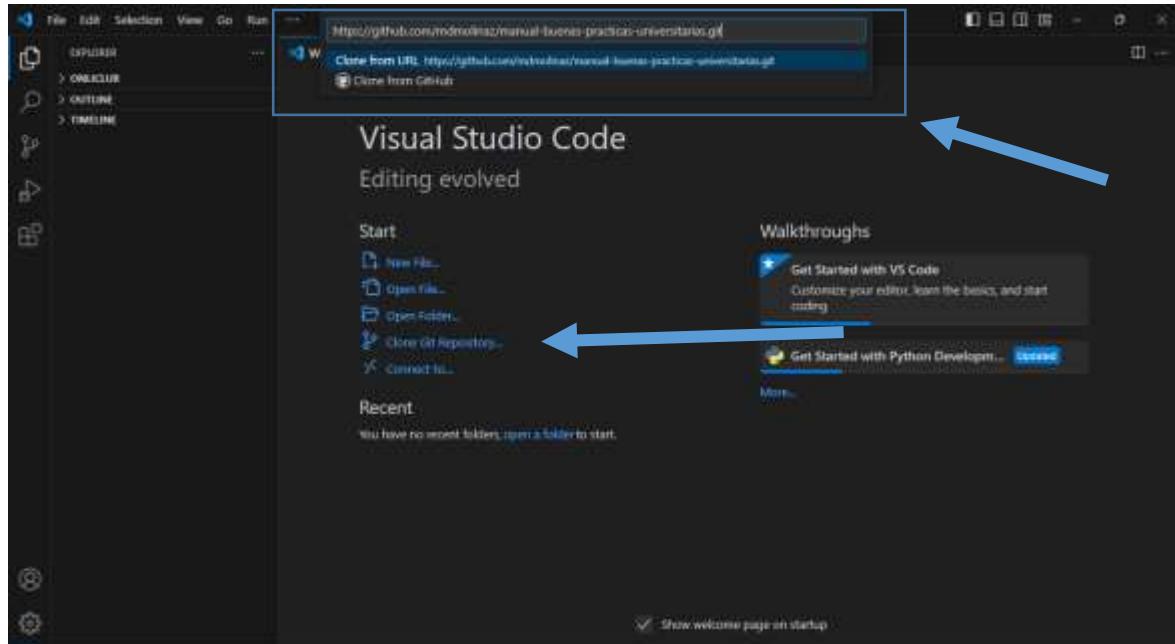
No license

Create repository

© 2025 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Community](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

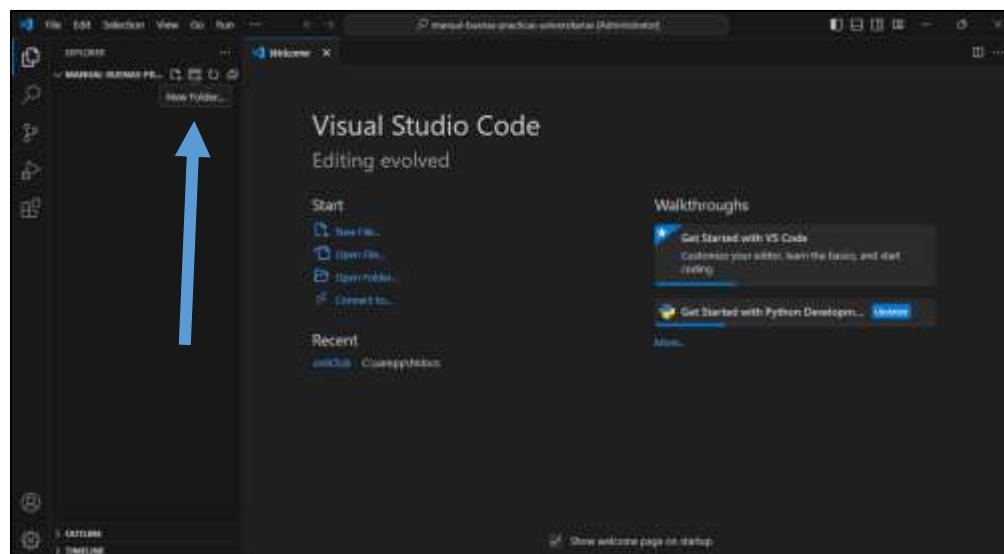
Paso 2: Clonar el repositorio localmente

- Abrimos visual studio code y le damos “clone git repository”, y en la parte de arriba colocamos el enlace del repositorio que creamos previamente.

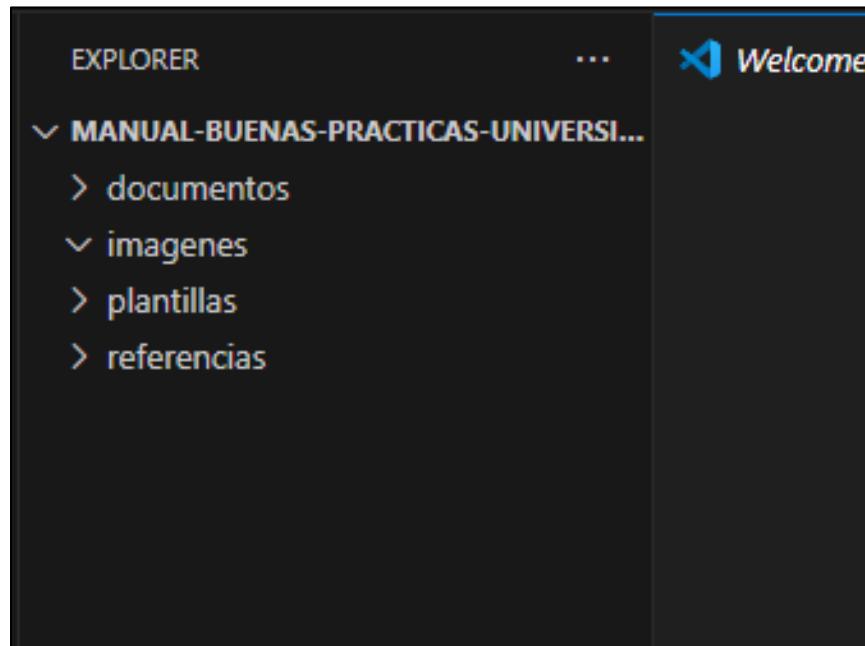


Paso 3: Crear la estructura

Debemos crear las carpetas correspondientes dentro del proyecto, lo podemos hacer directamente en visual studio code, dando en la opción “new folder”

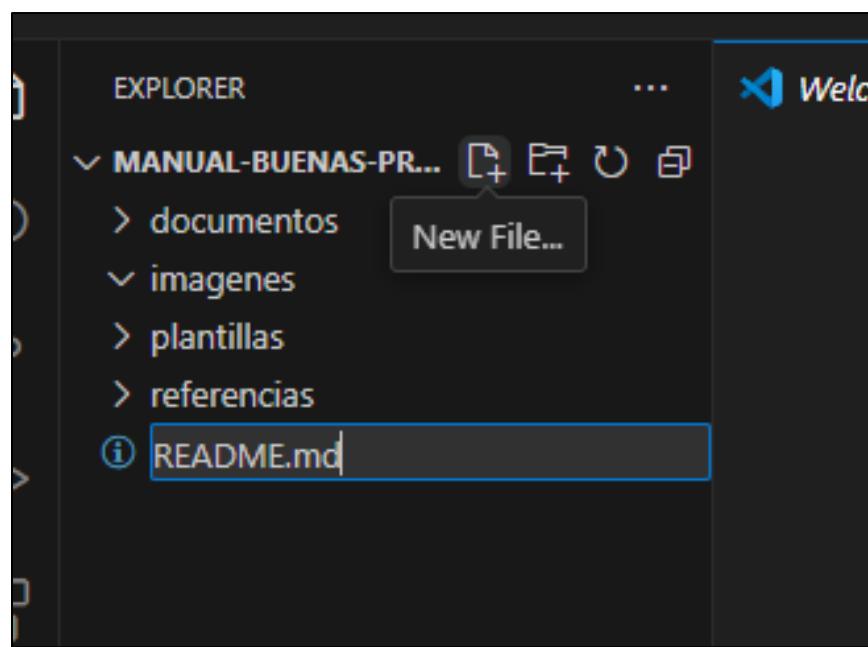


Creamos las carpetas que consideremos necesarias:



Paso 4: crear README.md

Creamos README.md dando en la opción “new file” y colocando el nombre correspondiente



Luego de haber creado el archivo, vamos a seleccionarlo y escribir en el editor lo que queremos que tenga de contenido:

```

# README.md - Manual de Buenas Prácticas para Trabajos Universitarios

## Descripción
Este manual recopila buenas prácticas, consejos y metodologías para realizar trabajos universitarios de manera efectiva.

## Objetivo
Servir como guía de referencia para organizar, planificar y ejecutar trabajos académicos.

## Estructura del Manual
- [x] 1. Organización Personal
- [ ] 2. Gestión de Tiempo
- [ ] 3. Herramientas Digitales
- [ ] 4. Formatos de Documentos
- [ ] 5. Normas de Citación APA
- [ ] 6. Presentaciones Efectivas
- [ ] 7. Estructura de Informes

## Cómo Usar Este Manual
1. Consulta las secciones según tu necesidad
2. Adapta las plantillas a tus proyectos
3. Mantén el manual actualizado con nuevas experiencias

## Contacto
[Maria Molina] <[molina@universidad.edu.es]>

```

Paso 5: Crear CHANGELOG.md

También crearemos un changelog.md, tal cual como lo hicimos, seleccionando “new file”:

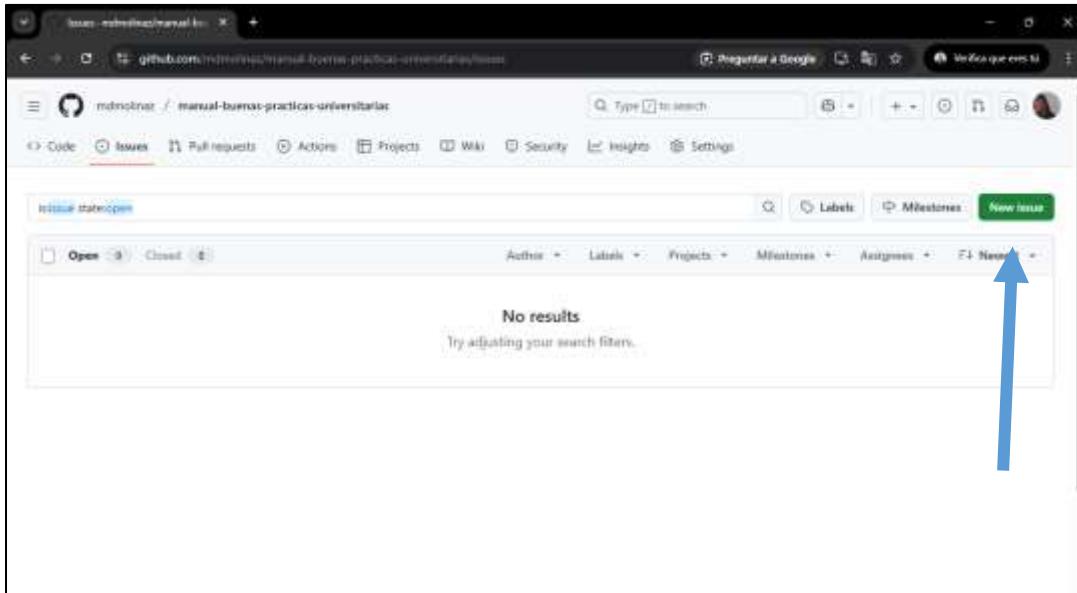
```

# CHANEGLOG.md - Registro de Cambios
## Registro de Cambios
- [v2.0] - FECHA_ACTUAL
    - Plantillas
        - Estructura inicial del manual
        - Definición de secciones
        - Configuración del repositorio

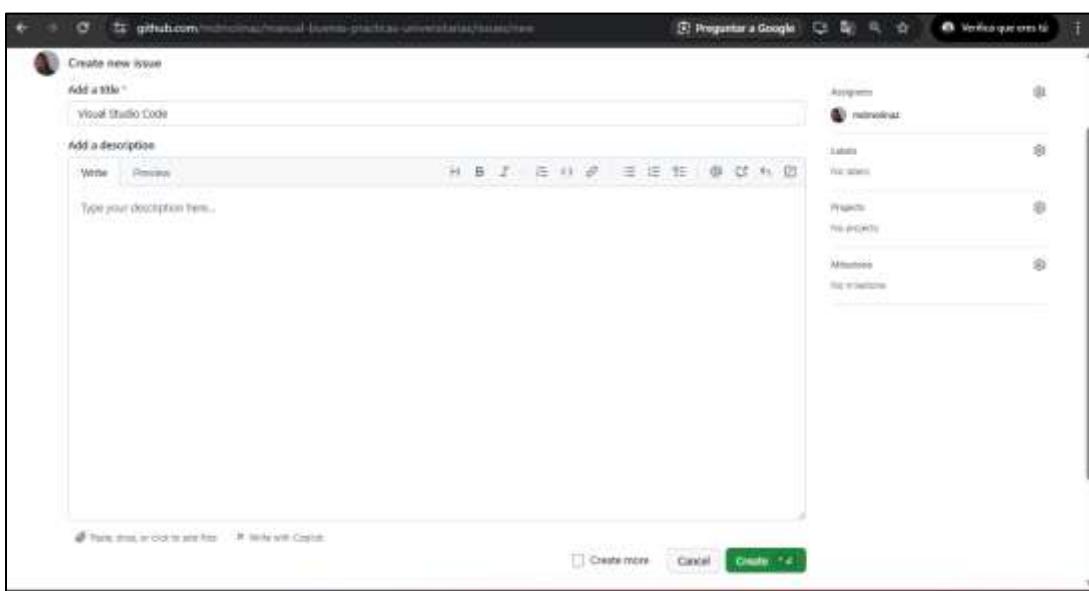
```

Paso 6: Crear Issues

Vamos a github y seleccionamos “new issue”

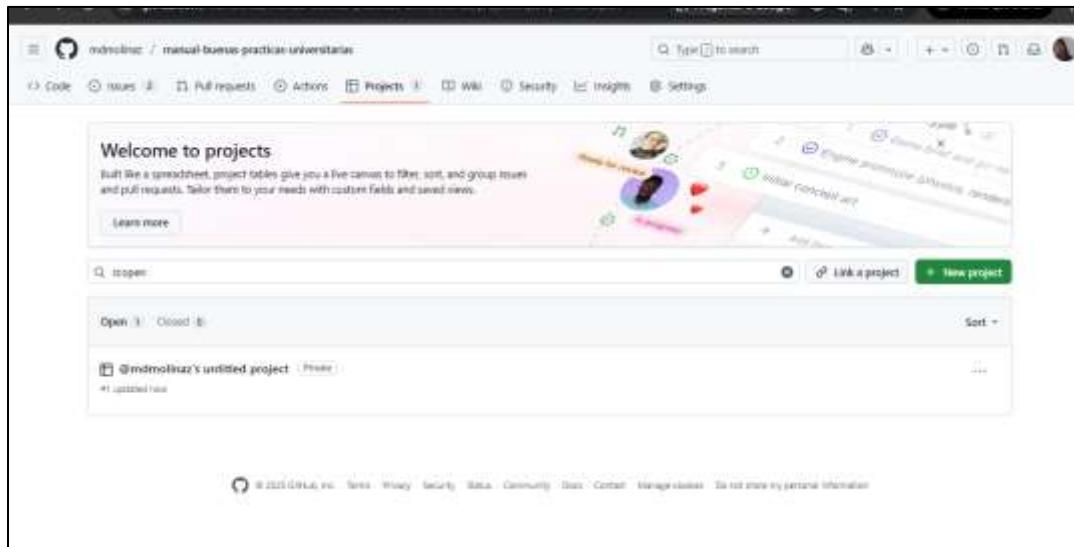


Luego colocamos el título del “issue” o asunto, nos lo asignamos a nosotros mismos o a quien corresponda y le damos a “crear”

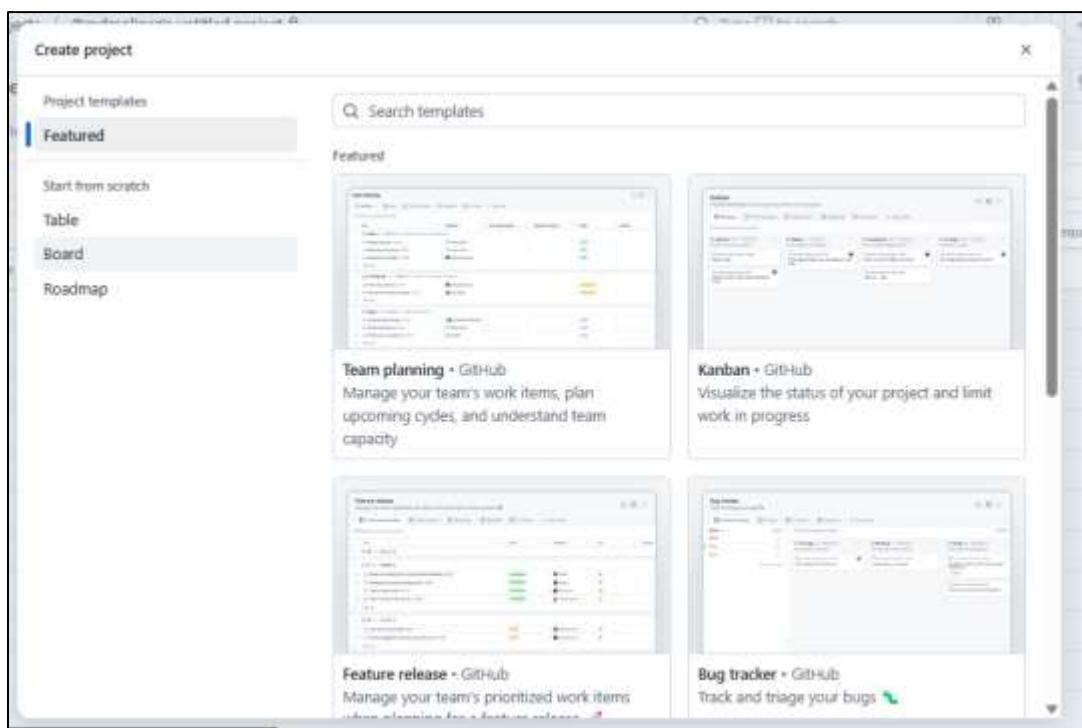


Paso 7: Configurar Project Board

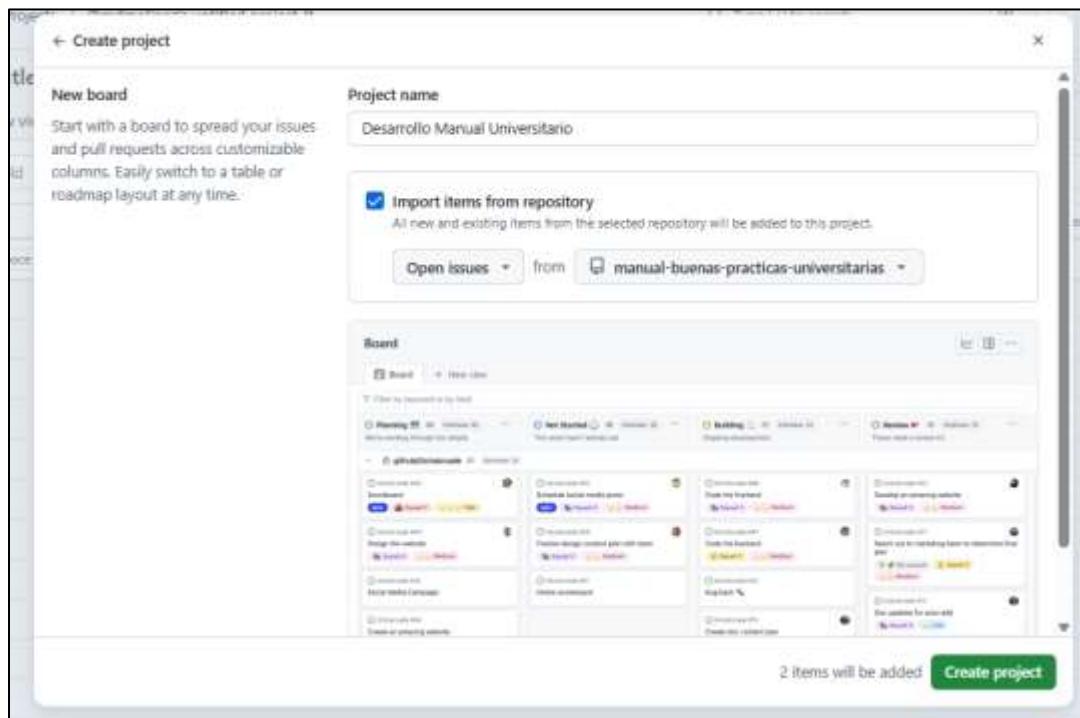
Ve a "Projects" y luego "New project"



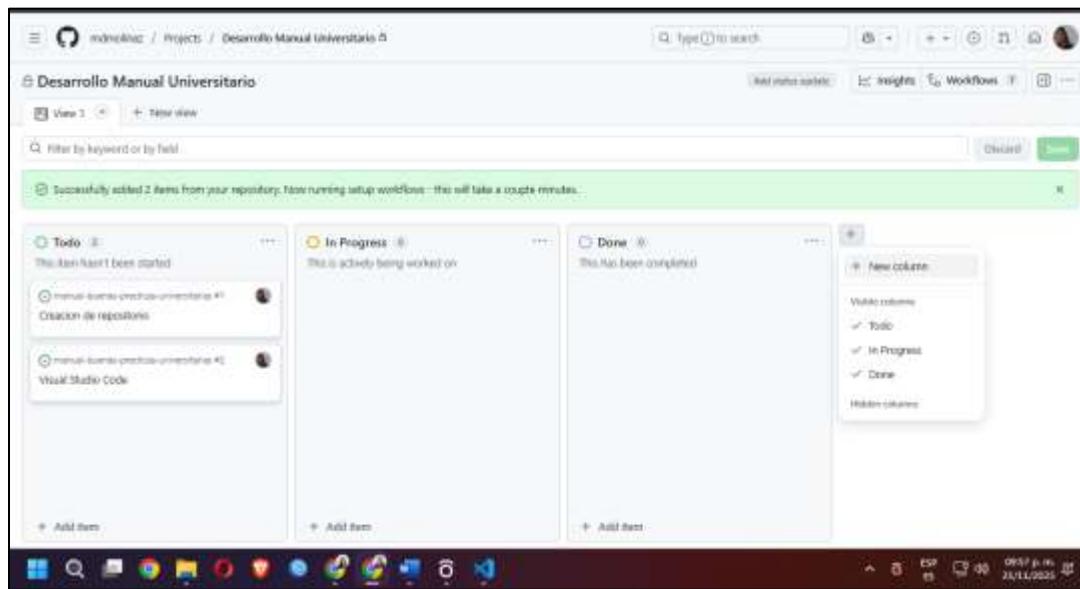
Elige "Board" (template)



Se debe nombrar (ejemplo): "Desarrollo Manual Universitario"



Crea estas columnas:



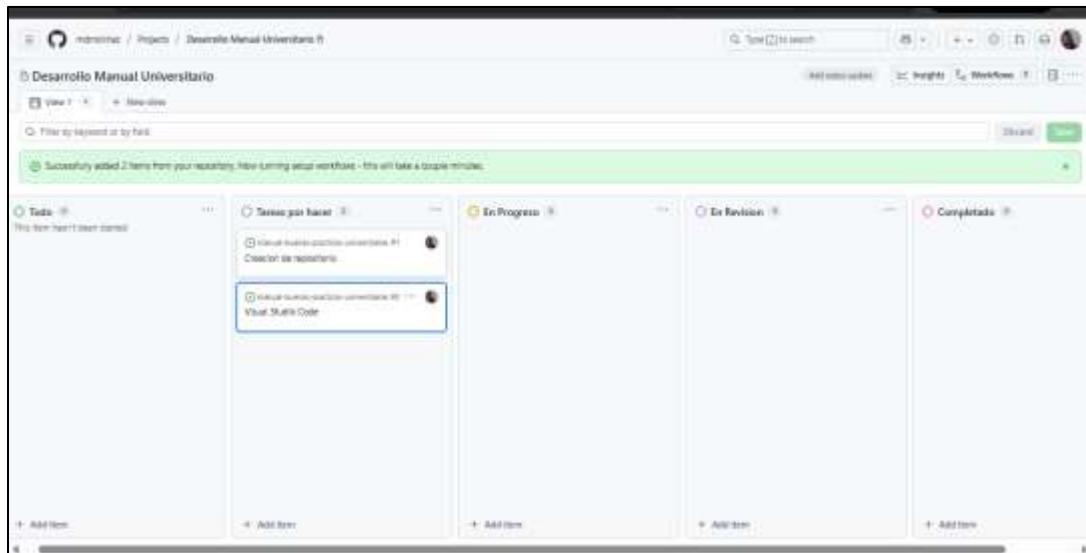
Backlog: Tareas por hacer

En Progreso: Trabajando actualmente

En Revisión: Para auto-revisión

Completado: Terminado

Arrastra los Issues a "Tareas por hacer"



Paso 8: Crear el archivo principal del manual

En la carpeta documentos, crea manual.md:

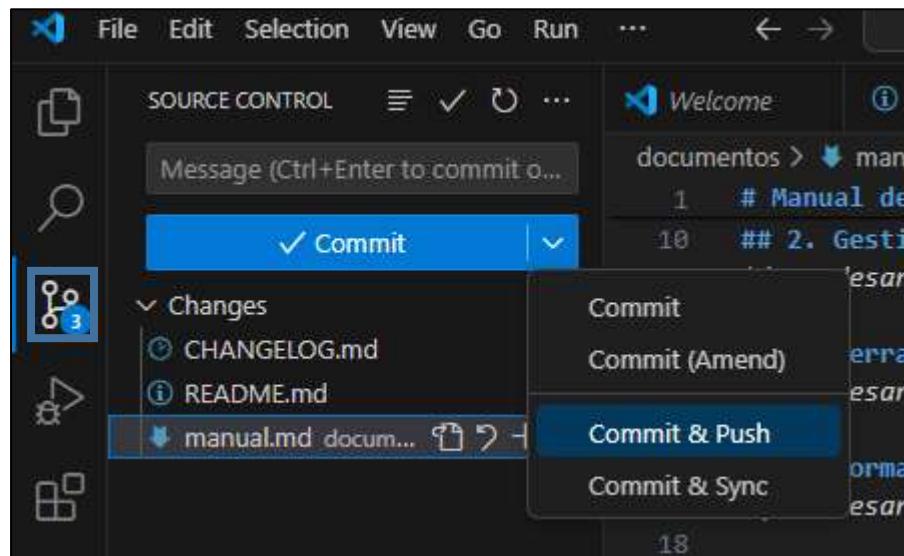
```

#> I. Manual de Buenas Prácticas Universitarias
#>   *por | Tu Nuevo |
#>
#>   #> I. Organización Personal
#>   #>   *(Esta sección está en desarrollo)*
#>
#>   #> II. Gestión de Tiempos
#>   #>   *(Por desarrollar)*
#>
#>   #> III. Herramientas Digitales
#>   #>   *(Por desarrollar)*
#>
#>   #> IV. Formatos de documentos
#>   #>   *(Por desarrollar)*
#>
#>   #> V. Normas de Citación APA
#>   #>   *(Por desarrollar)*
#>
#>   #> VI. Presentaciones Efectivas
#>   #>   *(Por desarrollar)*
#>
#>   #> VII. Estructura de Informes
#>   #>   *(Por desarrollar)*

```

Paso 9: Primer commit - Estructura inicial

Vamos a “source control” que es donde se esta señalando, allí tendremos los cambios realizados, que están pendientes por actualizar, escribimos un mensaje y damos en “commit & push”



Vemos que luego de esto en nuestro github vemos el commit realizado:

The screenshot shows a GitHub commit history for a repository. It displays one commit on the 'main' branch made on May 21, 2023, with the message 'Estructura de carpetas' and a note that it was 'automatically committed now'.

Paso 10: Trabajar en la primera sección

Editamos documentos/manual.md

Guardamos y hacemos “commit”

Paso 11: Crear Pull Request

- Vamos a "Pull requests" y damos en "New pull request"
- Base: main Compare: seccion-organizacion
- Título: "Añade sección de organización personal"
- Descripción: "Esta PR añade la primera sección completa del manual sobre organización personal"
- Hacemos clic en "Create pull request"
- Auto-revisión: Hay que revisarlo, comentar si algo puede mejorar
- Merge: Hacemos clic en "Merge pull request"

Paso 12: Actualizar Project Board e Issues

- En el Project Board, mueve el Issue #1 a "Completado"
- Cierra el Issue #1 con un comentario: "Sección completada y mergeada a main"

Paso 13: Crear primera versión

Cuando tengamos varias secciones completas:

```
# Volver a main y actualizar
git checkout main
git pull origin main
# Crear tag de versión
git tag -a v1.0 -m "Versión inicial con 3 secciones completas"
git push origin v1.0
```

```

PS D:\Mis documentos\UMV\III 2025C\Proyecto de sistema Paolini\actividad 4\manual-buenas-prácticas-universitarias> # Volver a main y
actualizar
>> git checkout main
>> git pull origin main
>>
>> # Crear tag de versión
>> git tag -a v1.0 -m "Versión inicial con 3 secciones completas"
>> git push origin v1.0
Already on 'main'
Your branch is up to date with 'origin/main'.
From https://github.com/mdmolinaz/manual-buenas-prácticas-universitarias
 * branch            main      -> FETCH_HEAD
Already up to date.
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 189 bytes | 189.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/mdmolinaz/manual-buenas-prácticas-universitarias.git
 * [new tag]         v1.0 -> v1.0
PS D:\Mis documentos\UMV\III 2025C\Proyecto de sistema Paolini\actividad 4\manual-buenas-prácticas-universitarias>

```

Paso 14: Actualizar CHANGELOG.md

```

# Registro de Cambios
## [v1.0] - 21/11/2025
### Añadido
- Sección 1: Organización Personal
- Sección 2: Gestión de Tiempos
- Sección 3: Herramientas Digitales
- Estructura completa del repositorio
### Cambiado
- README.md con instrucciones actualizadas

```

Paso 15: Generar PDF

Podemos guardar el manual directamente de Word a pdf y lo cargamos en la carpeta de “documentos”.