

**UNIVERSIDAD PERUANA LOS ANDES**

**FACULTAD INGENIERIA INGENIERÍA DE  
SISTEMAS Y COMPUTACIÓN**



**MANUAL**

**CREAR TU REPOSITORIO EN GITHUB  
DESDE VISUAL STUDIO**

**ASIGNATURA:** BASE DE DATOS II

**DOCENTE:** FERNÁNDEZ BEJARANO RAUL

**ESTUDIANTE:** Bonifacio Hilario Erick

**CÓDIGO:** S01238F

**HUANCAYO-2025**

**Introducción**

GitHub es una plataforma que permite almacenar, compartir y gestionar proyectos de programación usando **control de versiones** con Git. Crear un repositorio en GitHub te permite mantener un historial de tus archivos, colaborar con otras personas y acceder a tu trabajo desde cualquier lugar. Este manual te guiará paso a paso para crear tu cuenta en GitHub, generar un repositorio y subir tu primer proyecto desde Visual Studio, así como actualizar tu proyecto con los últimos cambios.

## 1. Objetivo

Aprender a crear un repositorio en GitHub, subir un proyecto desde Visual Studio y actualizarlo correctamente usando comandos de Git.

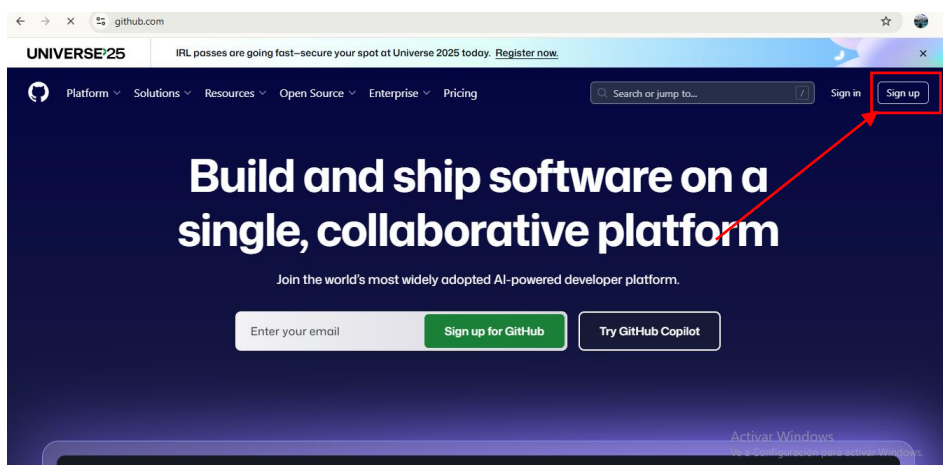
## 2. Requisitos

- Tener una cuenta de correo electrónico activa.
- Visual Studio instalado.
- Git instalado y configurado en tu computadora.

## 3. Procedimiento

### Paso 1: Crear tu cuenta en GitHub

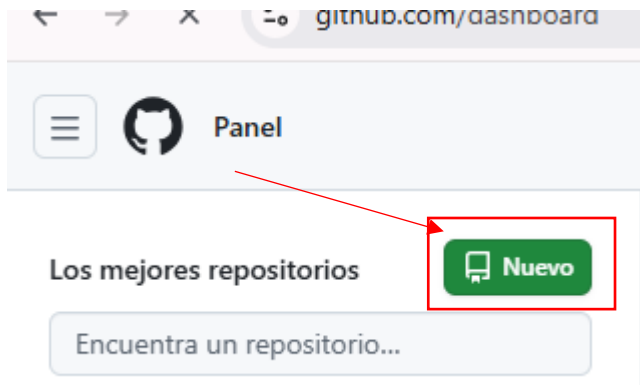
1. Abre el navegador y ve a <https://github.com>.
2. Haz clic en **Sign up** (Registrarse).
3. Ingresa tu correo electrónico, nombre de usuario y contraseña.
4. Confirma tu cuenta siguiendo el correo que GitHub te enviará.



### Paso 2: Crear un nuevo repositorio en GitHub

1. Inicia sesión en GitHub.

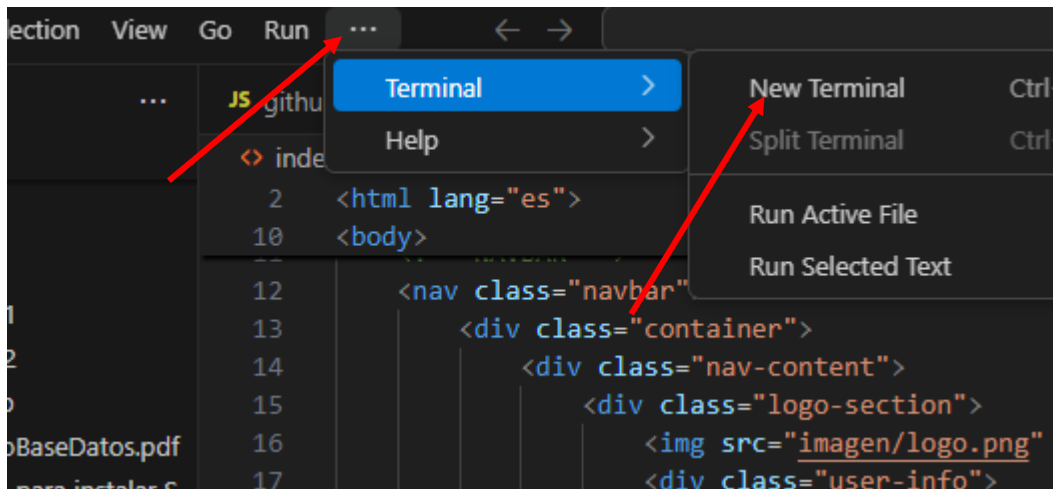
2. Haz clic en el botón **+** en la esquina superior derecha y selecciona **New repository** (Nuevo repositorio).



3. Asigna un nombre a tu repositorio.
4. Opcional: agrega una descripción.
5. Selecciona si quieres que sea **Público** o **Privado**.
6. Haz clic en **Create repository** (Crear repositorio).

A screenshot of the GitHub 'Create new repository' form. It has two tabs: '1 General' and '2 Configuración'. The 'General' tab is active. It contains fields for 'Dueño \*' (owner, 'Erick BH'), 'Nombre del repositorio \*' (repository name), and 'Descripción' (description, with a 0/350 character limit). Below the 'General' tab is the '2 Configuración' tab. It contains three sections: 'Elija visibilidad \*' (visibility, with a dropdown set to 'Público'), 'Añadir README' (add README, with a toggle set to 'Apagado'), and 'Agregar .gitignore' (add .gitignore, with a dropdown set to 'Sin .gitignore'). At the bottom right of the 'Configuración' section is a green button labeled 'Crear repositorio'.

1. Paso 3: Abrir la terminal en Visual Studio
2. Abre tu proyecto en Visual Studio.
3. En el menú superior, selecciona **Ver → Terminal** o presiona **Ctrl + `** (tecla debajo de Esc).
4. Se abrirá la terminal integrada en Visual Studio, lista para ejecutar comandos de Git.



## Paso 4: Subir tu proyecto a GitHub por primera vez

1. Navega a la carpeta de tu proyecto:

```
cd ruta/de/tu/proyecto
```

2. Inicializa Git en tu proyecto:

```
git init
```

3. Agrega todos los archivos al control de versiones:

```
git add .
```

4. Haz tu primer commit:


```
git commit -m "Primer commit"
```

5. Conecta tu proyecto con el repositorio de GitHub (reemplaza TU\_URL\_DEL\_REPOSITORIO por la URL de tu repositorio):

```
git remote add origin TU_URL_DEL_REPOSITORIO
```

6. Sube tu proyecto al repositorio:

```
git push -u origin main
```

<a href="#">←</a> <a href="#">→</a> <a href="#">🔄</a> <a href="#">github.com/ErickBH/BASE-DE-DATOS</a>		
<a href="#">principal</a>	<a href="#">1 sucursal</a>	<a href="#">0 etiquetas</a>
<input type="text" value="Q Ir al archivo"/>	<a href="#">t</a>	<a href="#">Agregar archivo</a>
<a href="#">&lt;&gt; Código</a>		
<div>  <span>Erick BH</span> <span>Actualizar datos de semanas</span> <span>✓</span> </div> <div> <span>97e431c · hace 7 horas</span> <span>🕒 85 confirmaciones</span> </div>		
<div>📁 API</div>	<div>Compromiso inicial con token seguro</div>	<div>la semana pasada</div>
<div>📁 CSS</div>	<div>actualizar cambios</div>	<div>la semana pasada</div>
<div>📁 datos</div>	<div>Actualizar datos de semanas</div>	<div>hace 7 horas</div>
<div>📁 imagen</div>	<div>actualizar cambios</div>	<div>la semana pasada</div>
<div>📁 js</div>	<div>corrección: corregir rutas de carpetas y visor de PDF</div>	<div>la semana pasada</div>
<div>📁 Semana 1</div>	<div>actualizar cambios</div>	<div>la semana pasada</div>
<div>📁 Semana 10</div>	<div>Agregar carpetas de semanas con .gitkeep</div>	<div>la semana pasada</div>
<div>📁 Semana 11</div>	<div>Agregar carpetas de semanas con .gitkeep</div>	<div>la semana pasada</div>
<div>📁 Semana 12</div>	<div>Agregar carpetas de semanas con .gitkeep</div>	<div>la semana pasada</div>
<div>📁 Semana 13</div>	<div>Agregar carpetas de semanas con .gitkeep</div>	<div>la semana pasada</div>
<div>📁 Semana 14</div>	<div>Agregar carpetas de semanas con .gitkeep</div>	<div>la semana pasada</div>
<div>📁 Semana 15</div>	<div>Agregar carpetas de semanas con .gitkeep</div>	<div>la semana pasada</div>

## Paso 5: Actualizar tu proyecto cuando hagas cambios locales

Si realizas cambios en tu proyecto localmente y quieres subirlos a GitHub:

1. Agrega los archivos modificados:

```
git add .
```

2. Crea un commit con un mensaje describiendo los cambios:

```
git commit -m "Descripción de los cambios"
```

3. Sube los cambios al repositorio remoto:

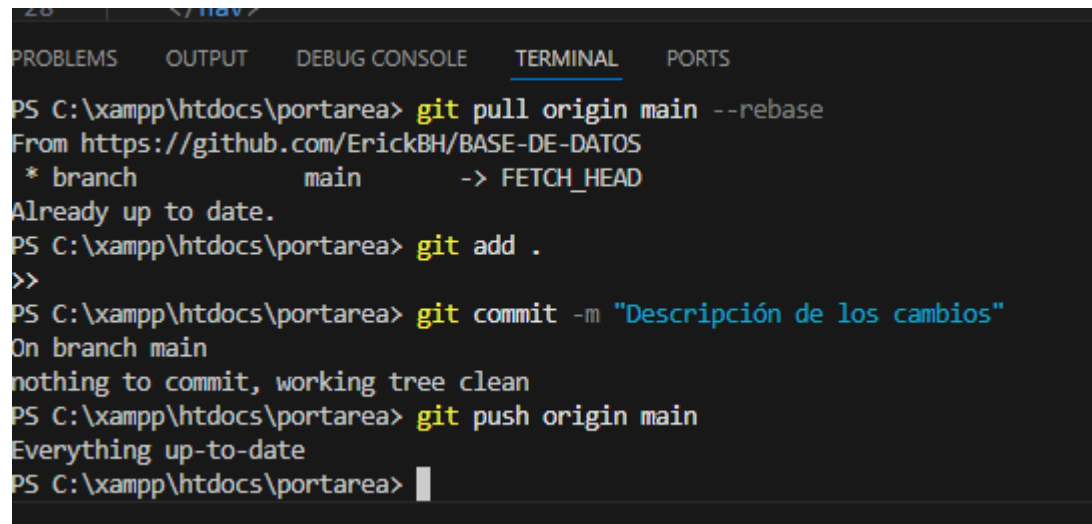
```
git push origin main
```

## Paso 6: Actualizar tu proyecto con cambios del repositorio remoto

Si hay cambios en GitHub que no están en tu proyecto local, utiliza:

```
git pull origin main --rebase
```

Esto hará que tus cambios locales se integren sobre los cambios más recientes del repositorio remoto, evitando conflictos.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\xampp\htdocs\portarea> git pull origin main --rebase
From https://github.com/ErickBH/BASE-DE-DATOS
* branch          main      -> FETCH_HEAD
Already up to date.
PS C:\xampp\htdocs\portarea> git add .
>>
PS C:\xampp\htdocs\portarea> git commit -m "Descripción de los cambios"
On branch main
nothing to commit, working tree clean
PS C:\xampp\htdocs\portarea> git push origin main
Everything up-to-date
PS C:\xampp\htdocs\portarea>
```