

Guía de Colaboración en Proyecto con GitHub y VS Code

Esta guía explica paso a paso cómo trabajar de forma colaborativa en un proyecto de código usando Git y GitHub desde Visual Studio Code (VS Code).

Requisitos Iniciales

Antes de comenzar, asegúrate de tener lo siguiente:

- **Git** instalado en tu computador: <https://git-scm.com/>
- Una cuenta en **GitHub** activa: <https://github.com/>
- **Visual Studio Code (VS Code)** instalado: <https://code.visualstudio.com/>
- Conexión a internet.

Opcional (pero útil):

- Extensión **GitLens** en VS Code para facilitar el seguimiento de cambios.
 - Extensión **Live Share** para editar el mismo proyecto en tiempo real con otros compañeros.
-

Primera vez que te sumas al proyecto (única vez por computador)

1. **Abre VS Code.**
2. Abre la **terminal** de comandos en VS Code (atajo `Ctrl + ñ` o desde el menú Terminal > Nuevo terminal).
3. Navega hasta la carpeta donde quieras guardar el proyecto en tu computador. Por ejemplo:

```
cd Desktop
```

1. **Clona el repositorio remoto** (el proyecto que está en GitHub):

```
git clone https://github.com/Jalbert70/proyecto-inventario-alimento.git
```

1. Entra a la carpeta del proyecto recién descargado:

```
cd proyecto-inventario-alimento
```

¡Ya tienes una copia del proyecto lista para comenzar a trabajar!

Antes de empezar a trabajar (SIEMPRE hacerlo primero)

Para evitar sobrescribir o perder el trabajo de otros compañeros:

```
git pull
```

Esto actualizará tu copia local con los últimos cambios del repositorio remoto (GitHub).

Editar archivos

- Puedes editar cualquier archivo dentro de la carpeta del proyecto desde VS Code.
- Asegúrate de **guardar** los archivos después de hacer cambios (`Ctrl + S`).

Subir tus cambios a GitHub (al terminar de trabajar)

Cuando finalices tus cambios, debes subirlos al repositorio remoto para que los demás los vean:

1. Agregar todos los archivos modificados al área de preparación (staging):

```
git add .
```

1. Crear un commit (confirmar los cambios localmente con un mensaje explicativo):

```
git commit -m "Describe brevemente el cambio realizado"
```

Ejemplo:

```
git commit -m "Agrega gráfico de ventas por temperatura"
```

1. Subir los cambios a GitHub:

```
git push
```

17 Buenas prácticas de colaboración

- Haz `git pull` **antes** de comenzar a trabajar cada día.
- Haz `git push` **después** de terminar de trabajar.
- Evita editar el mismo archivo al mismo tiempo que otro miembro del equipo.
- Comenta bien tu código y usa nombres claros en los archivos.
- Si creas nuevas funciones, explícalas en comentarios o en un README.

- Usa la extensión **Live Share** si necesitas trabajar en tiempo real con otro compañero (ambos deben tenerla instalada).



Problemas comunes y cómo solucionarlos

Problema	Causa común	Solución
<code>error: src refspec main does not match any</code>	No hiciste <code>git commit</code> antes de <code>git push</code> .	Asegúrate de hacer <code>git commit -m "mensaje"</code> antes de subir.
<code>Merge conflict</code>	Tú y otro usuario editaron la misma parte de un archivo.	Edita manualmente el archivo afectado y resuelve el conflicto.
<code>Permission denied</code>	No estás autenticado en GitHub o no tienes acceso.	Revisa que estés logueado y que tienes permisos sobre el repositorio.
<code>remote origin already exists</code>	Ya habías configurado la conexión con GitHub.	Usa <code>git remote set-url origin <URL></code> para cambiar la dirección.

Tips recomendados

- Instala la extensión **GitLens** para ver quién modificó cada línea de código.
- Instala **Live Share** si quieres colaborar en tiempo real en VS Code.
- Si vas a trabajar desde otro computador, repite los pasos de clonación (`git clone`), y siempre haz `git pull` antes y `git push` al finalizar.
- Usa mensajes de commit descriptivos y específicos.



¿Tienes dudas?

- Consulta a quien esté liderando el proyecto.
- Visita la documentación oficial: <https://git-scm.com/docs>

¡Listo! Ya sabes cómo trabajar colaborativamente con Git y GitHub desde Visual Studio Code. 🚀

Si todos siguen estos pasos, el trabajo en equipo será más ordenado, eficiente y sin conflictos innecesarios.