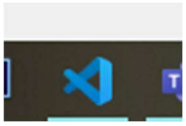
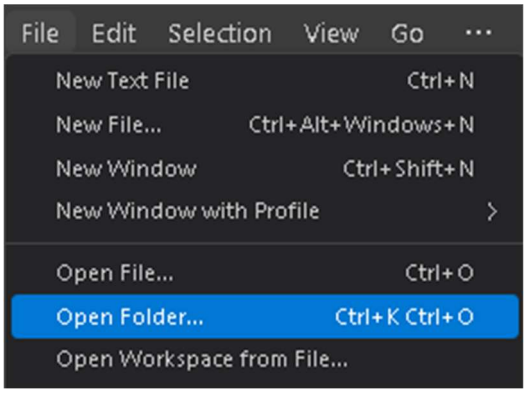
VISUAL STUDIO CODE

1. Crear un Proyecto

1. Abre **Visual Studio Code**.

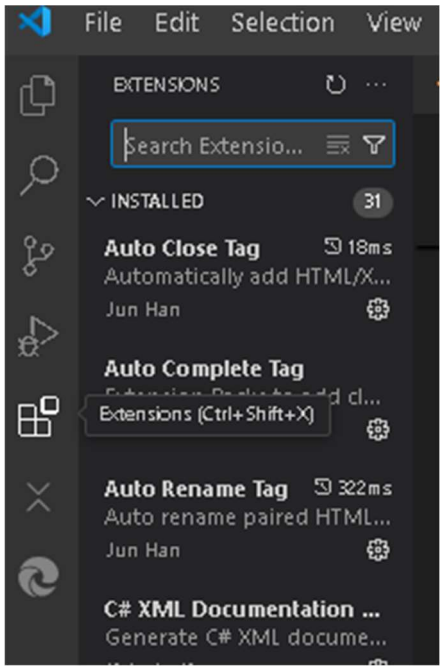


2. Ve a **File** → **Open Folder** (Archivo → Abrir Carpeta).



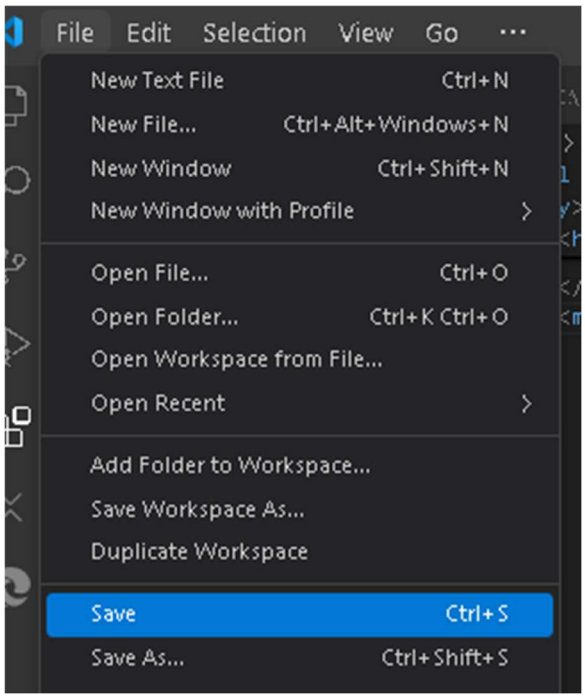
3. Selecciona o crea una carpeta para tu proyecto.

4. Si trabajas con un lenguaje específico (Java, Python, JavaScript, etc.), instala las extensiones necesarias desde la pestaña **Extensions** (Ctrl + Shift + X).



2. Guardar Proyecto

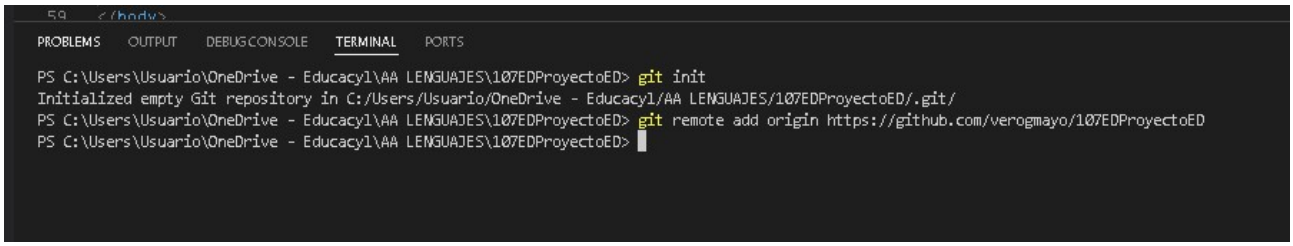
- VS Code guarda automáticamente los cambios en los archivos abiertos.
- Para forzar el guardado, presiona **Ctrl + S** o ve a **File → Save**.



---

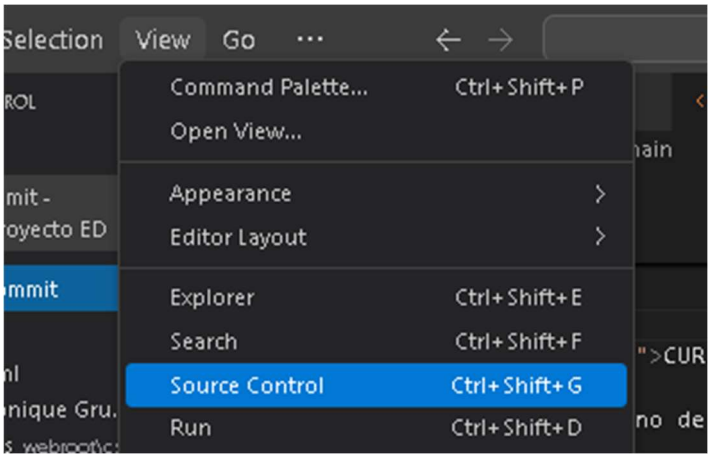
3. Versionar (Git)

1. desde la terminal de Visual Studio



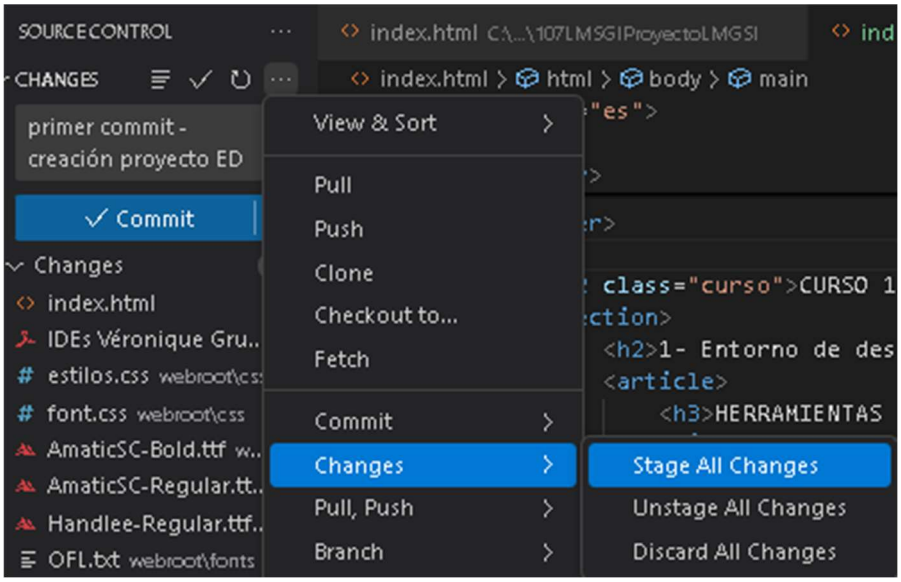
o

Abre **View → Source Control** (Ctrl + Shift + G).



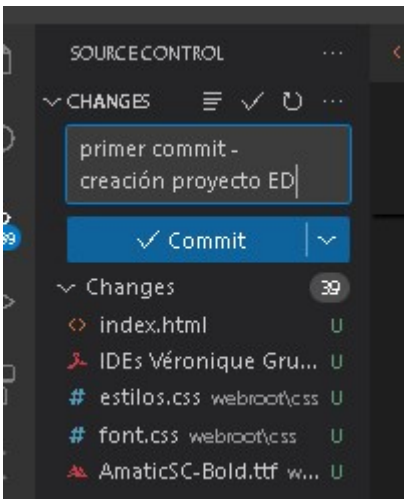
2. Si tu carpeta no está versionada, haz clic en **Initialize Repository**.

3. Preparar todos los archivos modificados para ser incluidos en el próximo commit.

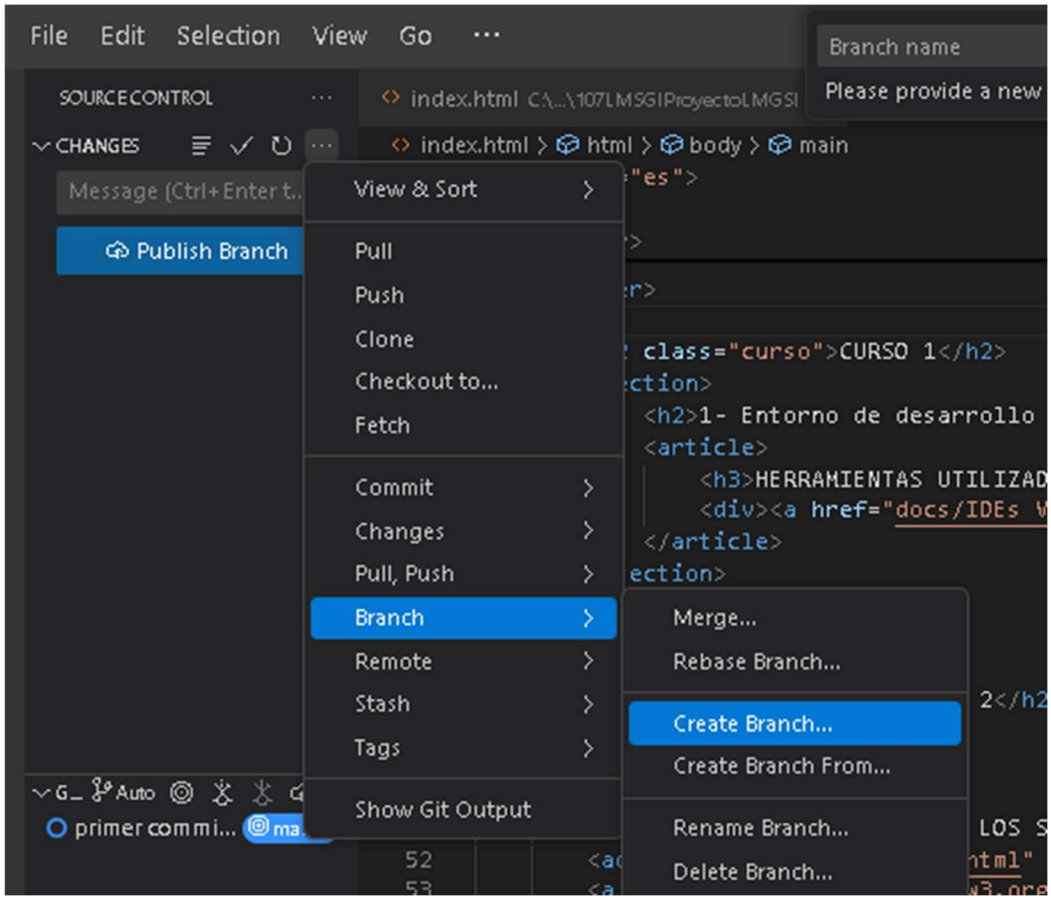


En la teminal será así

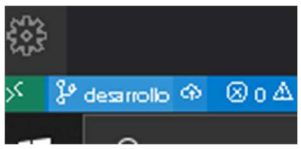
4. Escribe un mensaje en la caja de texto y presiona **Commit** (Ctrl + Enter).



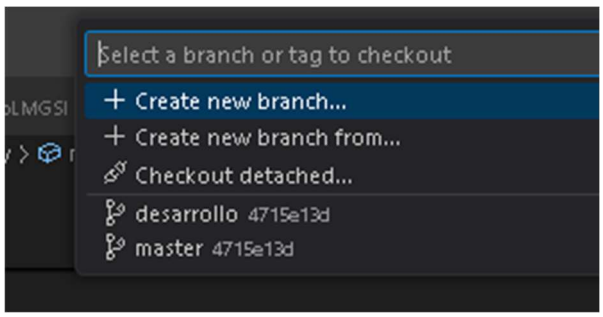
5. Crear una nueva rama .



6. Cambiar de rama. En la esquina inferior izquierda haz clic en la rama actual



7. Saldrá un menú arriba donde elegir la rama que quieras



8. Para subir cambios, haz clic en **Push** o usa git push en la terminal.

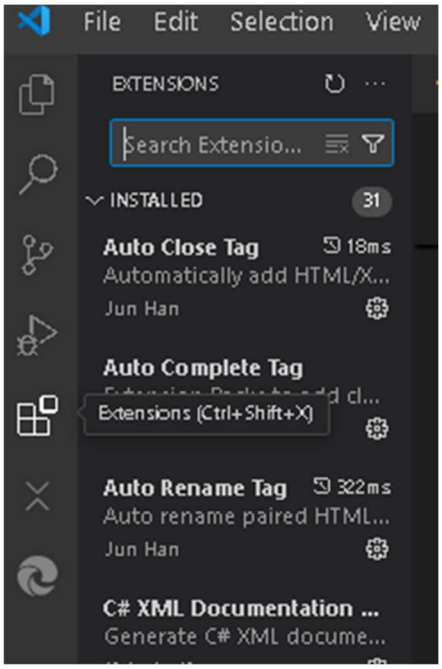
4. Recuperar desde el Repositorio

1. Abre **View** → **Source Control**.
2. Haz clic en **Clone Repository** y pega la URL del repositorio (GitHub, GitLab, etc.).
3. Selecciona la carpeta donde se clonará el proyecto.
4. Abre la carpeta en VS Code.

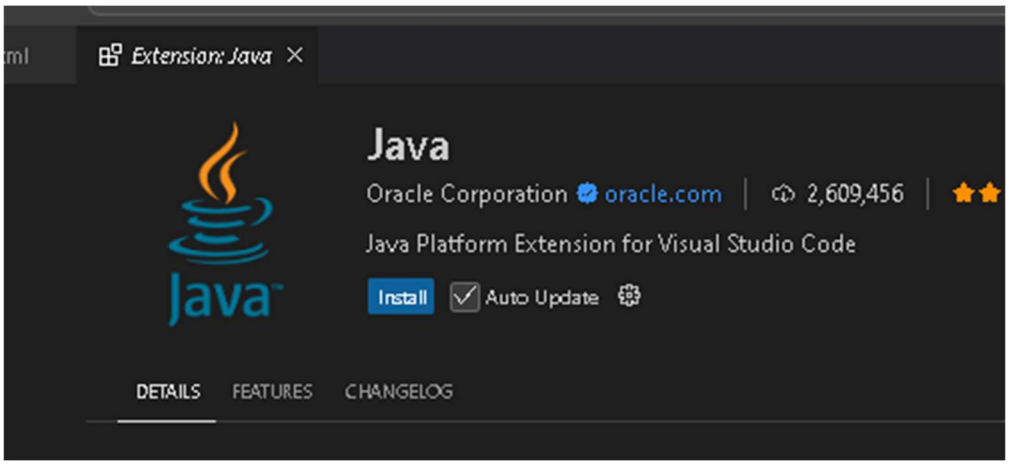
---

5. Instalar Plugins (Extensiones)

1. Ve a **View** → **Extensions** (Ctrl + Shift + X).



2. Busca el plugin necesario (ej. "Java Extension Pack" para Java).
3. Haz clic en **Install**.



6. Desarrollar (Escribir Código y Gestionar Archivos)

1. Usa el explorador de archivos (**Explorer**, Ctrl + Shift + E) para crear y abrir archivos.
2. Escribe tu código con soporte de autocompletado y resaltado de sintaxis.
3. Usa la terminal integrada (**View** → **Terminal**, `Ctrl + ``) para ejecutar comandos.

---

7. Compilar

- **Para Java:** Instala **Java Extension Pack**, luego compila con javac MiClase.java.
- **Para C/C++:** Instala **C/C++ Extension Pack** y usa gcc o g++.
- **Para Python:** Instala **Python Extension** y ejecuta python archivo.py.
- **Para JavaScript/TypeScript:** Usa **Node.js** (node archivo.js).

---

8. Ejecutar Código

1. Abre el archivo de código.
2. Presiona **F5** para ejecutar en modo depuración o usa Ctrl + F5 para ejecutar sin depuración.
3. También puedes usar la terminal integrada (`Ctrl + ``) para ejecutar comandos específicos según el lenguaje.

---

9. Probar (Manual y Automática)

- **Manual:** Ejecuta el programa y verifica la salida.
- **Automática:**
    1. Instala una extensión de pruebas (ej. **JUnit** para Java, **PyTest** para Python).
    2. Crea un archivo de prueba (ej. test_miprograma.py).
    3. Ejecuta las pruebas con pytest test_miprograma.py o desde la opción **Run Tests** en VS Code.

---

10. Documentar

- Usa comentarios en el código:

11. Generar Documentación (Javadoc, Doxygen, etc.)

- **Java:** Usa javadoc -d doc miarchivo.java para generar documentación en HTML.
- **Python:** Usa pydoc o **Sphinx**.
- **C/C++:** Usa **Doxygen** (doxygen Doxyfile).

12. Modelar (Diagramas UML y más)

1. Instala la extensión **PlantUML**.
2. Crea un archivo .puml y dibuja el diagrama, por ejemplo:
3. Abre la vista previa con Alt + D.

---

13. Depurar (Debugging)

1. Configura el depurador en **Run → Add Configuration**.
2. Presiona **F5** para iniciar el depurador.
3. Usa **breakpoints** (clic en el margen izquierdo del código).

---

14. Inspección de Variables

1. Activa el modo de depuración (F5).
2. Usa la pestaña **Variables** en la vista de depuración.
3. También puedes escribir en la **Consola de Depuración** para inspeccionar valores.

---

15. Ejecución Paso a Paso

- **F10:** Ejecutar línea sin entrar en funciones.
- **F11:** Entrar dentro de una función.
- **Shift + F11:** Salir de una función.

---

16. Comparar Código

1. Haz clic derecho sobre un archivo en el **Explorador**.
2. Selecciona **Select for Compare**.
3. Luego, haz clic derecho sobre otro archivo y elige **Compare with Selected**.

---

17. Refactorizar

1. Selecciona el código a refactorizar.
2. Usa el atajo **Ctrl + Shift + R** o haz clic derecho → **Refactor**.
3. Opciones comunes: **Renombrar (F2)**, **Extraer método**, **Mover código**.

---

18. Generar Código Automáticamente

- **Java:** Usa **Alt + Shift + S** para generar constructores, getters y setters.
- **JavaScript/Python:** Usa **Code Snippets** (Ctrl + Space para autocompletar código).

19. Ingeniería Inversa (Generar UML desde Código)

1. Instala la extensión **Java Extension Pack** o **PlantUML**.
2. Usa la opción de **Diagrama UML** para visualizar clases automáticamente.

---

20. Administrar Base de Datos

1. Instala la extensión **SQL Server** o **MySQL** según la base de datos.
2. Usa el panel **View → SQL Explorer** para conectar bases de datos.

---

21. Ejecutar Script sobre la Base de Datos

1. Abre un archivo .sql.
2. Conéctalo a una base de datos desde el panel de extensiones.
3. Ejecuta el script con **Run Query** (Ctrl + Shift + E).