



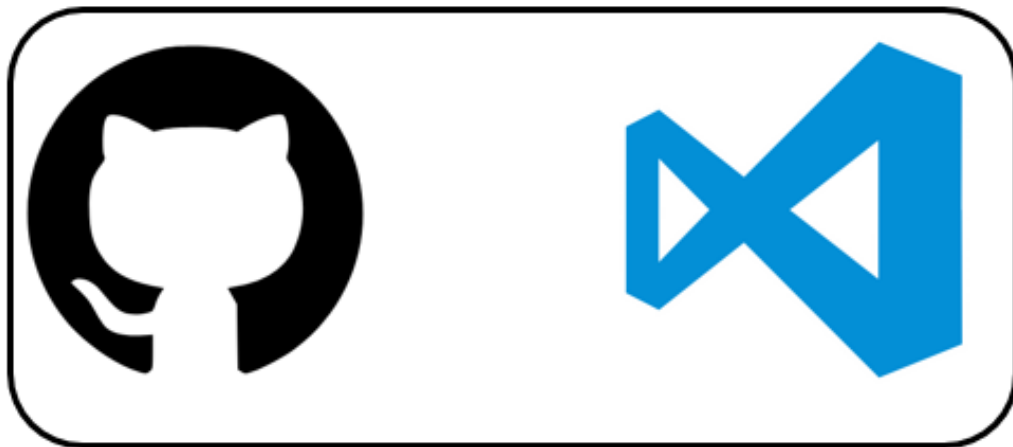
UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Ingeniería, Arquitectura y Diseño

Profesor. Raúl Ramírez Ramírez.

Materia: Elaboración de Documentación Técnica.

MANUAL DE ENLACE ENTRE VISUAL STUDIO CODE Y GITHUB.



Realizado por:

Beritain Navarro José Abraham 369226

Rojas Aripez Josue Francisco 369718

Ingeniería en Computación.

Grupo: 432

Fecha de realización: Lunes, 23 de Mayo del 2023.

Contenido

Introducción.....	3
¿Qué es Visual Studio Code?.....	3
¿Qué es GitHub?	4
Crear un repositorio en GitHub	5
Enlazar repositorio de GitHub con Visual Studio Code.....	13
Guardar Cambios	21

Introducción

Bienvenido al manual de enlace entre ‘Visual Studio Code’ (“VS code”) y “GitHub”. Este manual ha sido creado con el propósito de brindarte instrucciones lo más claro y conciso posible para que puedas lograr establecer una conexión exitosa entre dos herramientas fundamentales para el desarrollo de software: “Visual Studio Code” y “GitHub”.

En el mundo actual del desarrollo de software, la colaboración y el control de versiones son fundamentales. “GitHub”, como plataforma líder en alojamiento de repositorios de código, ofrece un entorno eficiente y colaborativo para el desarrollo de proyectos. Por otro lado, “VS Code”, uno de los editores de código más populares y potentes, proporciona una interfaz intuitiva y herramientas avanzadas para la escritura y edición de código.

En este manual, aprenderás cómo llevar a cabo la integración entre “VS Code” y “GitHub”, permitiéndote gestionar de una manera más eficiente tus proyectos de código fuente y facilitando la colaboración con otros desarrolladores a través de la plataforma “GitHub”. A lo largo de estas páginas, exploraremos los pasos necesarios para configurar tu entorno de desarrollo en “VS Code” y establecer la conexión con tu repositorio en “GitHub”. Además, te familiarizarás con las funcionalidades clave que te permitirán gestionar tus repositorios, realizar seguimiento de versiones, colaborar en proyectos y optimizar tu flujo de trabajo.

No importa si eres un principiante que está dando sus primeros pasos en el desarrollo de software o si ya tienes experiencia en el campo, este manual te brindará los conocimientos necesarios para establecer una sólida conexión entre “VS Code” y “GitHub”, lo que te permitirá trabajar de manera más eficiente y colaborativa en tus proyectos.

¿Qué es Visual Studio Code?

“VS Code” es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) de código abierto, multiplataforma y altamente personalizable desarrollado por Microsoft. Está diseñado para ser liviano, rápido y compatible con una amplia gama de lenguajes de programación y tecnologías.

“VS Code” se ha convertido en una opción popular entre los desarrolladores debido a sus numerosas ventajas y características como:

- **Depuración integrada:** El IDE ofrece una depuración integrada que permite a los desarrolladores rastrear y solucionar errores de manera eficiente. Puedes establecer puntos de interrupción, inspeccionar variables

en tiempo real, ejecutar paso a paso y realizar un seguimiento del flujo de ejecución de tu código.

- **Integración con herramientas y extensiones:** “VS Code” se puede ampliar aún más mediante la instalación de extensiones. Hay una amplia gama de extensiones disponibles que ofrecen características adicionales, como integración con control de versiones, administración de paquetes, integración con servicios en la nube y mucho más.
- **Integración con “Git” y “GitHub”:** “VS Code” tiene una excelente integración con el sistema de control de versiones “Git”. Puedes realizar todas las operaciones de control de versiones directamente desde el IDE, como confirmar cambios, crear ramas, fusionar y resolver conflictos. Además, puedes conectar tu repositorio de GitHub y aprovechar las funciones de colaboración y seguimiento de problemas directamente desde “VS Code”.

¿Qué es GitHub?

“GitHub” es una plataforma web de desarrollo de software que utiliza el sistema de control de versiones “Git”. Permite a los desarrolladores y equipos de desarrollo almacenar, gestionar y colaborar en proyectos de software de manera remota y eficiente.

En “GitHub”, los proyectos se almacenan en repositorios, que son espacios virtuales donde se guarda el código fuente, así como otros archivos y recursos relacionados. Estos repositorios pueden ser públicos, lo que significa que son visibles para todos los usuarios, o privados, donde solo las personas autorizadas tienen acceso.

Una de las principales características de “GitHub” es su capacidad de control de versiones. Utiliza “Git”, un sistema distribuido de control de versiones, que permite realizar un seguimiento de los cambios realizados en el código a lo largo del tiempo.

“GitHub” se ha convertido en una opción popular entre los desarrolladores debido a sus numerosas ventajas y características como:

- **Colaboración y trabajo en equipo:** Proporciona una plataforma para la colaboración efectiva entre desarrolladores y equipos. Varios desarrolladores pueden trabajar en paralelo en un proyecto, crear y fusionar ramas de código y resolver conflictos de manera sencilla.
- **Repositorios públicos y privados:** Ofrece la opción de tener repositorios públicos o privados. Los repositorios públicos son visibles y accesibles para todos, lo que fomenta la colaboración abierta y permite que otros desarrolladores contribuyan a tus proyectos. Los repositorios privados, por

otro lado, son visibles solo para las personas autorizadas, lo que los hace adecuados para proyectos comerciales o privados que requieren confidencialidad.

- **Respaldo de datos:** Se encarga de la administración y el respaldo de tus repositorios en la nube, lo que proporciona una mayor seguridad y redundancia de datos. Esto significa que tus proyectos están protegidos contra la pérdida de datos en caso de fallas de hardware o errores humanos.

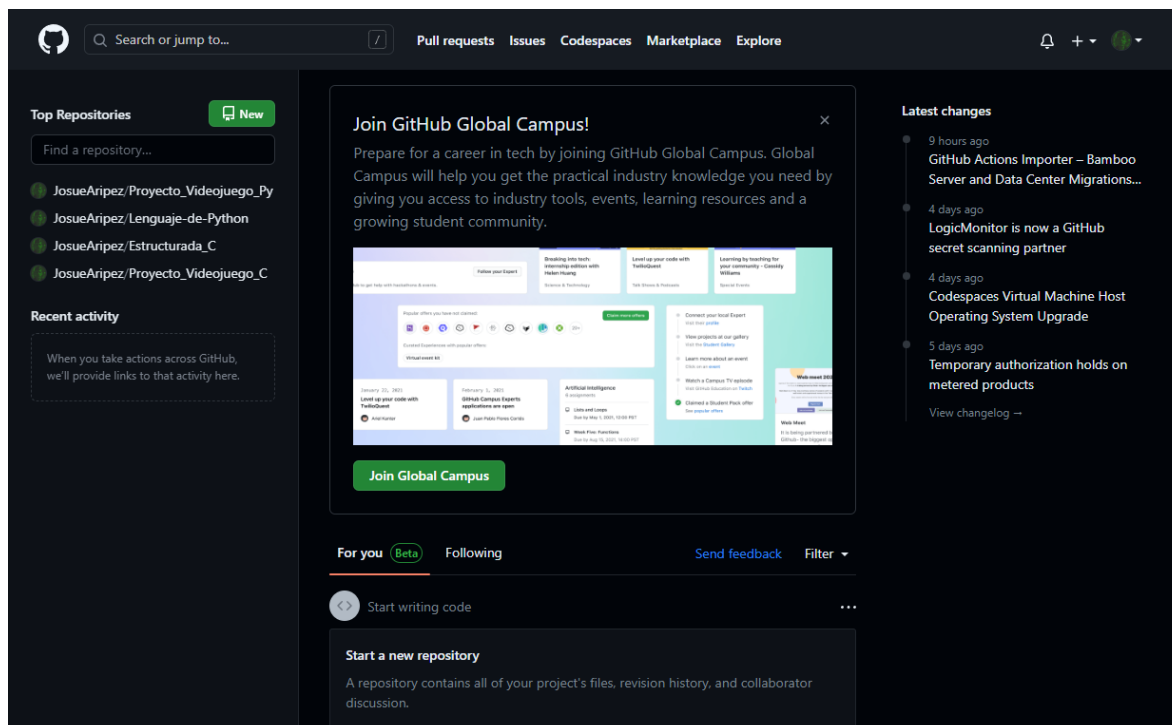
Crear un repositorio en GitHub

Este manual asume que tienes conocimientos mínimamente básicos de “VS Code” y que cumples con unos requisitos básicos que son:

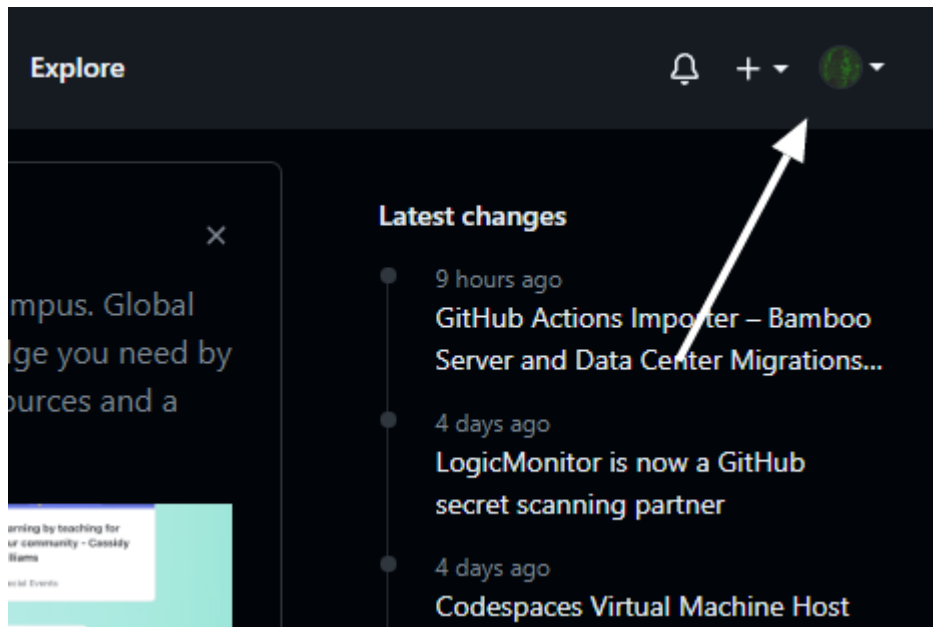
- Tener una cuenta de [“GitHub”](#)
- Tener instalado [“Visual Studio Code”](#)
- Tener [“Git”](#) instalado

Una vez tengamos los requisitos anteriores nos situamos en la página [principal de “GitHub”](#).

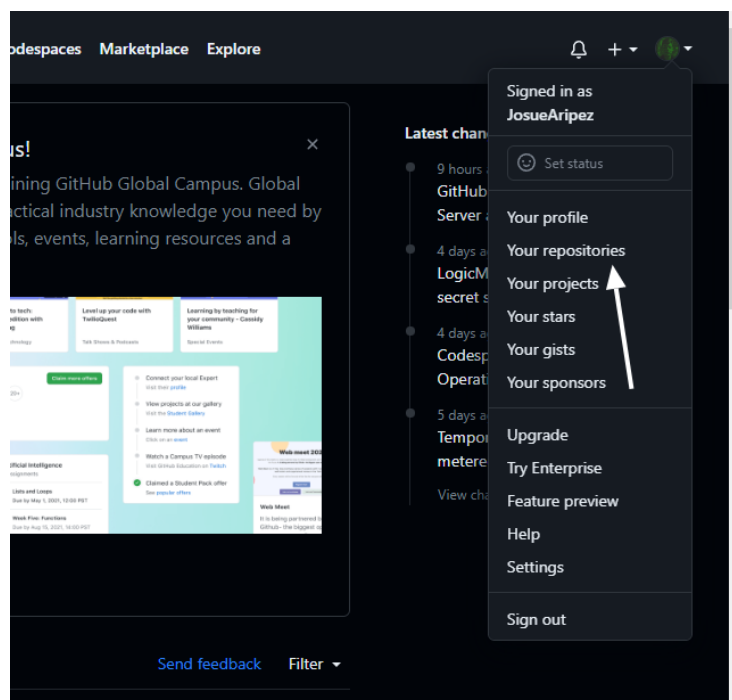
A continuación presentaremos todos los pasos a seguir en orden por el cual se deben de ir realizando



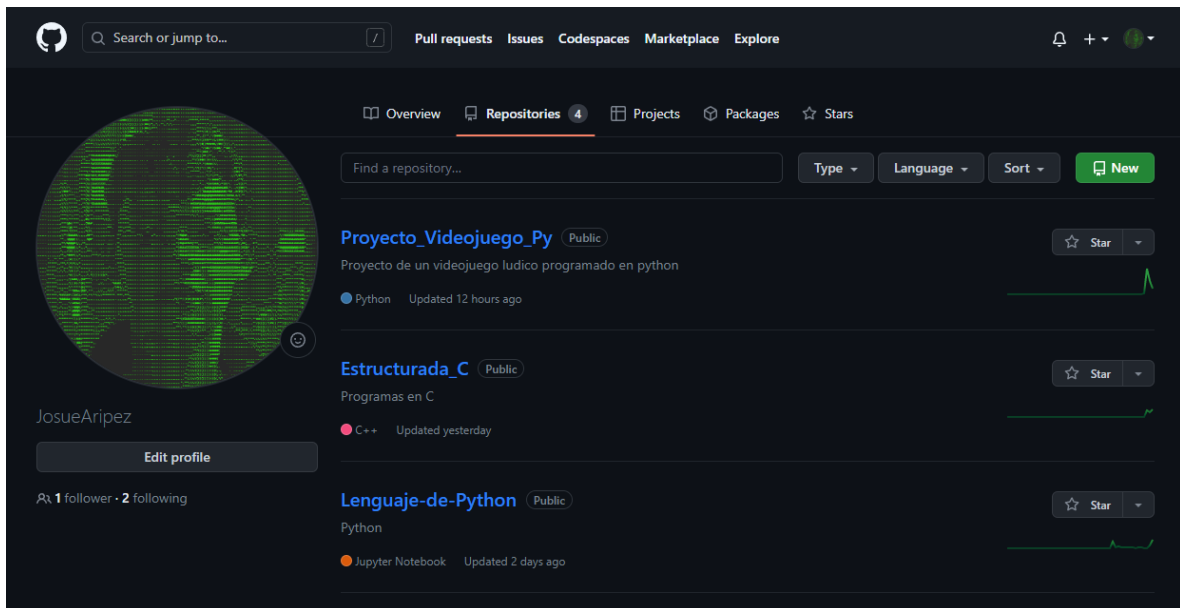
Nos fijaremos en la parte superior derecha justo donde está situada nuestra foto de perfil.



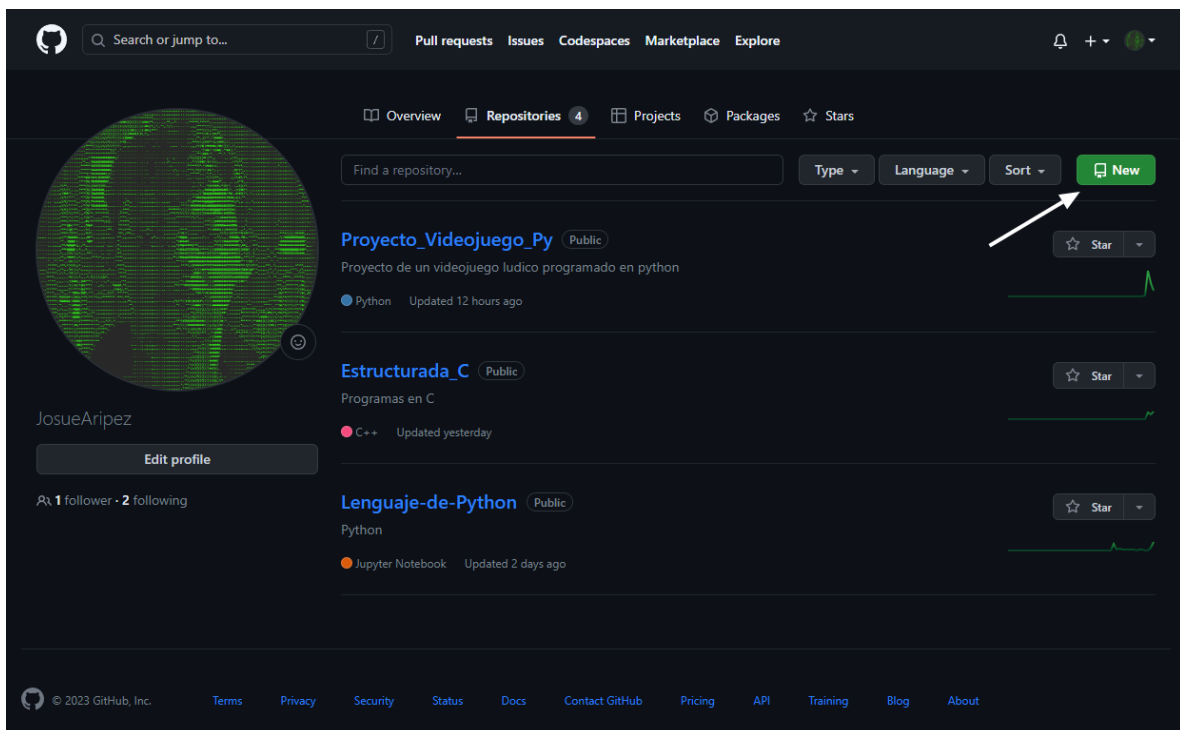
Damos “click” en nuestra foto de perfil y se desplegará un menú donde seleccionaremos la opción “Your repositories” (Tus repositorios).



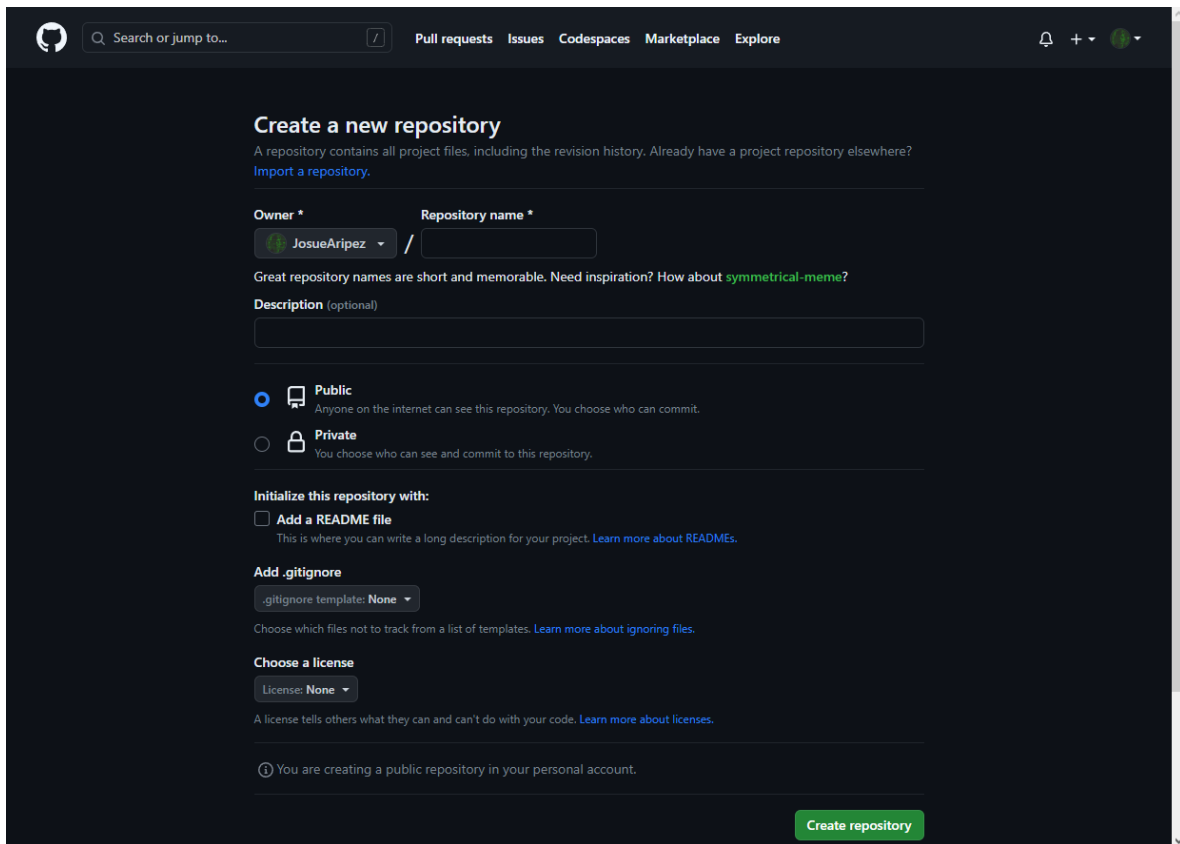
Se nos cargará la página donde aparecerán todos los repositorios que hemos creado con anterioridad



Nos situaremos en la parte medio arriba de la página justo donde está un botón verde con la palabra que dice “New” (Nuevo) y daremos “click”.



Se cargará la página para crear un nuevo repositorio.



The screenshot shows the GitHub 'Create a new repository' page. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Codespaces', 'Marketplace', and 'Explore'. The main heading is 'Create a new repository', followed by a subtext: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. Below this, there are two input fields: 'Owner *' with a dropdown menu showing 'JosueAripez' and 'Repository name *' with an empty text box. A tip below these fields says: 'Great repository names are short and memorable. Need inspiration? How about [symmetrical-meme?](#)'. There's a 'Description (optional)' text area. Below that, there are two radio button options: 'Public' (selected) with the description 'Anyone on the internet can see this repository. You choose who can commit.' and 'Private' with the description 'You choose who can see and commit to this repository.'. Under 'Initialize this repository with:', there's a checkbox for 'Add a README file' with the text 'This is where you can write a long description for your project. [Learn more about READMEs.](#)'. Below that is a section for 'Add .gitignore' with a dropdown menu showing '.gitignore template: None' and the text 'Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)'. Then, there's a 'Choose a license' section with a dropdown menu showing 'License: None' and the text 'A license tells others what they can and can't do with your code. [Learn more about licenses.](#)'. At the bottom, there's a note: 'You are creating a public repository in your personal account.' and a green 'Create repository' button.

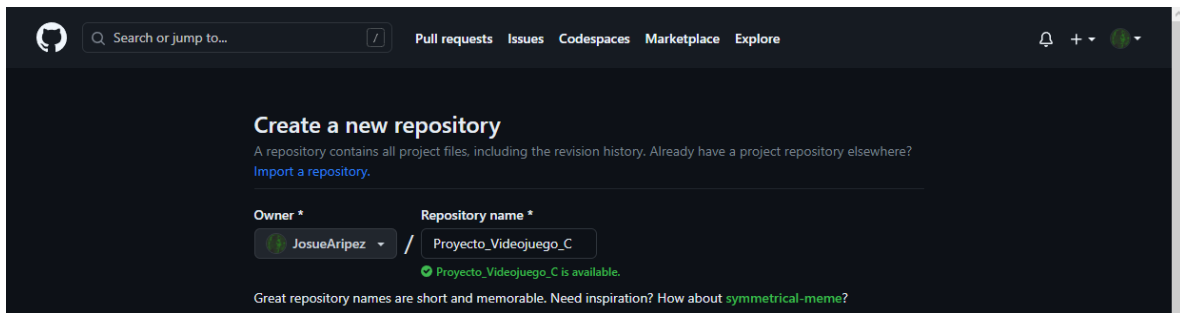
Tendremos que llenar los datos pedidos.

En "Owner" (Dueño o propietario)

Saldrá nuestra cuenta por defecto (en caso de tener mas cuentas le daremos "click" a nuestra cuenta por defecto y se desplegaran todas las cuentas añadidas y seleccionamos una).

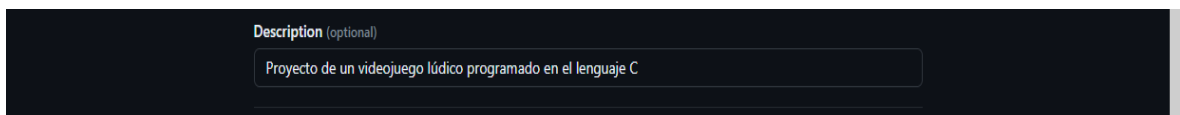
"Repository name" (Nombre del repositorio)

Debajo de esta estará el campo para ingresar el nombre en nuestro caso será: Proyecto_Videojugo_C. Cabe recalcar que la misma página nos proporcionará un pequeño texto en caso de que el nombre cumpla o no con algunos de sus parámetros.



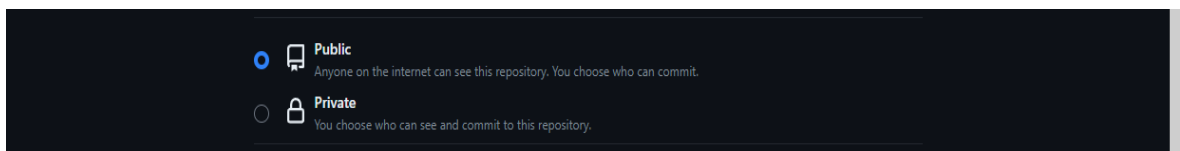
“Description” (Descripción)

Este campo es para rellenar por si deseamos agregar información acerca de nosotros o nuestro repositorio, esto es de manera opcional, de momento nosotros le pondremos una breve descripción puesto que luego se puede modificar.



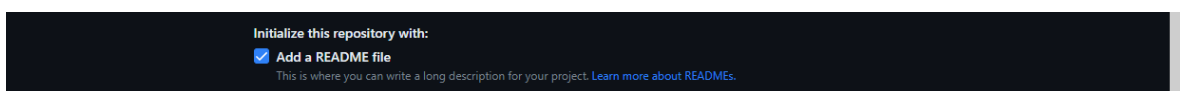
“Public” (Público) y “Private” (Privado)

El siguiente apartado tendremos dos opciones esto es por si queremos que nuestro repositorio este disponible para la comunidad de GitHub o cualquiera pueda acceder a ella o si queremos que se quede privado solamente para nosotros o a quien decidamos brindarle el acceso y ya nadie mas lo pueda ver. Nosotros lo dejaremos en público.



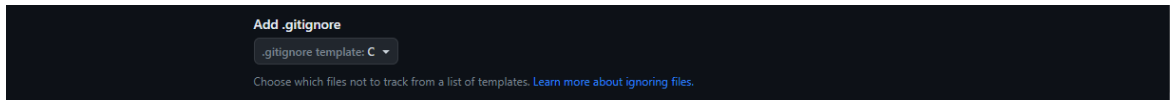
“README file” (Archivo para leer)

Sí deseamos agregar este archivo ya que es opcional podemos agregar información extra ya sea de nosotros, del repositorio, o algún otro mensaje que deseamos este archivo se descargar de manera automática a la hora de descargar todos los recursos del repositorio.



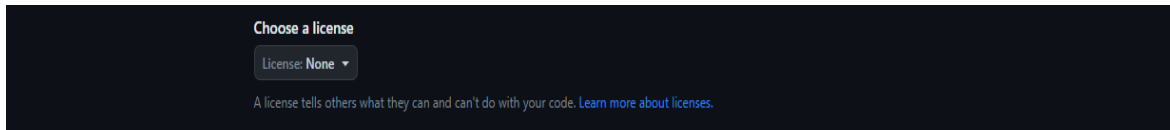
“Add.gitignore” (Agregar lenguaje)

Desde aquí podemos especificar en que lenguaje estará escrito nuestro código, aunque también la página lo detectara de forma automática con el tiempo. Esto sirve como referencia a la hora de buscar programas o repositorios, podemos especificar en concreto que lenguaje estamos buscando. De igual manera que pasos anteriores, este es de manera opcional.



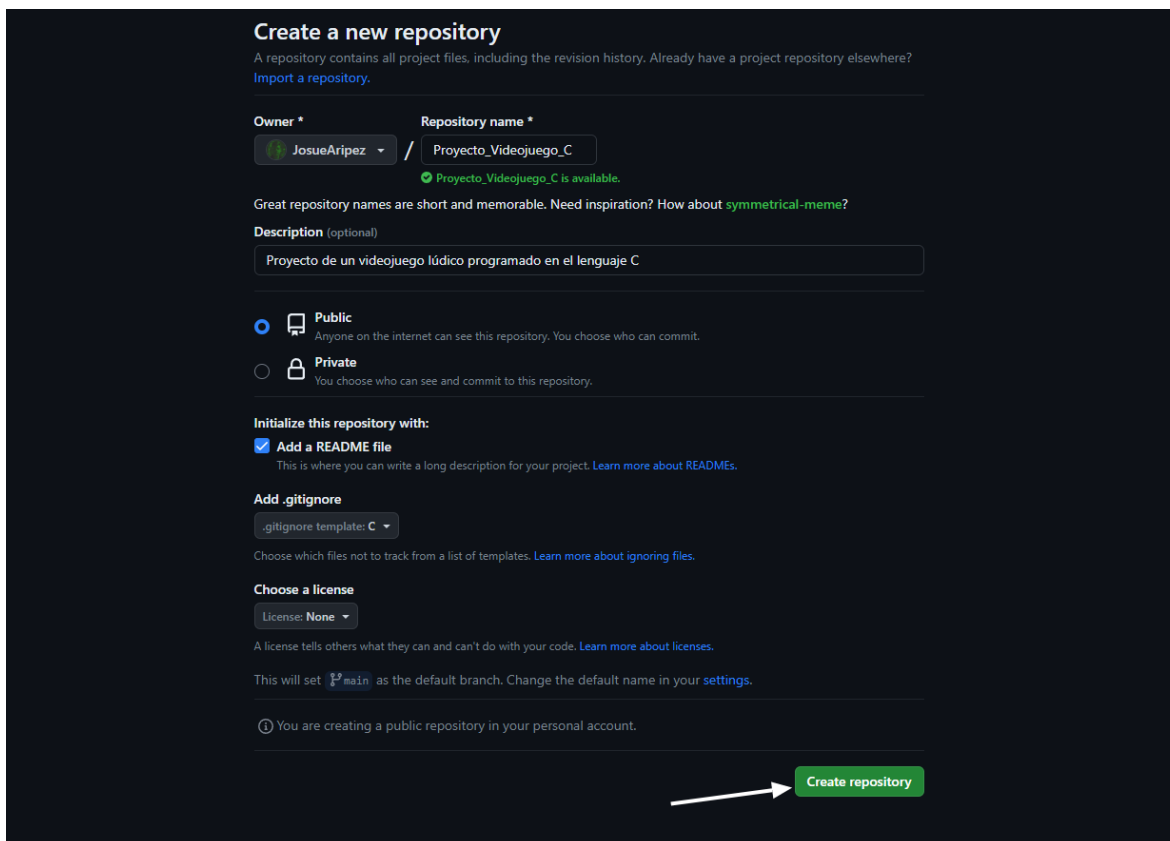
“Choose a license” (Elija una licencia)

En este apartado se nos ofrece a elegir una lista de licencias en caso de contar con alguna, en caso de no contar con alguna dejar la opción ya definida (None).



Como se puede apreciar, crear un repositorio en “GitHub” no es una tarea complicada, pero si te quedan dudas respecto a un paso debajo de cada opción la página te brinda información útil respecto a cada función.

Para terminar de crear el repositorio solamente queda dar “click” en el botón verde que dice “Create repository” (Crear repositorio) y el repositorio estará creado y listo para usarse.



Una vez dado el “click” en el botón verde “Create repository” nos cargará la página principal de nuestro nuevo repositorio, donde ya de inicio tendremos varias funcionalidades entre las que destacan:

1. **“Settings” (Ajustes):** Aquí podremos modificar las diferentes opciones de nuestro repositorio entre las que más destacan es la posibilidad de agregar colaboradores.
2. **“main” (principal):** Esta es la rama donde nos encontramos, en este caso el “main” viene por defecto. Posteriormente podremos agregar más ramas pero de momento lo dejaremos con el “main”.
3. **“Go to file” (Ir al archivo):** Al dar “click” en este botón nos llevará a la página donde tendremos cargados todos los archivos que vayamos agregando a nuestro repositorio.
4. **“Add file” (Agregar archivo):** Aquí podremos adjuntar archivos que ya tengamos en nuestro ordenador.
5. **“Code” (Código):** En este botón verde podemos obtener algunas cosas interesantes de nuestro repositorio, entre lo mas resaltante es el link de la pagina y la posibilidad de descargar todo lo que tengamos en nuestro repositorio por medio de un archivo .zip
6. **“Initial commit” (Confirmación inicial):** En este recuadro nos aparece los archivos cargados en el “main” de inicio tendremos dos archivos “.gitignore” donde vendrán unos datos ya cargados por la propia pagina y el “README.md” esta solo estará si fue seleccionada durante la creación.
7. **“README.md” (Archivo para leer):** En el recuadro aparecerá la información que agregamos con anterioridad.

The screenshot shows the GitHub repository page for 'JosueAripez / Proyecto_Videojuego_C'. The interface is in dark mode. Numbered callouts highlight the following elements:

- 1**: The 'Settings' tab in the repository navigation bar.
- 2**: The 'Code' tab in the repository navigation bar.
- 3**: The 'Go to file' button.
- 4**: The 'Add file' button.
- 5**: The 'Code' button with a dropdown arrow.
- 6**: The 'main' branch selector.
- 7**: The 'README.md' file in the file list.

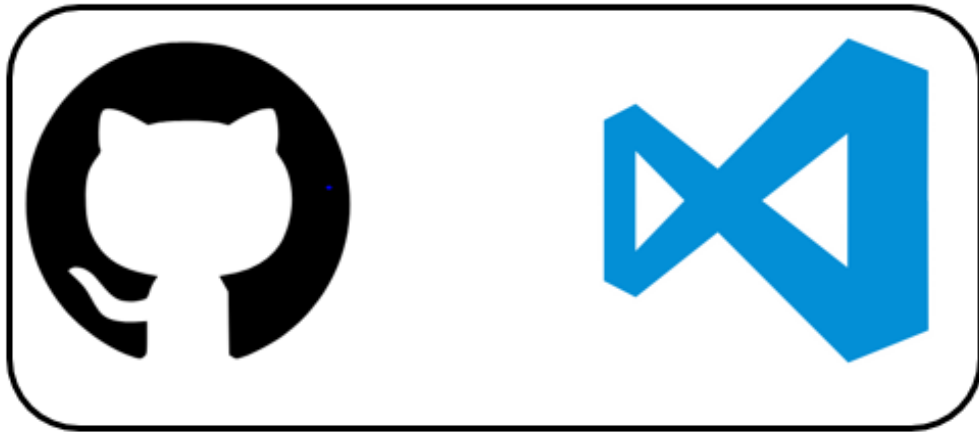
The repository details show an initial commit by JosueAripez 22 minutes ago. The README content is:

Proyecto_Videojuego_C

Proyecto de un videojuego lúdico programado en el lenguaje C

The right sidebar shows repository statistics: 0 stars, 1 watching, 0 forks. It also includes sections for 'Releases' (No releases published) and 'Packages' (No packages published).

Enlazar repositorio de GitHub con Visual Studio Code

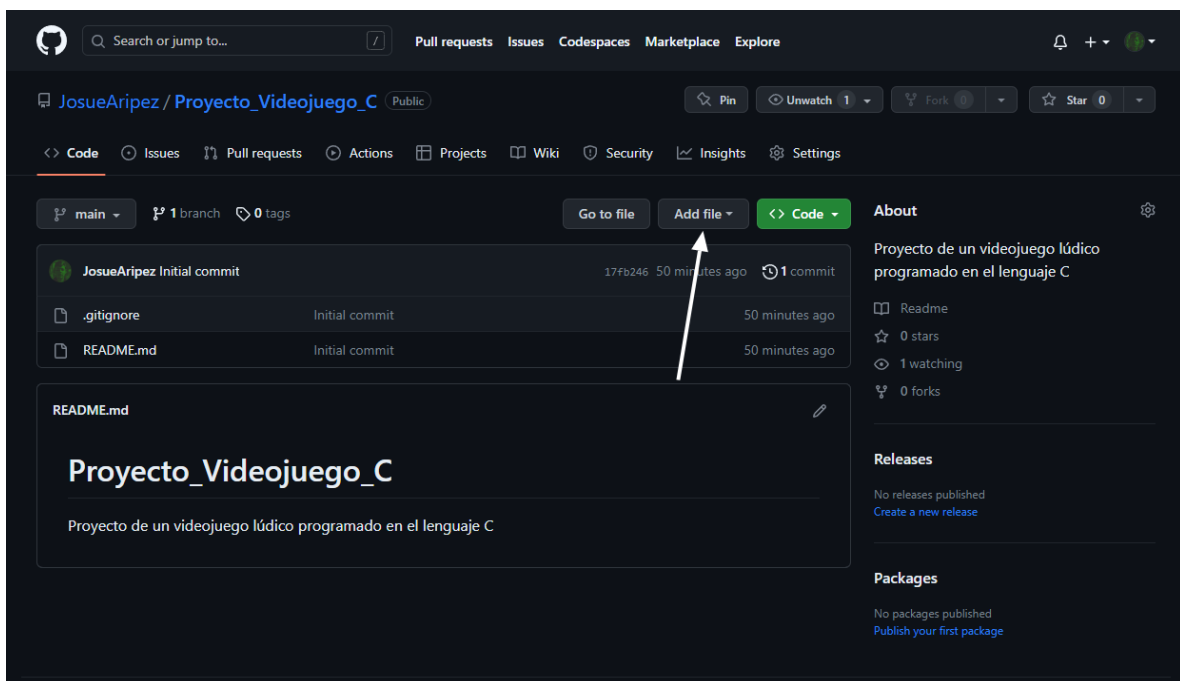


Requisitos previos:

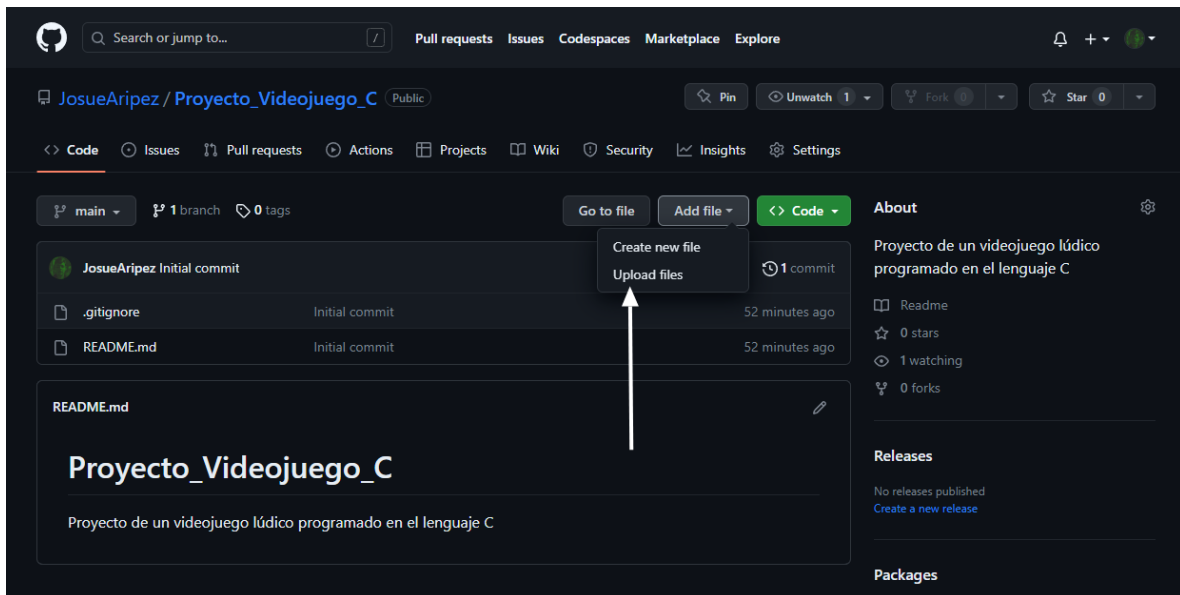
Para este paso del proceso sería necesario contar con un par de objetos previos, que son los siguientes:

- Tener una carpeta o archivo donde se encuentre el código, esta carpeta es necesario que esté dentro de tu equipo de cómputo.
- Tener en activo la misma cuenta en “VS Code” y “GitHub”.

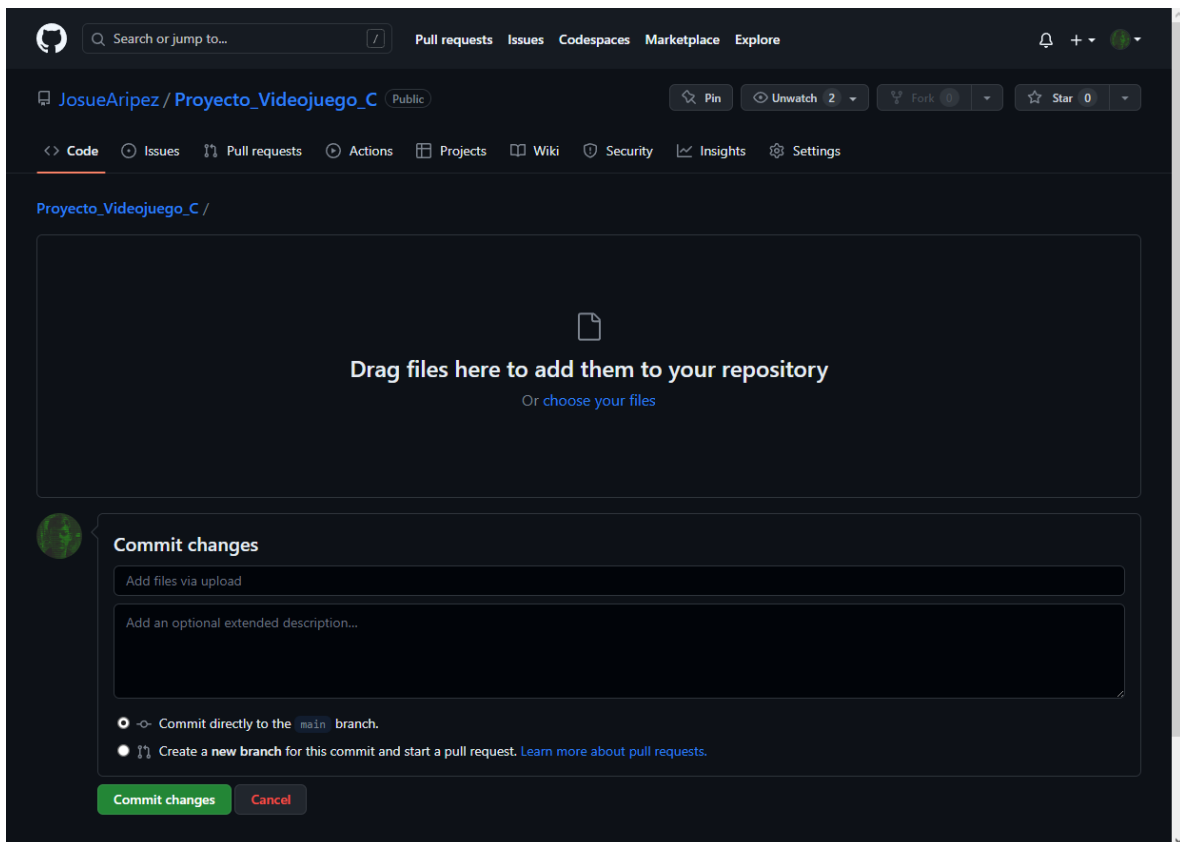
Para enlazar Visual Studio Code con GitHub tendremos que estar en la página principal de nuestro repositorio de “GitHub” y daremos clic en el botón “Add file”



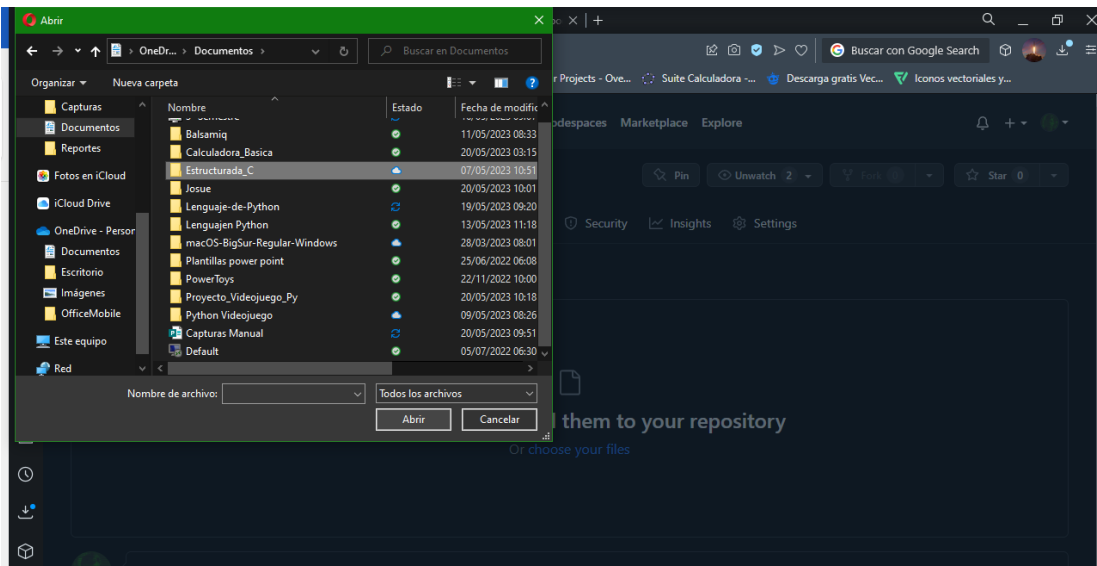
Al dar “click” se nos desplegará un menú donde aparecerán dos opciones “Create new file” (Crear nuevo archivo) y “Upload files” (Subir archivos) seleccionaremos la segunda opción “Upload files”



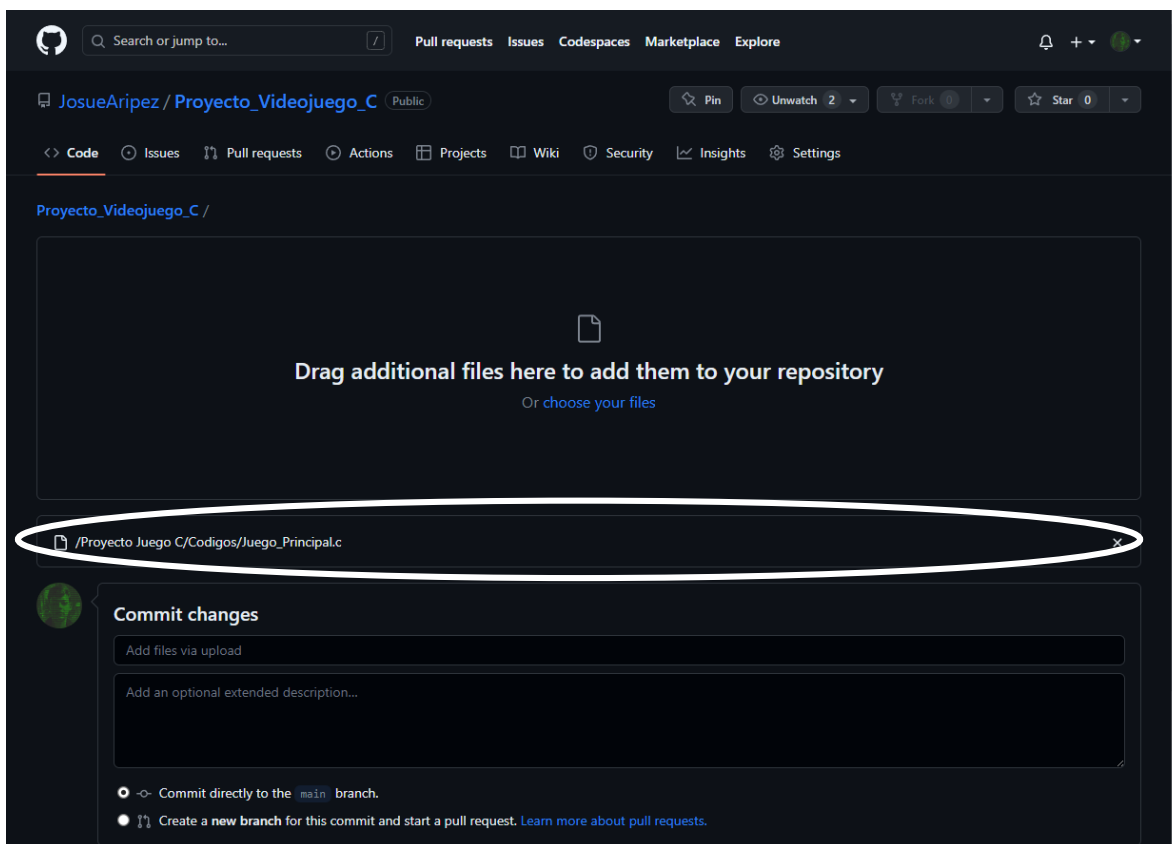
Nos aparecerá la pagina para subir los archivos.



En el primer recuadro nos ofrecera dos formas de subir los archivos la primera es arrastrándolos desde el origen en los archivos o donde sea que los tengas posicionados hasta la pagina de “GitHub”. La segunda opcion es buscándolos desde el directorio. Nosotros los buscaremos a trevaz del directorio.



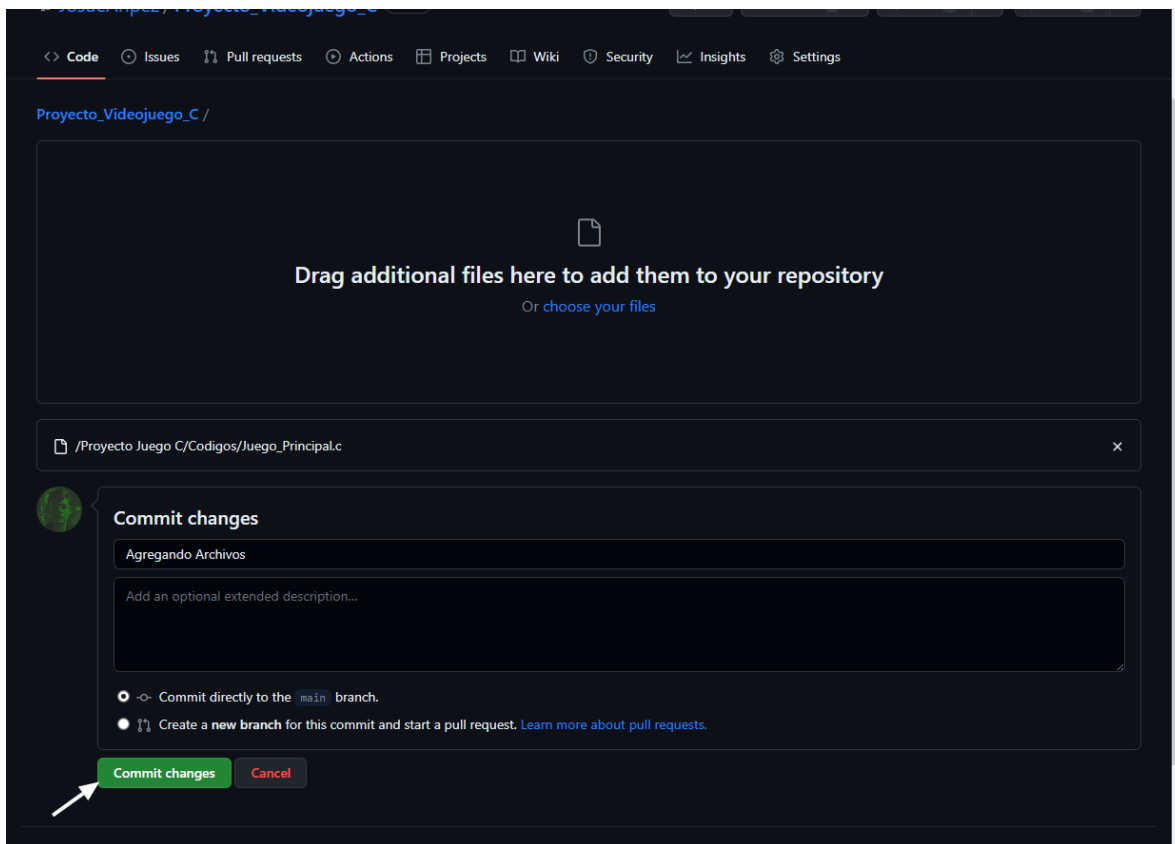
Eso será necesario para cargar los archivos deseados.



En el segundo recuadro titulado “Commit changes” (Cometer cambios) tendremos dos campos: en el primero será ingresar un nombre que haga referencia a lo realizado. En este caso nosotros pondremos ‘Agregando Archivos’. El segundo campo se utiliza para poder agregar alguna descripción acerca del cambio pero esto es opcional.

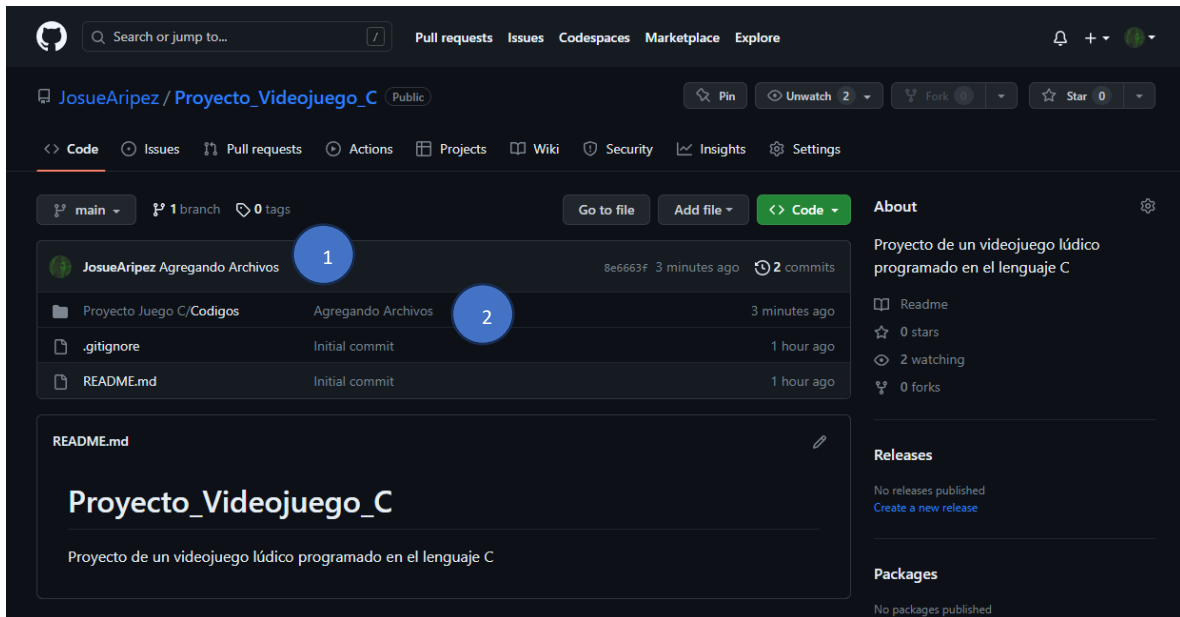
Por último aparecen dos opciones donde tenemos que elegir donde queremos cometer los cambios si en el “main” o si deseamos crear una segunda rama. De momento nosotros lo dejaremos en el “main”.

Ya solo debemos hacer click en el botón verde “Commit changes”

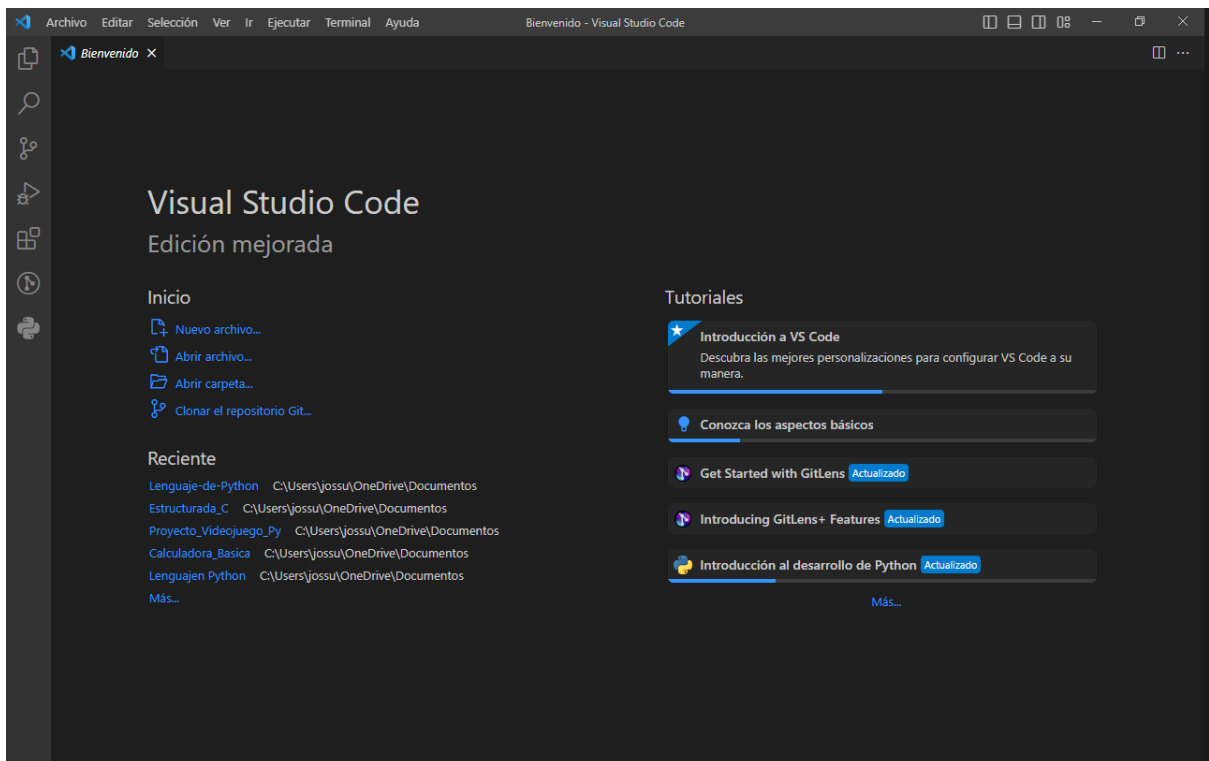


A continuación nos aparecerá una pantalla de carga idicando que se están agregando los archivos, posteriormente aparecerá la página principal de nuestro repositorio y en ella aparecerá:

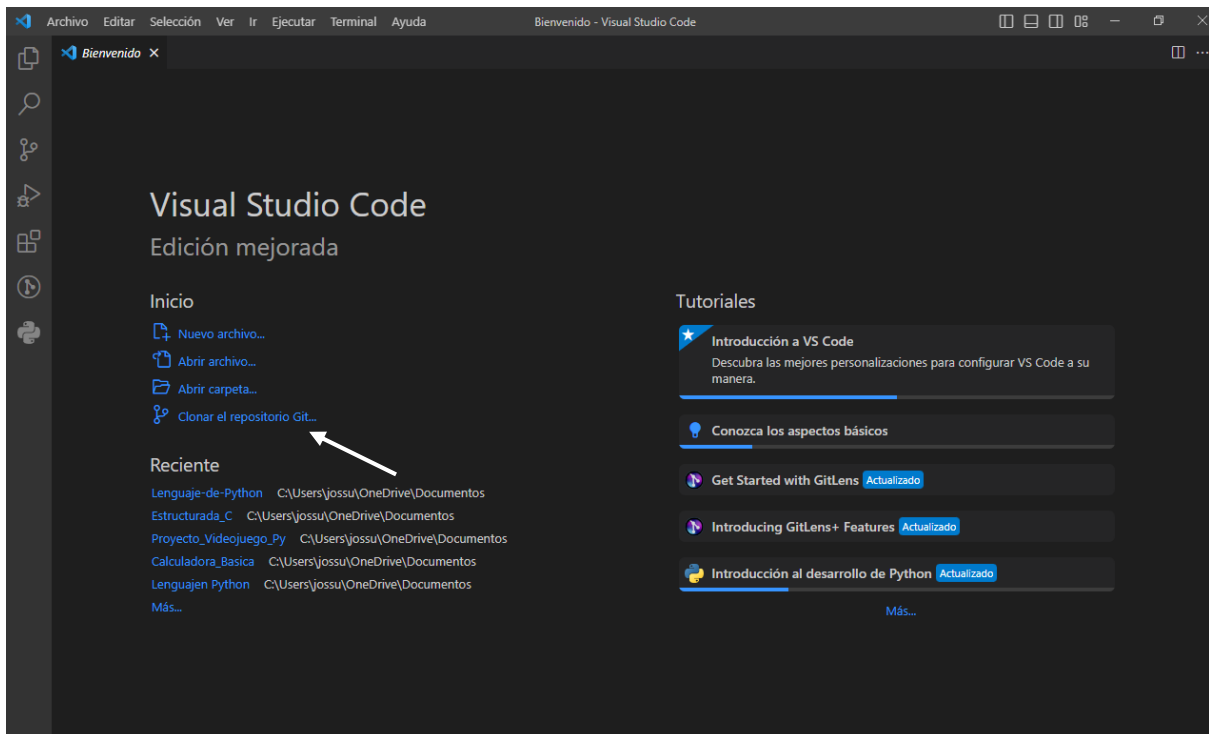
1. El nombre que agregamos.
2. Los archvos cargados.



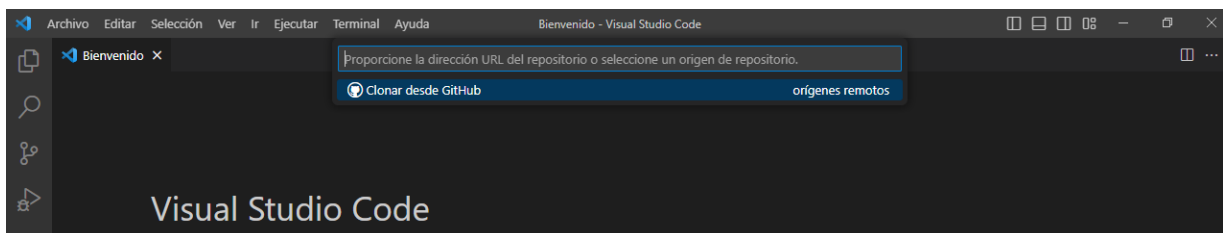
Una vez hecho esto ya podemos cerrar “GitHub” y dirijirnos a “VS Code”.



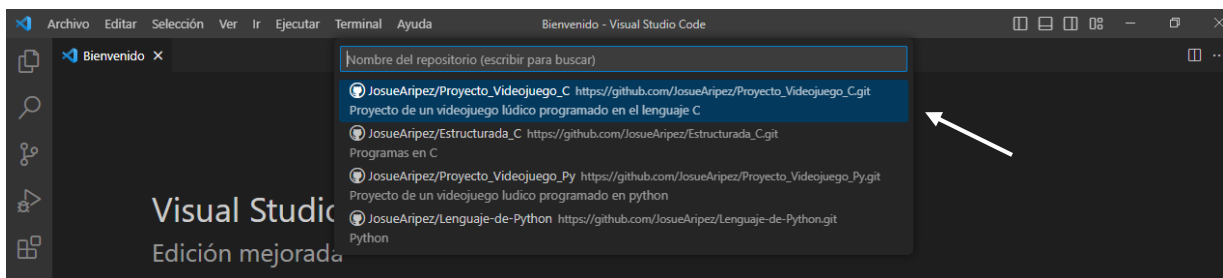
En la pantalla principal de “VS Code” le tendremos que dar “click” a la opción que dice: Clonar repositorio “Git”



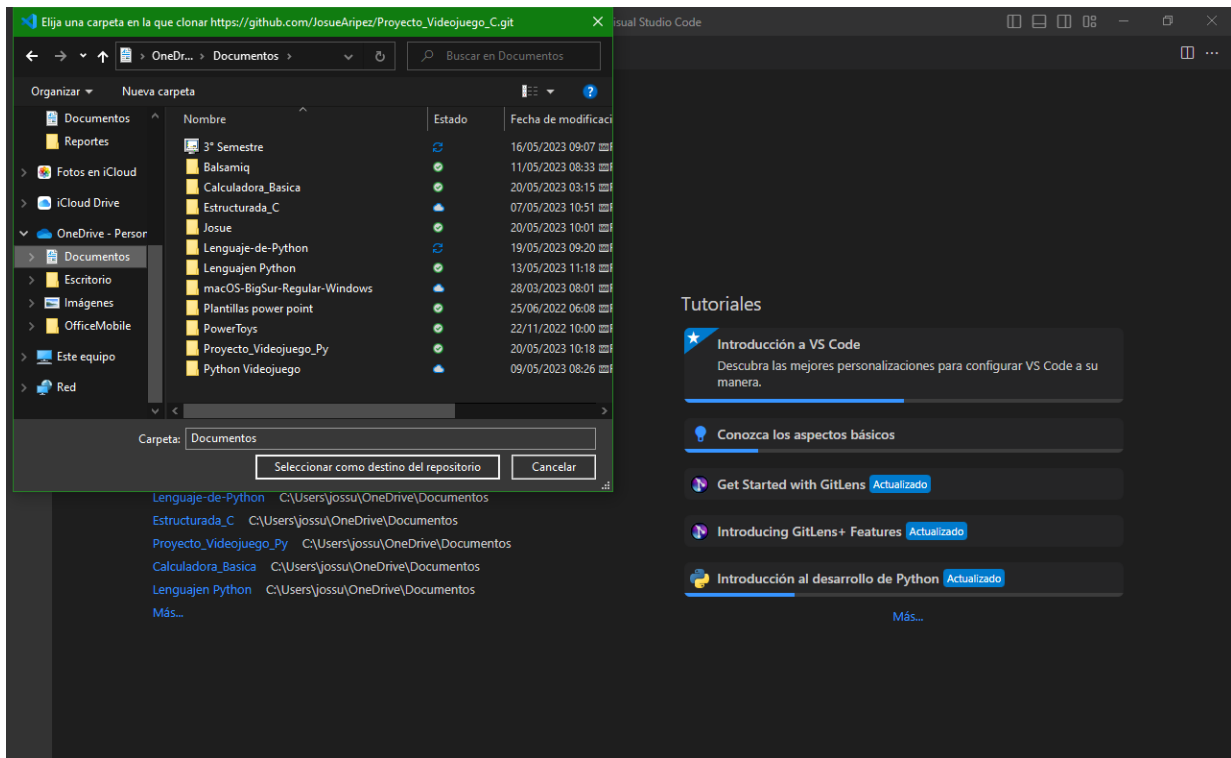
Se desplegará un menú en la parte superior con el nombre Clonar desde “GitHub” la cual seleccionaremos.



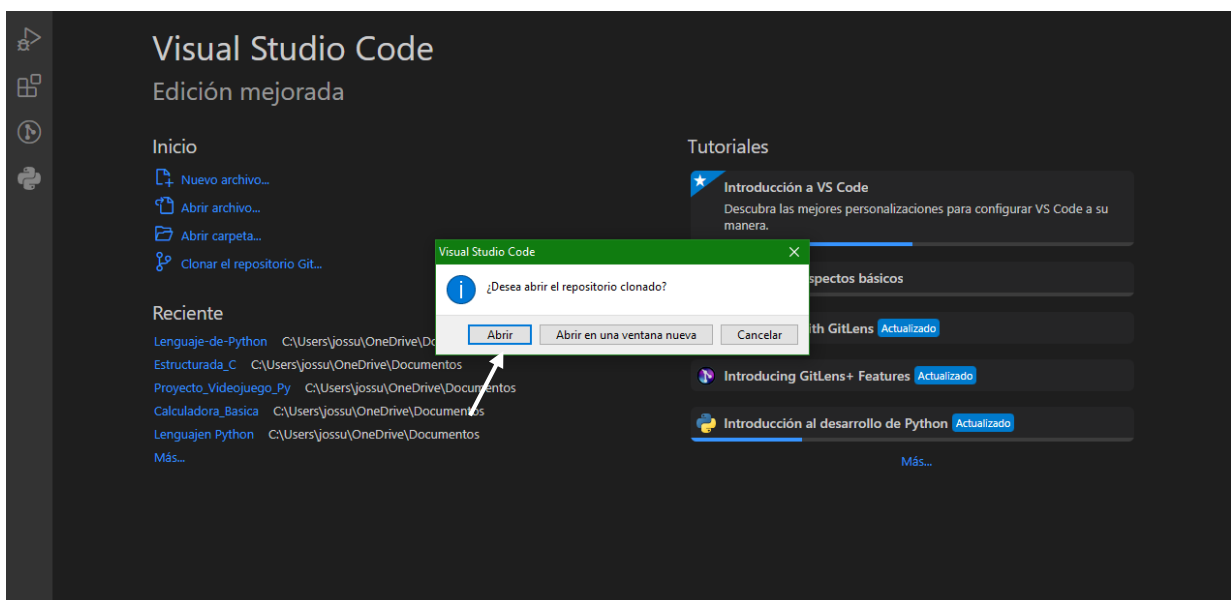
Luego de hacer “click” se nos desplegará otro menú donde aparecerán todos los repositorios creados por nosotros y seleccionamos el que queramos.



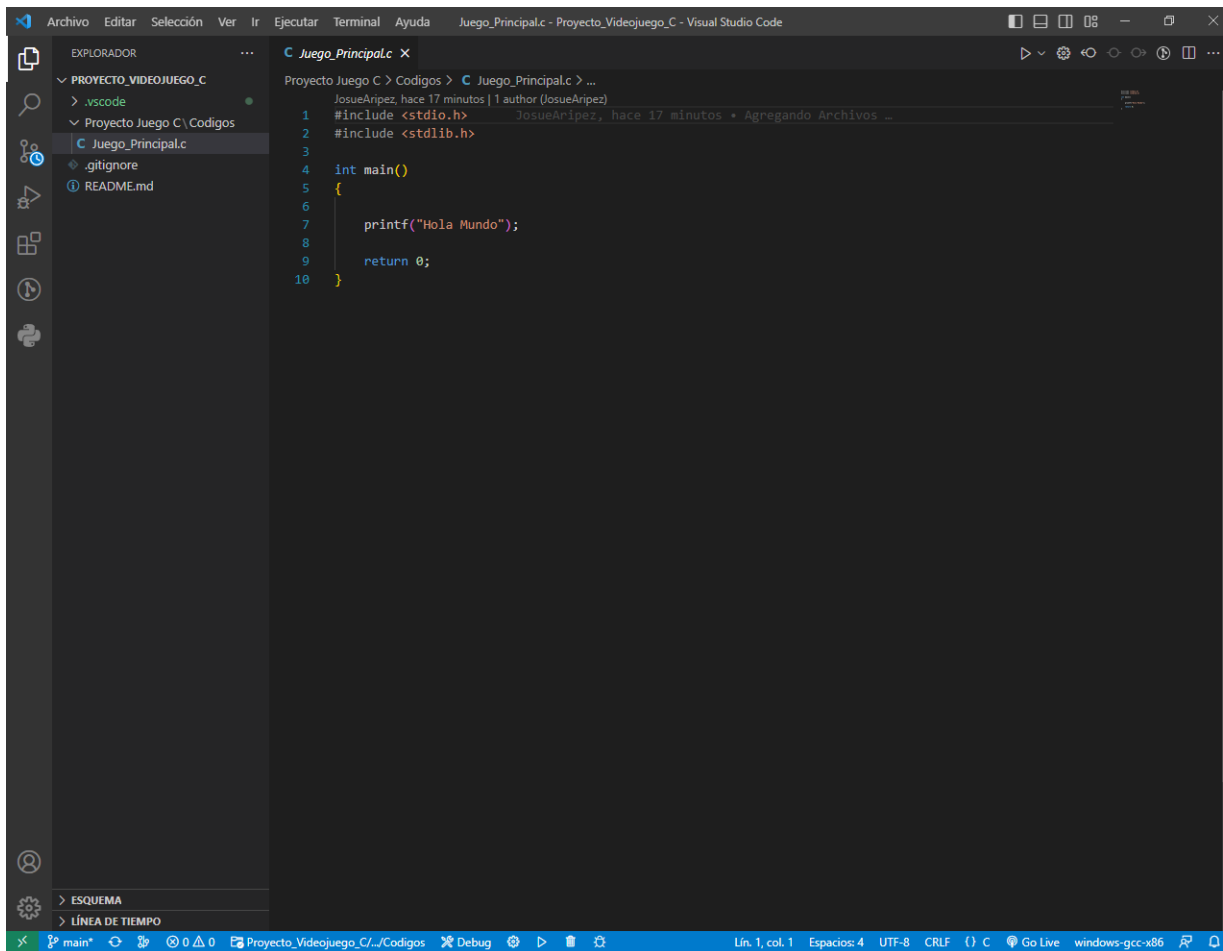
A continuación se nos abrirá una carpeta en donde debemos especificar dónde queremos que se haga la clonación de nuestro repositorio, nosotros la dejaremos en Documentos



Nos saldrá una ventana emergente donde se nos pregunta si deseamos abrir el repositorio clonado a la cual le daremos en Abrir.



Con eso terminaríamos de clonar nuestro repositorio.



The image shows a screenshot of the Visual Studio Code editor interface. The title bar at the top reads "Juego_Principal.c - Proyecto_Videojuego_C - Visual Studio Code". The interface is divided into several panels:

- EXPLORADOR (Explorer):** Located on the left, it shows a file tree for "PROYECTO_VIDEOJUEGO_C". The files listed are ".vscode", "Proyecto Juego C\Codigos", "Juego_Principal.c", ".gitignore", and "README.md".
- EDITOR:** The central area displays the code for "Juego_Principal.c". The code is as follows:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     printf("Hola Mundo");
7
8     return 0;
9 }
10
```
- TERMINAL:** A terminal window is open at the bottom, showing the command prompt "Proyecto Juego C > Codigos > Juego_Principal.c > ...". It also displays a message: "JosueAripez, hace 17 minutos | 1 author (JosueAripez)".
- ESQUEMA (Outline):** Located at the bottom left, it shows the structure of the code, including "main".
- LINEA DE TIEMPO (Timeline):** Located at the bottom right, it shows the current line of code being edited, "main".

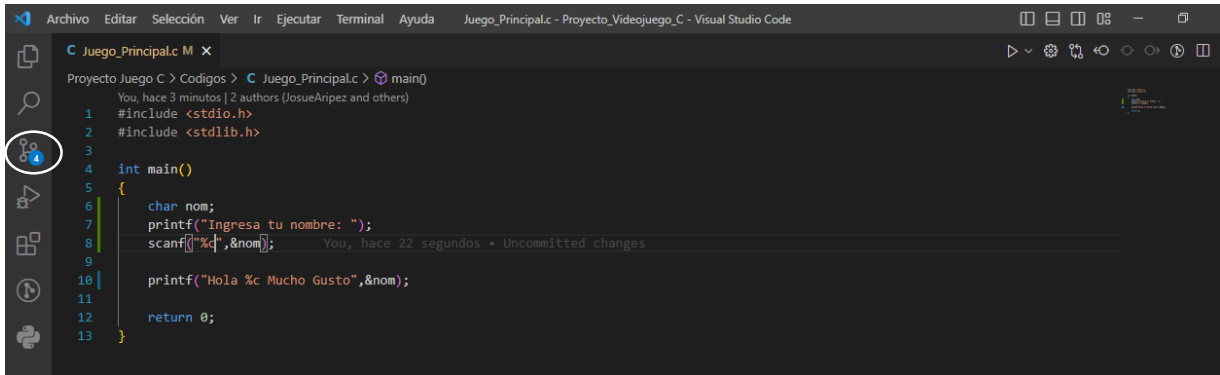
The status bar at the bottom indicates the current line and column: "Lin. 1, col. 1". It also shows the file encoding as "UTF-8", the line ending as "CRLF", and the active window as "main".

Guardar Cambios

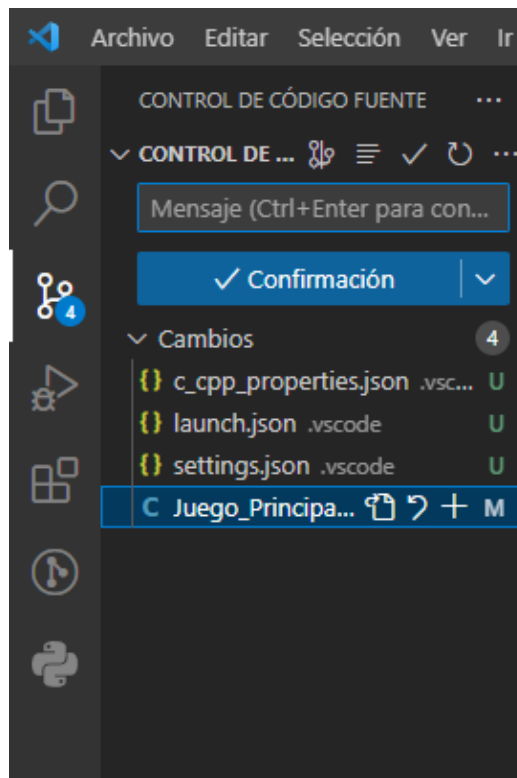
Requerimientos previos:

- Tener modificaciones en el código

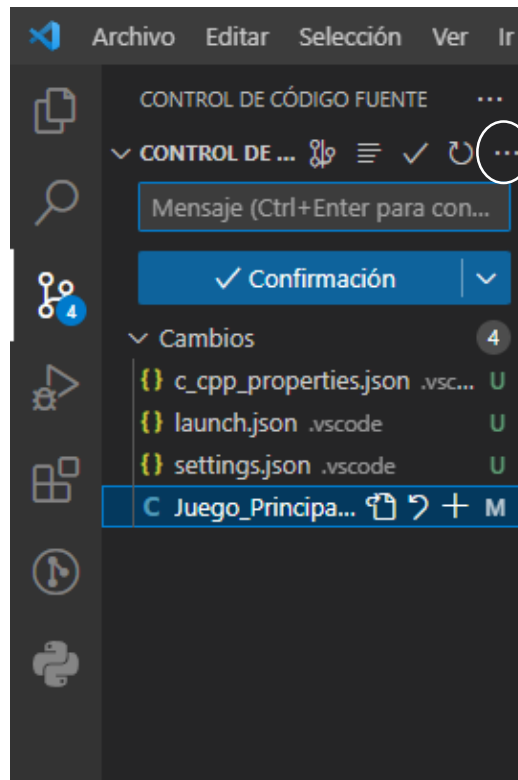
Para guardar los cambios realizados en el código nos dirigimos al panel lateral (por defecto está en la izquierda).



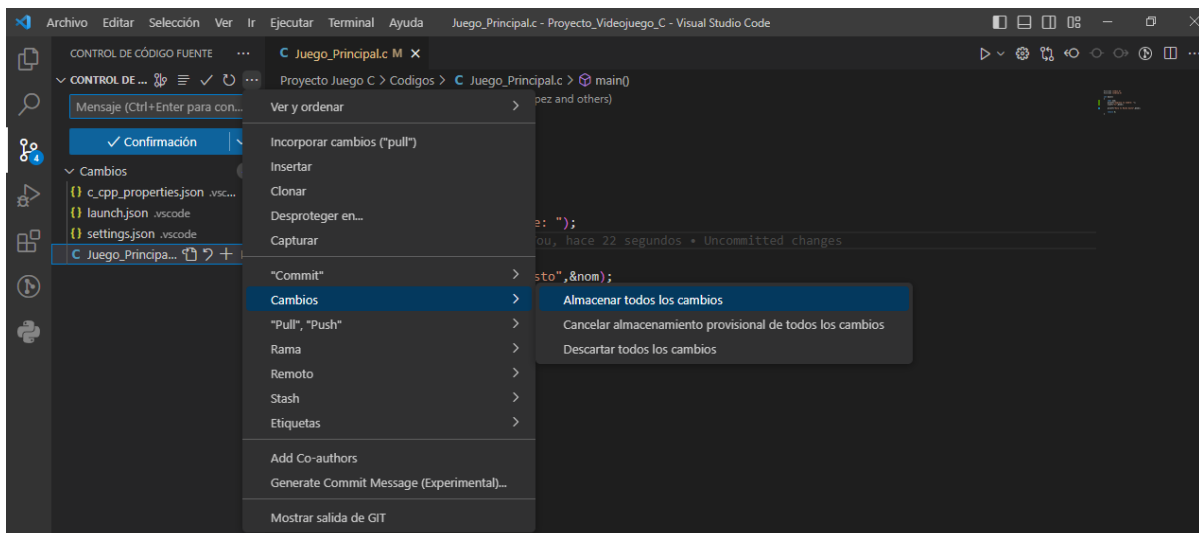
Aparecerá el siguiente menú:



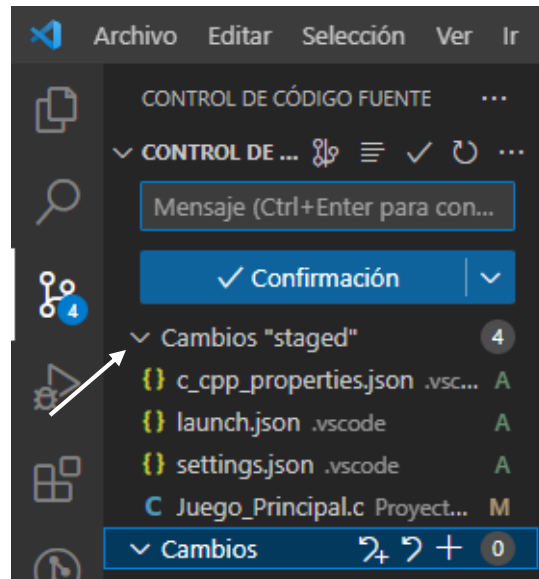
Le daremos “click” a los tres puntitos donde está la barra que dice CONTROL DE...



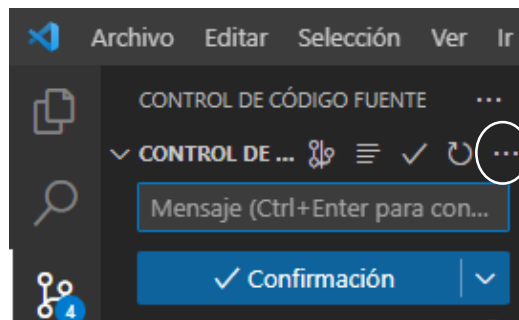
Nos aparecerá un menú desplegable donde buscaremos la palabra Cambios. Fijamos el cursor encima de el y se nos desplegará un sub'menu donde seleccionaremos: Almacenar todos los cambios.



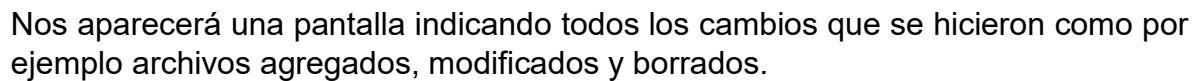
Tendremos un cambio en el menú.



Nuevamente daremos "click" en los tres puntos donde esta la barra que dice: CONTROL DE...



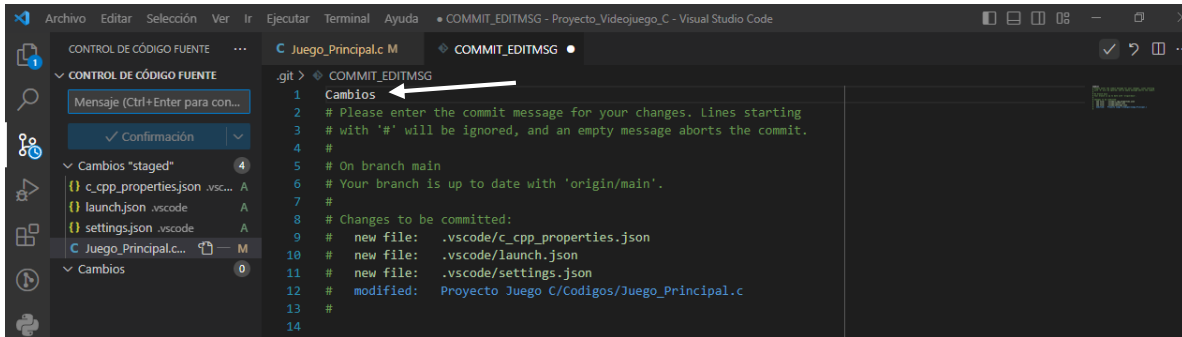
Y damos click.



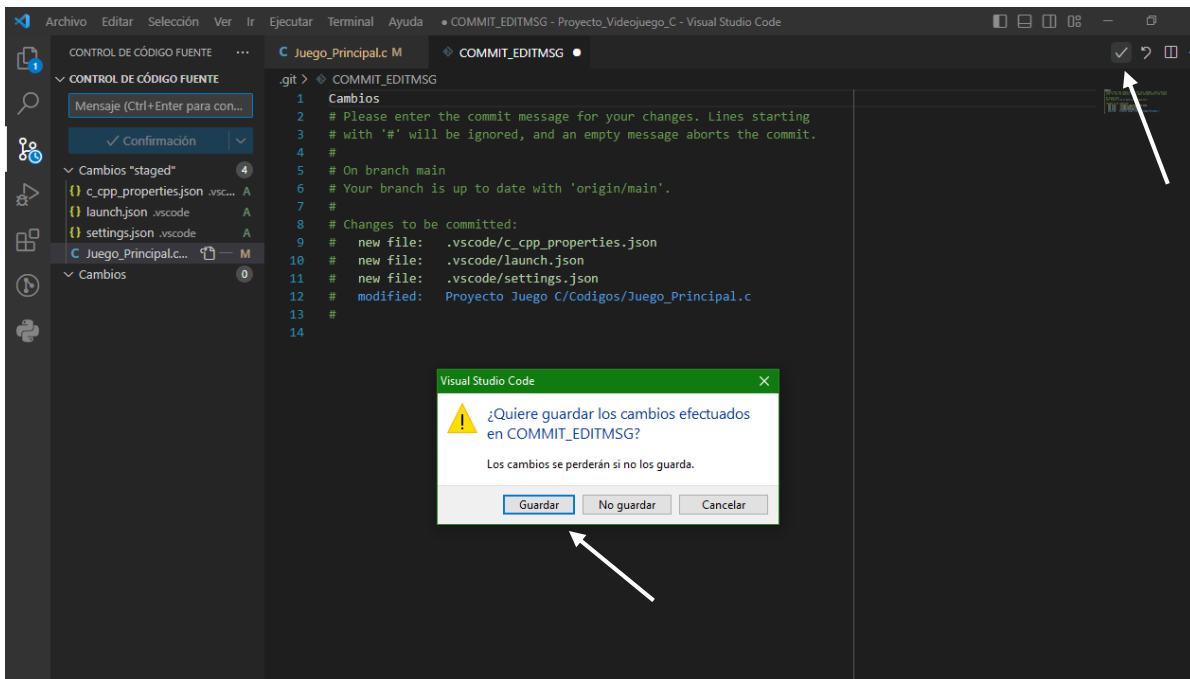
En el primer renglón escribiremos un mensaje donde se puede escribir lo que quieras. Por lo general hacemos referencia a lo realizado en los cambios.

En nuestro caso escribimos cambios.

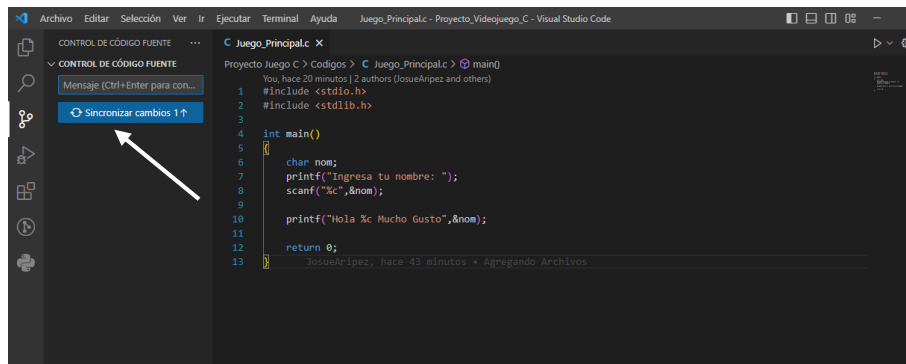
Esta parte de escribir un mensaje es de carácter obligatorio.



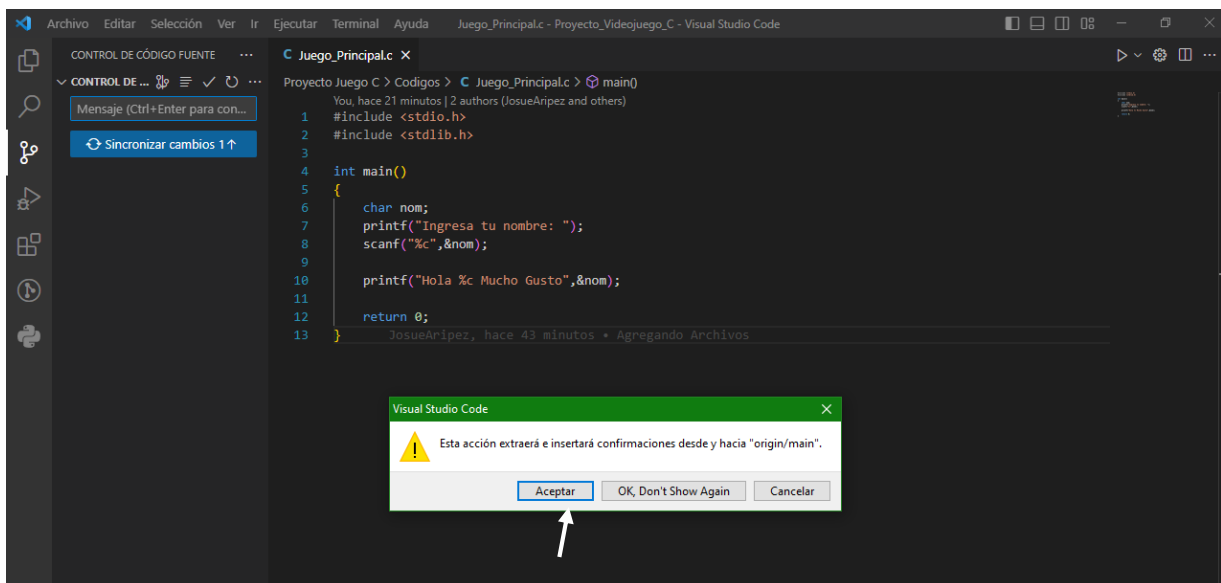
Nos colocamos en la flecha de confirmación, ubicada en la esquina superior derecha. Le daremos “click” y nos saldrá una ventana emergente preguntándonos que si queremos guardar los cambios.



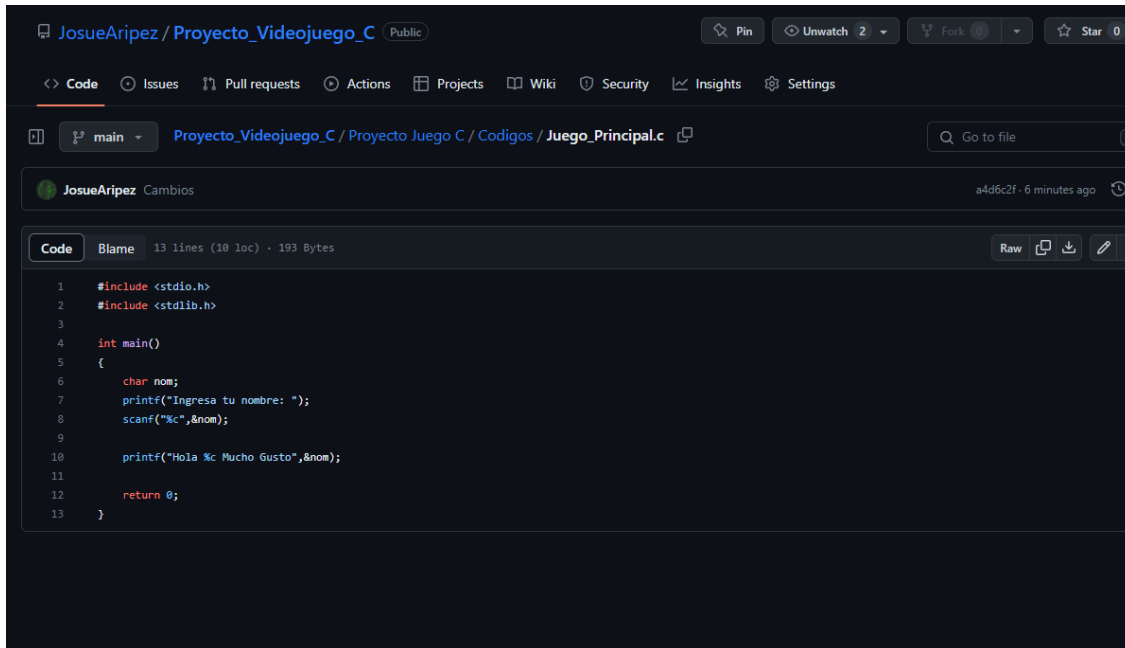
En el menu de la drecha nos aparecerá una nueva opción que indicará el Sincronizar Cambios a la cual le daremos “click”.



Nos preguntará si queremos hacer los cambios en el “main” y le daremos en Aceptar.



Esperamos y estarían listos los cambios efectuados. Solo para confirmar que todo esto se haya realizado de manera exitosa nos dirigimos a nuestro repositorio de “GitHub” y vamos al archivo donde se hicieron los cambios para corroborar que se hayan realizado correctamente.



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     char nom;
7     printf("Ingresa tu nombre: ");
8     scanf("%c",&nom);
9
10    printf("Hola %c Mucho Gusto",&nom);
11
12    return 0;
13 }
```

En este manual, hemos explorado de manera detallada el proceso de enlace entre “VS Code” y “GitHub”, dos herramientas fundamentales para el desarrollo de “software”. A lo largo de estas páginas, hemos aprendido cómo configurar correctamente nuestro entorno de desarrollo en “VS Code” y establecer una conexión sólida con un repositorio en “GitHub” y aprovechar al máximo la integración entre ambas plataformas.

Al enlazar “VS Code” con “GitHub”, hemos descubierto una serie de ventajas significativas que nos permiten optimizar nuestro flujo de trabajo y potenciar nuestra productividad. Mediante la integración de “Git”, podemos realizar un control de versiones efectiva, lo que nos ayuda a rastrear y administrar los cambios en nuestro código fuente de manera eficiente. Además, hemos aprovechado las capacidades de colaboración y seguimiento de problemas de “GitHub”, lo que facilita la colaboración en equipo y mejora la comunicación en el desarrollo de proyectos.

La capacidad de personalización de “VS Code” nos ha permitido adaptar el IDE a nuestras preferencias y necesidades específicas, brindándonos una experiencia de desarrollo más eficiente y cómoda. Además, hemos aprendido a utilizar las

herramientas integradas en “VS Code” para interactuar con el repositorio de “GitHub”, lo que nos ha permitido ampliar las funcionalidades de “VS Code” y adaptarlo aún más a nuestros requerimientos individuales, lo que facilita la colaboración en equipo y mejora la comunicación en el desarrollo de proyectos.

Esperamos que este manual te haya proporcionado los conocimientos y las habilidades necesarias para establecer una conexión sólida entre “VS Code” y “GitHub”, y que te sientas más capacitado para aprovechar al máximo estas dos herramientas esenciales en tus tareas o trabajo diario como desarrollador de software o practicante en el área de “software”.