



**UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE**

**SEGUNDO SEMESTRE  
CARRERAS TECNICAS**

**DEPARTAMENTO DE CIENCIAS EXACTAS**

**PRIMER PARCIAL**

**“INFORME DE PROGRAMACION ORIENTADA A OBJETOS”**

**AUTOR:**

**Loachamin Tipan Nayeli Scarleth**

**PARALELO:**

**1323**

**DOCENTE:**

**LUIS ENRIQUE JARAMILLO MONTAÑO**

**SANGOLQUI – ECUADOR**

## **1. Introducción**

GitHub se ha consolidado como una de las plataformas más importantes en el ámbito del desarrollo de software ofreciendo un entorno colaborativo en la nube que facilita tanto a desarrolladores individuales como a equipos de trabajo la creación, gestión y mantenimiento de proyectos de software. Su integración con Git, un sistema de control de versiones distribuido, permite un manejo eficiente y seguro del código fuente lo que hace que la plataforma sea indispensable en la actualidad para desarrolladores de todo el mundo. A lo largo de los años, GitHub ha evolucionado más allá de un simple repositorio de código, incorporando una variedad de herramientas colaborativas que fomentan la comunicación y la eficiencia en los equipos de trabajo. Funciones como revisiones de código, gestión de incidencias, integración continua y despliegue automático permiten que los proyectos sean gestionados de manera más fluida y sin los obstáculos tradicionales que enfrentan los desarrolladores en entornos más aislados. Además, GitHub ha logrado crear una comunidad global de desarrolladores que comparten conocimientos, resuelven problemas y colaboran en proyectos de código abierto, convirtiéndolo en un espacio clave para el intercambio de ideas y la innovación tecnológica. Gracias a su interfaz intuitiva y su amplia documentación, GitHub es accesible tanto para principiantes como para profesionales experimentados. Desde proyectos pequeños hasta grandes aplicaciones empresariales, esta plataforma ofrece una flexibilidad que se adapta a diversas necesidades, permitiendo que los equipos gestionen desde simples scripts hasta desarrollos de gran escala. De este modo, GitHub se ha establecido no solo como una herramienta poderosa para la gestión de código fuente, sino también como un referente esencial en el ecosistema de desarrollo de software moderno, apoyando la colaboración y el progreso de proyectos de diversa índole en todo el mundo.

## 2. Objetivos

### 2.1 Objetivo General

Desarrollar competencias fundamentales en el uso de GitHub, enfocándose en la creación de archivos locales, la gestión de repositorios mediante comandos en la terminal y la documentación del proceso, con el fin de fomentar habilidades prácticas en colaboración y control de versiones.

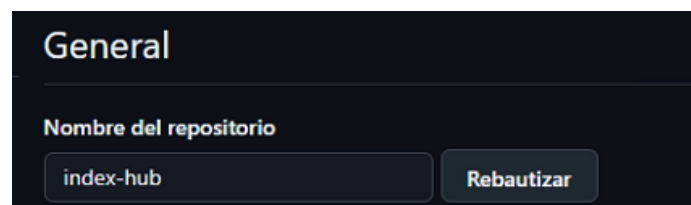
### 3. Objetivos Específicos

- Crear una cuenta en GitHub para comenzar a gestionar proyectos y repositorios en línea.
- Perfeccionar el manejo de comandos básicos en Git para iniciar proyectos, realizar confirmaciones y cargar actualizaciones a repositorios en GitHub, mejorando así las destrezas en control de versiones.
- Registrar de forma clara y organizada el proceso de creación y carga de archivos en GitHub, promoviendo la ordenación y la comunicación efectiva en proyectos colaborativos.

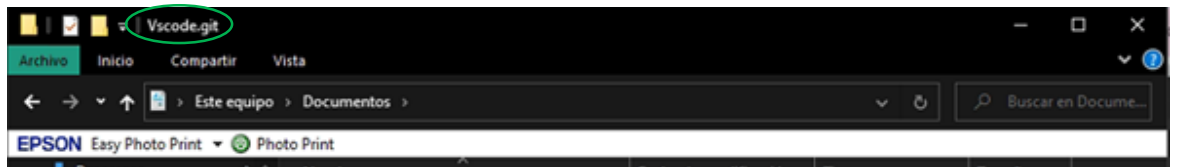
## 4. Marco Teórico

A continuación, vamos a explicar cómo sincronizar la plataforma GitHub con Visual Studio Code y crear un repositorio en GitHub.

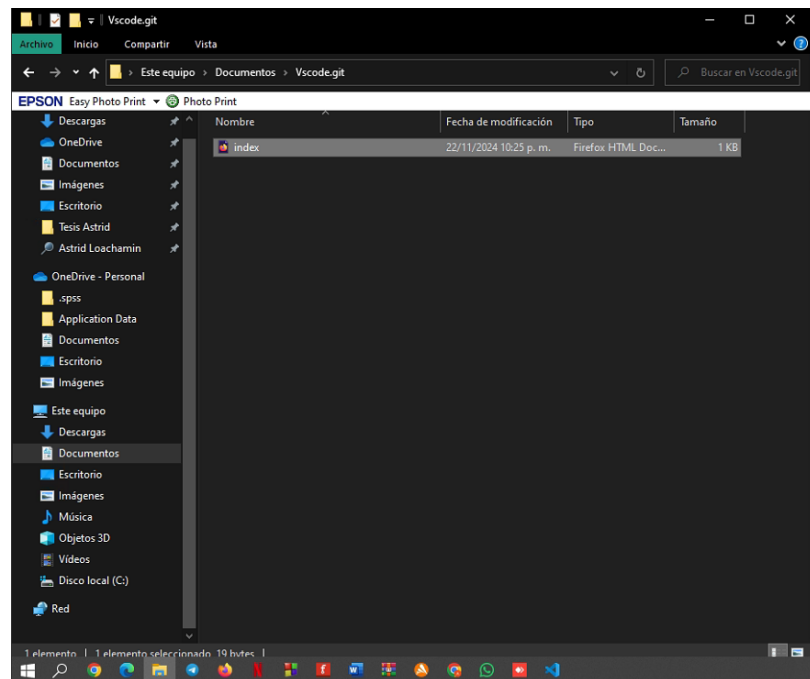
- El primer paso que realizamos es asignarle un nombre al repositorio, en este caso "Index-Hub", y luego crearlo.



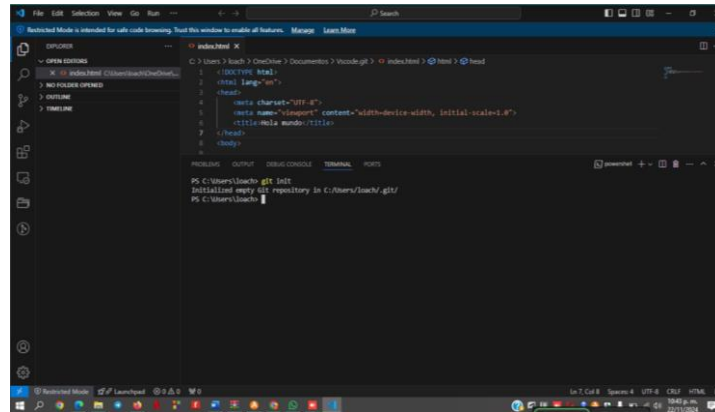
- Una vez creado, se generó la carpeta llamada "Vscode.git" con un archivo HTML vacío en su interior.



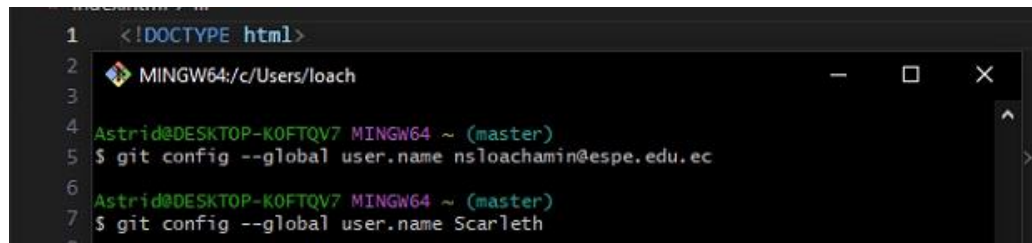
- El siguiente paso es abrir la carpeta con Visual Studio Code.



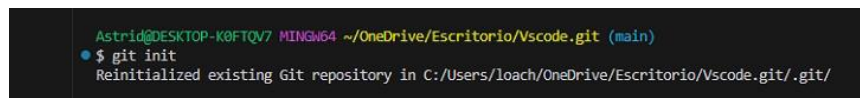
- Después de abrir Visual Studio Code, creamos un terminal y utilizamos el código *git init*.



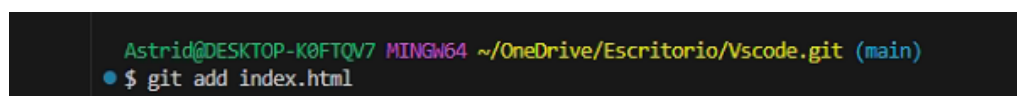
- Después de iniciar ese código, use Git Bash con los siguientes comandos para sincronizar la cuenta de GitHub con Visual Studio Code.



- En la consola, utilizamos los códigos que nos otorgó la página para sincronizar con el Origin/main.



- Usamos los comandos *pull* y *push*.



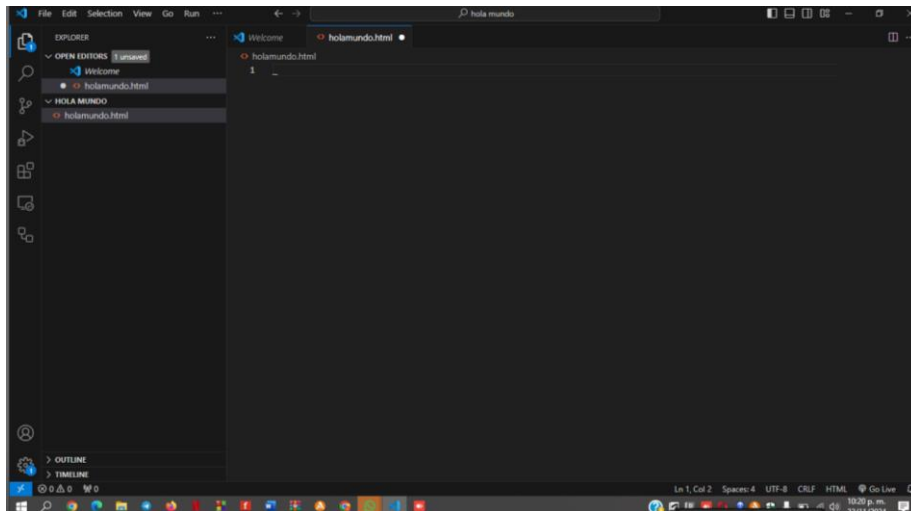
- Use este código para poder rebasar y realizar modificaciones entre GitHub y

Visual Studio Code: *git rebase origin/main*

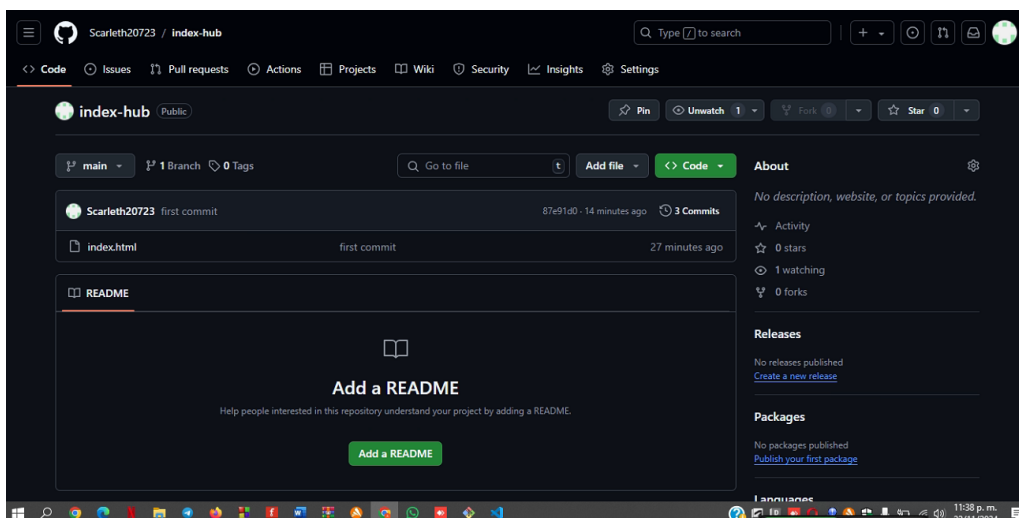
```
Astrid@DESKTOP-K0FTQV7 MINGW64 ~/OneDrive/Escritorio/Vscode.git (main)
$
```

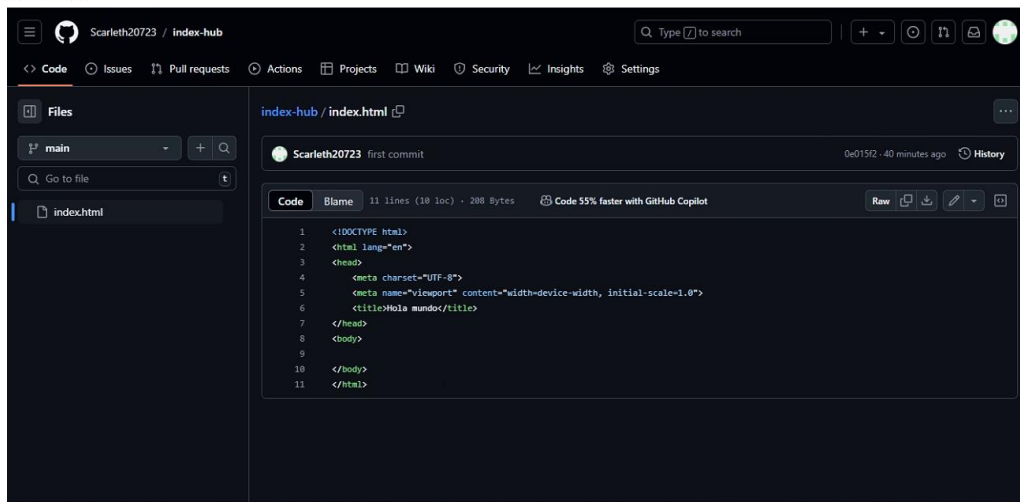
Esto funciona para evitar que el origin/main dé el error de **"refused"**.

- Creamos una nueva carpeta o un nuevo archivo con el nombre **"Hola Mundo"**.



- Y verificamos que se haya subido correctamente a GitHub.





```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Hola mundo</title>
7 </head>
8 <body>
9
10 </body>
11 </html>
```

## 5. Conclusiones

- Perfeccionar el manejo de comandos básicos en Git permitirá comenzar proyectos de manera eficiente, realizar confirmaciones de cambios y cargar actualizaciones a un repositorio en línea en GitHub. Este proceso mejorará nuestras destrezas en el control de versiones, lo que facilitará la gestión adecuada del código y optimizará la colaboración en proyectos.
- Registrar de forma clara y organizada el proceso de creación y carga de archivos en GitHub fomentará la ordenación y mejorará la comunicación eficaz entre los miembros de un equipo. Este enfoque contribuirá a que los proyectos colaborativos se gestionen de manera más fluida y sin contratiempos, permitiendo que todos los participantes estén alineados en el desarrollo del código.

## 6. Recomendaciones

- Practicar el uso de comandos de Git y GitHub de manera constante para fortalecer las habilidades y mejorar la eficiencia en el manejo de proyectos.

- Mantener una documentación precisa y detallada del proceso para garantizar que todos los miembros del equipo puedan seguir los pasos correctamente y evitar confusiones.
- Explorar herramientas adicionales de GitHub para optimizar la colaboración y facilitar la gestión de proyectos a través de funciones como revisiones de código, gestión de incidencias y flujos de trabajo automáticos.

## Bibliografía

- Chacon, S., & Straub, B. (2021). *Pro Git* (2nd Edition). Apress.
- GitHub Docs. (2024). *Documentación oficial de GitHub*.
- Loeliger, J., & McCullough, M. (2012). *Version Control with Git: Powerful tools and techniques for collaborative software development*. O'Reilly Media
- Scott Chacon. (2014). *Git Pocket Guide*. O'Reilly Media.
- Fernández, D. (2018). *GitHub para desarrolladores: Aprende a colaborar y compartir código de manera efectiva*. Marcombo.

Link de GitHub: <https://github.com/Scarleth20723/index-hub.git>