

# UNIVERSIDAD TECNOLÓGICA DEL SURESTE DE VERACRUZ

## TECNOLOGÍAS DE LA INFORMACIÓN



### Actividad V.- Primera actividad en Git

#### **PRESENTA**

HERNÁNDEZ CONTRERAS JORGE ANDRÉS

#### **CUATRIMESTRE Y GRUPO**

NOVENO, "901"

#### **MATERIA:**

PROGRAMACIÓN WEB INTEGRAL

#### **NOMBRE DEL DOCENTE**

ING. JUAN JOSÉ PACHECO REYES

Cd. Nanchital, Ver., a 12 de julio de 2023

---

## ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS .....	2
OBJETIVO.....	3
INTRODUCCIÓN.....	4
DESARROLLO DE LA INVESTIGACIÓN .....	5
NOMBRE DEL TEMA A INVESTIGAR .....	6
CONCLUSIÓN .....	11

## **OBJETIVO**

El alumno realizará un proyecto y lo subirá a GitHub con ayuda de los comandos vistos en clase.

# INTRODUCCIÓN

GitHub es una plataforma líder en el mundo del desarrollo de software y la colaboración entre programadores. Es ampliamente utilizada para el control de versiones y la gestión de proyectos, brindando una infraestructura sólida para el trabajo en equipo y la contribución abierta.

La utilidad de GitHub radica en su capacidad para facilitar el desarrollo de software de manera colaborativa y transparente. Permite a los desarrolladores almacenar y administrar sus proyectos en repositorios, que son como carpetas virtuales donde se almacenan todos los archivos, código fuente, documentación y otros recursos relacionados con un proyecto. Los repositorios en GitHub son accesibles para todo el equipo de desarrollo, lo que fomenta la colaboración y la coordinación.

Uno de los aspectos más destacados de GitHub es su sistema de control de versiones distribuido, basado en Git. Esto permite a los desarrolladores rastrear y gestionar los cambios en el código fuente de manera eficiente, lo que facilita la colaboración en equipos grandes y la gestión de ramas de desarrollo paralelas.

Además de ser una plataforma de colaboración, GitHub también se ha convertido en una comunidad vibrante y activa, donde los desarrolladores pueden descubrir proyectos interesantes, colaborar en proyectos de código abierto y aprender de otros programadores. Los usuarios pueden contribuir a proyectos existentes mediante solicitudes de extracción, informar problemas o errores a través de los "issues" y colaborar en el desarrollo de software en un entorno abierto y transparente.

Otra característica clave de GitHub es su integración con una amplia variedad de herramientas y servicios, lo que brinda a los desarrolladores la flexibilidad de construir sus propios flujos de trabajo personalizados. Esto incluye integraciones con sistemas de construcción y pruebas automatizadas, herramientas de despliegue continuo, servicios de integración continua y mucho más.



1

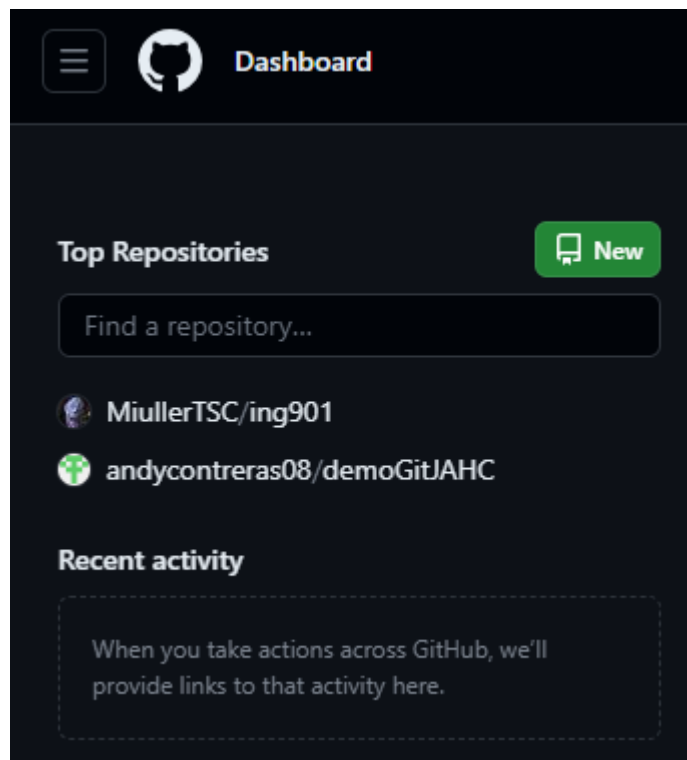
---

---

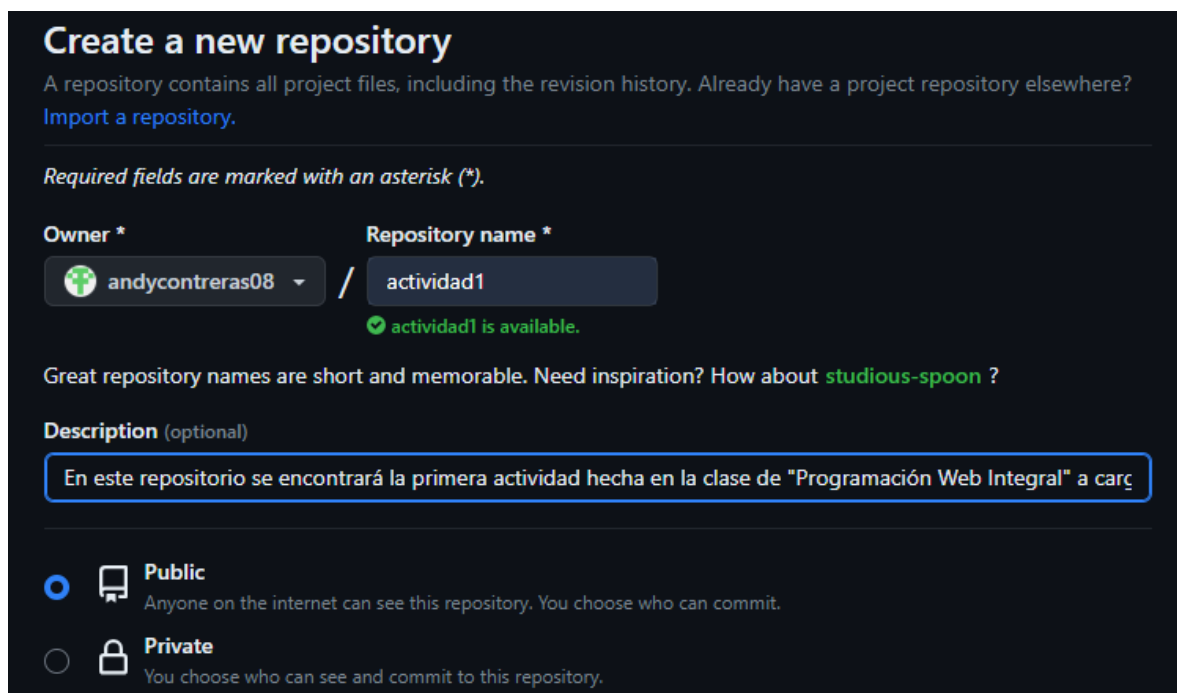
## **DESARROLLO DE LA INVESTIGACIÓN**

## PROYECTO EN GIT HUB

1.- Iniciamos sesión en GitHub & hacemos click en el apartado que dice “Top Repositories” y le damos al botón de “New”.



2.- Colocamos un nombre y una descripción a nuestro repositorio, además de darle una privacidad pública.

A screenshot of the 'Create a new repository' form in GitHub. The title is 'Create a new repository' in bold. Below it is a subtitle: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. A note states: 'Required fields are marked with an asterisk (\*).' The form has two main sections: 'Owner \*' and 'Repository name \*'. The 'Owner' dropdown is set to 'andycontreras08'. The 'Repository name' input field contains 'actividad1', and a green checkmark message below it says 'actividad1 is available.'. Below these fields is a suggestion: 'Great repository names are short and memorable. Need inspiration? How about **studious-spoon** ?'. The 'Description (optional)' section has a text area with the content: 'En este repositorio se encontrará la primera actividad hecha en la clase de "Programación Web Integral" a car...'. At the bottom, there are two radio button options for visibility: 'Public' (selected) and 'Private'. The 'Public' option is described as 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is described as 'You choose who can see and commit to this repository.'

3.- Procedemos a crear nuestro reposito dando click en el botón de “Create Repository”-


## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*


Owner \*

Repository name \*

 andycontreras08

 / 


actividad1

 actividad1 is available.


Great repository names are short and memorable. Need inspiration? How about [studious-spoon](#) ?

**Description** (optional)

En este repositorio se encontrará la primera actividad hecha en la clase de "Programación Web Integral" a carg

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

**Initialize this repository with:**

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**


.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**

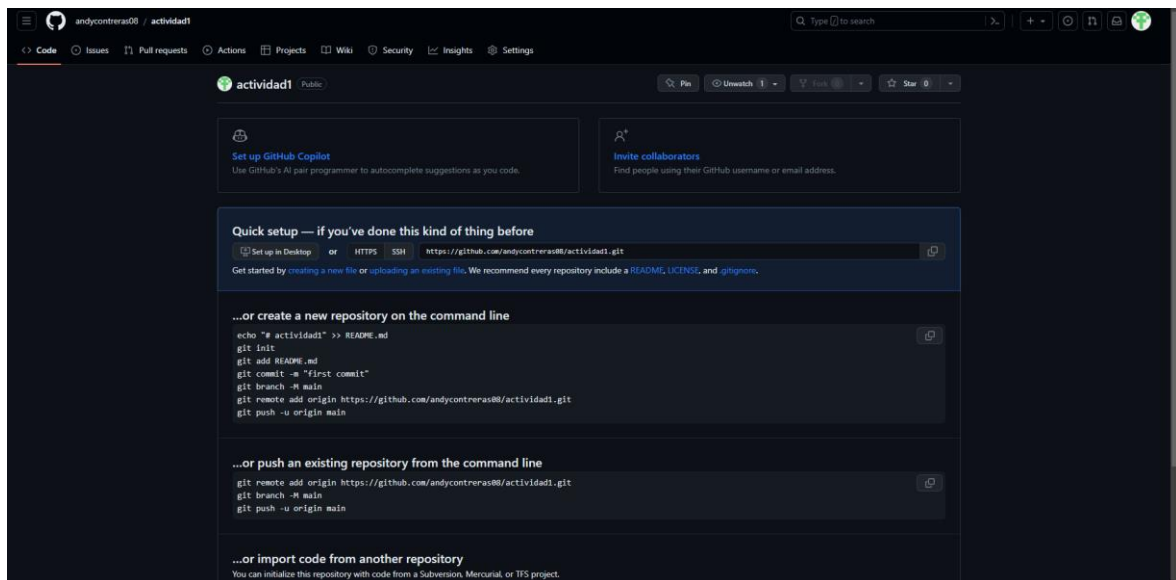
License: **None**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

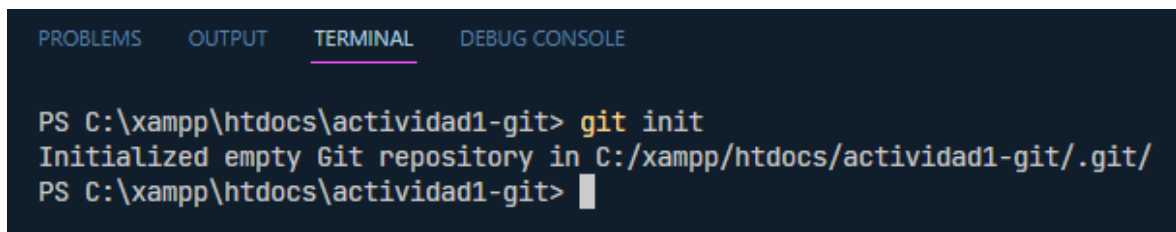
 You are creating a public repository in your personal account.

Create repository

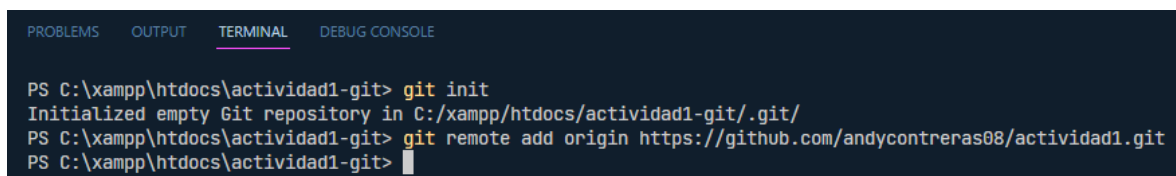
4.- Nos arrojará a la interfaz principal de nuestro repositorio.



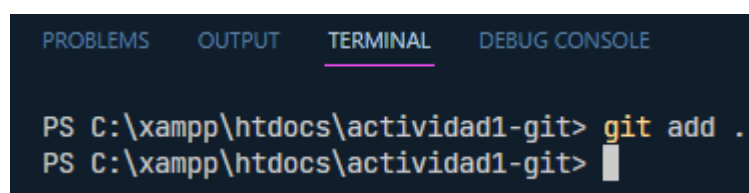
5. Procederemos ir a Visual Studio Code, donde pondremos abriremos una nueva terminal e iniciaremos Git con el comando “git init”.



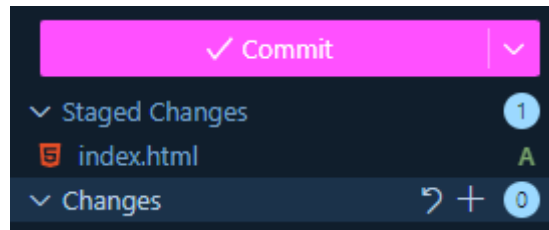
6.- Continuamos creando un repositorio remoto, conectándolo con el que ya tenemos en GitHub, utilizando el siguiente comando:



7.- Podemos ver que los archivos tienen una letra U al lado, así que realizamos el comando “git add .” para que se guarden todos los cambios que se deban de guardar.







8.- Una vez hecho esto, realizamos el primero commit para que todo este hecho utilizando el comando “git commit -m "Primer commit"”.

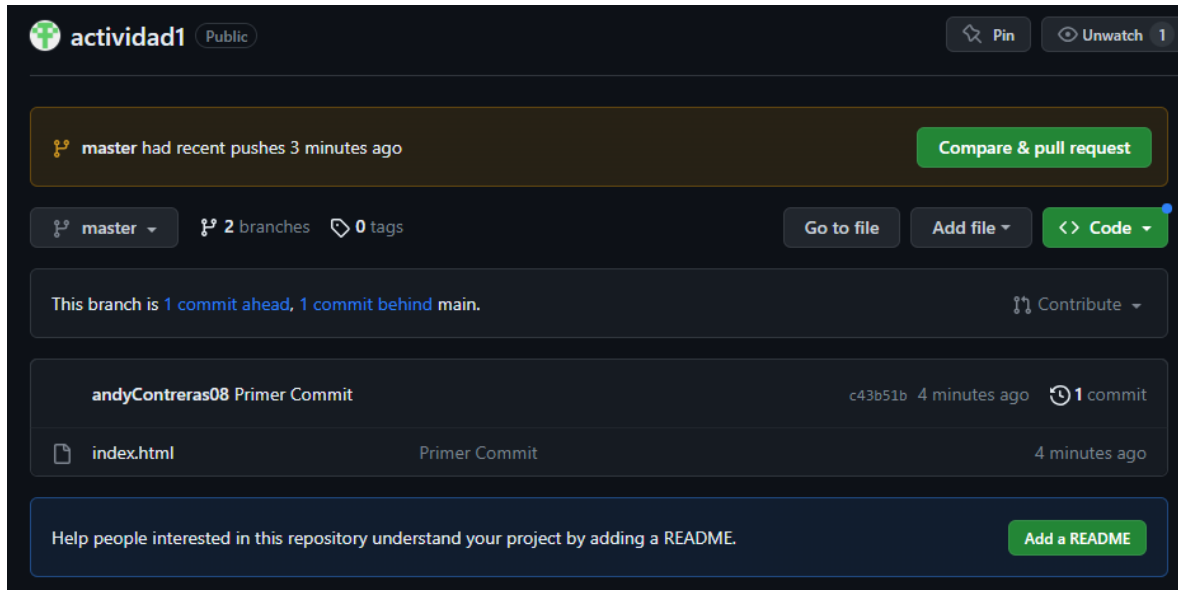
```
PS C:\xampp\htdocs\actividad1-git> git add .
PS C:\xampp\htdocs\actividad1-git> git commit -m "Primer commit"
[master (root-commit) 10922d6] Primer commit
 1 file changed, 15 insertions(+)
 create mode 100644 index.html
PS C:\xampp\htdocs\actividad1-git> █
```

9.- Ahora para subir nuestro proyecto a GitHub, utilizaremos el siguiente comando.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\xampp\htdocs\actividad1-git> git push -uf origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 523 bytes | 523.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/andycontreras08/actividad1/pull/new/master
remote:
To https://github.com/andycontreras08/actividad1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\xampp\htdocs\actividad1-git> █
```

10- Una vez hecho esto, checamos en nuestro repositorio en GitHub & efectivamente, se encuentra el archivo con el código.



# CONCLUSIÓN

En conclusión, el uso de GitHub proporciona numerosos beneficios y se ha convertido en una herramienta fundamental para el desarrollo de software y la colaboración entre programadores. Algunas de las principales conclusiones sobre el uso de GitHub son las siguientes:

**Colaboración efectiva:** GitHub facilita la colaboración entre desarrolladores al proporcionar una plataforma centralizada para almacenar, administrar y colaborar en proyectos de software. Los repositorios permiten un seguimiento preciso de los cambios, el trabajo en equipo y la coordinación eficiente.

**Control de versiones sólido:** GitHub utiliza Git como sistema de control de versiones distribuido, lo que permite a los desarrolladores rastrear y administrar cambios en el código fuente de manera eficiente. Esto facilita la gestión de ramas de desarrollo paralelas, fusiones de código y la reversión de cambios.

**Comunidad y código abierto:** GitHub es más que una plataforma de desarrollo, es una comunidad activa y colaborativa. Permite a los desarrolladores descubrir proyectos interesantes, colaborar en proyectos de código abierto y aprender de otros programadores. Esto fomenta la transparencia, el intercambio de conocimientos y el crecimiento profesional.

**Integración con herramientas y servicios:** GitHub ofrece una amplia gama de integraciones con herramientas y servicios populares utilizados en el desarrollo de software, como sistemas de construcción, pruebas automatizadas y despliegue continuo. Esto permite a los desarrolladores personalizar sus flujos de trabajo y aprovechar las mejores prácticas de la industria.

**Visibilidad y reconocimiento:** Publicar proyectos en GitHub puede dar visibilidad a tu trabajo y permitir a otros colaborar y contribuir. Además, GitHub proporciona herramientas para seguir el progreso del proyecto, revisar cambios y recibir aportes valiosos de la comunidad.

En general, el uso de GitHub mejora la eficiencia, la colaboración y la calidad del desarrollo de software. Es una plataforma esencial para los desarrolladores que desean gestionar sus proyectos, trabajar en equipo, aprender y contribuir a la

comunidad global de desarrollo de software., experiencia y juicio adecuados para aplicarlos de manera efectiva.