



Uso y configuración de GIT.

Nombre: Angel Daniel Mendoza Melendez

Jesús Xavier Valdés Rocha

Matrícula: 440265059, 860075191

Materia: Métodos y módulos para el
desarrollo de software

Profesor: Julio Cesar Villa López

Campus: Cumbres

Fecha de entrega: 06/13/24

Investiga y contesta:

1. ¿Cómo se establece el nombre de usuario en git?
En forma de comando. Para establecer el nombre es:
`git config --global user.name "Nombre del usuario"`
2. ¿Cómo se establece el correo del usuario en git?
En forma de comando.
`git config --global user.email "Correo"`
3. ¿Cómo se activa el coloreado de salida en git?
En forma de comando.
`git config --global color.ui auto`
4. ¿Cómo se muestra el estado original en los conflictos de git?
En forma de comando.
`git checkout --conflict=diff3`
5. ¿Cómo se muestra la configuración?
En forma de comando.
`git config --list`
6. ¿Cómo se crea un repositorio en GIT?
En forma de comando.
`git init`
7. ¿Cómo se muestra el estado de un repositorio en git?
En forma de comando.
`git status`
8. ¿Cómo se muestra las diferencias entre versiones en git?
En forma de comando.
`git diff`
9. ¿Cómo se puede volver a un directorio de trabajo o versión anterior?
En forma de comando.
`git reset --hard HEAD`
10. ¿A qué se refiere con un pull en git?
Este comando nos permite actualizar el repositorio local con los cambios del repositorio remoto.
11. ¿A qué se refiere con un push en git?
Este comando nos ayuda a enviar los commits locales al repositorio remoto.

Practica:

- Crea un repositorio local en Git
- Comparte ese repositorio con cada uno de tus compañeros de equipo del directorio remoto.
- Que cada compañero tenga su escritorio remoto y añada los cambios necesarios.
- Añadan los cambios al stage.
- Registren los cambios.
- Suban los cambios al remoto.
- Realiza las siguientes operaciones sobre la rama:
 - Crea su rama.
 - Ver en que rama se encuentra.
 - Cambiar de rama.
 - Ver los cambios entre ramas.
 - Fusionar ramas
 - Eliminar rama.
- Cada compañero deberá de mostrar su rama y su usuario además de que comparte el mismo repositorio.
- Para un pull request:
 - Crear la rama.
 - Añada su nombre de usuario en README.md en el apartado Colaboradores.
 - Sube los cambios en local.
 - Subir los cambios.
 - Ir a github y solicitar una pull request.
 - Agregar un comentario.

Codigo.

Ser requiere que se genere lo siguiente:

Codificación de botón de eliminar y botón limpiar.

Botón eliminar:

Debe de buscar en base al código y si no existe mencionar que asi es.

Botón Limpiar:

Debe de limpiar todos los campos de texto, esto lo realiza llamando una función que se debe de generar en el código, es decir no escribimos el código en el botón si no que generamos una función y se manda llamar.

BD:

Deberás mostrar el antes y el después.

- Abre tu plantilla y genera las actividades que consideres necesarias para los pasos anteriores.

- Genera las historias de usuario.
- Pasamos a codificar.

A entregar:

- Planificación e historias de usuario en repositorio que compartirás con el docente.
- Captura de pantalla de cada uno de los pasos solicitados.
- BD
- Interfaz funcionando.
- Liga o enlace a tu repositorio.

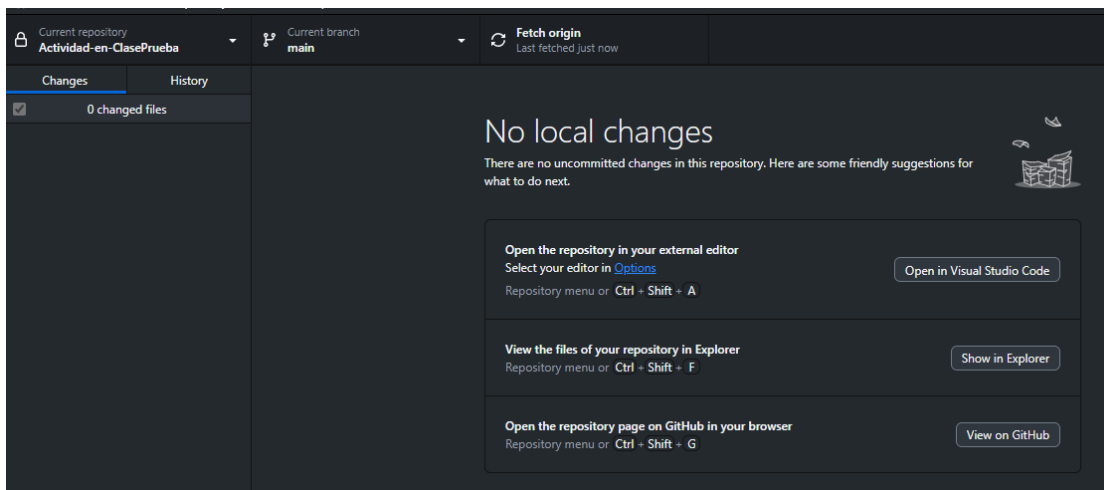
Historias de usuario.

Historia de Usuario: Botón eliminar	
Número: 1	Usuario: Usuario a cargo de la administración de los productos
Nombre historia: Dar funcionalidad al botón Eliminar en la interfaz	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 5	Iteración asignada: 1
Programador responsable: Angel	
Descripción: Dar funcionalidad al botón Eliminar que aparece en la interfaz. Primero es necesario que se busque el artículo, en caso de ser encontrado se indicará que se puede borrar, después pedirá una confirmación de borrar el artículo. En caso de que no se encuentre el artículo, se tendrá que indicar que le articulo a eliminar no existe.	

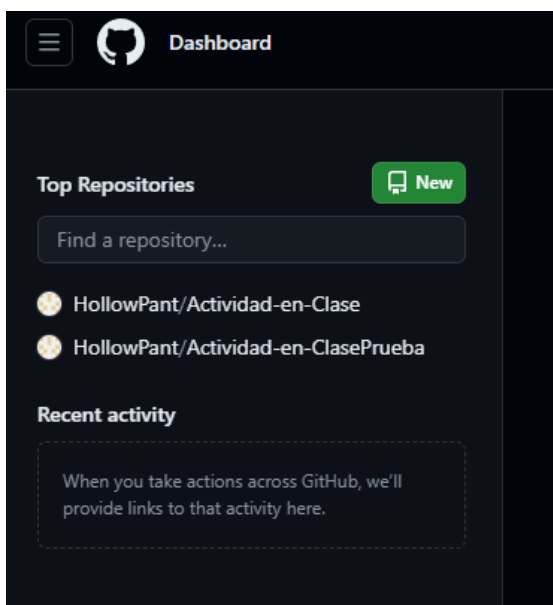
Historia de Usuario: Botón Limpiar	
Número: 1	Usuario: Usuario a cargo de la administración de los productos
Nombre historia: Dar funcionalidad al botón Limpiar de la interfaz	
Prioridad en negocio: Alta	Riesgo en desarrollo: bajo
Puntos estimados: 5	Iteración asignada: 1
Programador responsable: Xavier	
Descripción: Dar funcionalidad al botón Limpiar que aparece en la interfaz. Se necesita un botón que limpie o vacíe los campos de la interfaz.	

Capturas de pantalla.

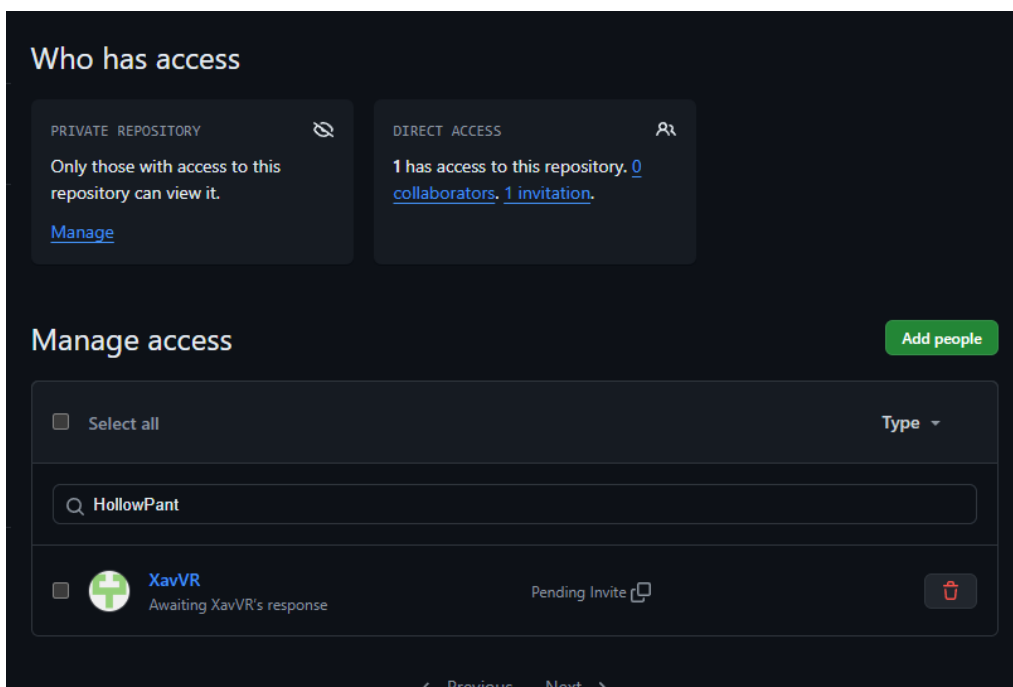
Creación del repositorio local.



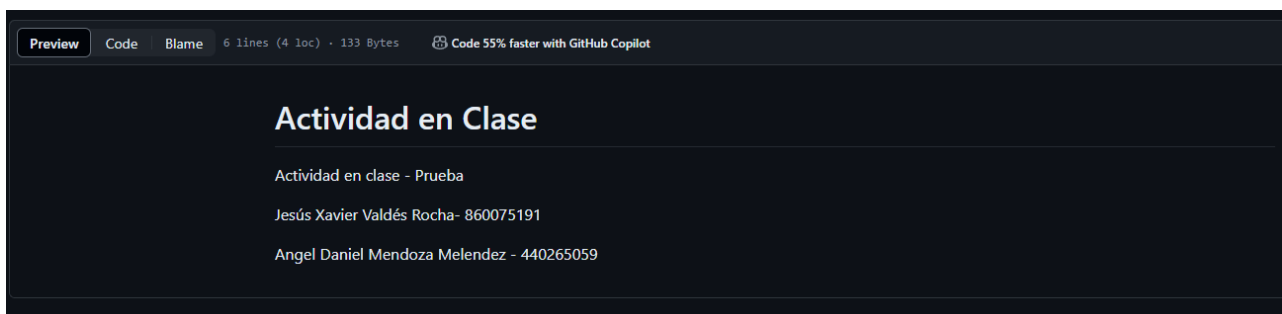
Repositorio en la página web de GitHub.



Compartiendo repositorio con mi compañero.

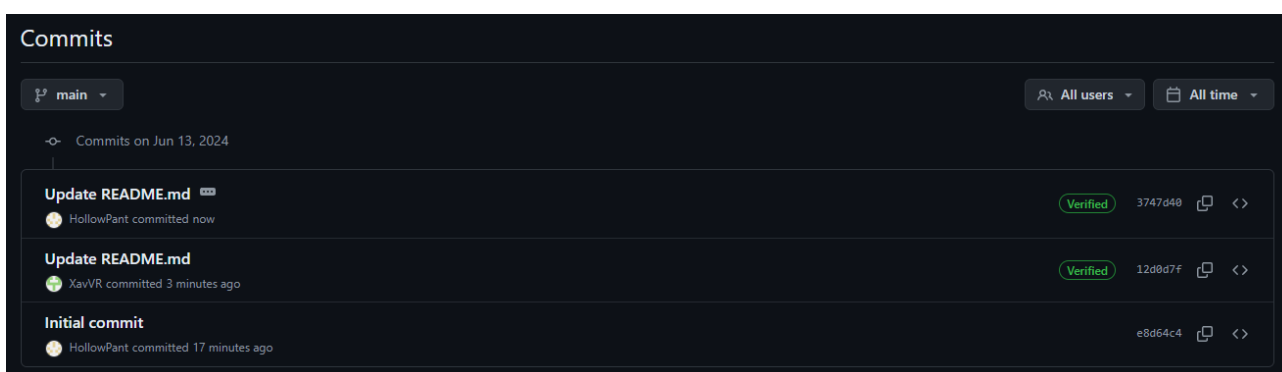


Datos ingresados.



The screenshot shows a GitHub file viewer interface. At the top, there are tabs for 'Preview', 'Code', and 'Blame'. The 'Preview' tab is active, displaying the content of a file named 'Actividad en Clase'. The file content includes the title 'Actividad en Clase', a subtitle 'Actividad en clase - Prueba', and two lines of text: 'Jesús Xavier Valdés Rocha- 860075191' and 'Angel Daniel Mendoza Melendez - 440265059'. The interface also shows file statistics: '6 lines (4 loc) · 133 Bytes' and a note 'Code 55% faster with GitHub Copilot'.

Comprobacion de cambios.



The screenshot shows a GitHub 'Commits' page. The page title is 'Commits'. Below the title, there are filters for 'main' branch and 'All users' and 'All time' time ranges. The page displays a list of commits for the date 'Jun 13, 2024'. The commits are as follows:

Commit Message	Author	Time	SHA-1	Actions
Update README.md	HollowPant	committed now	3747d40	Verified, Copy, Compare
Update README.md	XavVR	committed 3 minutes ago	12d8d7f	Verified, Copy, Compare
Initial commit	HollowPant	committed 17 minutes ago	e8d64c4	Copy, Compare

Link del repositorio.

<https://github.com/HollowPant/Actividad-en-ClasePrueba>