**Git y GitHub**

**Git –versión, muestra la versión que se tiene de git**

**Se crea una carpeta con el formato mkdir**

**C:\Windows\System32>mkdir miPrimerProyecto**

**Se usa el cd para dirigirse a ese proyecto**

**Interfaz de usuario gráfica, Sitio web

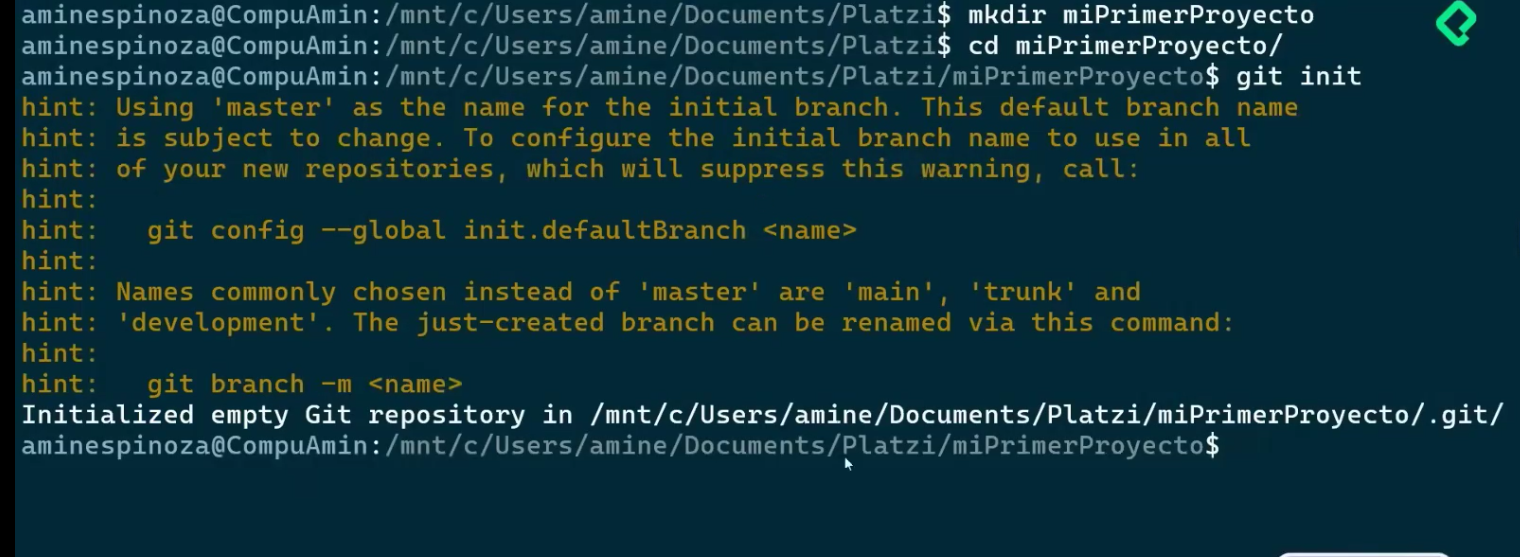
El contenido generado por IA puede ser incorrecto.**

**El primer comando para usar de git es**

**Git init .**

**Nota: se debe tener en cuenta que ahora se usa el main y no el master**

**Si se usa el master podría pasar lo siguiente**

****

**Git config –global init.defaultBranch main**

**Después del git init ., esto funciona para configurar la rama principal por defecto cuando se inicializa el repositorio.**

**Existe un tip, implementar el git –help, para saber como usar comandos, es simple pero eficiente.**

**Git config –global: hace que la configuración que se agrego previamente en el proyecto tambien entre a la configuración global de git, ósea todos los repos**

**Git config –global user.name “Nombre”: genera el nombre de usuario**

**Git config –global user.email “correo”: con ese correo se accede a que todos los repositorios sepan cual es el correo de la persona que esta trabajando en el proyecto**

**Git config --list: nos Brinda la configuración de lo que estábamos configurando, nos da el email, el name y la denominación de la rama sea master o main.**

**Observar la hoja de referencia:** [**git-cheat-sheet-education**](https://education.github.com/git-cheat-sheet-education.pdf)

**Además de la hoja tener en cuenta la documentación de git**

**El git init crea una carpeta oculta y con el comando en git ls -a puede verse**

**Git config –list**

**Texto

El contenido generado por IA puede ser incorrecto.**

**nano: **

**se usa para generar un tipo de archivo**

**git status: Texto

El contenido generado por IA puede ser incorrecto.**

**genera una descripción de los archivos trabajados si esta en rojo es por que fue creado pero no agregado,**

**git add: genera la adición del proyecto o el ombre del archivo, si es para el archivo se genera con:**

**git add testing.txt: git + add + nombre del proyecto.tipo de archivo**

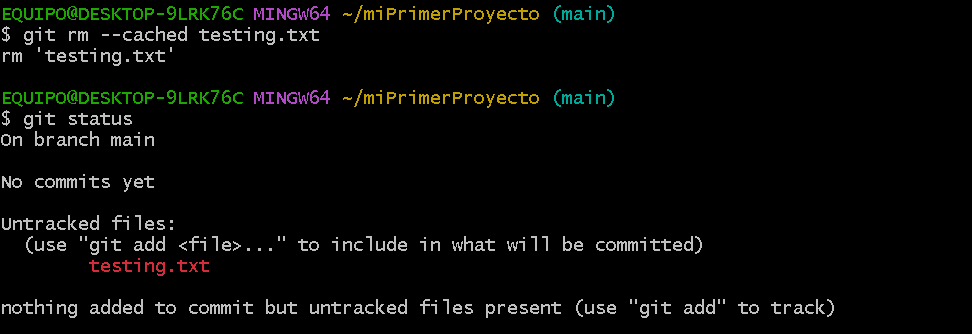
**Texto

El contenido generado por IA puede ser incorrecto.**

**Si ya esta en verde es por que se subio a un stage toca generarle un commit después de subido con un git commit**

****

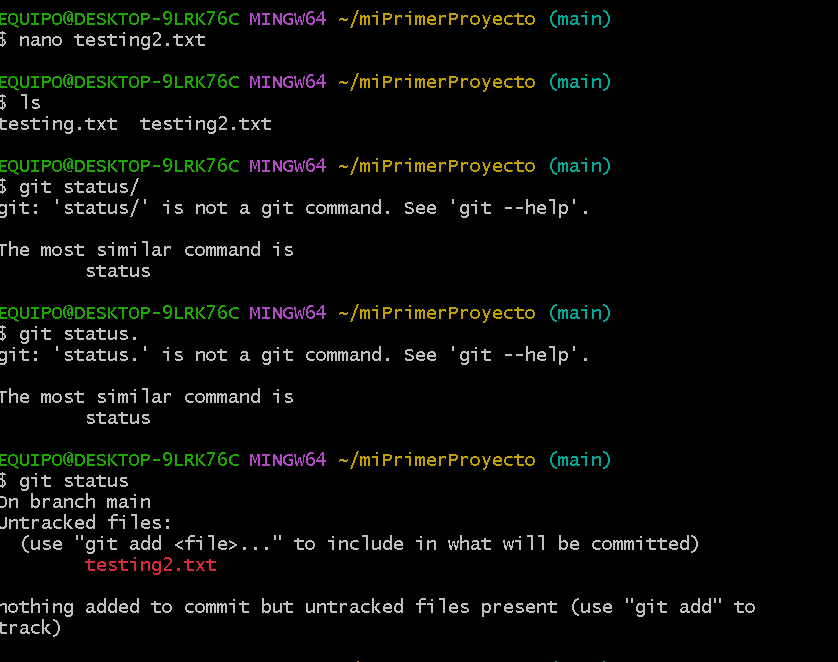
**El stage permite que archivos si y que no permite que se guarden en el control de versiones.**

**El comando git rm—cached nombre del archivo: **

**Permite que se saque un archivo antes de generarle el commit por tal motivo se genera que otravez el testing este en rojo**

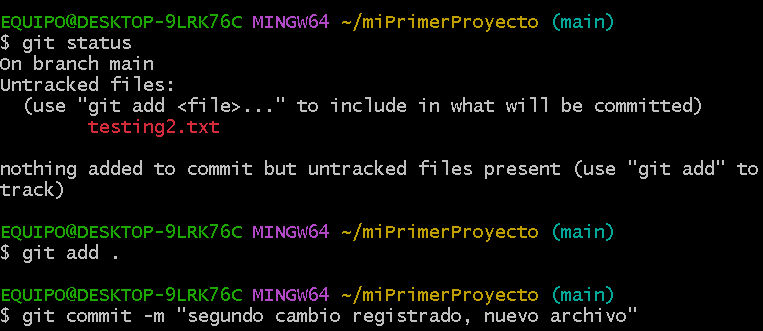
**Con el commit se pone un mensaje con la que se debe indicar que se hizo de manera simple Texto

El contenido generado por IA puede ser incorrecto.el caso aquí no es rtm, sino rm ten cuidado con este código**

****

**Con git add . se manda el testing2.txt a stage**

**Para subirlo se usa el commit**

****

**Git log: se genera la bitácora con los cambios que se han hecho**

**Texto

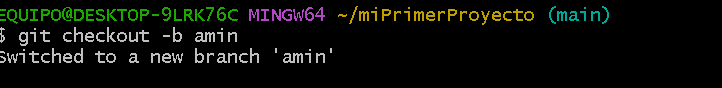
El contenido generado por IA puede ser incorrecto.**

**Texto

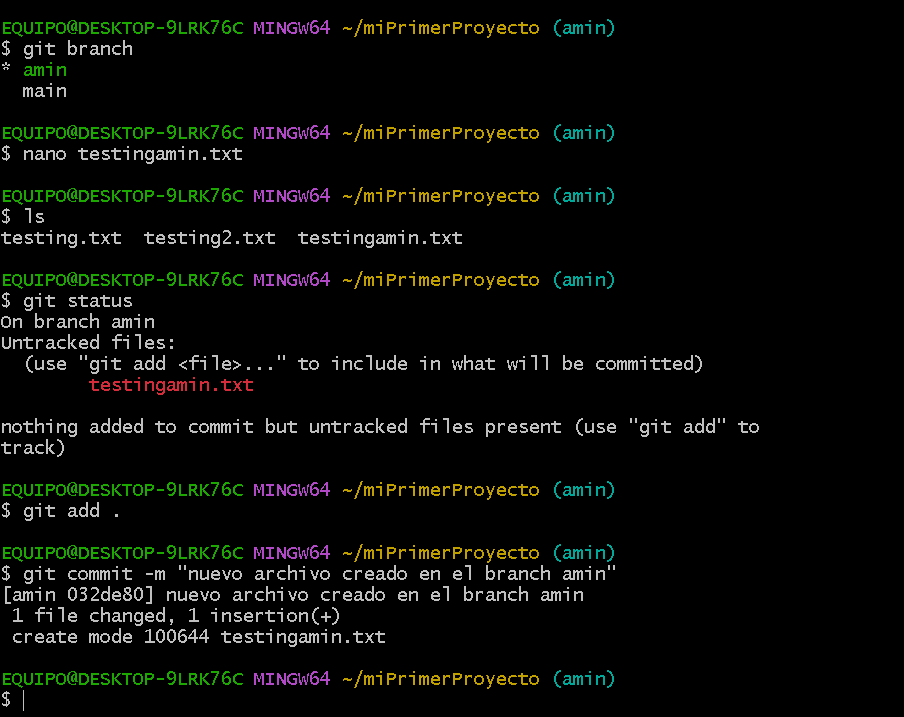
El contenido generado por IA puede ser incorrecto. nótese que en este archivo modificamos un primer archivo y luego usamos el git add y luego el commit para que se suba, al hacer un git add se manejan de manera igual y al hacer un commit se hace esto sin importar sus extensiones**

**//Conflicto de Ramas: ramas y fusión de cambios Branch marge switch y checkpoint:**

**Git checkout -b amin: se genera un cambio a una nueva rama llamada amin**

****

**Git Branch: genera la ubicación del Branch que se va a usar si antes se pasó otro comando**

****

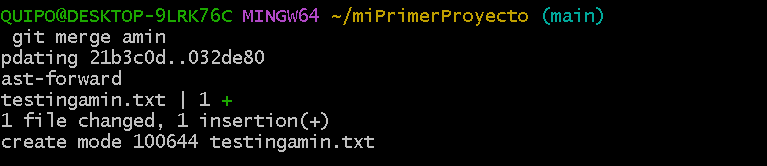
**Ramas y merge: con merge combina una rama principal eneste caso main y una rama individual amin,   
para cambiar de ramas se puede usar**

**Git checkout main**

**O**

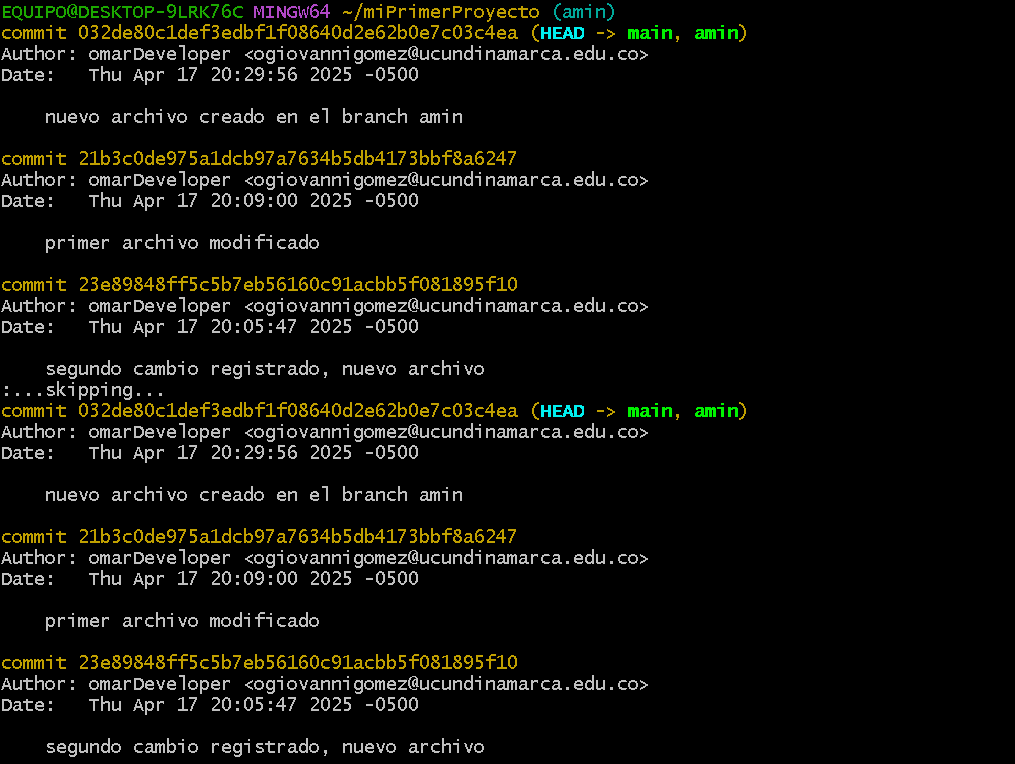
**Git switch amin, cualquiera de los dos hacen exactamente lo mismo**

**Git merge=**

****

**El resultado dice quer a un fast -foward que es un método de unioficacion con una línea un cambio de archivo y una inserción ha sido generada y unificado**

**Con git log se vuelve a mirar el proceso**

****

**Para borrar una rama**

**Git Branch -D amin**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

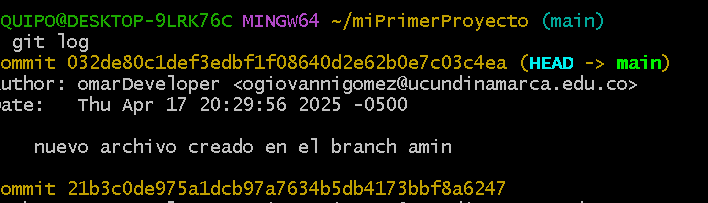
**Cuando una rama se deja de utilizar y se une con el main, ósea si su propósito ya se cumplió se elimina para no tener problema o conflicto con ramas anteriores**

**Comando reset y revert:**

**Sacan de situaciones cuando se tiene problema**

**Reset: se trata de un comando que devuelve a un commit anterior, eliminando los cambios en el historial como si nunca hubieran ocurrido, no solo revierte cambios, sino ayuda a recordar lo que se hizo en commits anteriores, se pued regresar a commits anteriores ponerlo como puntero yn ver los cambios de los archivos y permite explorar el historial de los commits**

**Revert: crea un nuevo commit que revierte los cambios realizados por un commit especific, por un nuevo commit me lleva a uno anterior,**

****

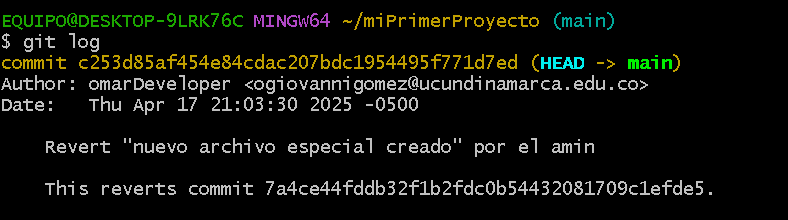
**Esto es un puntero aquella flecha que señala el head al main,**

**El numero después del commit es un hash identifcador sirve para saber cual vamos a eliminar o a modificar etc.**

**Para usar el revert**

**Git revert numero de hash:**

****

****

**Se revierte el commit**

**Despues de usar el revert se puede ver que aparece los archivos que aparecieron antes del error.txtTexto

El contenido generado por IA puede ser incorrecto.**

**Uso del git reset –hard**

****

**El parámetro hard permite que el reset elimine no solo el arhcivo sino el historial de los commits antes de este identificador o después de este identificador, se tienen 3 opciones de parámetros**

**Soft: que permite eliminar archivos.**

**Mix: que me permite regresar los commits**

**Hard: que permite deshacer todos los cambios**

**Los comandos git reset hard debe ser la ultima opción por que está afectando el trtabajo de alguien del equipo.**

**Esto sucede al ejecutar el reset hard**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Recordar para que sirven**

**Git reverrt funciona para poder desbaratar un commit realizado y todos sus cambios dentro de sus archivos**

**El git reset permite deshacer el historial de commits volviendo a un punto previo de las actualizaciones**

**Ambos funcionan para la corrección de errores limpieza de historial y manejo de conflictos**

**La corrección de errores sis se sube un archivo quye no me funciona se usa el git revert y si algunos compañeros que hayan hecho cambios se puede usar mejor el reset, estos comando se debe usar en comunicación con el equi[o**

**Al hacer limpieza con el historial si se tiene un historial largo con un proyectyo solido se quiere deshacer commits para navegar con el comando hgit log**

**Si hayu un conflicto entre los archivos que no se puede solucionar se udsa git reset, el git reset es el ultimo parámetro que se deba usar en los recursis**

**Git tag y git checkout, para el gestionado de versiones:**

**Git tag: es una etiqueta que se aplica a un commit para identificarlo fácilmente en el futuro, se usa para generar una etiqueta a versiones lts, es bueno para usarlo en verisonados, en hitos,**

**El git chekout cambia de una rama para evaluar ciertos cambios en una rama antes de integrarlos a una rama principal**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Esto permite verificar para saber en donde esta el hito de las versiones de un avance mayor, lo ideal es que cada commit tenga un tag, sin embargo genera trabajo extra, la idea es controlar las versiones y se determinaría cuando usar tags**

**Git tag:**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Se implementa para ver el listado de tags q ue tenemos**

**Texto

El contenido generado por IA puede ser incorrecto. con el git tag show se muestrta el commit de referencia, además muestera mas información de lo que sucedió en ese commit**

**Texto

El contenido generado por IA puede ser incorrecto. eesta -d sirve para eliminar el tag**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

**No altera el historial de commits ni los archivos, solo altera la etiqueta**

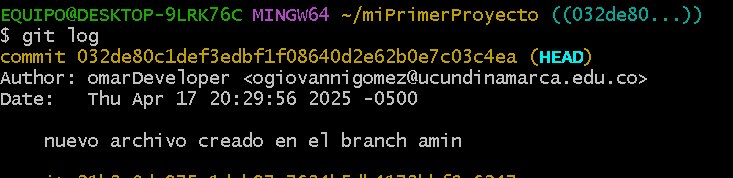
**Git checkout:**

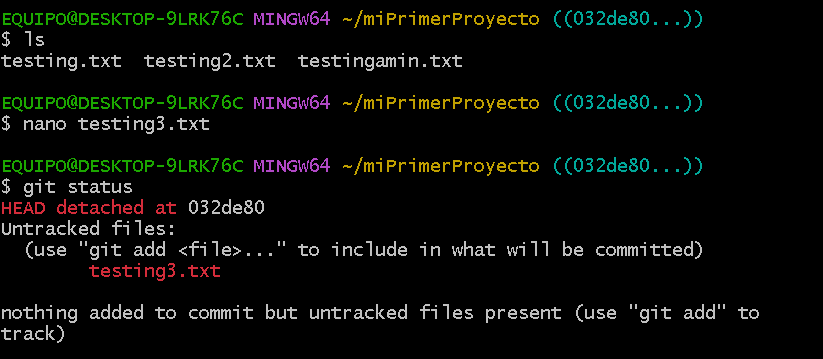
**Si se quiere hacer una revisiond e un punto en particular sin modificar el trabajo actual se puede usar el checkout con el hash a usar**

**Ejp:**

**Texto

El contenido generado por IA puede ser incorrecto.**

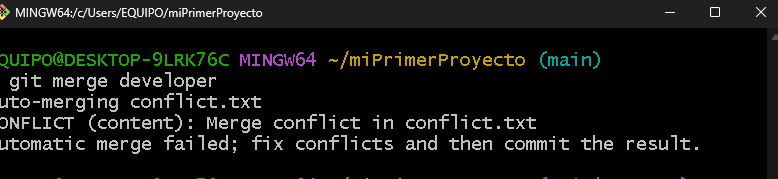
** si se ve no hay un puntero y esto es por que se explorando el historial**

**al ver esto se puede hacer pruebas incluso dentro de ese espacio temporal**

**Texto

El contenido generado por IA puede ser incorrecto. como se puede ver se regresa a la realidad sin afectar el proyecto, por lo que es bueno usar el checkout**

**Resolución de conflictos en git**

****

**Se generaron dos archivos con el mismo nombre en dos rams diferentes**

**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.el error radica en que en ambas ramas hay un archivo tipo conflict.txt y no va a dar solución hasta que se arregle el error**

** esto se genera al vewr el archivo con el conflicto metiéndose con el comando nano conflicto.txt**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Se modifica lo del gnu nano a esto**

**Interfaz de usuario gráfica, Texto, Aplicación

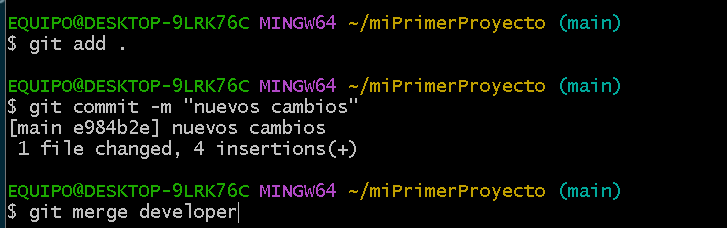
El contenido generado por IA puede ser incorrecto.**

**De acuerdo a esa modificación se tiene:**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Después de realizado los cambios y resolver el conflicto se tiene:**

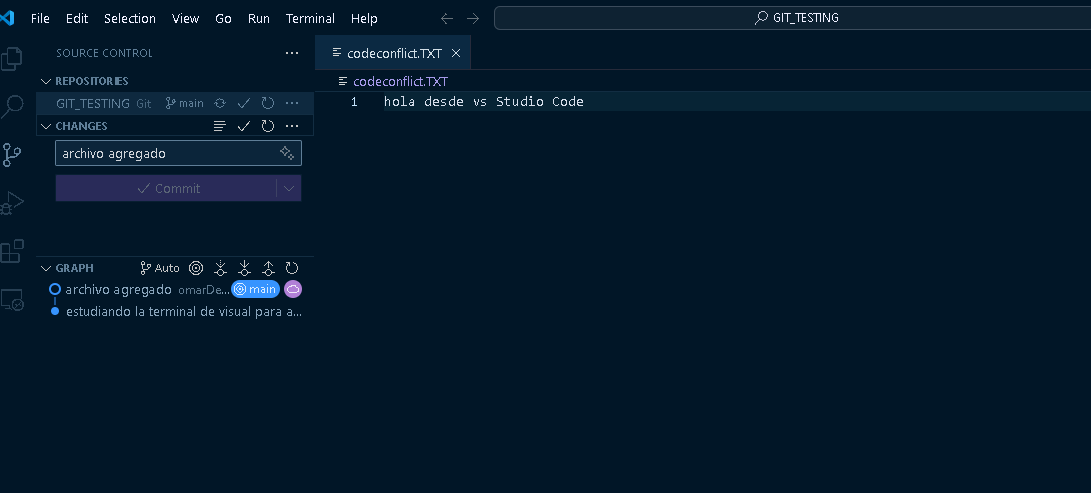
****

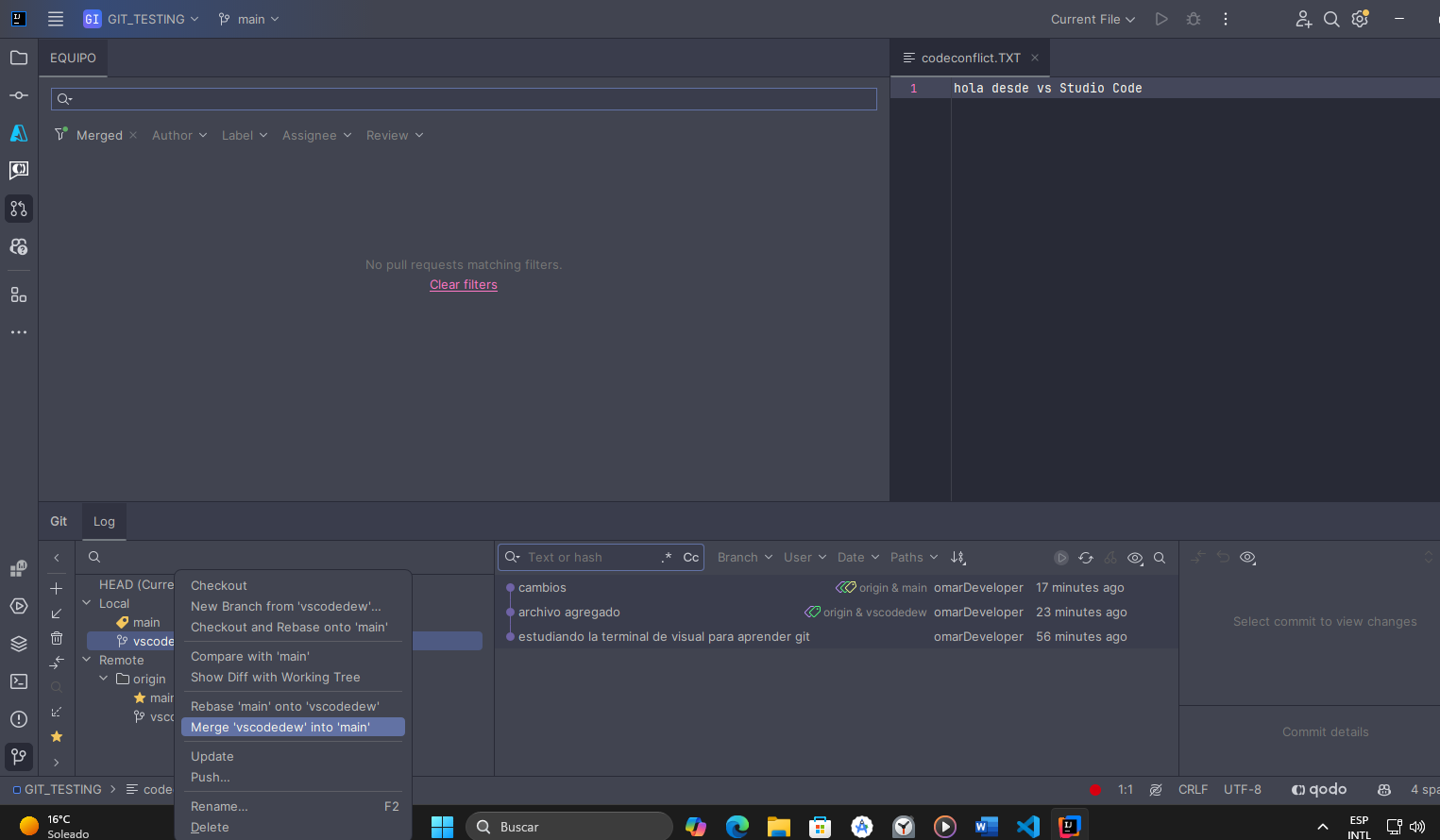
**Texto

El contenido generado por IA puede ser incorrecto.**

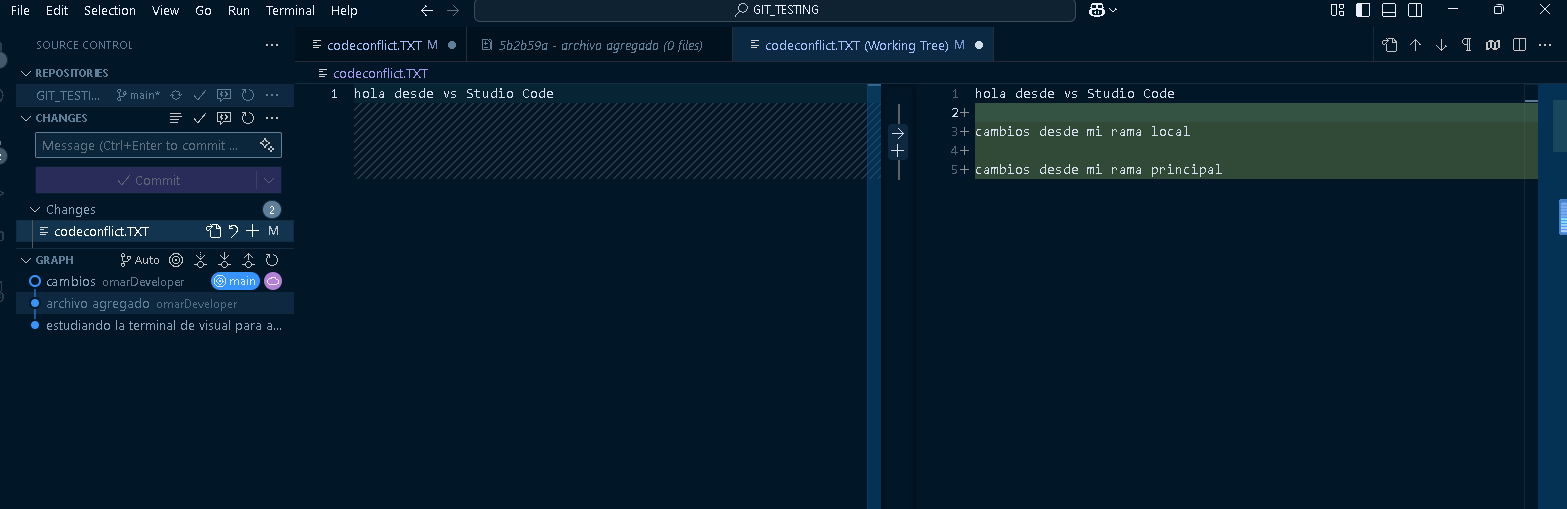
**Lo esencial en este caso después de usada la rama secundaria se elimina para no tener problemas de conflicto en los commits posteriores**

**Con intefaz de programación**

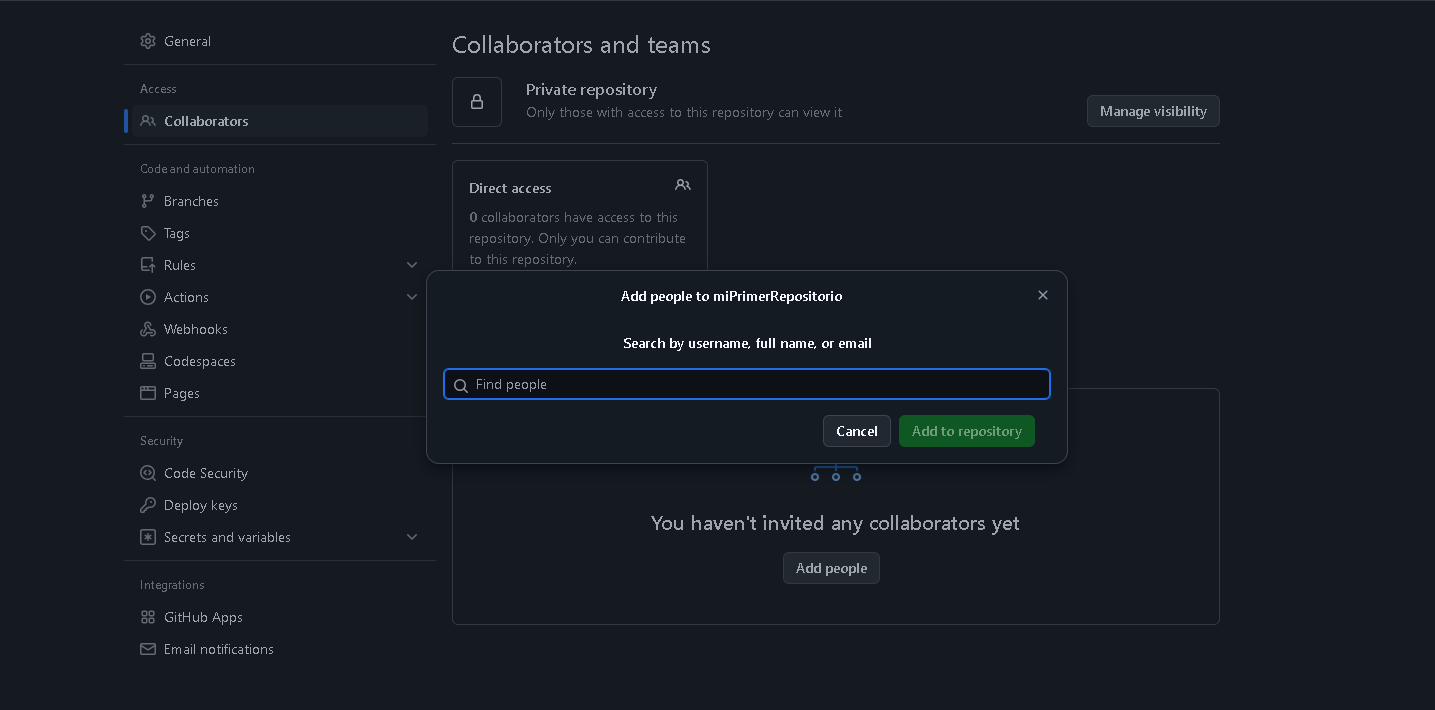
**se genera un historial automáticamente usando el github copilot,**

****

**El merge realizado en intellij se ejecuta dándole click a las rama a la que se le va a realizar el merge**

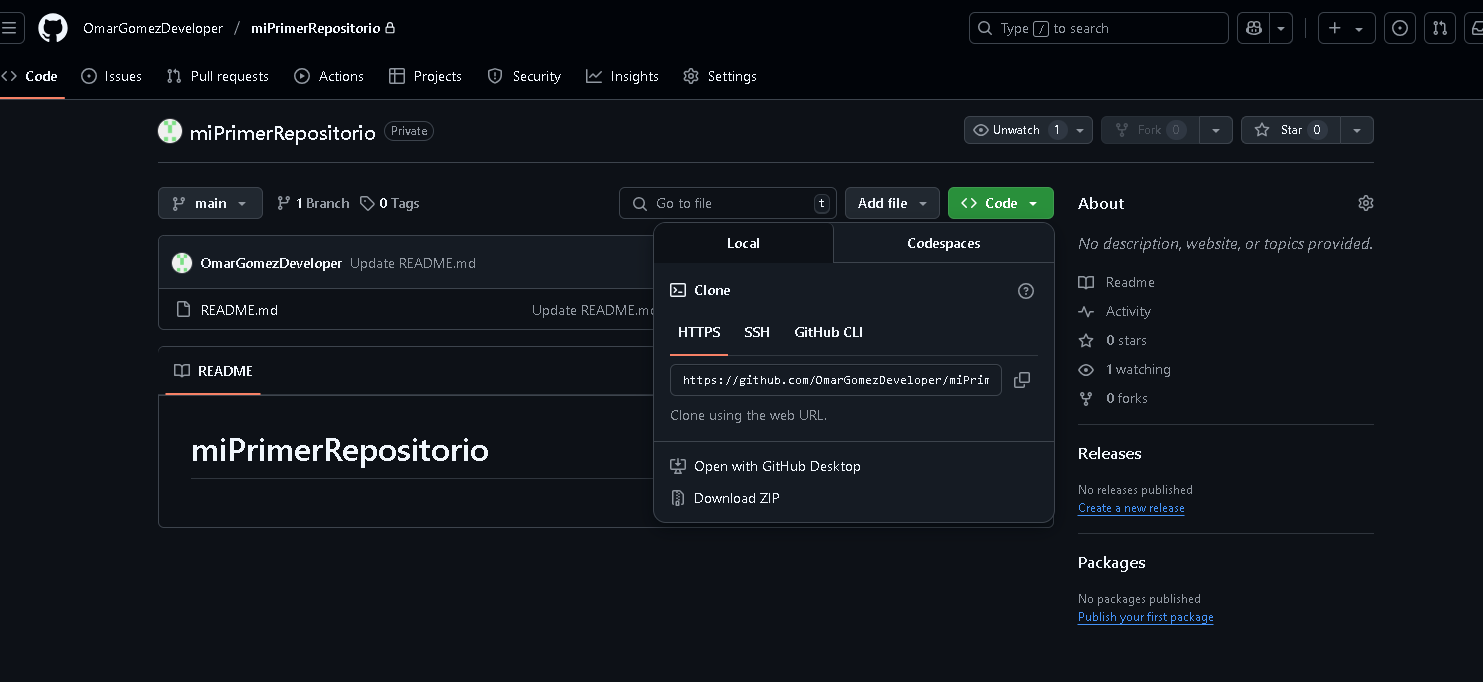
** de la misma forma cambiándolo queda desde el visual**

**Uso de GitHub:**

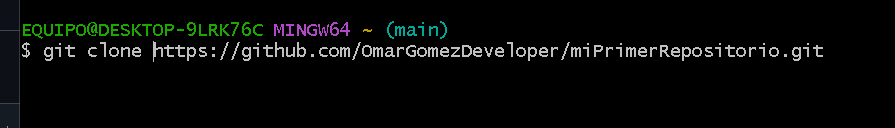
****

**Ir a settings luego a colaborators para generar el ingreso de personas al proyecto,**

**Con el code**

** se usa para clonar el repositorio para trabajar en el**

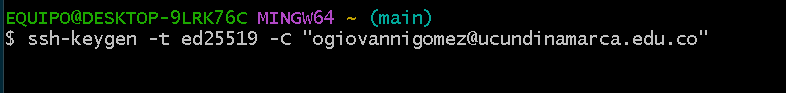
**Con esto se clona el repositorio**

****

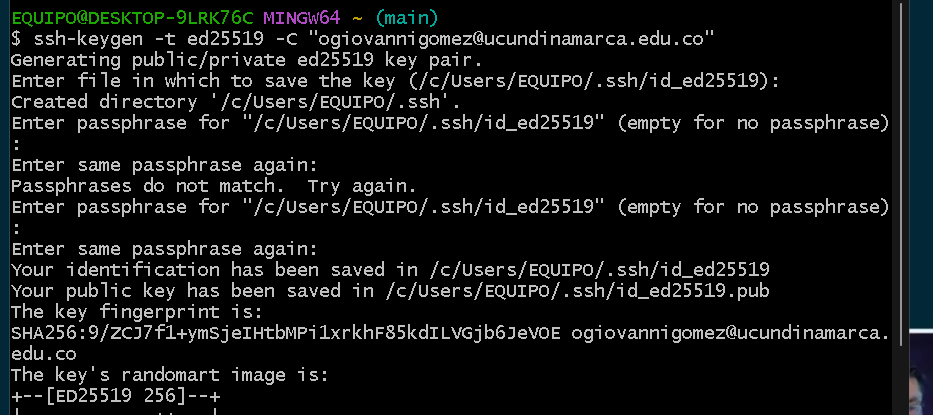
**Texto

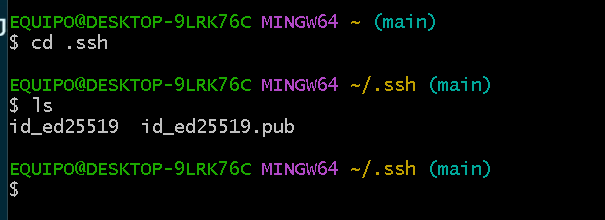
El contenido generado por IA puede ser incorrecto.**

**Generandolo con ssh**

**Se va directamente a la terminal **

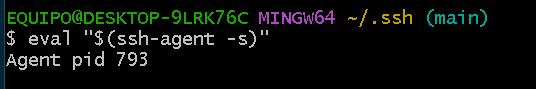
**Al momento de generar esto el ssh gen crea herramientas por excelencia el -t es el nivel de encriptación y luego se agrtega el correo de github**

** se genera el nombre de usuario por defecto o se le da enter sin poner nombre y la clave se genera**

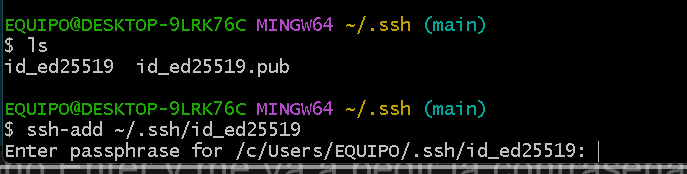
****

**Se genera la ubicación, además se muestra el archivo con el id del ssh**

**Generar el agente de ssh**

****

**Generar el ssh para que no se ponga el password a cada momento:**

****

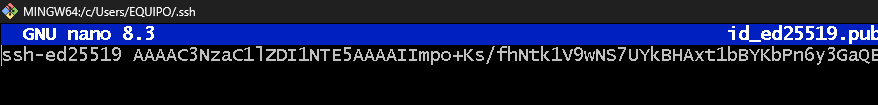
**Aquí la contreaseña ya se ingreso**

**Texto

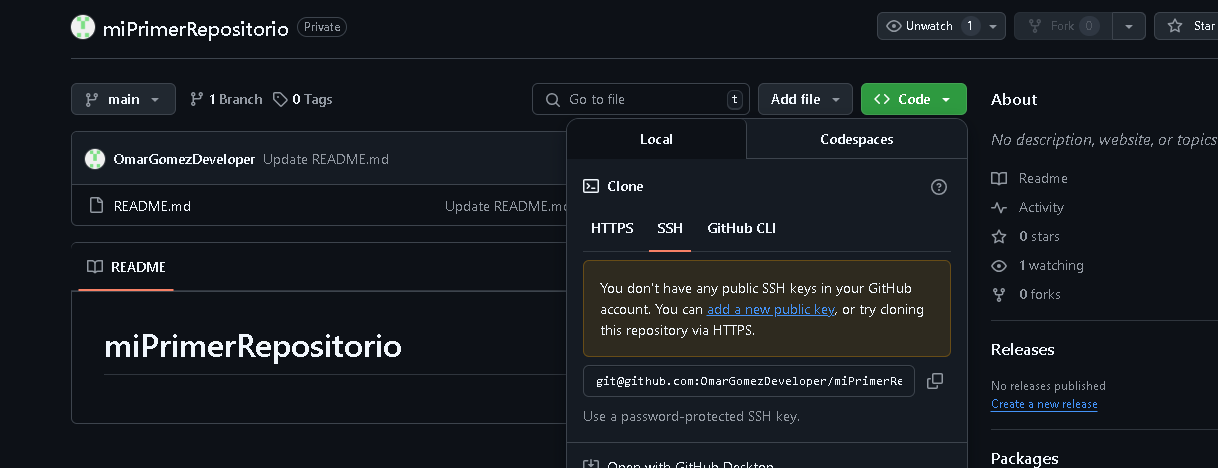
El contenido generado por IA puede ser incorrecto.**

**Sea Windows con wsl o Linux se abre lo asiguiente para copiar y poder continuar co el github**

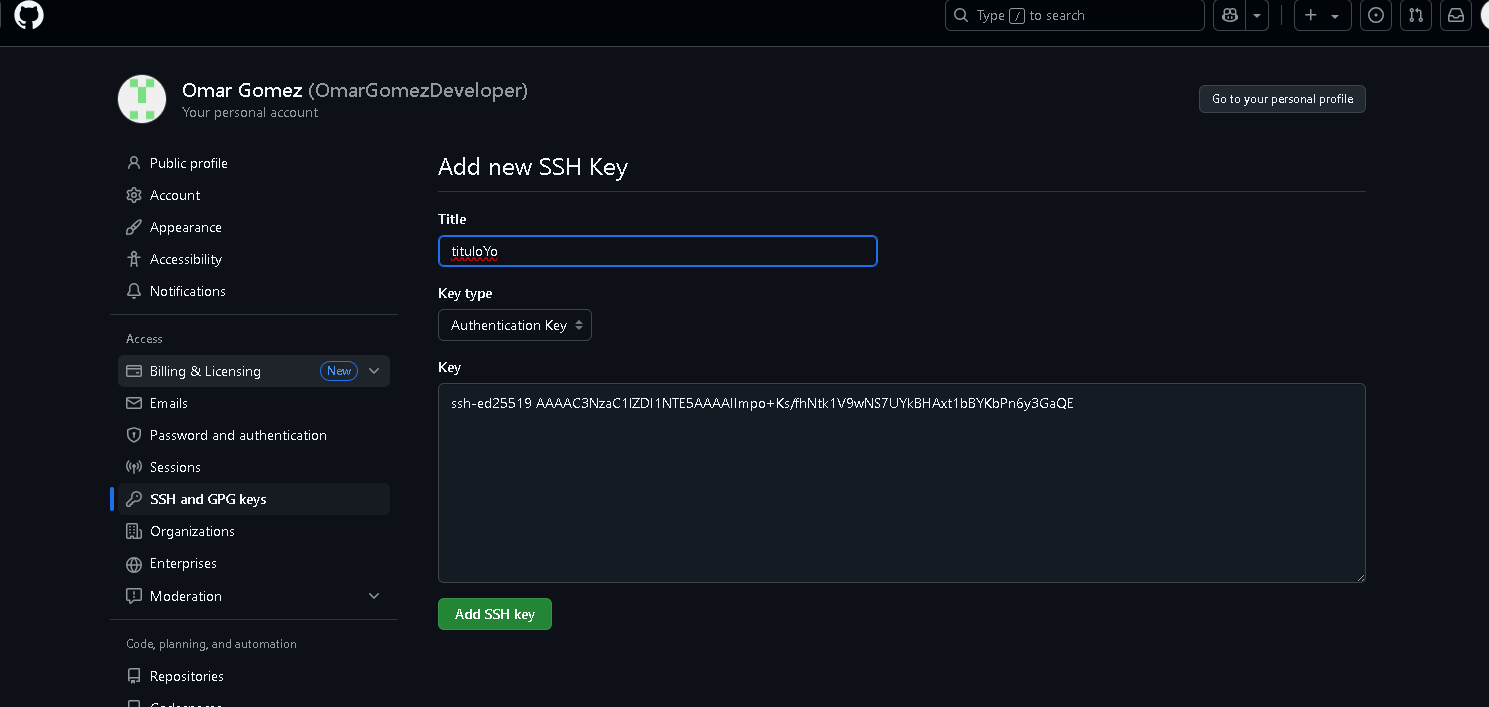
** debe ser la clave pub, para saber que hacer**

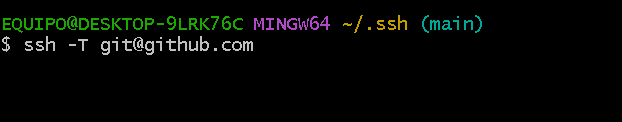
****

**Este es el ejemplo de la llave ssh**

**Luego der tener la llave darle click al add new publckey para generarla ahi**

**Luego de ello se va a github a deploy key y se mete la llave ssh**

****

****

**Con este comando se establece una conexión ssh para ver que todo este fluyendo bien**

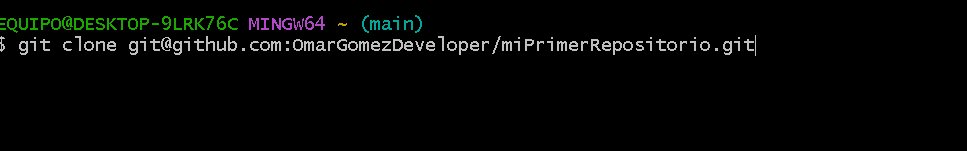
**Texto

El contenido generado por IA puede ser incorrecto.**

**Se debe poner yes para ingresar**

**Ahora se ha ingresado **

**Para clonar el repositorio con git ssh:**

****

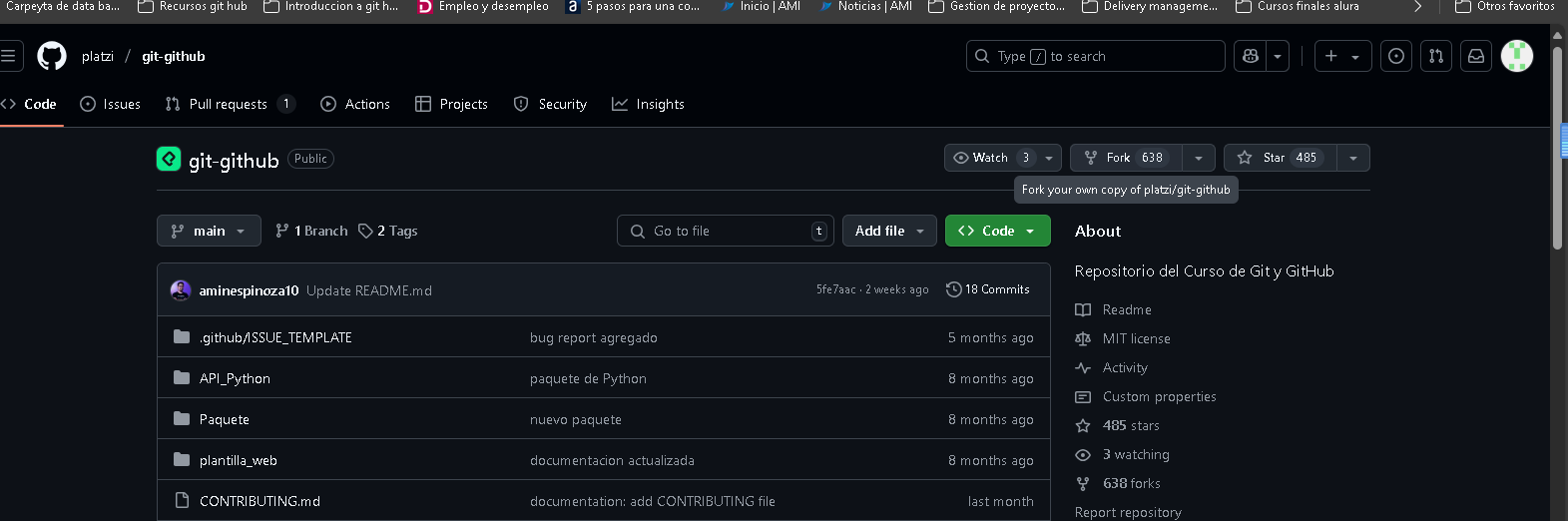
**La clave ssh no se comparte con nadie, además un solo comutador se conecta desde el ssh, y al ingresar desde varias computadoras se debe hacer ese proceso para todas**

**Clone fork y estrellas a repositorios:**

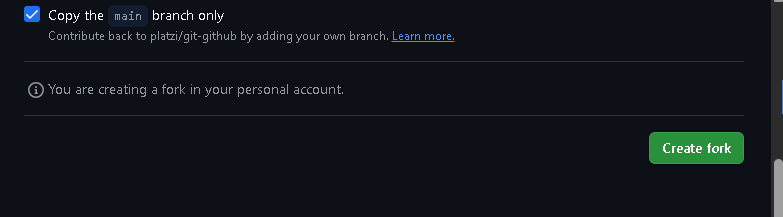
**Forks:**

**Es un repositorio encontrado en la cuenta de alguien más y se copio a nuestra cuenta, apartir de ponerlo en nuestra cuenta se encuentra como congelado, si hay nuevas actualizaciones remoto no va a ser visto en el repositorio forkeado**

**La estrella genera segmentación en los repositorios favoritos, con esto usamos un repositorio como referencia**

****

**Se da click en fork para forquear**

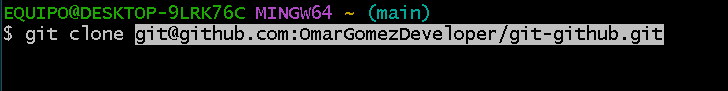
**Luego dse da click al créate fork **

**Start**

**Asparece luego de fork dar click Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Para saber el star ir al perfil y poner your starts**

**Después de forqueado se busca copiar el ssh para activarlo **

**Texto

El contenido generado por IA puede ser incorrecto.**

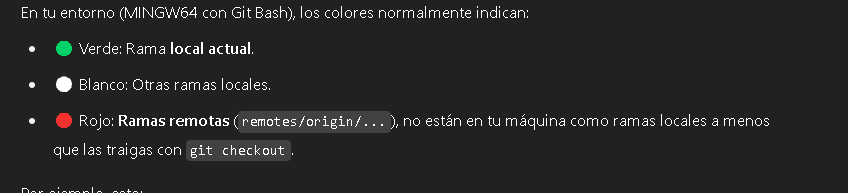
**Con el forqueado podemos hacer los cambios que queramos con los propioscambios que deseo sin que se afecte al original**

**Trabajos con repositorios remotos push pull y fetch:**

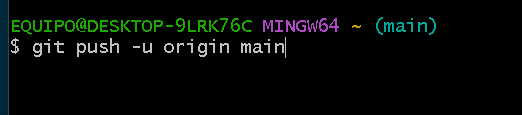
**Git pull jala informacion actualizada del repositorio y actualizar el local**

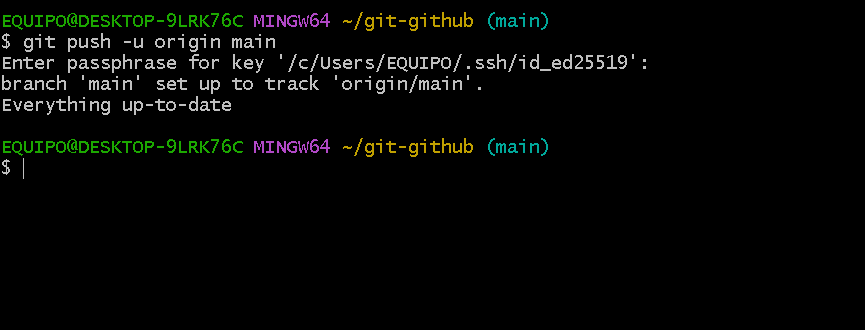
**Git push: sube cambios del entorno local al github o nube**

**Git fetch:**

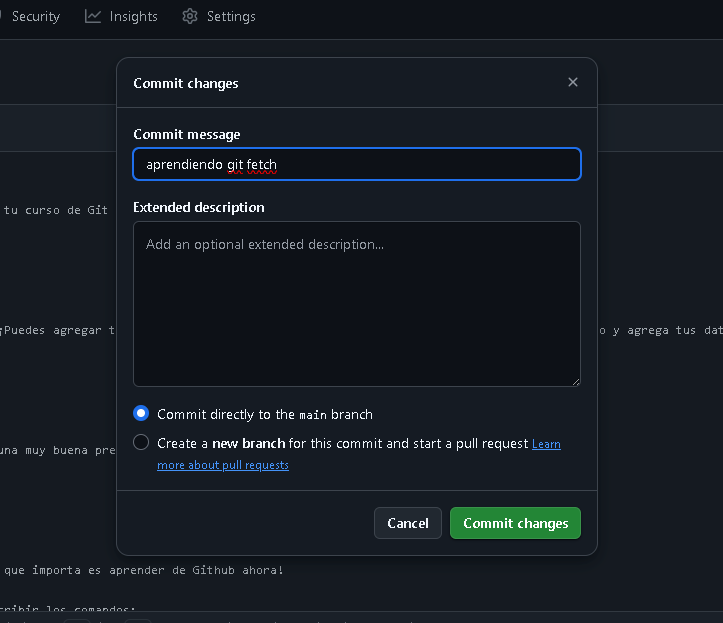
****

**Commando push solo se hyace una ocasión dado que se quiere mandar una rama que ya esta en línea pero con información recién ingresada**

****

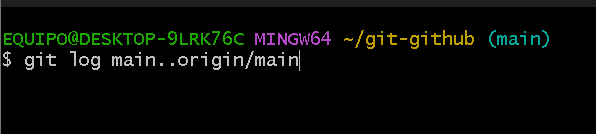
****

**No se puede hacer un git push si el repositorio de git hub no esta sincronizado en el repositorio local, es decir primero se debería hacer el git pull p[ara mantener el repositorio local actualizado   
git fetch:**

****

**Texto

El contenido generado por IA puede ser incorrecto. se usa la rama original para jalar los cambios pero no los va a fusionar con el main,**

****

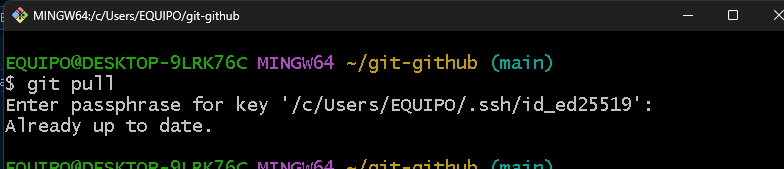
**Se genera un registro de la rama local main con la rama origin main**

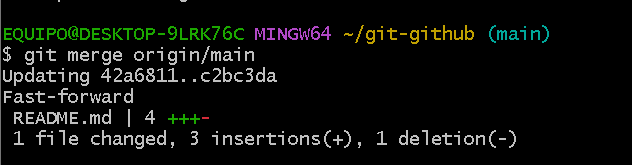
**Texto

El contenido generado por IA puede ser incorrecto.**

**Con el comando anterior se ve el comando y el commit hecho con el fetch, si nos interesa procedemos a usar el merge para fusioinarlo con la rama main, si no nos interesa no se usa el merge**

**Git pull: al escribir git pull por naturaleza se traen los últimos cambios de la rama por defecto que es main**

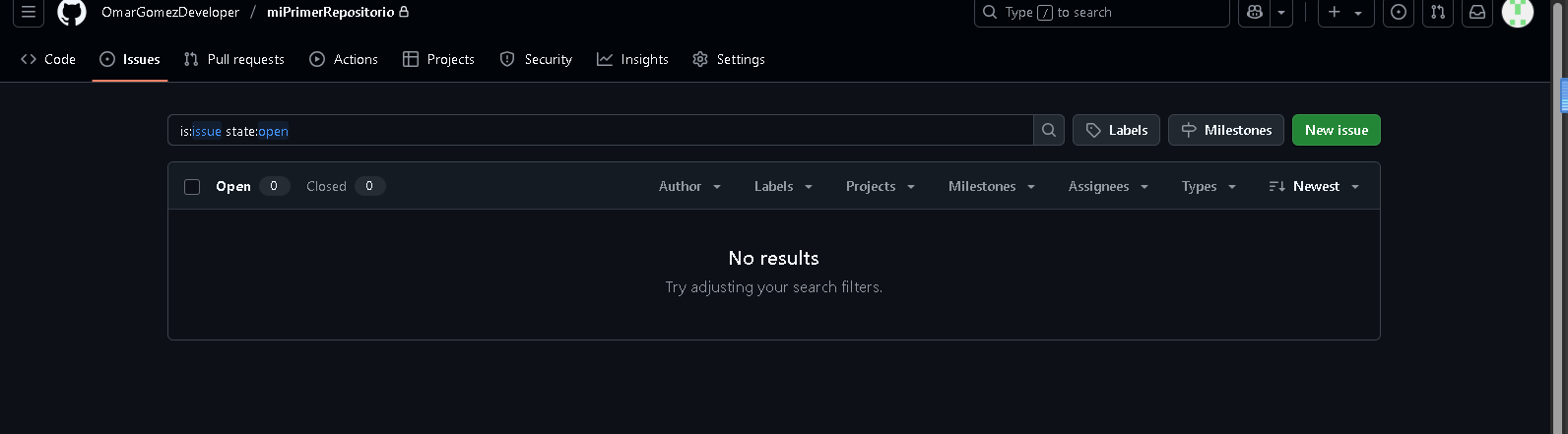
****

**Con el commando mostrado acontinuacion miramos el commit que se generó y ya se usa el merge si nos interesa **

**En un escenario real es necesario que se use el git pull dado que genera los cambios de manera inmediata, se vea lo que se quiere sincronizar y luego se haga un git push para sincronizar los cambios recibidos como los cambios que yo haya hecho para el repositorio de git hub**

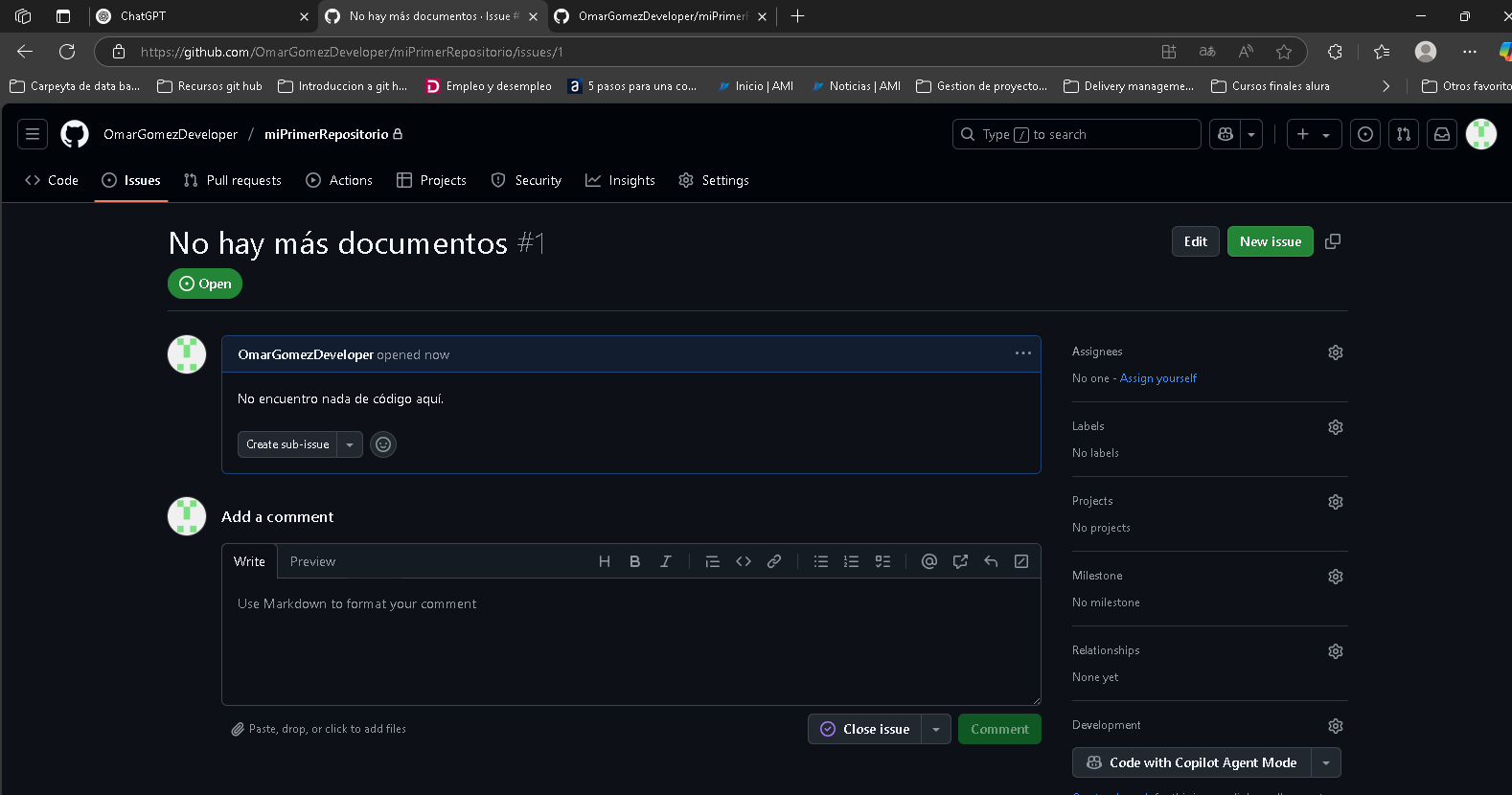
**Gestion de issues y discussions en github:**

**El issue permite esclarecer un defecto dentro del repositorio, esos defectos pueden ser simples comom un error en la documentación o una falla en el repositorio dse una manera esperada, con el issue puedo demostrar al autor alguna anomalia**

****

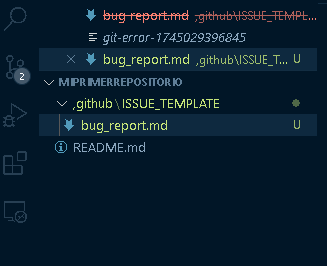
**Se da click en el icono verde lo mas importante en un issue es el titulo y la descripción del problema**

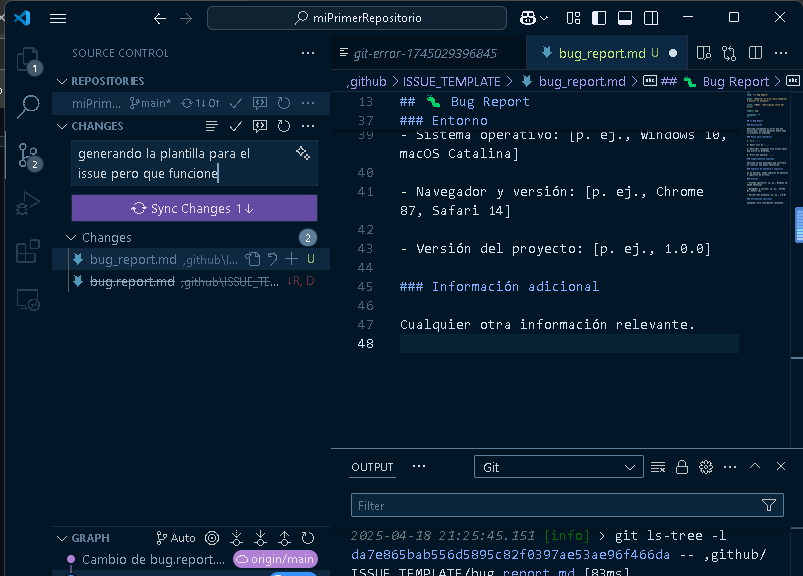
**Ya después de creado se genera el issue**

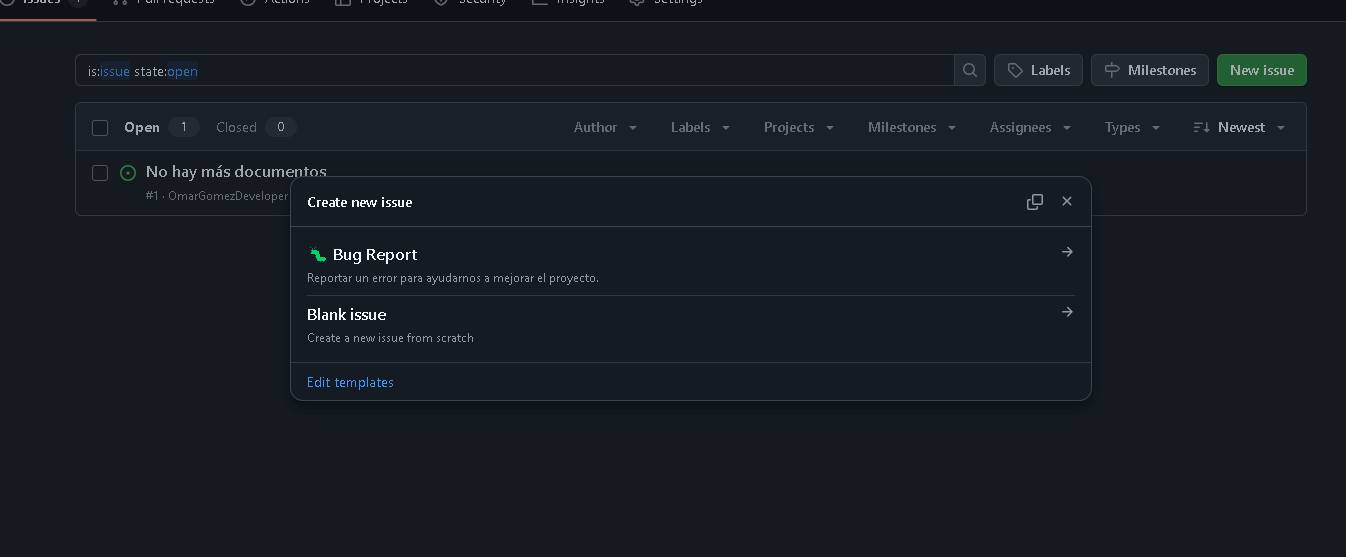
****

**Creando formulario para explicar que sucedió desde el issues**

**Se genera una carpeta con las denominaciones .github, dentro de esta una que se denomine .ISSUE\_TEMPLATE, luego de esto se crea un archivo bug\_report.md**

****

**Luego de creada la plantilla dse genera el commit en este caso con visual studio code **

**Luego de generado aparece un icono con el siguiente comadno **

**Se le da crear y listo**

**Colaboración sin errores Pull Requests en github:**

**Los cambios directamente no se suben al main directamente,por lo que es necesario usar rama locales para luego subirlas a la principal, por tal motivo usamos el pull requests**

**Los pullrequests, permiten fusionar los cambios de una rama local hacia una rama principal**

**De acuerdo al tener el cambio se crea un pullrequests, al generar esto se notifica al equipo de algún cambio, y pueden entrar y revisar los cambios, este proceso es un codereview, luego de que lo hayan aprobado esos cambios se ponen en la rama**

**Primero ase debe crear la rama local en este caso es la rama developer01, se genera los cambios y luego se sube con git add . y luego con git commit -m “”**

****

**Después de ello se ira al portal de github para generar el pullrequest**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

**La descripción debe ser detallada para el equipo**

**Captura de pantalla de un celular

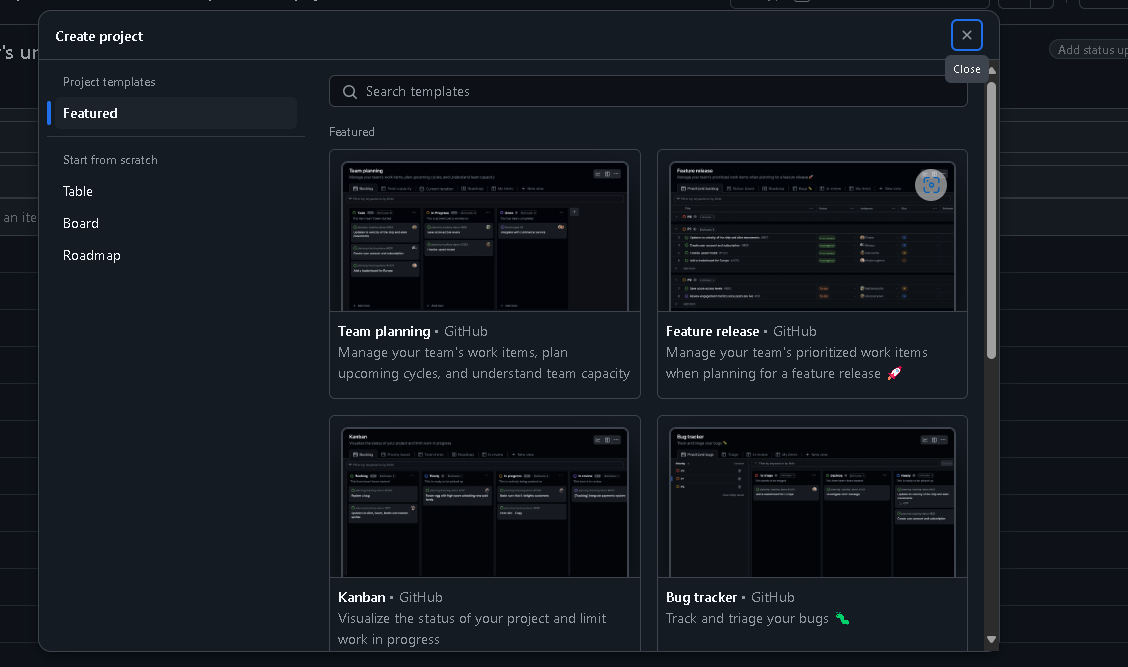
El contenido generado por IA puede ser incorrecto.tiene que ver esto para que no haya conflicto y luego hay que esperar el codereview**

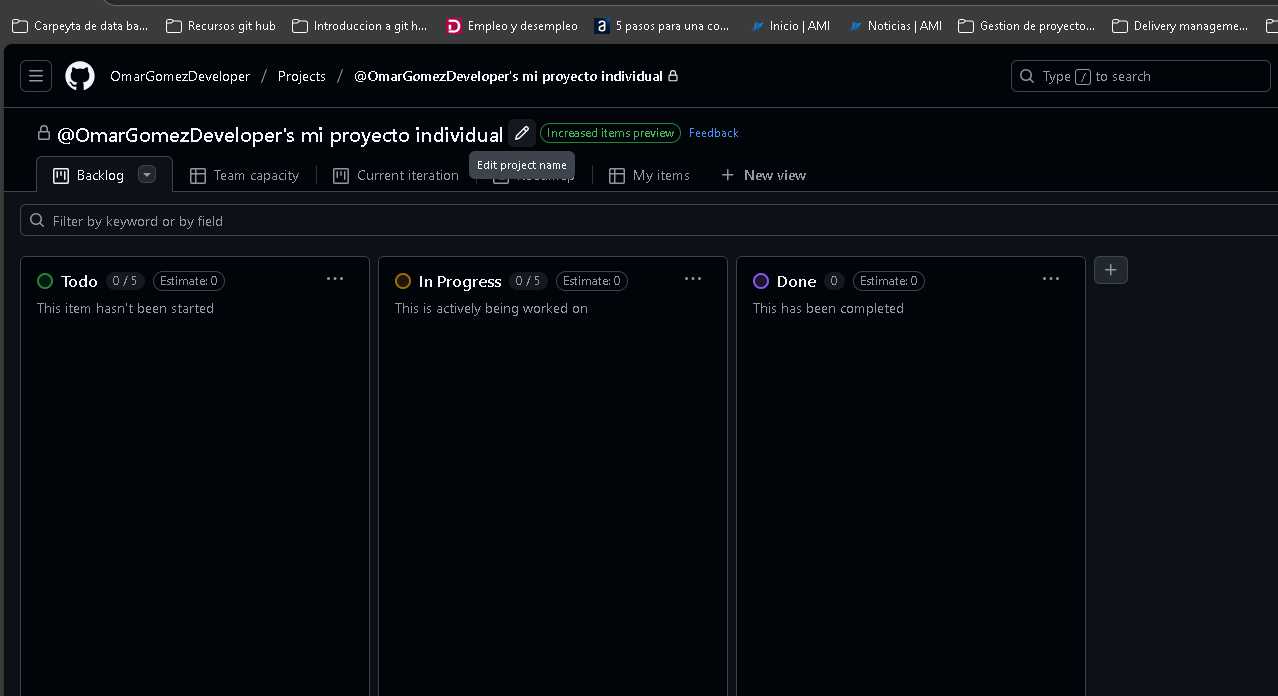
**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto. con esto se elimina la rama para poder ser usada en futuros archivos**

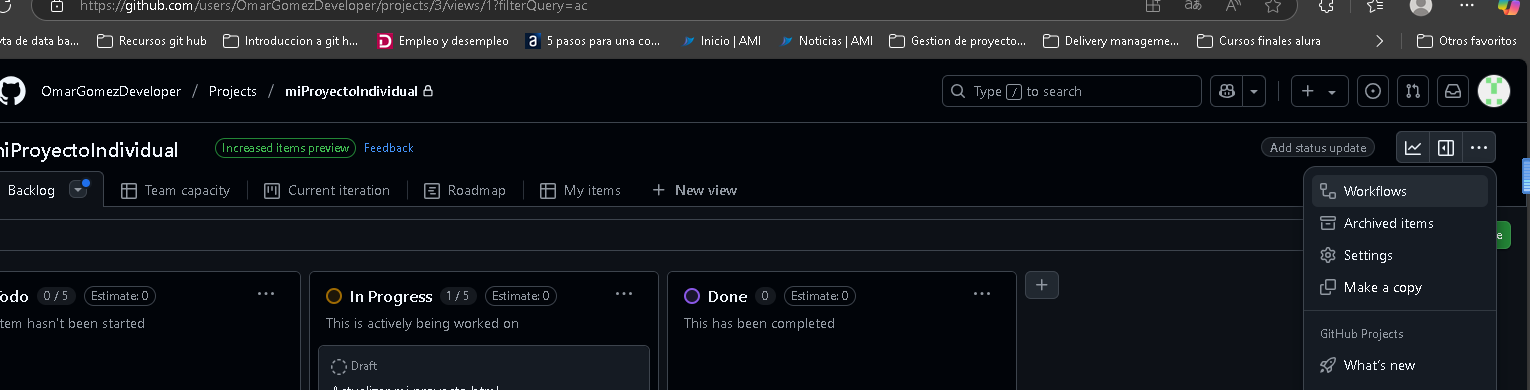
**Introducción a gitHubProjects**

**Ir a github y luego ir a Project, luego se puede crear un proyecto, es como usar jira**

****

** se genera la edición dándole click al lado derecho del nombre**

**Con status chart se mide la estadística por tarea**

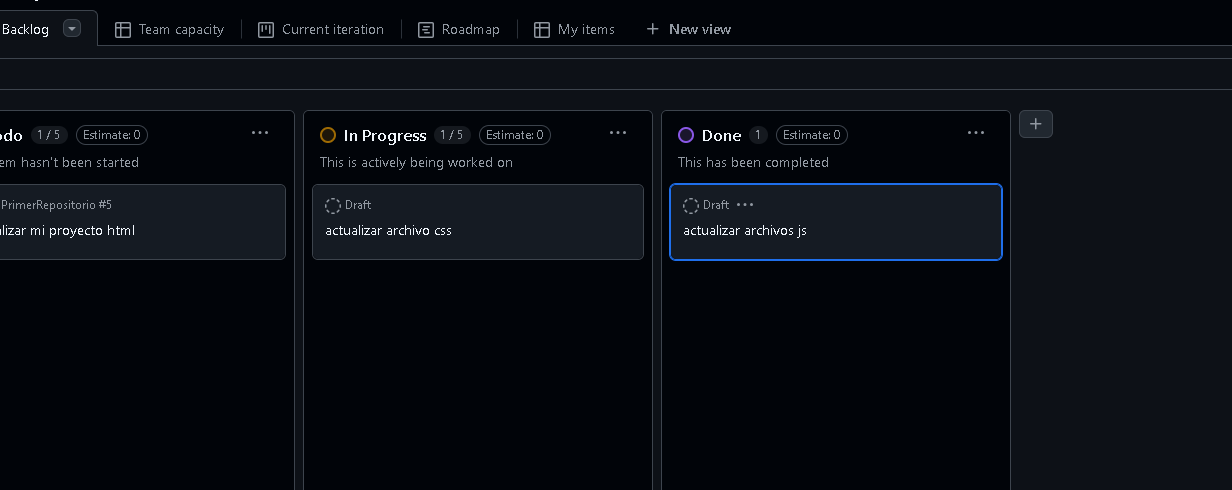
** en los flujos de trabajo se muestra lo que se estya haciendo**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto. con el flujod e trabajo se camnbia el estado es más dinámico**

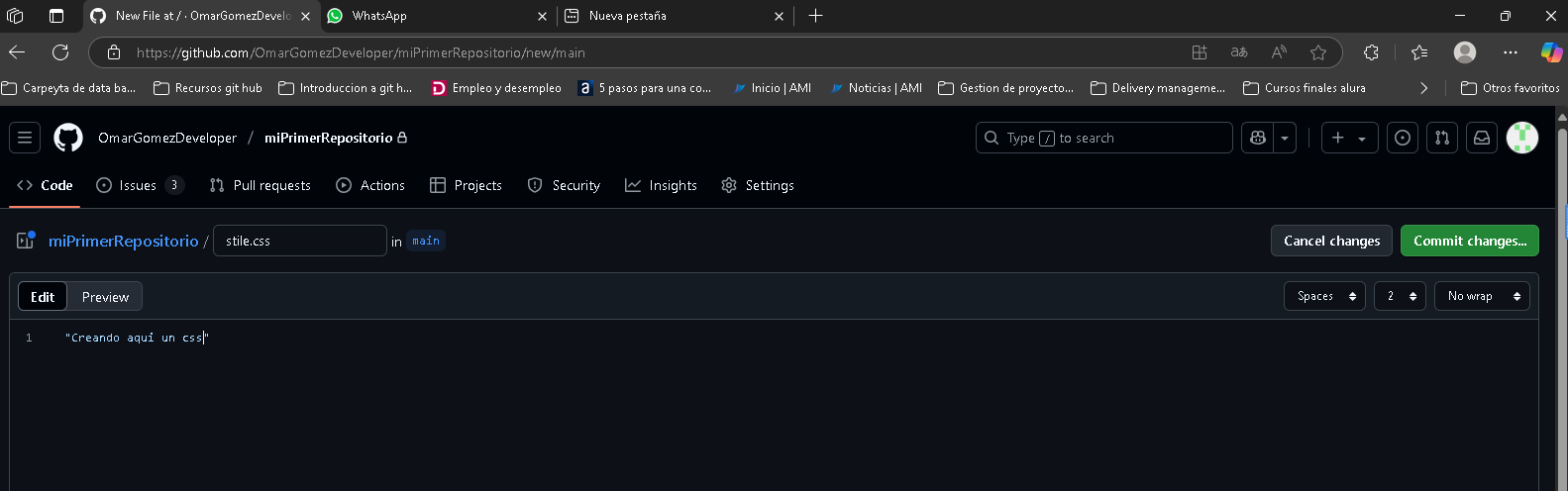
**para generar las tareas hay que crear elitem desde la parte que dice todo he ir editándolo de acuerdo al flujo de trabajo**

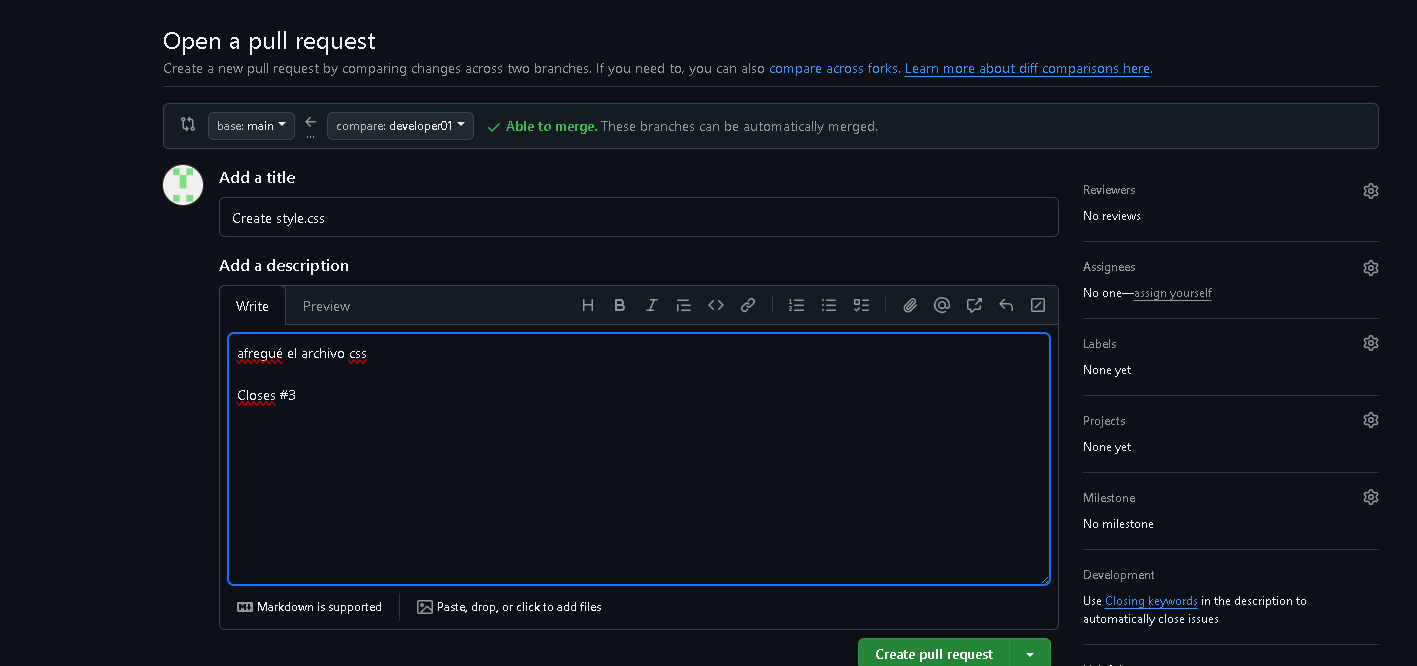
**Creando un archivo desde el girt hub**

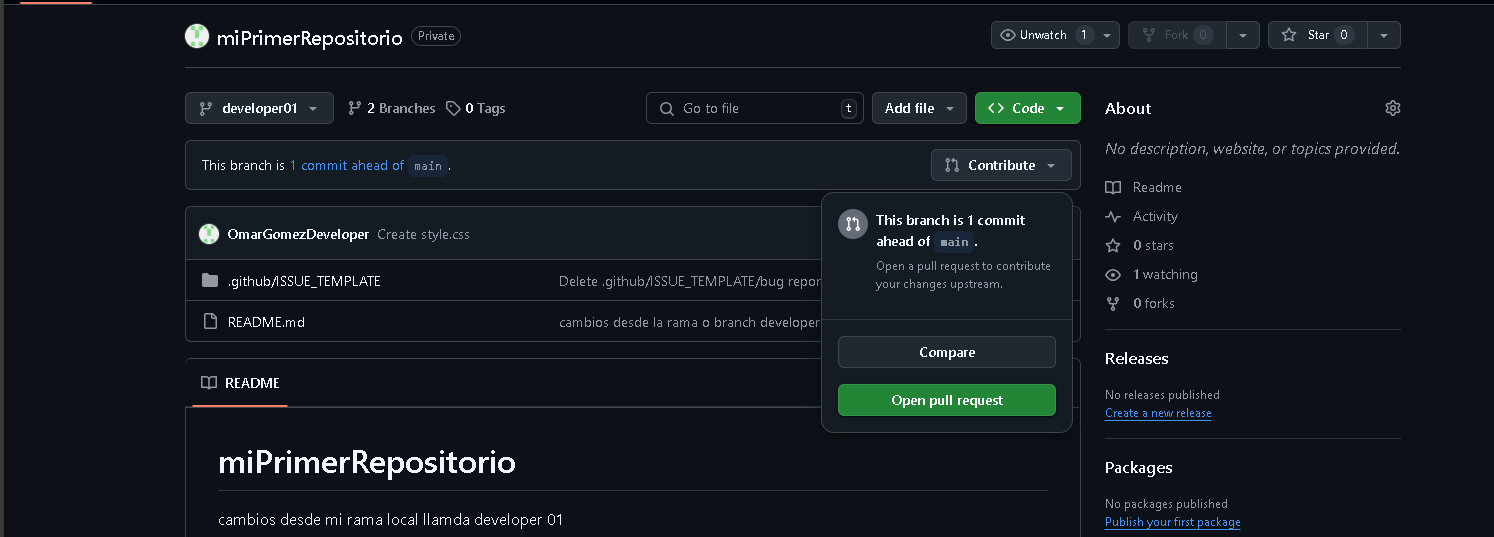
**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

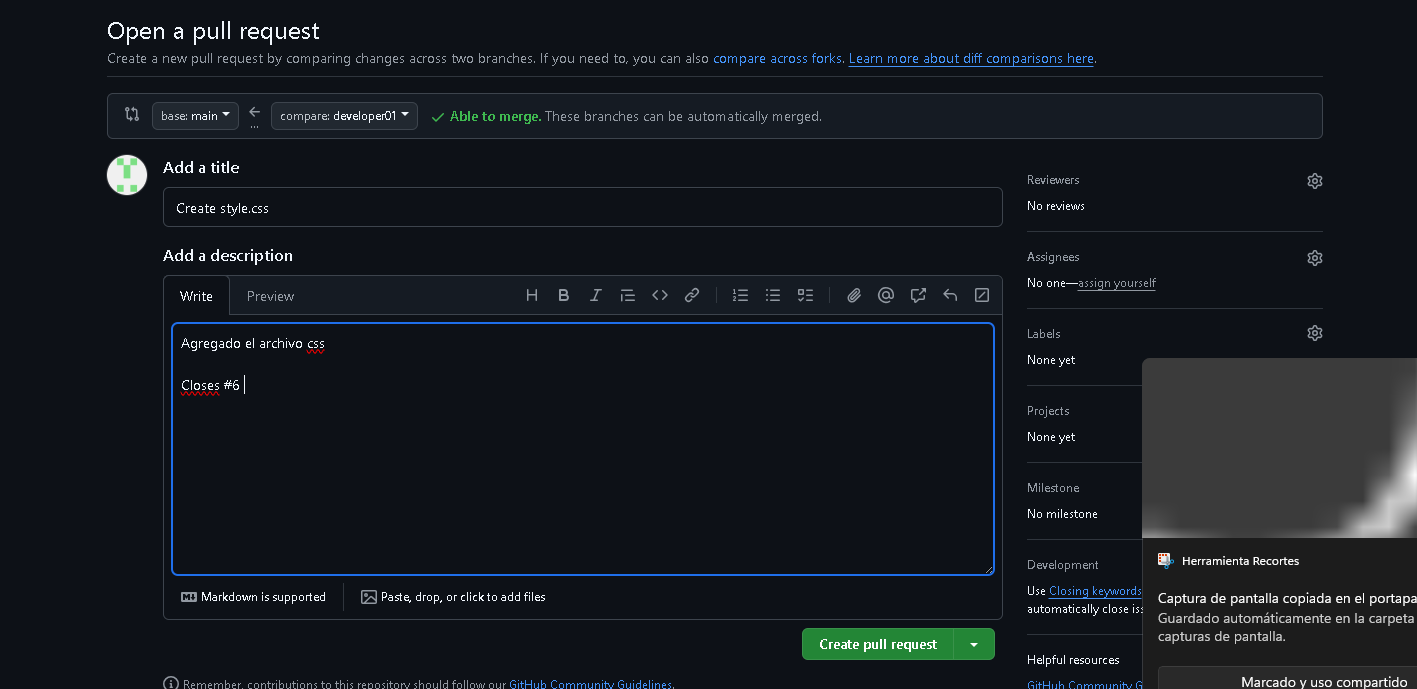
**Se genera el archivo**

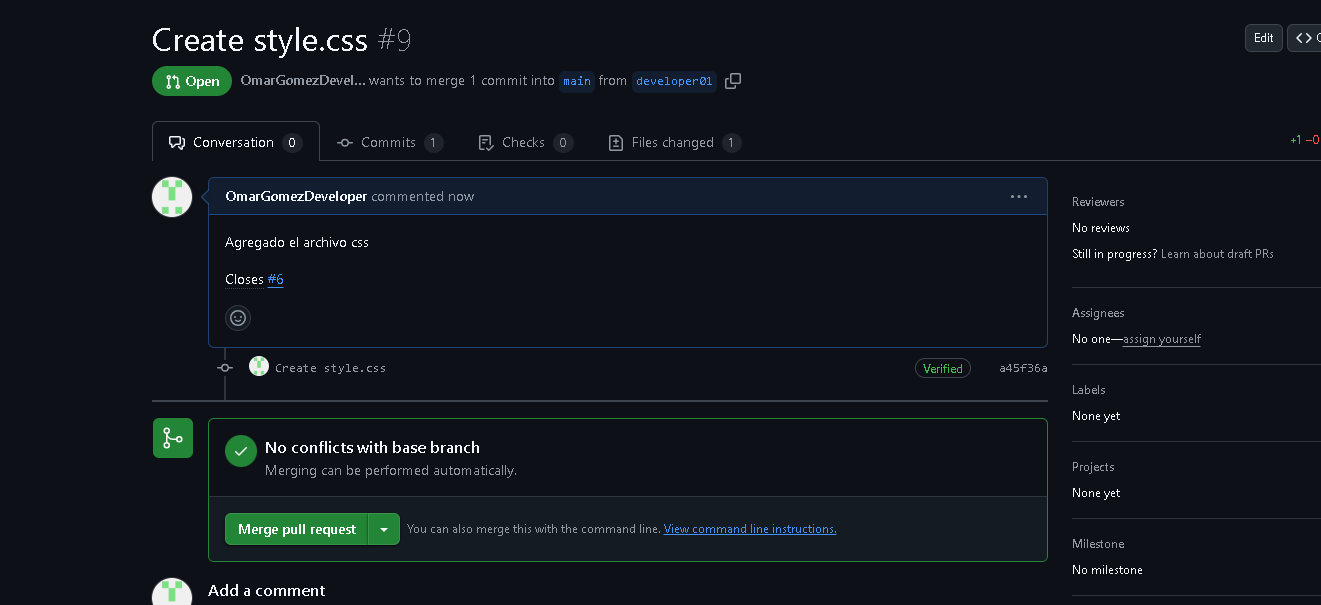
****

**Se genera un pullrequest con uno de los itewms y al generarle un closes # se identifica en que parte del proceso está **

****

**Al ver el comentario se genera el open pullrequest para hacer el pullrequest**

**al generar esto con el closes y el # se obtiene el desarrollo con el Project**

****

**estol ver el check significa que el merge s epuede hacer**

**la idea al generar el uso del github Project es poder automatizar el workflow**

[**Markdown Guide**](https://www.markdownguide.org/) **se debe tener en cuenta la documentación**

**Personalización de Github con Markdown:**

**para generar la portada de perfil en github, se hace lo siguiente:**

**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.**

**Generar un nombre de repositorio igual al nombre de propietario, con el fin de crear la portada**

**Texto

El contenido generado por IA puede ser incorrecto. obligatoriamente se debe dejar en público y darle check al readme**

[**Static Badge | Shields.io**](https://shields.io/badges/static-badge) **tener en cuenta esta pagina para generar el static badge, Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto. el badges con el markdoqwn genera insignias dinámicas y estáticas**

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto. se genera la insignia de acurdp a la documentación**

**Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto. se descarga el plugin para saber que hacer y que no hacer con markdown**

**Luego de generar los shields, se genera una dirección, en ese caso se usa la markdown, se copia la dirección ,, luego de tener los shields.io se implementa con el git**

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**GitHub Gist:**

**Se entra a la pagina de github gist**

[**Create a new Gist**](https://gist.github.com/)

**En este evento se muestra un pedazo del código y un comentario para podr pasarlo**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

**La idea es comentar con el equipo los cambios que se están generando con el código, además, se genera una url para copiarlo y pergarlo y enviarlo al equipo**

**Si hay un gist que no se usa o ya se implemento se genera un delete, esto solo cuando ya se termino el trabajo en el gist**

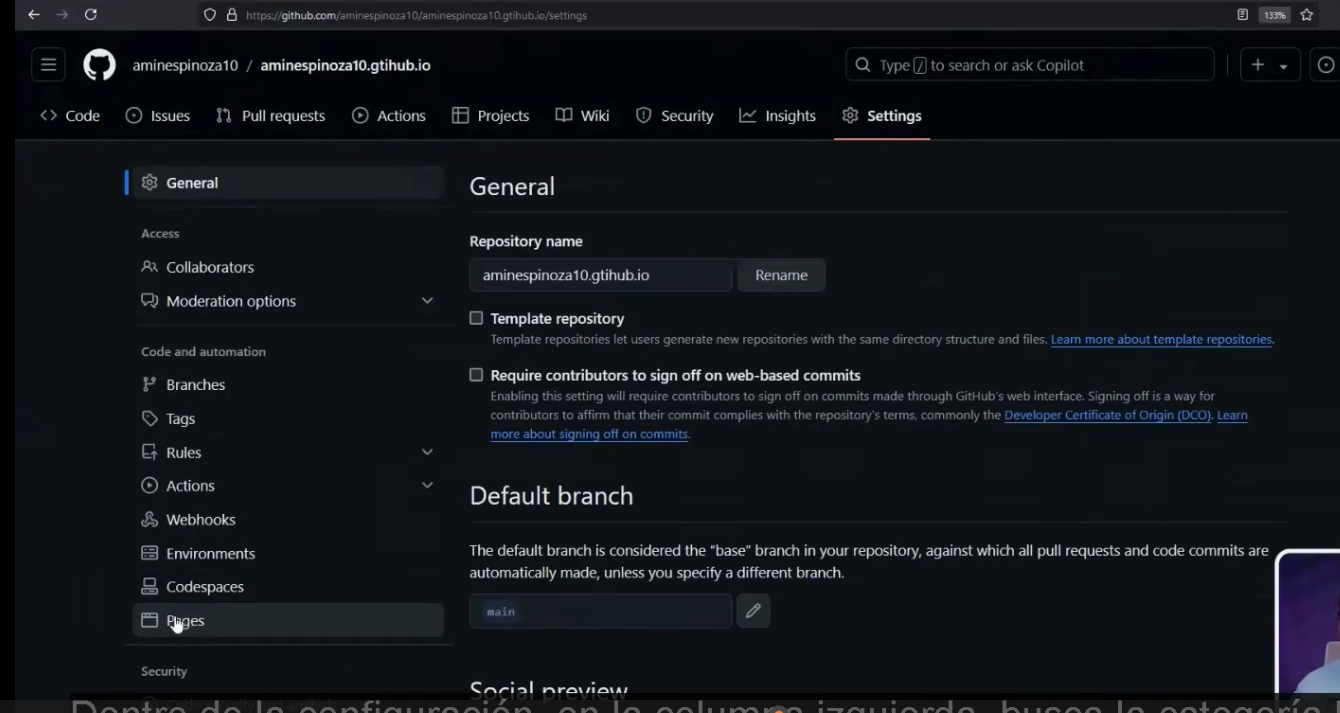
**Lo bueno de gist es que ayudan mucho el las sesiones de peeer programing**

**Tambien como individuo si usa ciertos fragmentos de código de manera recurrente en los proyectos**

**GihubPages:**

**Es una herramienta de github para crear sitios web para desplegarlos y hacerlos accequibles a todo el mundo sin importar que el entorno de frontend y backend sea dinámico o estático, lo primero que se debe hacer es crear un repositorio**

**Después de tener el repositorio se busca en el git la pestaña de git pages, esta se encuentyra primero dándole click al settings del github**

****

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

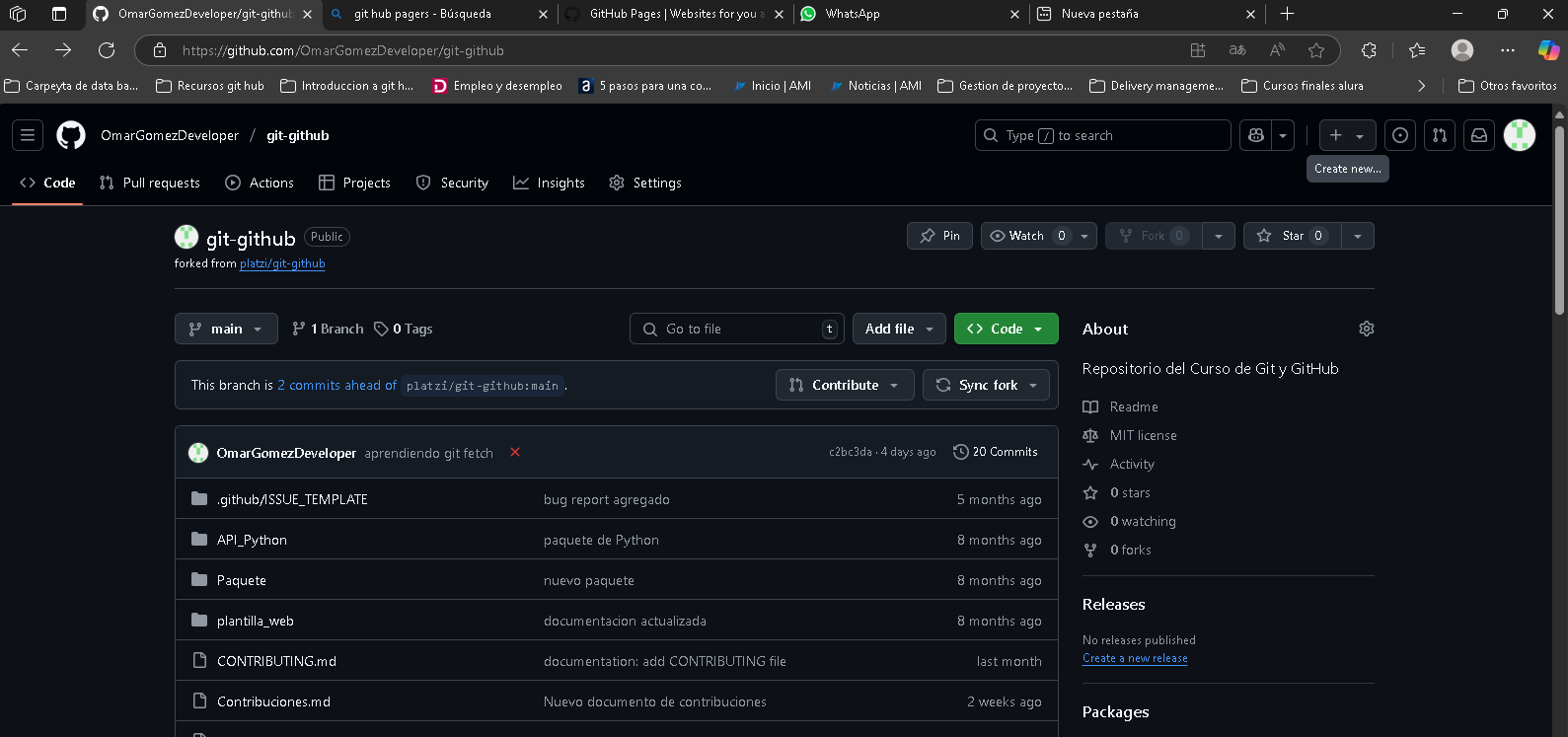
**Después de esto se pone si se trabaja con copilot o con un Branch, en este caos se trabaja con un Branch, donde main es la rama principal**

**Texto

El contenido generado por IA puede ser incorrecto. si se elige el Branch, se debe poner la carpeta como docs**

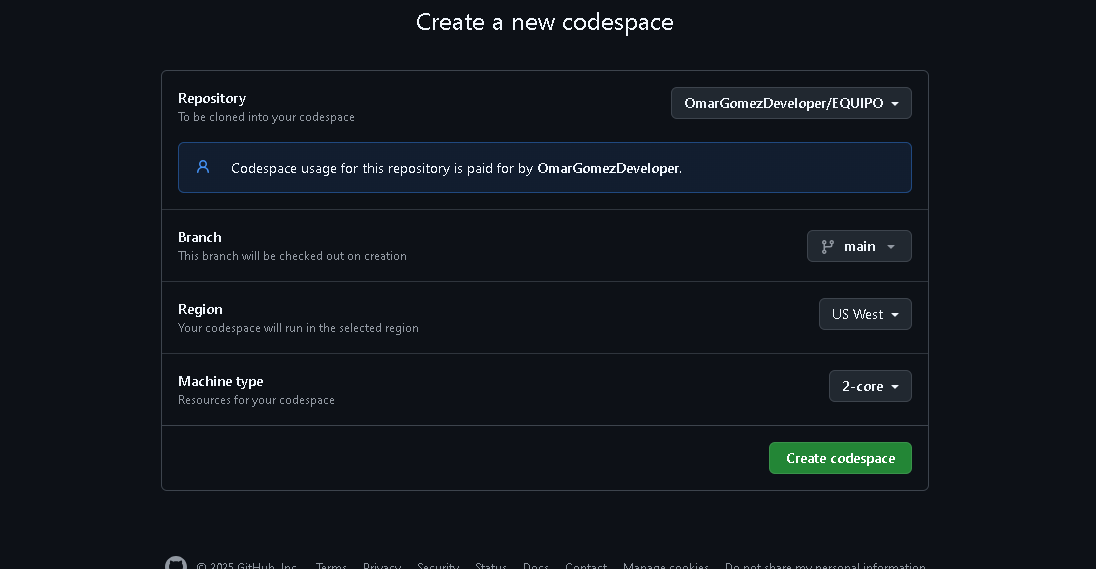
**Como usar GitHub codespaces: configuración y desarrollo en la nube:**

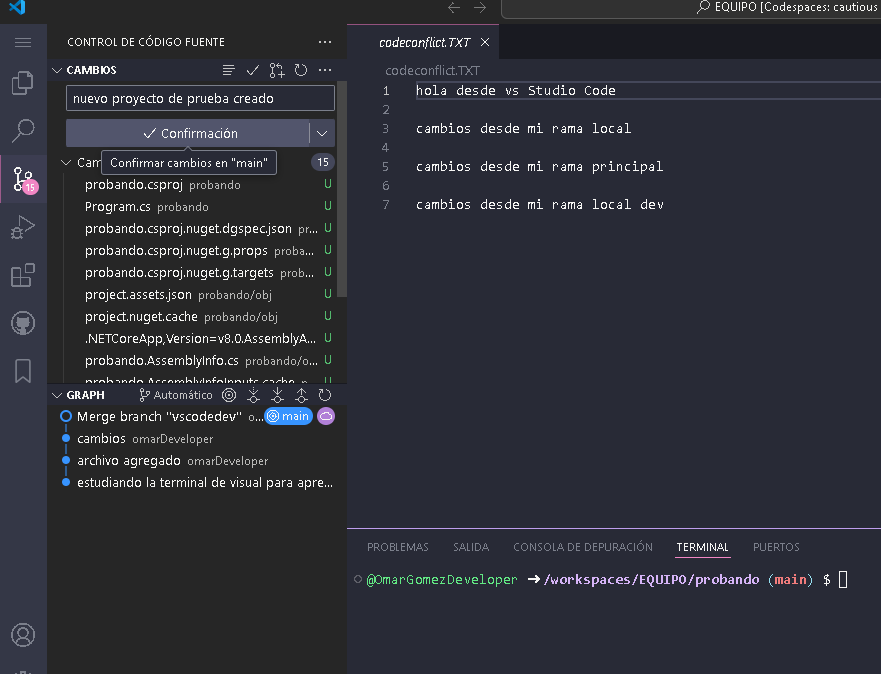
**Es un ambiente de desarrollo en maquina virtual que se encuentra en la nube para poder trabajar en los proyectos, con el se puede trabajar desde cualquier dispositivo, celular, Tablet, computador**

** para iniciarlo se le da click al más y se le dá click al new codespaces**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

** se busca el repositorio y se debe tener cuidado en el Branch para saber en que rama se va a trabajar**

**Como se trabaj desde la nube los cambios son más sencillos **

**La misma nube presenta espacios de código aquí hay algunos que se pueden dar dando click al icono de la parte superior izquierda**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

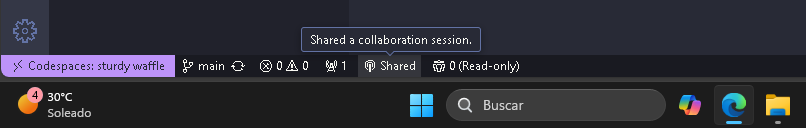
**Es necesario luego de usarlo es eliminarlo, congithub solo se puede entrar en 120 horas, por lo que se debe usar cuando haya una emergencia, cuando se tenga que configurar algo de ultimo momento, concluir actividades etc.**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

**Pairprogramming con visual studio code y codespaces:**

**Se genera con el plugin de liveshare**

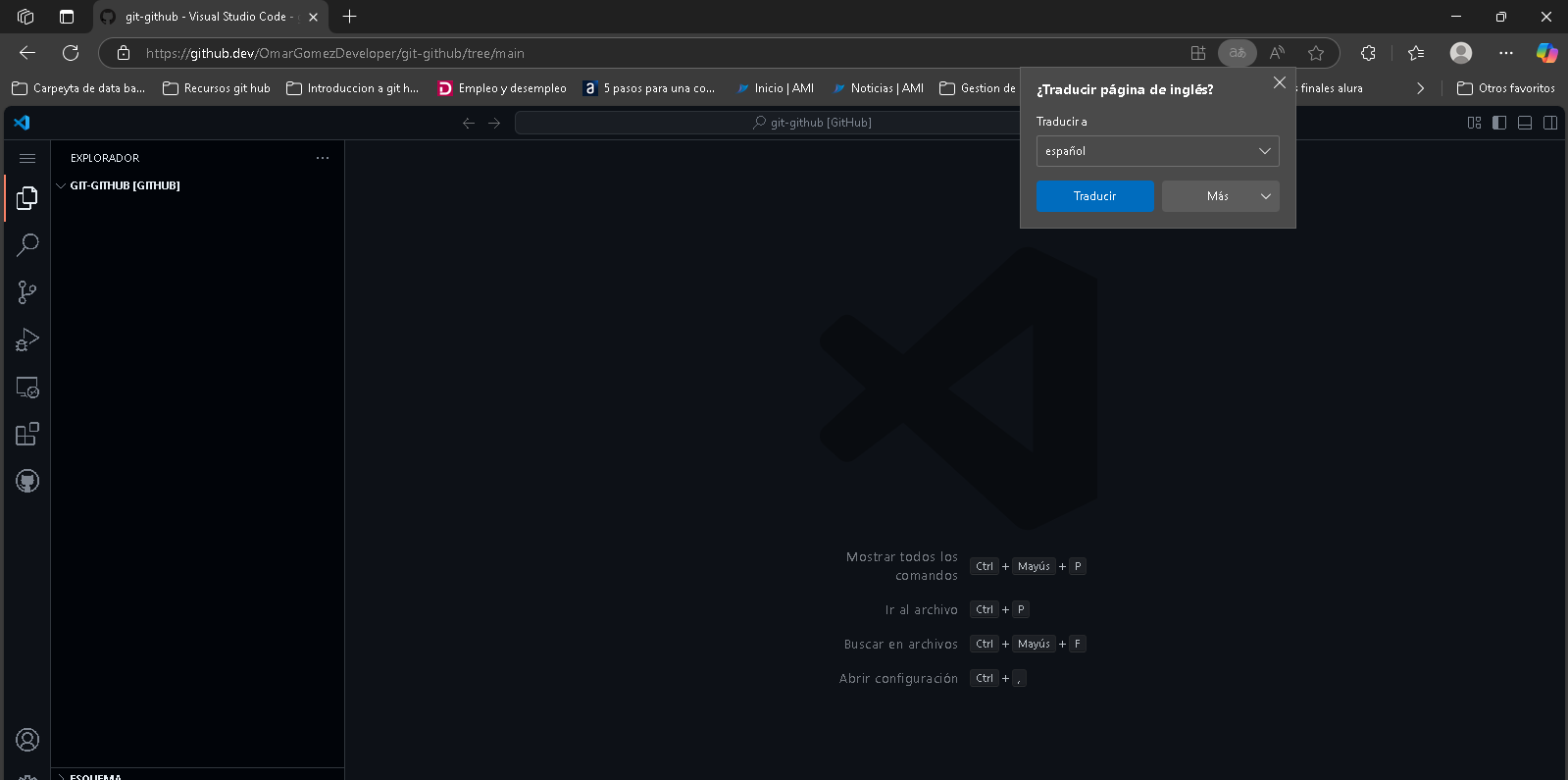
** se encuentra en esta sección, para optimizar recursos es mejor que los que van a hacer pairprogramming lo hagan desde el vscode, sino que uno lo haga desde vscode y el otro desde el entorno virtual de vscode de github**

**Como usar github.dev Editor:**

**Desde la parte del repositorio**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto. darle click a la tecla “.”, punto, para generar un entorno de desarrollo**

****

**Interfaz de usuario gráfica, Texto

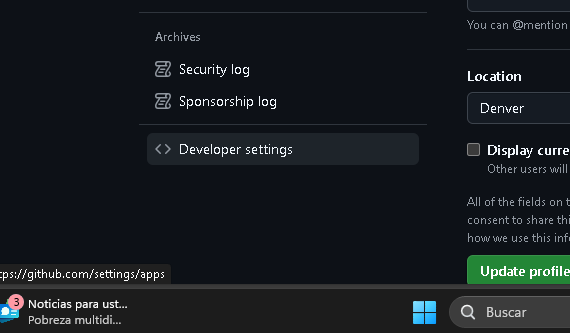
El contenido generado por IA puede ser incorrecto.**

**Texto

El contenido generado por IA puede ser incorrecto. ya con esto se abre el dev editor**

**Como usar tokens en github para acceso seguro a repositorios privados:**

**Primero con el repositorio creado se va a la esquina derecha se irá a settings y se ira a la opción de developer settings**

****

**Se va a la parte izquierda y se pone los tokens, en este caso se usa el oquen clásico**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Y se le genera el new token**

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto. en el token no poner token de expiración ilimitado**

**Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.**

**Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.**

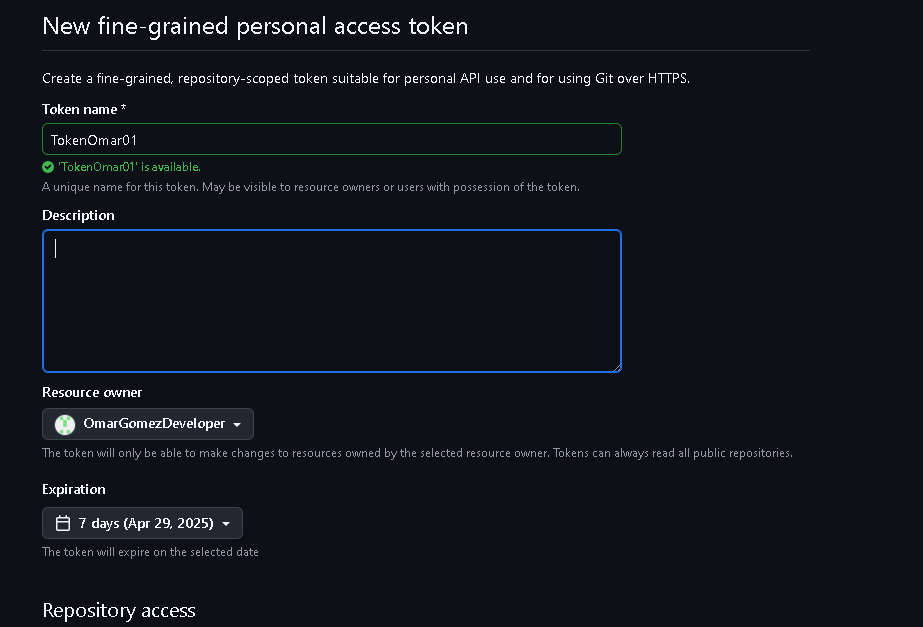
**Texto

El contenido generado por IA puede ser incorrecto.**

**Luego de generar los permisos se pone el token,**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto. se debe copiar el token inmediatamente por que solo se mostrara una vez, de no ser así toca eliminar de nuevo ese token para generar uno nuevo**

****

**Texto

El contenido generado por IA puede ser incorrecto.**

**Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.luego de generado los permisos**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto. se genera un token este es ecesario grabarlo en un block de notas**

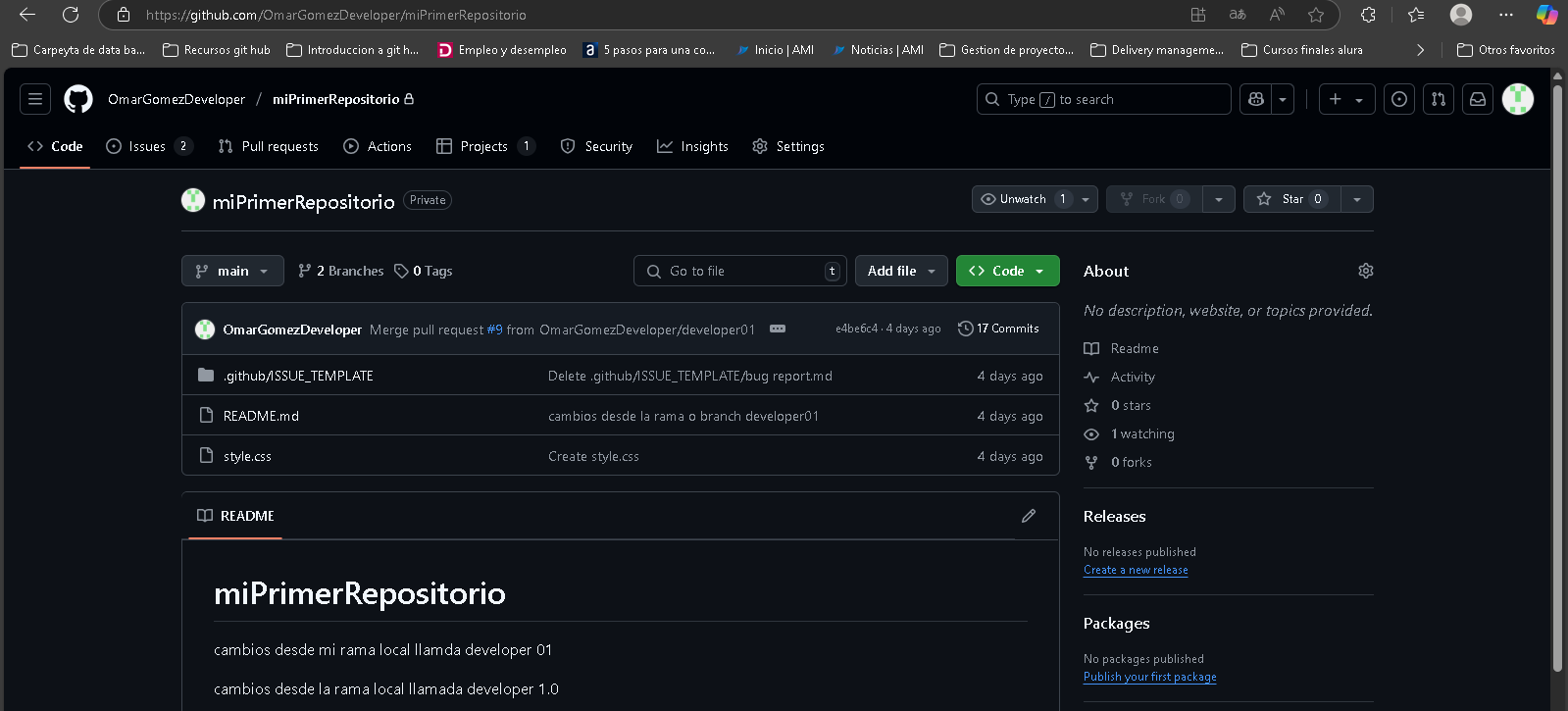
**github\_pat\_11BOSVGRY0T6JwjutGYySi\_l8Cl5Ujaq1z92RQrrqtKChxiFdfZ9iOCuje00QJD53ZQ5I5NEYYKS8rFm0i dado que se uso luego**

**Texto

El contenido generado por IA puede ser incorrecto. después de clonarlo se empieza a usar el rtoken para poder trabajar**

**La implementación de los tokens debe: ser seguros expirar, tener acceso a lo que se le va a dar al desarrollador y que funcionen para lo qwue se necesita.**

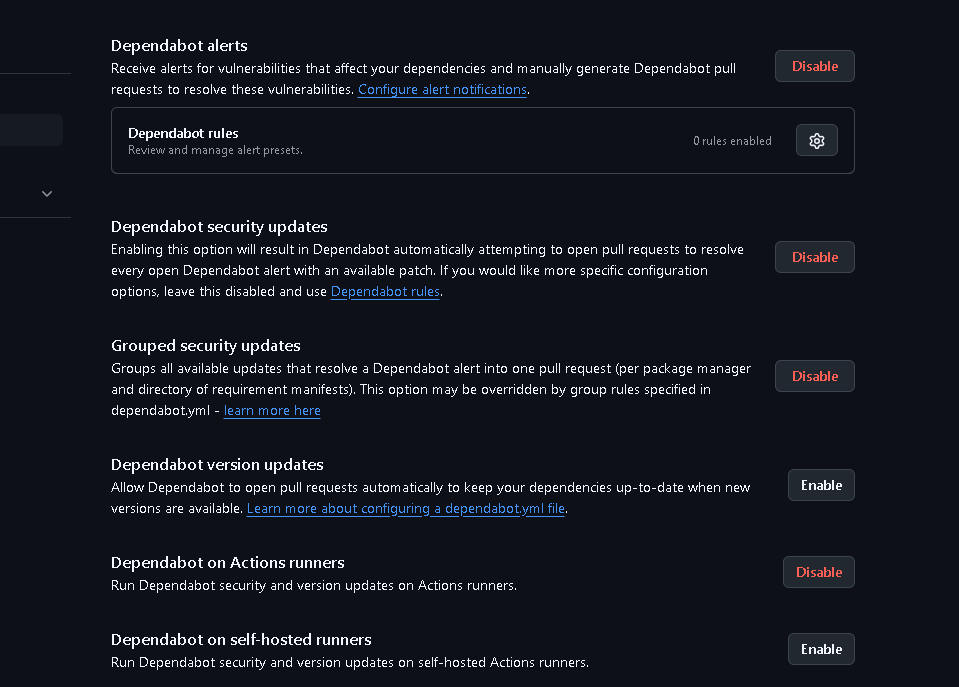
**Gestión de dependencias y seguridad con dependabot en github:**

** lo pprimero es entrtar a security o a settings   
Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto. y se le da click al dependabot**

**En settings se le da click a csecurity uy a code security y luego al dependabot, después de esto se le da habiliatar a las alertas del dependabot**

**Luego de esto se genera la habilitación del dependabot alerts y lasmopciones que aparecen**

**,. Al generar el dependabot versión updates se habilita un archivo yaml**

**Texto

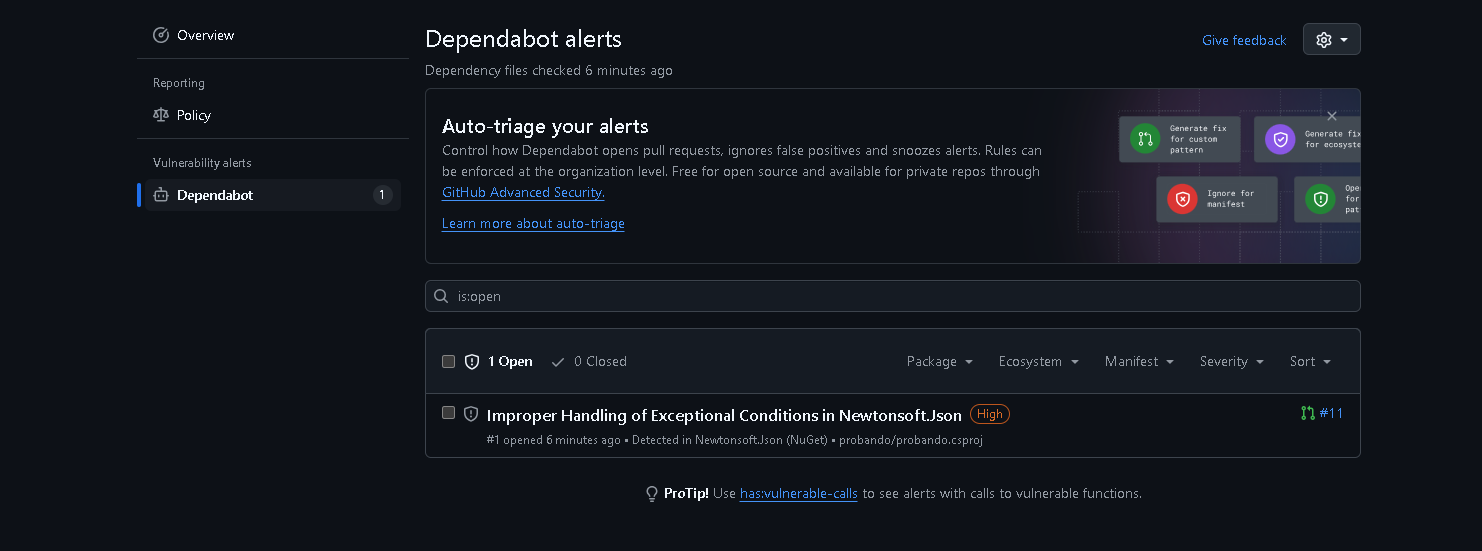
El contenido generado por IA puede ser incorrecto. en el archivo yaml, se genera un intervalo de escaneo que aquí esta semanal pero se puede cambiar, por mi parte lo genero diario**

**Luego de genrado esto se le da click en el commitchange para que capte los cambios.**

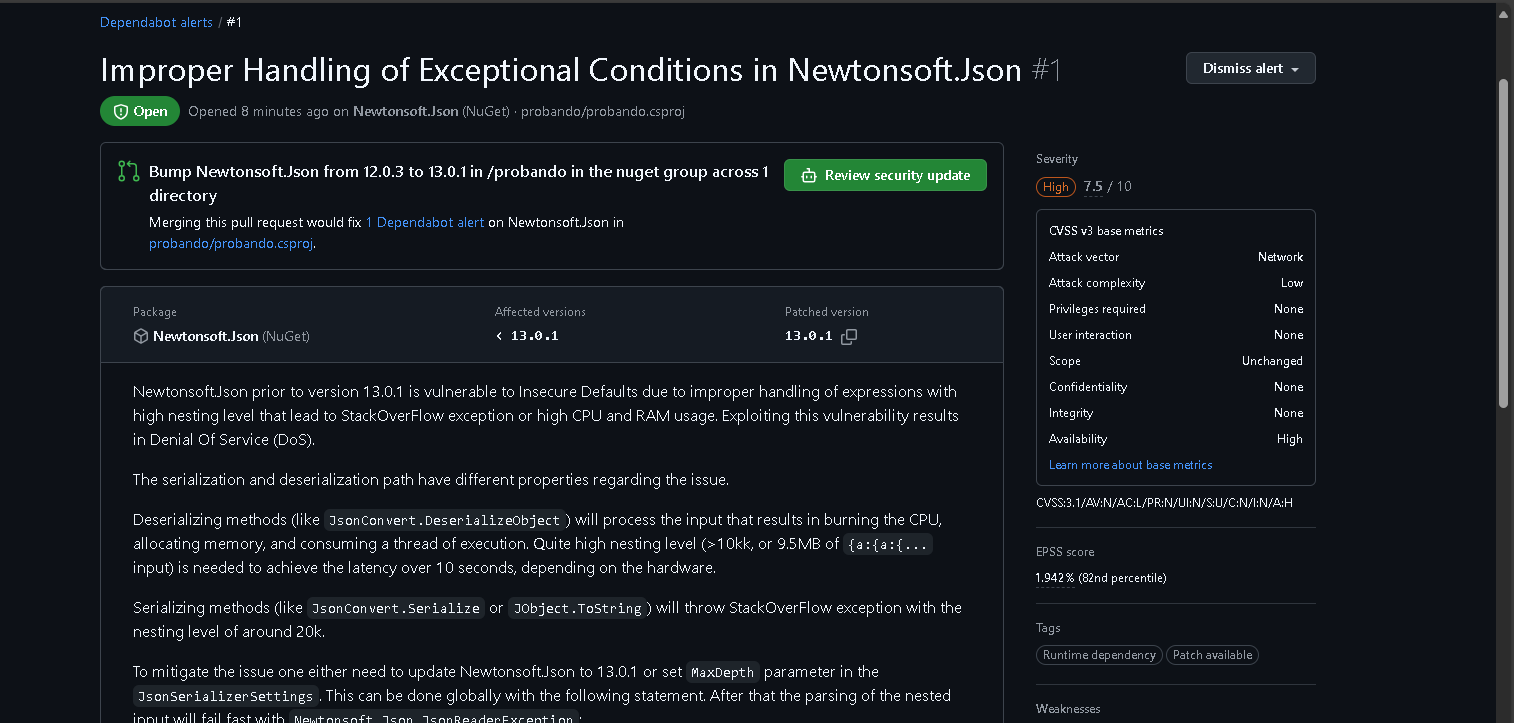
**Luego de esto se guarda en la rama main.**

**Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto. p[ara probar se genera un package con algunas vulnerabilidades**

****

**Después de puesta una dependencia con vulnerabilidades, dependabot lo analizó y verifico el problema**

**con el dependabot, se genero los cambios, generó un analisis de los paquetes y la solución,**

**El depemndabot genera una solución en un pullrequest para que se genera en orden, el dependabot va a ser muy útil por que a mayor aumento del proyecto, mayores packages se usan, dependabot puede automatizar la actualización de packages**

**Mantenimiento de repositorios seguros:**

**Ir a setting desde el repo**

**Y luego a la zona de peligro**

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Con el archivo .gitignore los archivos con el patron designado en este archivo nop van a ser subidos al github repositorio**

**Tener en cuenta aquí para saber que archivos ignorar de acuerdo a el tipo de lenguaje que se va a usar** [**github/gitignore: A collection of useful .gitignore templates**](https://github.com/github/gitignore)

**Se debe generar el archivo gitIgnore dentro del proyecto y con el repositorio mostrado se genera los archivos que no se van a usar,**

**Luego se ponen los archivos git con los comandos usuales.**

****

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto. para optimizar desde aquí se coloca los ignores que se van implementar**

**Gestión de datos sensibles y políticas de seguridad:**

**Dentro del primer paso el repo debe ser publico**

**Luego se va a code scanning Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

**Y a codeQl análisis por default**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto. se genera esto**

**EQUIPO@DESKTOP-9LRK76C MINGW64 ~/miPrimerRepositorio (main)**

**$ git push**

**Enumerating objects: 7, done.**

**Counting objects: 100% (7/7), done.**

**Delta compression using up to 4 threads**

**Compressing objects: 100% (4/4), done.**

**Writing objects: 100% (4/4), 548 bytes | 548.00 KiB/s, done.**

**Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)**

**remote: Resolving deltas: 100% (1/1), completed with 1 local object.**

**remote: error: GH013: Repository rule violations found for refs/heads/main.**

**remote:**

**remote: - GITHUB PUSH PROTECTION**

**remote: —————————————————————————————————————————**

**remote: Resolve the following violations before pushing again**

**remote:**

**remote: - Push cannot contain secrets**

**remote:**

**remote:**

**remote: (?) Learn how to resolve a blocked push**

**remote: https://docs.github.com/code-security/secret-scanning/working-with-secret-scanning-and-push-protection/working-with-push-protection-from-the-command-line#resolving-a-blocked-push**

**remote:**

**remote:**

**remote: —— Stripe Test API Secret Key ————————————————————————**

**remote: locations:**

**remote: - commit: 61689eee6a102565dfe4c764a6b0055f4cadd76a**

**remote: path: probando/Program.cs:4**

**remote:**

**remote: (?) To push, remove secret from commit(s) or follow this URL to allow the secret.**

**remote: https://github.com/OmarGomezDeveloper/miPrimerRepositorio/security/secret-scanning/unblock-secret/2w6TjDjdKmWWOjZ1LlsWdLe9Sc4**

**remote:**

**remote:**

**remote:**

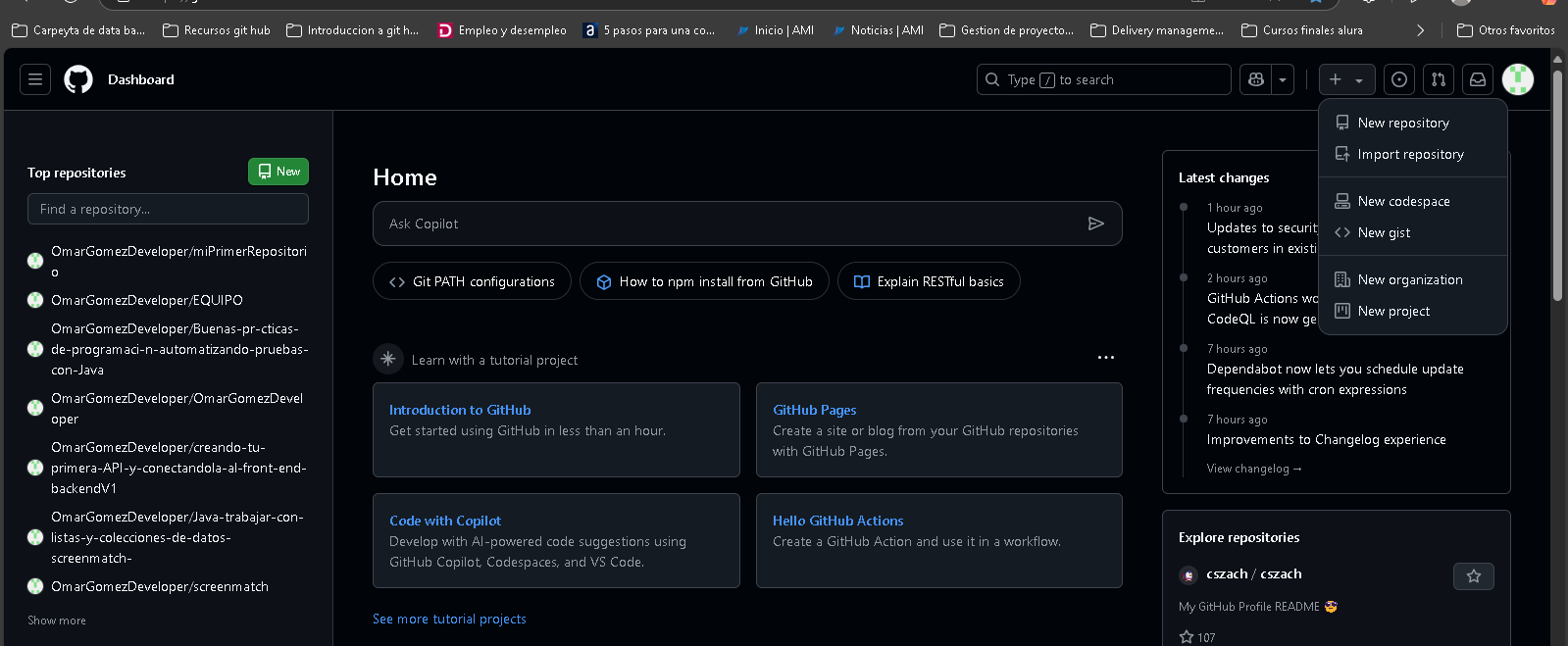
**To https://github.com/OmarGomezDeveloper/miPrimerRepositorio.git**

**! [remote rejected] main -> main (push declined due to repository rule violations)**

**error: failed to push some refs to 'https://github.com/OmarGomezDeveloper/miPrimerRepositorio.git'**

**esto infiere a que el git observo el posible secreto activo o el token mostrado**

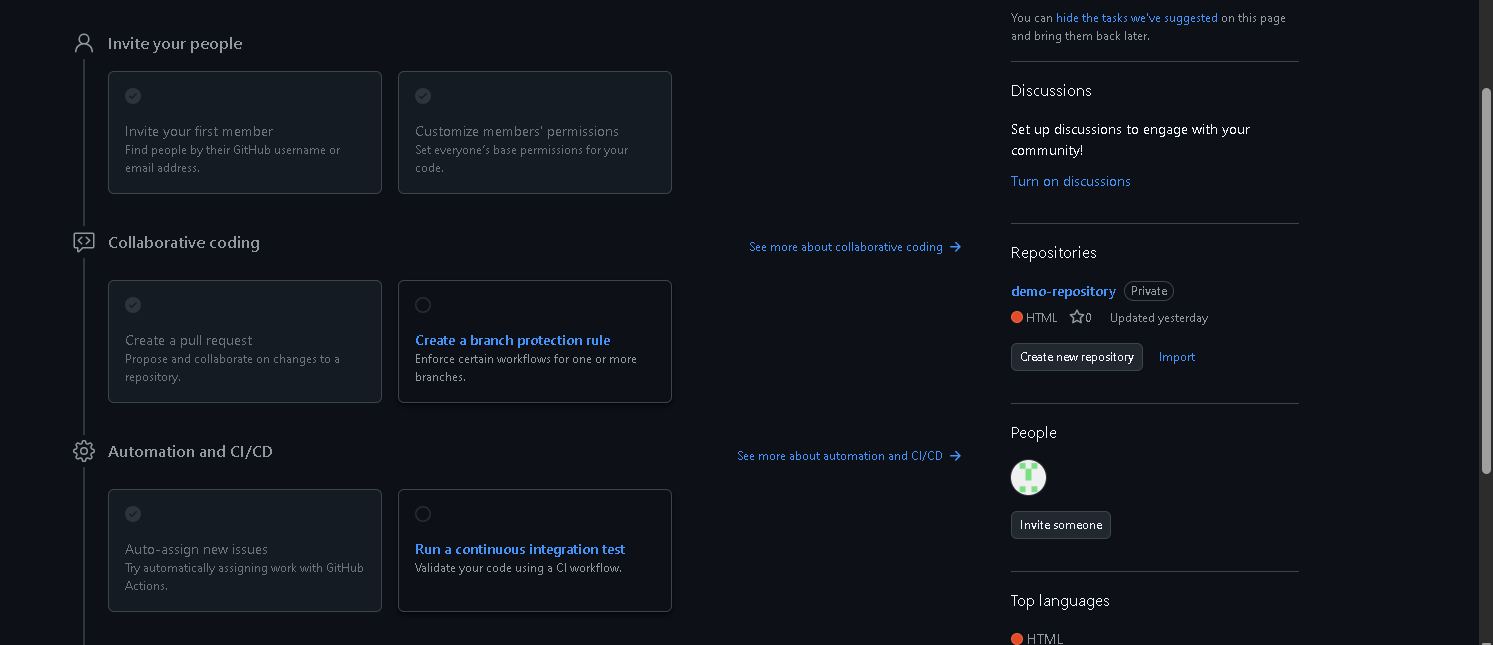
**Administraci;ón de organizaciones en github:**

**se va al plus**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto. ya creado se genera los cambios que se deseen**

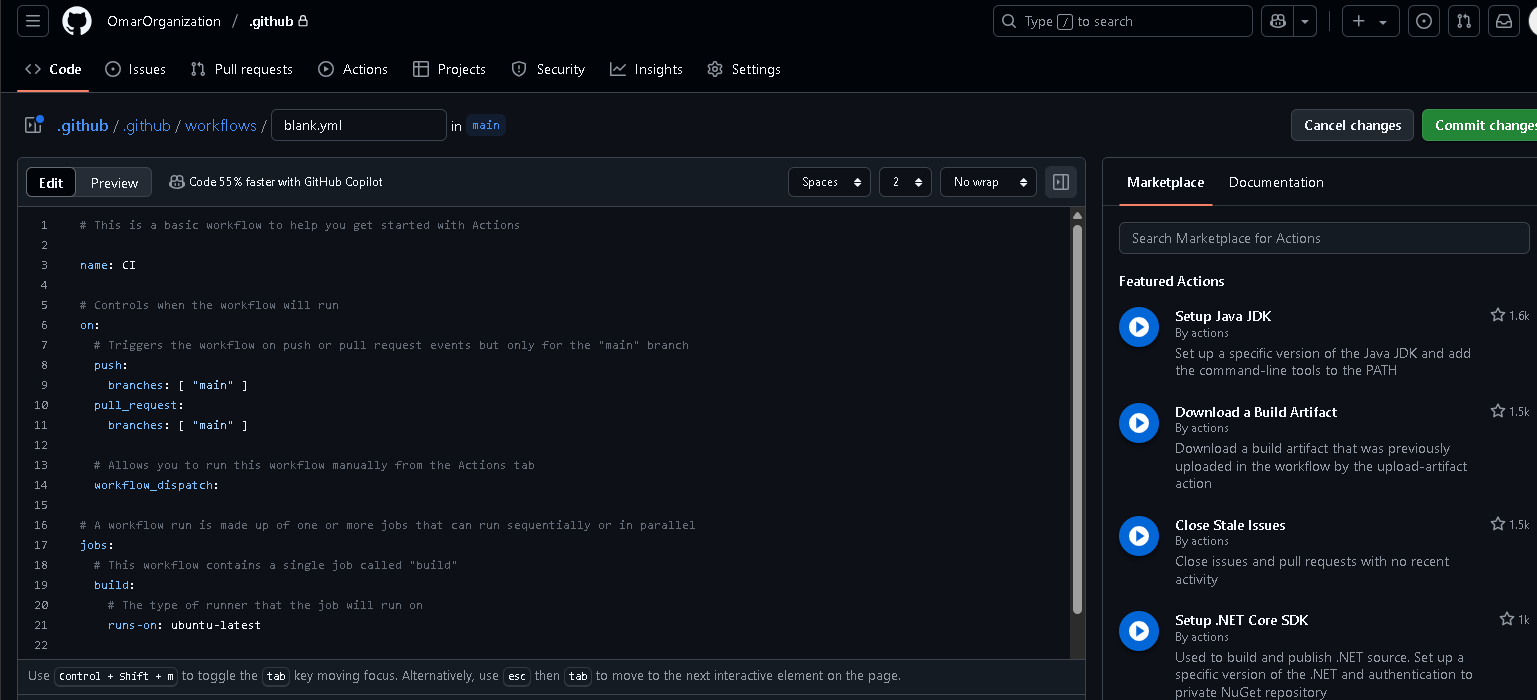
**Desde la part de la organización se da click en el repositorio y se crea uno de los repositorios de la organización**

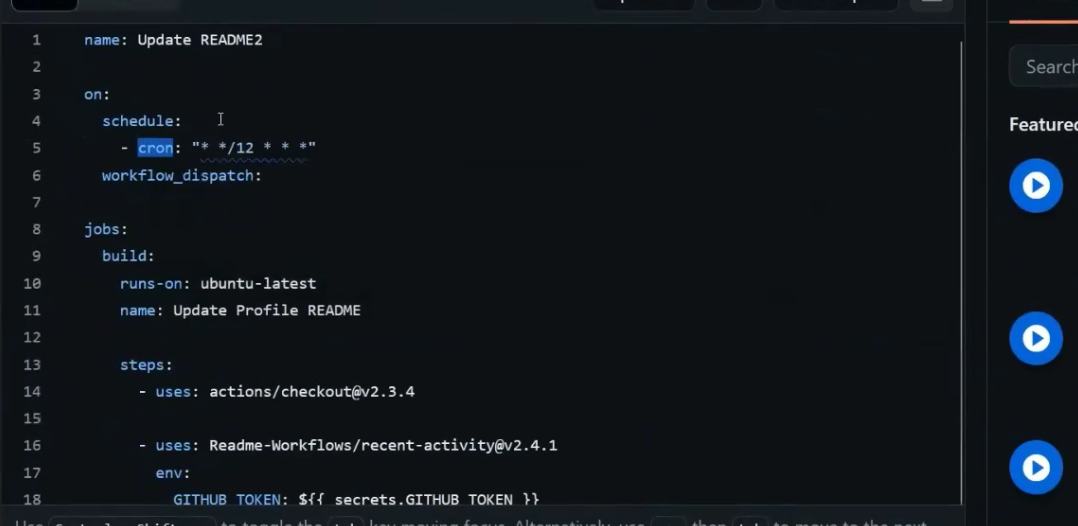
****

**Personalizar el perfil de github con github actions:**

**Se genera la automatización con CI/Cd con Github Actions,**

**al estar en el repo de la organización se da click en el actions, se genera iun workflow**

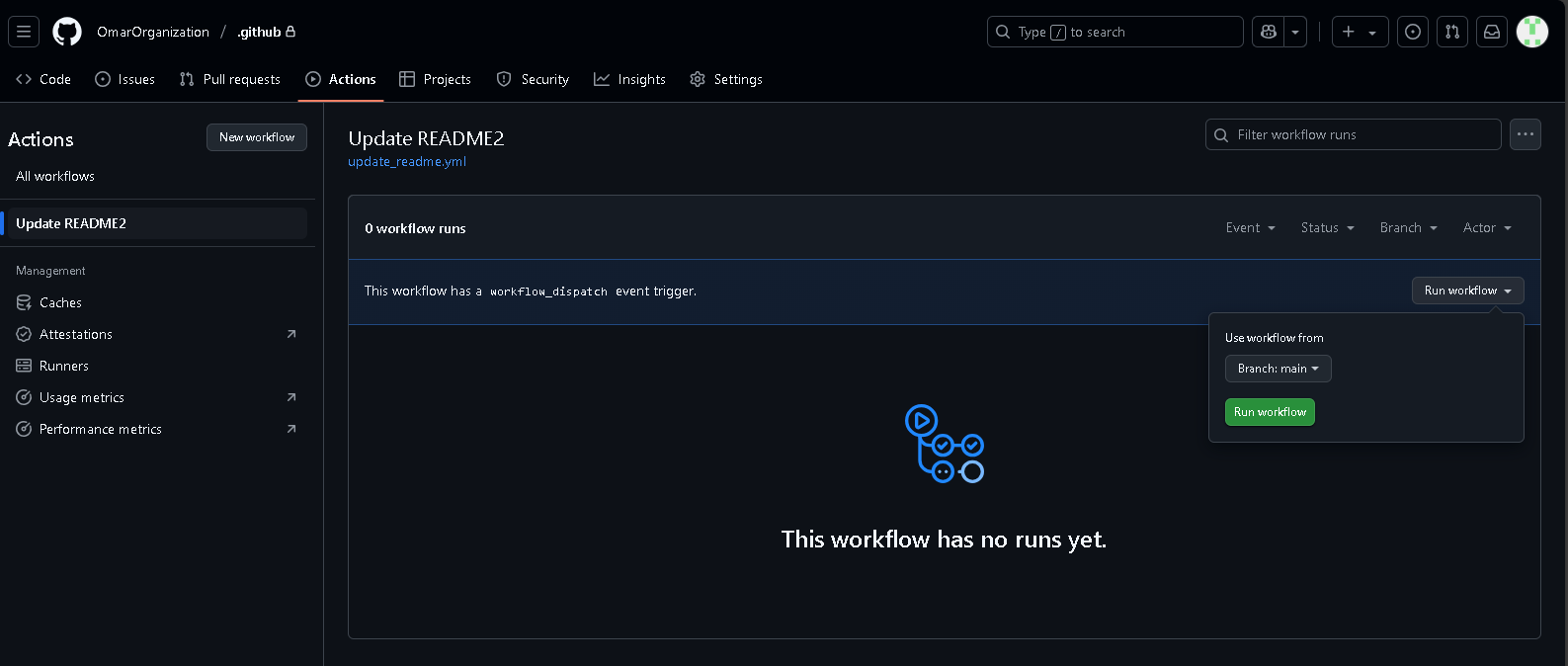
****

** con el cron se genera una periodización para generar un workflow, en este caso se genera por 12 horas,**

**Además se puede generar con Ubuntu**

**Lo que hace el código es en primera instancia nombre el flujo de trabajo, define en que momento se ejecta el flujo de trabajo, define un job llamado build qu ejecuta la maquina virtual con Ubuntu**

**En la pestaña de actions se ejecuta el workflow**

****

**El readme lyego de corrido se genera un analisis y el mismo actions, envia la respuesta de los errores**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

**Para volver a correrla se hace desde el run workflow**

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**V para que corra ir directamente al Marketplace, buscar el github actions readme y buscar el fragmento del código.**

**Administración de Github Cli:**

**Para instalación del github cli se sigue la documentación para la instalación desde la terminal**

[**cli/cli: GitHub’s official command line tool**](https://github.com/cli/cli?tab=readme-ov-file#installation)

**Texto

El contenido generado por IA puede ser incorrecto. dado que se trabaja con Windows s instala desde la terminal de powershell**

**El primer comando será gh auth login**

**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.después de unos pasos pertinentes aparecerá el código que hay que poner para actiar el github**

**Si es ssh hay que generarse antes de ponerla**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Para crear un repositorio se genera con:**

**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.**

**Luego se hacen las modificaciones a consideración del desarrollador**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Para optimizar el proceso anterior:**

****

**Para generar un issue**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

**Para generar el status**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Como desventaja de la cli es que conforme más la usamos más dependencia y arraigo se tiene a github, cuando git es universal y se puede usar con otros productos, si se genera con cli es para github pero git es más amplio**

**Pullrequest y ramas en github:**

**La política principal es nunca trabajar en main, se debe trabajar en una rama propia y luego subir los cambios**

** se genera la creación desde le Branch propio, se actuializa el peroyecto**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Texto

El contenido generado por IA puede ser incorrecto. s genera esto darle click para entrar directamente a generar el pullrequest**

**Revision y fusión de pull request:**

**Texto

El contenido generado por IA puede ser incorrecto. aquí se debe tener cuidado dado que se generó un conflicto en el trabajo dadas las ramas main y Omar-Tarea1**

**Git rebase estrategias de fusión de ramas:**

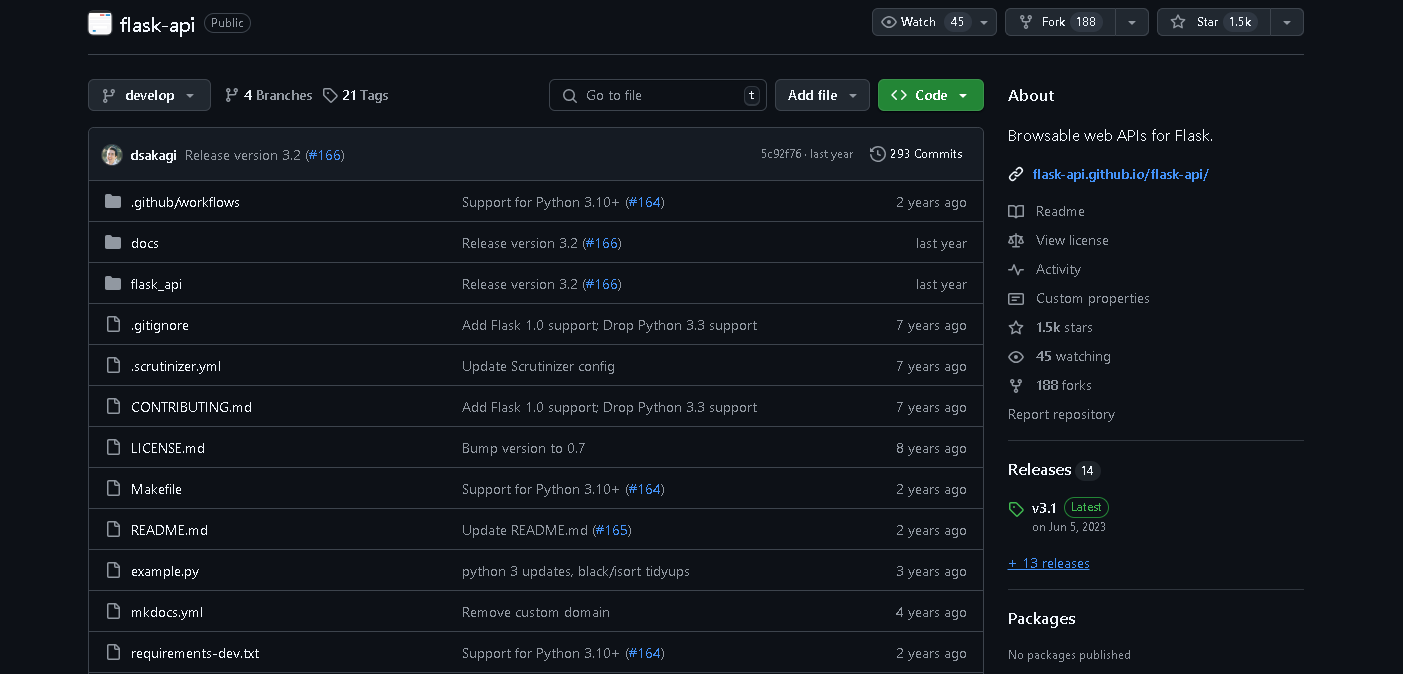
**Texto

El contenido generado por IA puede ser incorrecto.**

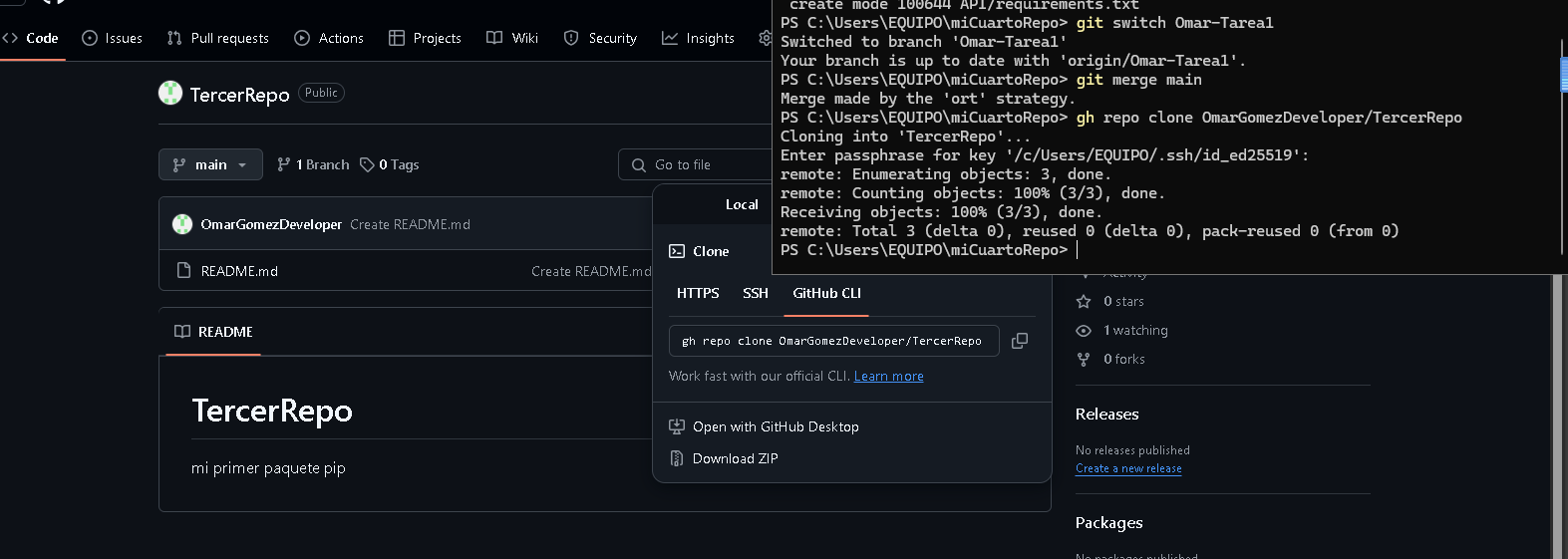
**Texto

El contenido generado por IA puede ser incorrecto.**

**Introducción a los github reales:**

** cuando se vaya a ver proyectos en git, se debe mirar si hay releases, en este caso se tiene 14 lanzamientos**

**Un reléase no puede ser privado, siempre será publico**

**después de tomado el repositorio cli ya se genera el clone para usar el proyecto con el cli.**

** con esto se genera la creación del paquete**

**Texto

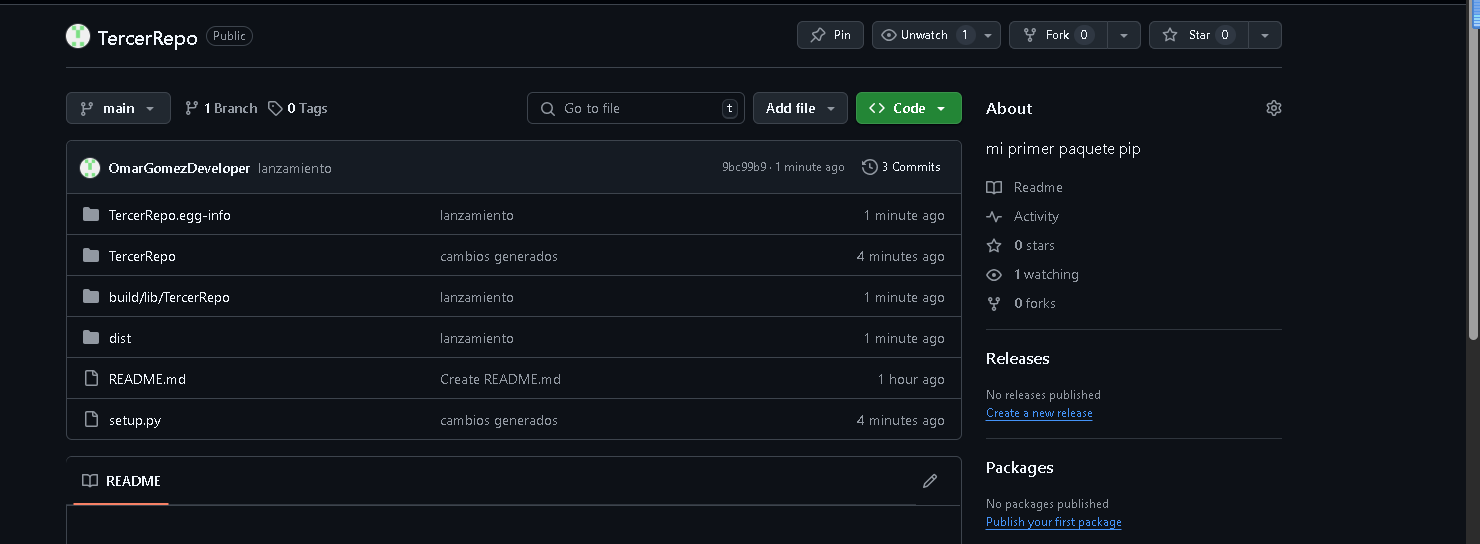
El contenido generado por IA puede ser incorrecto.se genera la subida del paquete con el comando**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Sin embargo hay que tener cuidado de que todo esté bien configurad**

**Publicación de paquetes en github y pypi:**

** lo primero es ir al reléase, luego se crea la tag, quees el versionado:**

**De no tener taag, se crea uno Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Luego darle click al atach binaries Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**En este caso se va a la ruta del proyecto y se busca la carpeta dist donde están los archivos**

**Y se suben tanto el tar.gz como el .whl**

**Una captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto. ya con esto se crea el versionado con el release**