GIT / GITHUB

RECORDATORIOS GENERALES

Git es un Sistema de control de versiones. Administra las distintas versiones de un programa(VCS).

Ideal para múltiples desarrolladores en un mismo proyecto.

Queda un registro de cada línea de código, por lo que siempre se puede volver a versiones anteriores. Por ello, pueden revertir cambios a la instancia que precises.

Puede trabajar con repositorios locales o remotos.

Para utilizar repositorios locales utilizo git, para trabajar con repositorios remotos utiliza github.

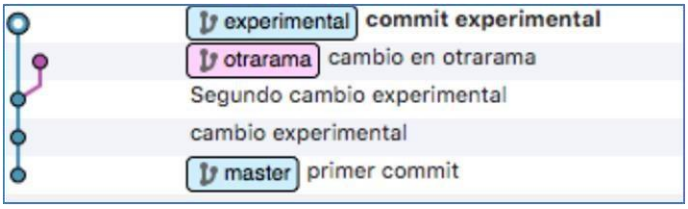
Github es una “nube” de repositorios. Existen varias empresas que ofrecen el mismo servicio

Términos más utilizados

- Repositorio → Proyecto que ya es seguido por GIT.
- Repositorio Remoto → La nube que contiene los repositorios
- Repositorio Local → El buck propio, estos se vinculan al remoto, Varias PC pueden unirse a un mismo “Repositorio Remoto”.
- Commit → Cambios registrados por los desarrolladores.
- Ramas → Bifurcaciones de un proyecto. Generalmente la rama principal se llama “master”. Esto es muy práctico para trabajar de forma independiente al proyecto principal y lograr así realizar pruebas y modificaciones sin riesgo de dañar o comprometer el proyecto principal.

Flujo básico de trabajo

- Crear un repositorio. Ya sea nuevo o clonado de otro repositorio.
- Trabajo en diferentes archivos, los voy pasando a Staging área “Add”.
- Envió los datos al repositorio.(Commit)
- Solo el líder de proyecto es quien debe utilizar la rama “Master”.
- Todos los otros integrantes de proyecto deben utilizar la rama “Develop”, es desde aquí que van a sacar sus propias ramas para trabajar en la tarea específica para luego ser unificadas a la rama. Esta tarea generalmente también es revisada por el líder de proyecto.



INDICE:

Dejar mis credenciales en GitBash.

Crear un repositorio REMOTO

Crear un repositorio LOCAL (Vinculado con Repositorio Remoto) Subir cambios desde Repositorio LOCAL al Repositorio REMOTO.

Ver los cambios en Repositorio Remoto.

Volviendo a algún punto en el tiempo (ubicarnos en un commit especifico)

Crear una Rama desde mi Repositorio REMOTO

Actualizar una Rama en mi Repositorio LOCAL

Crear una Rama en mi Repositorio LOCAL

Hacer Merge desde Consola

Hacer Merge desde GitHub

Clonar un Repositorio

Algunos comandos de GitBash

Dejar mis credenciales en GitBash

- Abro la consola propia de GIT
- Escribo → git config -global user.name “Mi nombre y Apellido”
- Escribo → git config -global user.email “Mi correo electrónico”
- De este modo, mi PC queda registrada bajo esas credenciales, ya que toda modificación que realice en un proyecto, debo “etiquetarlo”, es decir, dejar una referencia de quien y que cambio se realizó.

Crear un repositorio REMOTO

- Usar GITHUB → https://github.com
- Loguearse → En caso de no tener ya una cuenta, crear una.

Vista Repositorios:

- Extremo superior derecho, Your Repositories
- NEW
- Completo la información:
    - Repository name: Nombre que le quiero dar a mi repositorio
    - Description : Algún tipo de descripción que quiera añadir para tener sobre que contiene el repositorio que cree.
    - Acces : Público o privado
- Una vez creado, podre copiar una URL de ese repositorio puntual.

Crear un repositorio LOCAL (Vinculado con Repositorio Remoto)

RECORDAR TENER INSTALADO CON ANTELACION GITBASH

- Ir a la ubicación de mi Proyecto en mi PC.
- BOTON DERECHO Sobre mi proyecto → GitBash → Abre la consola propia de GIT
- Escribo → git init (indicando que voy a inicializar un repositorio)
- Retorna → La ruta donde se crea este repositorio
- Voy a vincular mi repositorio remoto con el local
- Voy a la pagina de GitHub, copio la URL de mi "Repositorio Remoto" (Ej: https://github.com/AdrianaBestilleiro/pruebaEgg.git)
- Vuelvo a la consola de GitBush
- Escribo → git remote add origin URL (Copio la URL especifica – Creando así la vinculación entre ambos repositorios)

Subir los cambios al Repositorio Remoto.

- Abro la consola propia de GIT
- Escribo → git status (para ver que esta en el local, pero no en el remoto)
- Escribo → git add . (para preparar todo lo nuevo/modificado)
- Escribo → git status (verificando que todo ya esta listo, en color verde, para vincularlo)
- Escribo → git commit -m "NombreMiCommit" (Este nombre es la ETIQUETA que va a tener este commit. Recordar utilizar etiquetas acordes y apropiadas) Queda lista la info para ser enviada al repositorio remoto.
- Escribo → git push origin master (Aclaracion: origin es el alias que utilice cuando cree mi repositorio remoto, master hace referencia a la "rama" donde va a estar la info)
- Va a pedir que verifiquemos las credenciales con el usuario que creamos en GITHUB
- Si voy a la pagina de GitHub, ejecuto CTRL+SHIFT+R, veo que se actualiza mi repositorio, creando la rama "master". Encontrando todo lo que mande desde mi Repositorio Local al Remoto.

- SI QUIERO INCORPORAR ALGO QUE MODIFIQUE , realizar otro Commit ○ → git add . (Preparo) ○ → git commit -m “NombreMiCommit” (Etiqueto y preparo) ○ → git push origin rama Especifica (Envio la info a la rama especifica)

Ver los cambios en mi Repositorio Remoto

- Ir GITHUB → https://github.com
- Recordar refrescar los cambios en la página si ya la tenía abierta, CTRL+SHIFT+R.
- Voy a ver Mis commit disponibles, si accedo a uno de ellos, me mostrara cual fue el cambio realizado.

Volviendo a algún punto en el tiempo (ubicarnos en un commit especifico)

- Abro la consola propia de GIT
- Escribo → git log (Listándome todos los commit y sus id específicos) □ Copio el id del punto al que me interesa volver.
- Escribo → git chekout xxxxxxxxx (Siendo xxxxxx el id copiado)
- Esto borrara los cambios realizados en mi Repositorio Local, es decir, los cambios que realice después del punto que elegi para volver.

Crear una Rama desde mi Repositorio Remoto

- Desde GitHub. Me posiciono sobre el proyecto especifico
- Click comando **Master**

- Se abre desplegable, puedo escribir el nombre de la nueva rama y crearla de forma directa → créate Branch : nombreNuevaRama from RamaOrigen.
- ES BUENA PRACTICA : Trabajar sobre Rama Develop, nunca sobre rama Master. Cada desarrollador tendrá a su vez, una rama creada desde Develop. La Rama Master debe contener solo proyectos iniciales o proyectos listos para producción o ya en funcionamiento
- Crea la NUEVA RAMA. La nueva rama tendrá todo aquello que tenia su rama de origen. Es decir, todo aquello que ya esta creado desde ese punto de creación.

Actualizar una Rama en mi Repositorio Local (Es decir, cree una rama desde GitHub y quiero que se actualice en mi PC)

- Escribo → git fetch (Para traer lo que esta en el Repositorio Remoto)
- Escribo → git branch -a (Voy a ver todas las ramas existentes tanto en mi repositorio remoto como en mi repositorio local)

Crear una Rama en mi Repositorio Local -(Para trabajar de forma segura)

- Primero debo posicionarme sobre la rama especifica donde va a nacer esta "nueva rama"
- Escribo → git chekout nombreRama (Para cambiar de rama).
- Escribo → git branch nombreNuevaRama (Para crear esa rama especifica). Por buena practica, se sugiere feature/nombrePersona/TareaEspicifica (Ej: git branch feature/adriana/login)

- Escribo → git Branch -a (Voy a ver todas las ramas existentes tanto en mi repositorio remoto como en mi repositorio local siendo la que estoy posicionado la que esta en color verde)
- Debo enviar la info al repositorio Remoto
- Escribo → git push origin ramaCreada ((Ej: git push feature/adriana/login) Verifico en la pagina de GitHub que se creo la nueva rama

Hacer un Merge desde consola

- Primero debo posicionarme en la rama especifica que va a recibir el merge.
- Escribo → git chekout nombreRama (Poniendo el nombre de la rama que va a recibir las novedades originadas en otra sub rama).
- Escribo → git pull origin nombreRama (Poniendo el nombre de la rama que va a recibir las novedades originadas). Esto lo hago para asegurarme que NADIE hiciera cambios sin yo tenerlos en cuenta con antelación . Me traigo la info nueva si existiera
- Escribo → git merge nombreRamaQueEnvia Debo enviar la info al repositorio Remoto
- Escribo → git push origin nombreRama (Poniendo en nombre de la rama que recibió

Hacer un Merge desde GitHub

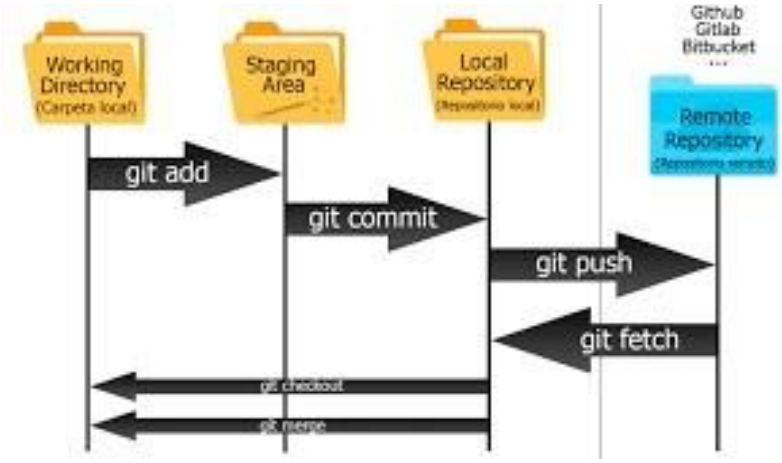
- Pull requests →New pull requests
- Elijo rama base (quien recibe) / Elijo rama compare (quien envía)
- Click Create pull requests
- Puedo dejar un mensaje, información adicional al respecto en ventana emergente que se abre.
- Confirm Merge

Clonar un Repositorio

- Creo una carpeta contenedora en mi PC, en la ruta que considere apropiada
- Desde la carpeta (que va a estar vacia), BOTON DERECHO, GitBashHere

- Escribo → git init
- Escribo → git remote add origin url (La copie desde GitHub, del repositorio que quiero hacer una copia)
- Escribo → git pull origin master (Traigo toda la info)
- Descargue el repositorio a mi PC
- Recordar luego, crear nuestra rama de trabajo

Algunos comandos de GitBush



- git init → Comando por consola para avisar que vamos a crear un repositorio nuevo local.
- git status→ Comando que permite conocer el estado de los archivos, informa que ha cambiado en la carpeta de trabajo.
- git remote -v→ Para verificar vinculaciones entre Repositorios Remotos y Locales
- git diff→ Comando que permite conocer las diferencias entre versiones de un archivo. Marca + o - (según si añadí o quite algo). Compara las versiones entre el ultimo commit y la situación actual.
- git push→ Comando para subirlo al repositorio remoto(Ej: Github). Va a solicitar usuario y contraseña.
- git pull→ Comando para traerte los cambios que realizaron otros desarrolladores. Sincroniza cambios de otros repositorios
- git checkout → Me permite saltar entre versiones, entre ramas o descartar cambios.
- git log→Retorno el historial de comit y el contenido de sus mensajes. Permite examinar el registro. Retorna el id, el autor, fecha, comentario de cada comit.
- git config -l→ Ver configuración general.
- git config user.name "Mi nombre y Apellido" → Declaro mi credencial

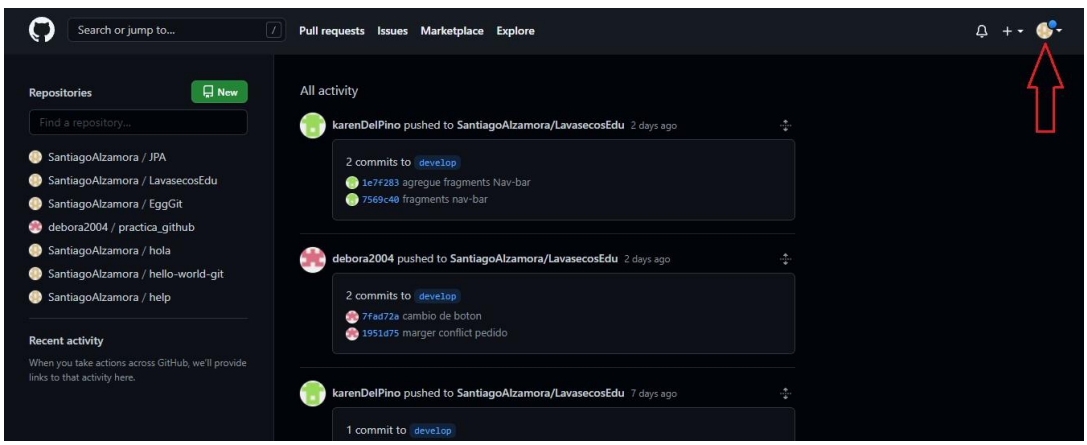
- git config user.email “Mi correo electrónico”→ Declaro mi credencial clear → Limpia la consola.
- git status → Muestra todo lo que este en mi repositorio local “nuevo”, es decir, todo aquello que sufrió cambios en mi PC y que no fue incorporado al Repositorio Remoto. Aparece en color rojo
- git commit → Para establecer el commit.
- git branch → Info sobre las ramas existentes en mi repositorio.
- git add→ Comando para ir pasando al Stagin Area los archivos con los que estoy trabajando. Puedo usar . (todos) o un archivo especifico.
- git help → Comando por consola para conocer todos los comandos disponibles. Si le agrego un comando especifico, solo me brinda información sobre ese.
- git merge→ Comando que permite unificar las ramas de distintos desarrolladores a una única rama. Se ejecuta desde la rama padre y se llama a la que vas a añadir. Ojo, puede suceder conflicto si dos usuarios diferentes hicieron cambios en el mismo código, el sistema pedirá que se elija y decida por uno de ellos.
- git clone→ Comando por consola para clonar un proyecto ya existente. (incorporo a la orden la url donde se encuentra el repositorio original .git clone <url>.

ESC :wq (para volver atrás), cuando sale una ventana emergente.

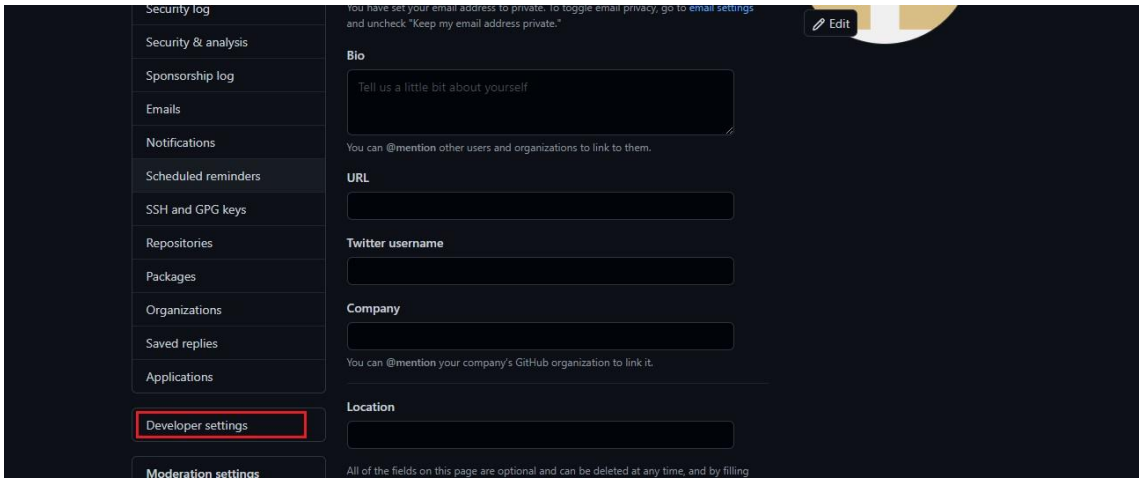
Creación de TOKEN en Github

Pasos a seguir:

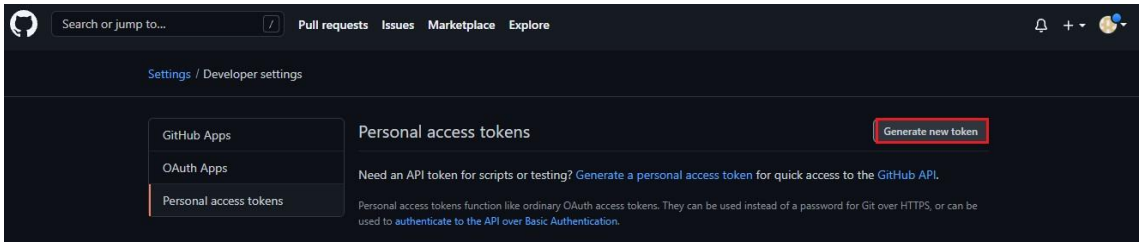
1 - Ingresar a tu cuenta de Github y presionar tu imagen para que despliegue el menu



2 - Ingresar a Settings y buscar la opción Developer settings

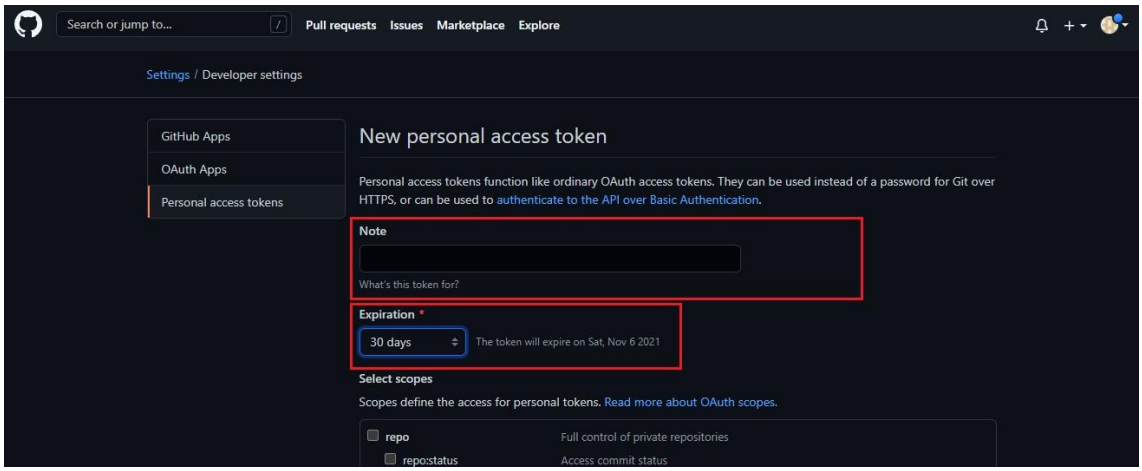


3 – Presionar sobre Personal access tokens y generar un nuevo TOKEN (generate new token)

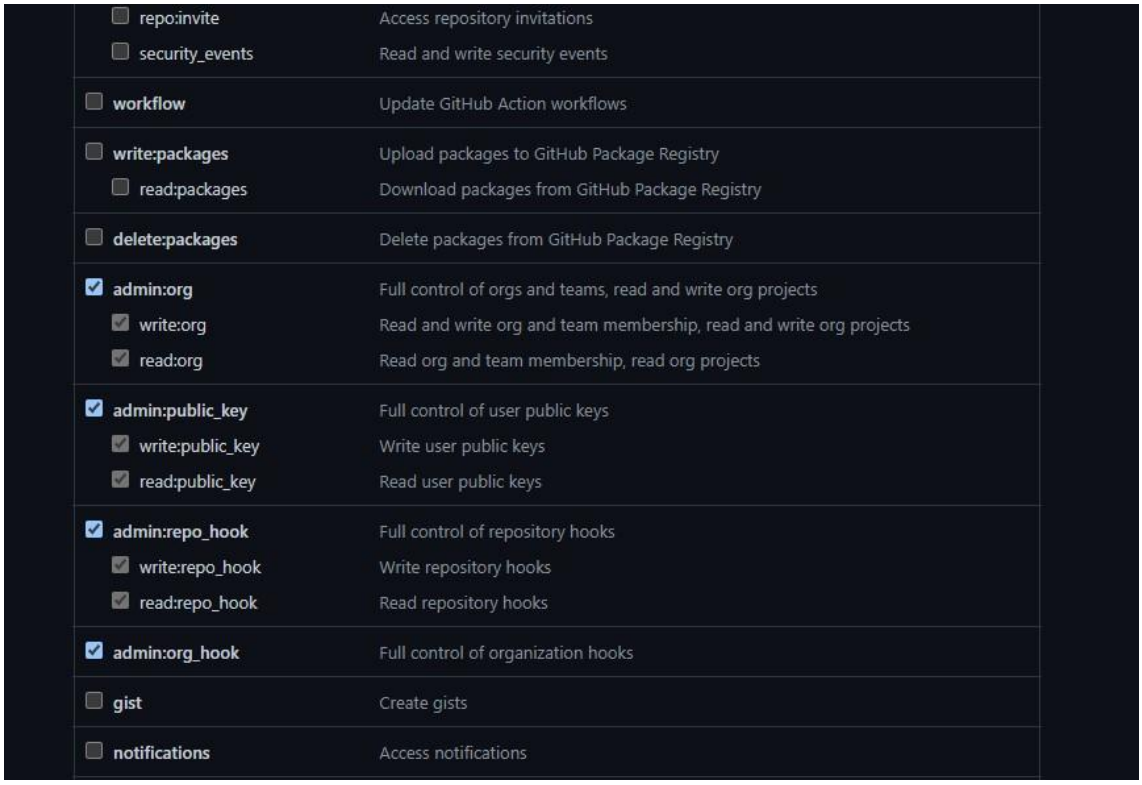


4 - Deben ingresar cualquier tipo de nota que ustedes quieran (es un nombre para identificarlo)

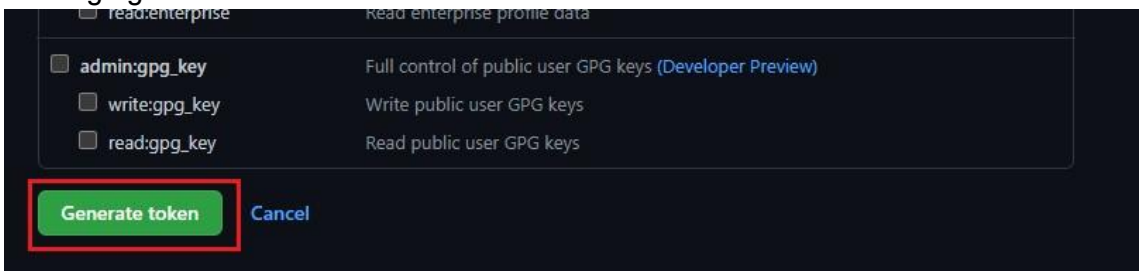
En Expiración pueden también decidir cuánto quieren que les dure el token, pueden poner la opción que quieran



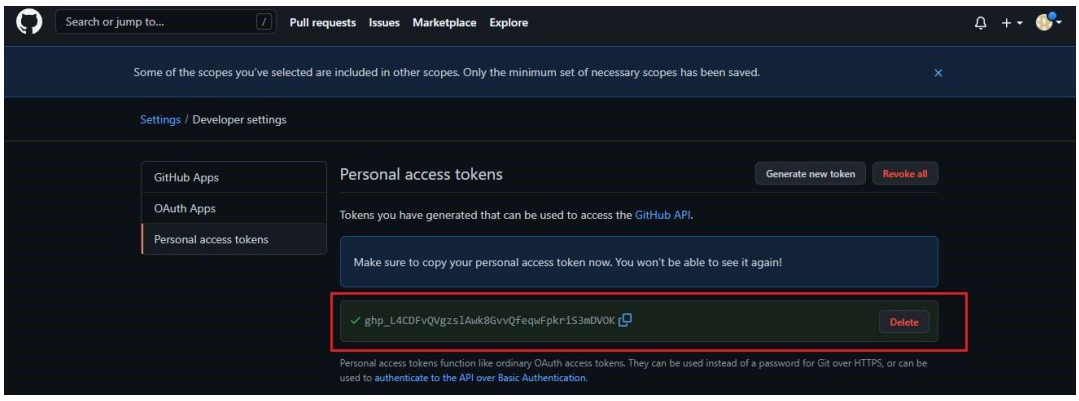
5 - Luego deben tildar estas 4 opciones



6 – Luego generan el token



7- Copian el token y guardenlo



8 - Luego en la consola, luego de haber hecho un git push y que les haya saltado un error ingresen lo siguiente:

Git remote set-url origin https://TOKEN@github.com/USUARIO-QUE-CREO-EL-REPOSITORIO/NOMBRE-DEL-PROYECTO.git

Siendo TOKEN el token generado.

Les dejamos un ejemplo de cómo quedaría con un token, usuario y repositorio:

Git remote set-url origin
https://kpq_brqZ6TfMr7xLRqVYCBZHJIARDTsKRg1mAFLa@github.com/Pepesuarez/miprimerproyectoengit.git
(ingresen todo en la misma línea)