Configuración VSCode y Git

En estos últimos años unos han manejado Git desde la consola (línea de comandos), otros han tirado de herramientas gráficas independientes como **SourceTree** o **GitKraken**, todos huyendo un poco de los IDE's integrados que al final te hacían cliente cautivo.

De un tiempo a esta parte se ha levantado *Visual Studio Code:*un editor ligero, potente, multiplataforma y open source, que va mejorando por semanas y además se ha ganado la simpatía de muchos sectores.

En este post vamos a aprender manejarnos con este cliente de Git.

Lo primero que tenemos que hacer es instalar el programa git

<https://gitforwindows.org/>

<https://git-scm.com/>

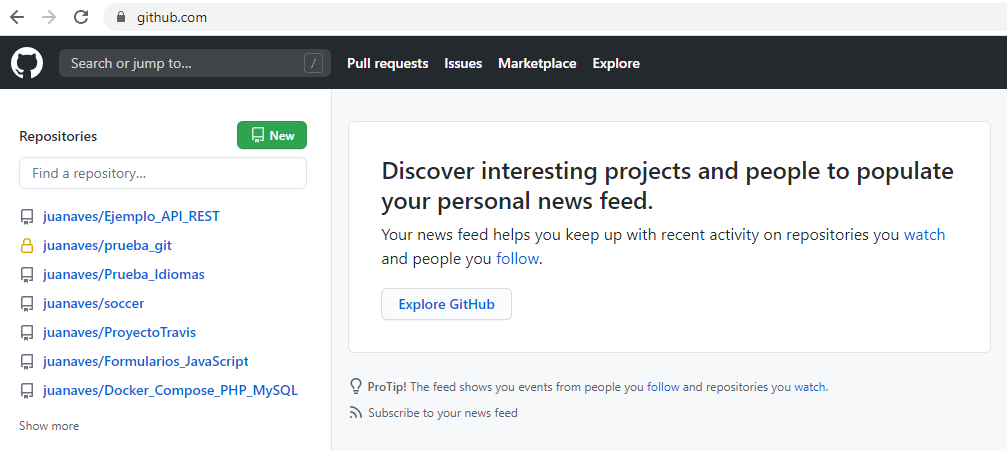
Nos creamos una cuenta en **GitHub**, **GitLab** o **Bitbucket** que son tres de los mejores servicios web de control de versiones.

GitHub es un entorno en la nube que nos permite crear repositorios remotos y, gracias a Git, podemos trabajar en local y coordinarnos con ese repositorio remoto. De ese modo tenemos la capacidad de salvaguardar nuestro trabajo (todas las versiones relevantes para nosotros) y coordinarlos con el de otra gente.

Primero veamos cómo podemos crear un repositorio en GitHub y luego explicamos otros conceptos básicos.

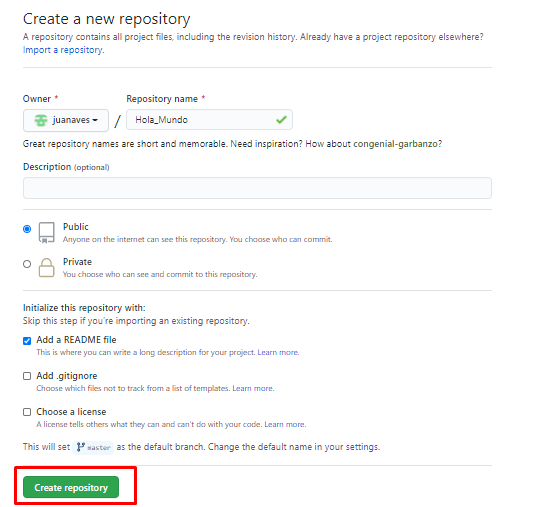
Vamos a

<https://github.com/> y nos creamos una cuenta o utilizamos la nuestra ya creada, como es mi caso.

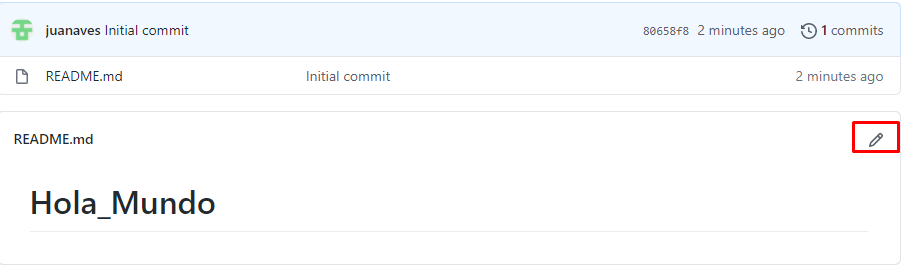


Vamos inicialmente a trabajar en la página Web de GitHub que es más sencillo para entender los conceptos sin otras complejidades.

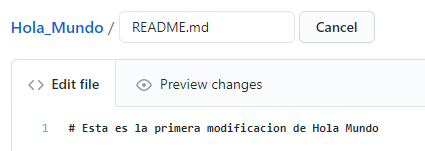
Creamos un repositorio (agrupación que representa un directorio realmente), público inicialmente. Le decimos que nos cree un fichero **README.TXT** con el que vamos a jugar.



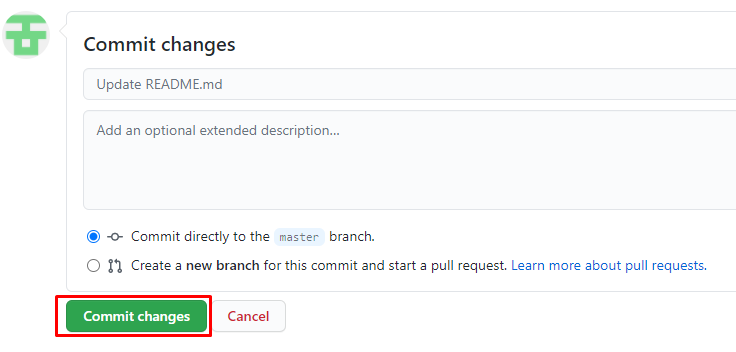
Vamos a visualizar el fichero y editarlo



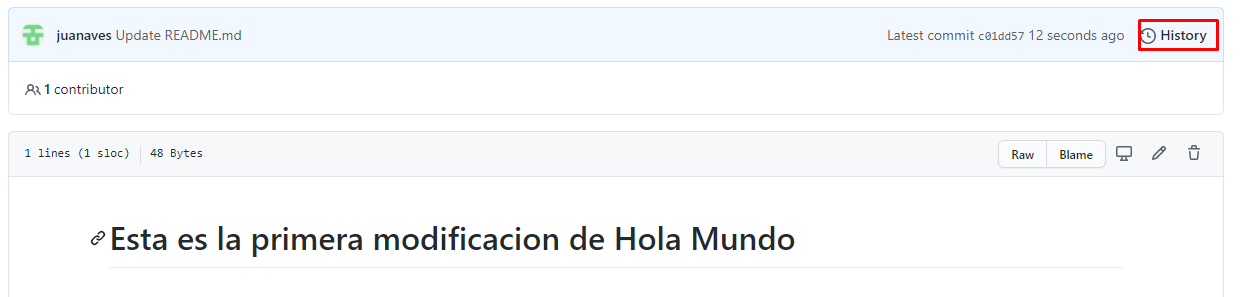
Modificamos su contenido



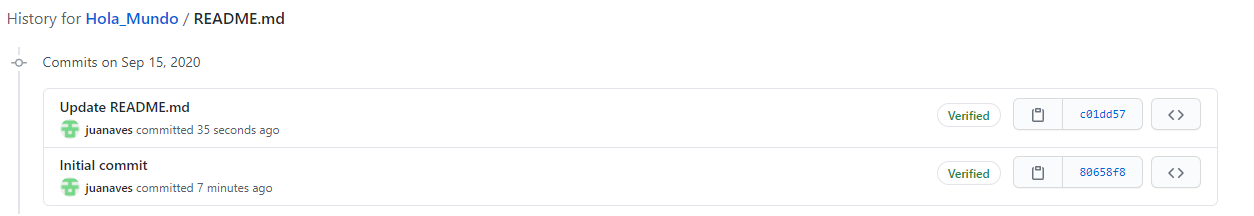
Al final de la página vamos a hacer commit y guardar el fichero de nuevo en el repositorio. Esto es la parte final de la pantalla anterior.



Podemos comprobar el histórico de cambios del fichero e incluso rescatar todas las versiones anteriores.



Y este sería el resultado

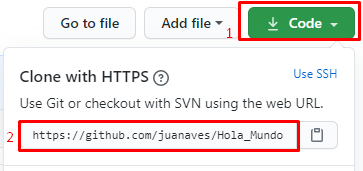


Una vez me he creado un repositorio voy a clonarlo (hacer una copia) para poder trabajar con ella desde VSCode.

Vamos a ver cómo trabajar desde VSCode para enlazarlo con alguno de los servicios de gestión de versiones que existen, en nuestro caso lo vamos a hacer con Github pero podríamos utilizar cualquiera.

***Clonar un repo***

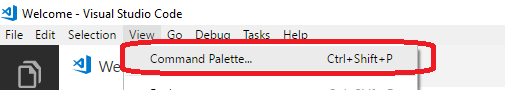
Arrancamos para **clonar** un **repositorio** para ello copiaremos la URL del repo (en este caso tenemos el repo en **github** pero funcionaría desde bitbucket, gitlab, vsts u otro).



Vamos a clonar un proyecto hospedado en GitHub.

 Para clonar un repo, lo podemos hacer desde la paleta de comandos.

* Para mostrar la paleta de comandos: podemos pulsamos la combinación de teclas *CTRL+SHIFT+P* o desde el menú view seleccionamos "command palette".

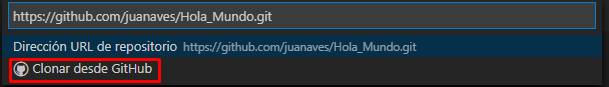


Mostrar la paleta de comandos (CTRL+SHIFT+P)

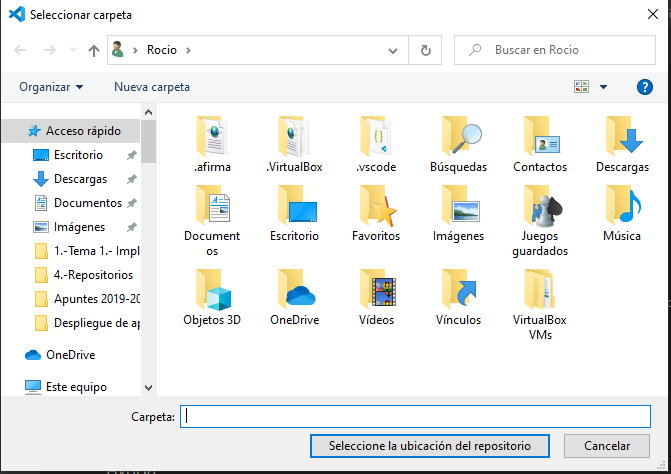
* Desde ahí tecleamos ***git clone***



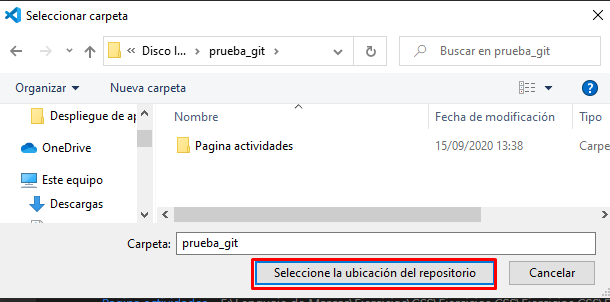
nos pide la ruta del repositorio y ya ponemos clonar (si es necesario nos pide credenciales).



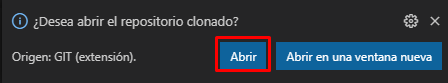
Ahora nos pedirá la carpeta en local donde vamos a tener nuestro proyecto.



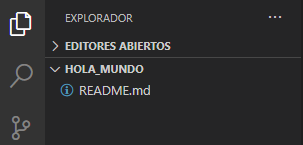
Seleccionamos una



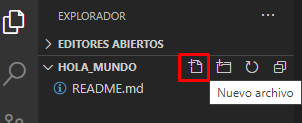
Nos aparecerá un mensaje en la parte inferior derecha para abrir el repositorio local



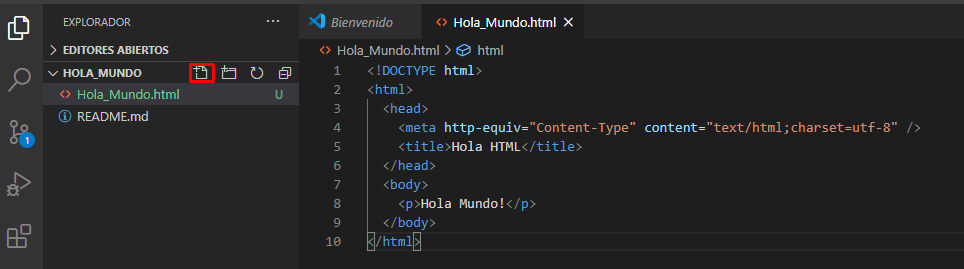
Al pulsar sobre el bobón Abrir ya nos aparecerá en nuestro espacio de trabajo de VSCode el readme que habíamos creado antes



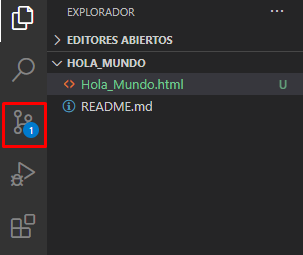
Voy a crear un archivo que se va a llamar Hola\_mundo.html



El resultado será este



Si os fijáis en la parte derecha ha aparecido sobre el icono que vamos a utilizar lara los repositorios un número, comunicándonos que ha habido un cambio en el proyecto.

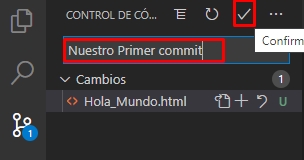


***Introduzco cambios, commit***

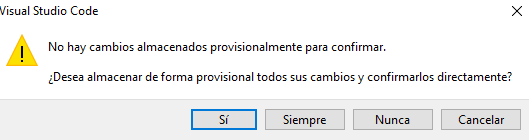
Los commit no son más que los cambios que hacemos en nuestro proyecto, el los guarda en nuestro repositorio local.

Conforme vamos introduciendo cambios en los ficheros estos se nos marcan como modificados o añadidos.

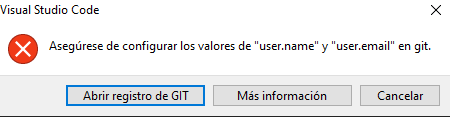
Si pulsamos sobre el icono de las ramas, nos aparecerá una pantalla en la cual nos pedirá que introduzcamos nuestro primer comentario para hacer un commit.



Pulsaremos sobre el botón confirmar para confirmar los cambios en local (**OJO en LOCAL**)



Si nos da un error del tipo



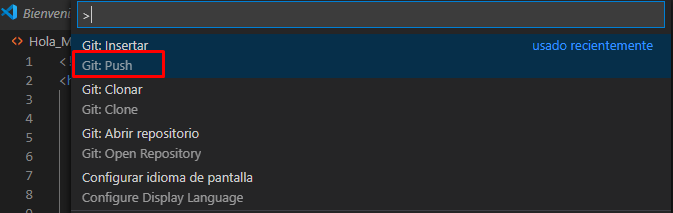
Mirar al final de este documento y ver cómo solucionarlo.

Una vez que estamos listos le damos al botón de **commit** (esto sólo guarda en local, tenemos que hacer un **push** para llevar los cambios a servidor).

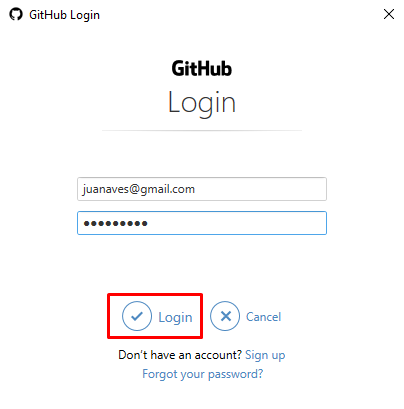
***Hacer push de los cambios***

Para hacer push de los cambios podemos hacerlo de dos maneras:

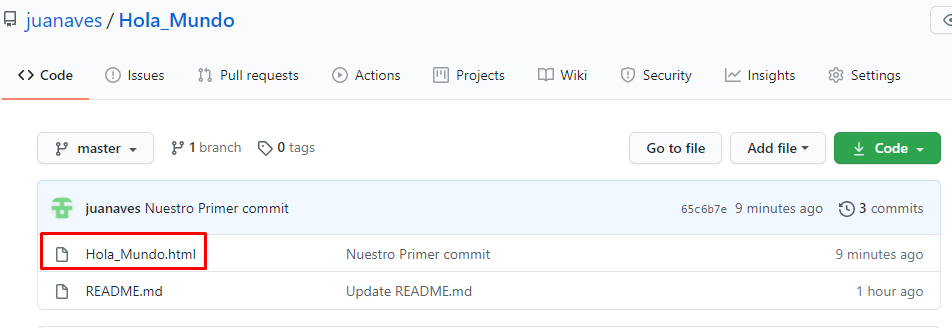
* bien pulsando en la parte derecha de la barra de git (icono tres puntos) y eligiendo la opción push



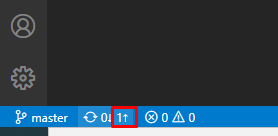
Nos aparecerá una pantalla para loguearnos el github



Y si nos vamos a Github veremos que los cambios en nuestro proyecto se han subido al repositorio.



* La segunda forma de hacer un **push** más fácil es pulsando sobre el botón que está en la parte inferior.

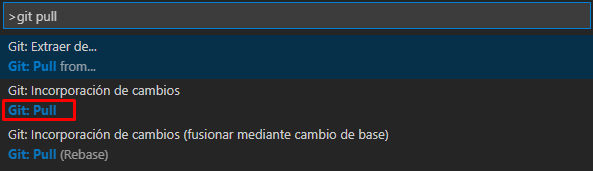


Veréis que los cambios estarán a 0



***Como hacer un pull***

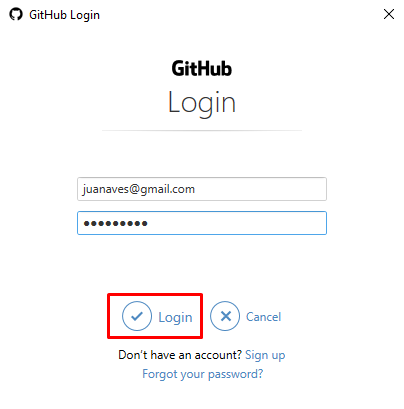
Para traernos los cambios que haya en servidor de la rama en la que estamos trabajando, podemos pulsar sobre los tres puntos (...) que hay en la parte superior derecha y pinchar sobre la opción *pull.*



Pull

 También podríamos hacer como en el paso anterior (pulsando en los iconos de la parte inferior, sincronizar). Está opción la suelo usar mucho, es cómoda.

 La primera vez que sincronicemos, deberemos introducir el usuario y contraseña de **GitHub**.



Una vez realizada la sincronización, el contador deja de mostrarse.

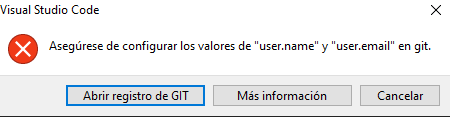


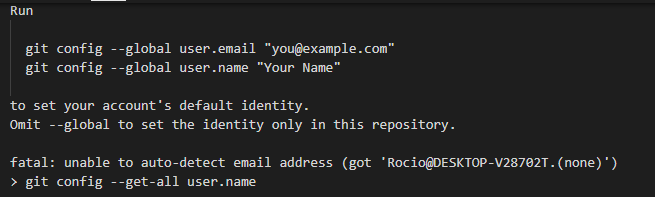
Puede comprobar visitando el repositorio remoto en el navegador que se han realizado los cambios.



Anexo

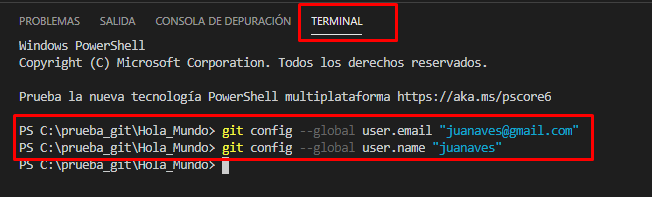
Si alguna vez nos da este mensaje





La salida muestra que Git debe configurarse con configuraciones globales y pocos comandos deben ejecutarse. Para corregir este indicador de CMD abierto, de forma predeterminada, git se instala en C *: \ Archivos de programa \ Git \ bin* . cd a la ubicación y ejecute el siguiente comando.

git config --global user.email "you@example.com"  
git config --global user.name "Su nombre"

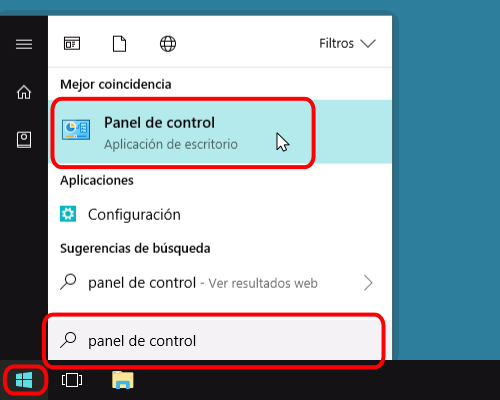


## Borrar credencial de Windows

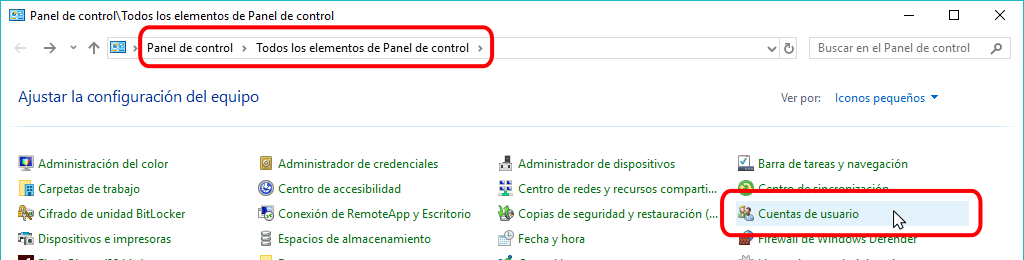
La identificación en GitHub que utiliza Visual Studio Code se guarda como credencial de Windows, por lo que si estamos utilizando un ordenador al que tienen acceso otras personas, es muy importante que eliminemos la credencial antes de abandonar el ordenador o nos arriesgamos a que otro usuario nos suplante y acceda a los repositorios remotos.

Para borrar la credencial:

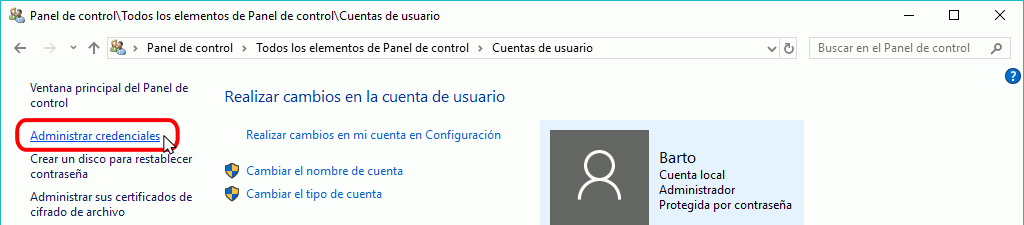
* Abra el panel de control clásico (Menú de inicio > panel de control)



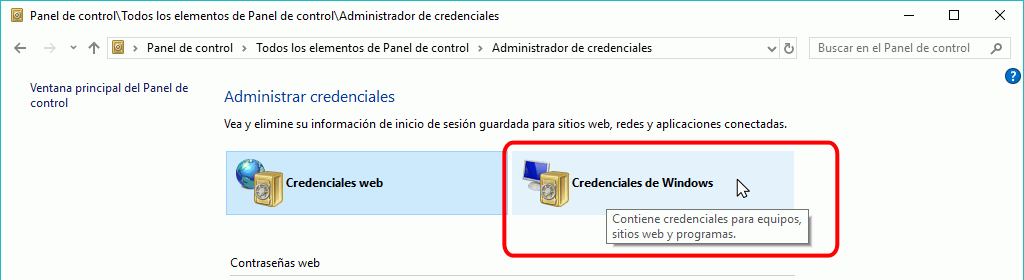
* Haga clic en el icono de Cuentas de usuario



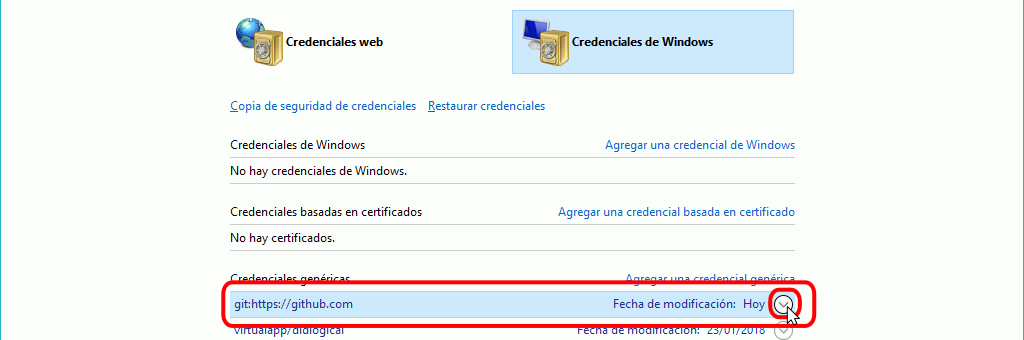
* Haga clic en el enlace "Administrar credenciales"



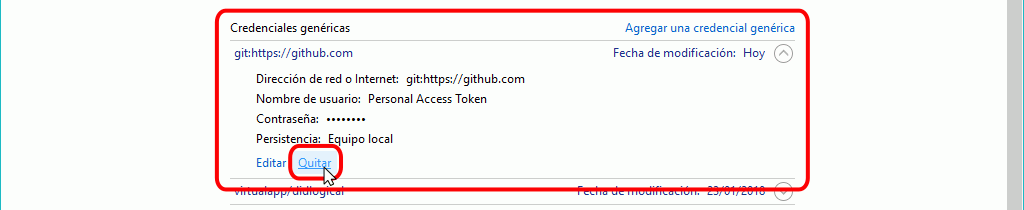
* Haga clic en el icono "Credenciales de Windows"



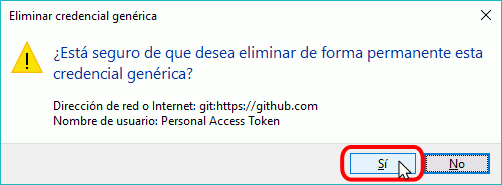
* Haga clic en la credencial genérica de github



* Haga clic en el enlace "Quitar"



* Confirme la eliminación



* Compruebe que la credencial ha sido eliminada

