**Tutorial Integración Continua.**

Git es un sistema de control de versiones creado por Linus Torvals un sistema de control de versiones nos registra los cambios realizados en un proyecto, teniendo las versiones para poder regresar a ciertos puntos en la elaboración del software, en conclusión Git lo que hace es monitorear todo lo que hacemos en nuestro proyecto, una de las ventajas es que podemos movernos en el tiempo y viajar de la versión 3 a la versión 1 etc.

Git tiene 3 estados

* **Working Directory**: aquí es donde editamos y trabajamos con nuestros proyectos
* **Staging** **área**: aquí es donde escogemos que archivos están listos para pasar al tercer estado, al igual que decidimos que archivos no están listos por el momento
* **Repository:** Aquí es el registro de todo nuestro proyecto

*Según flujo de trabajo*

* *Modificar, crear, editar una serie de archivos*
* *Escoger los archivos que estén listos*
* *Registro de nuestro proyecto*

Programas a utilizar:

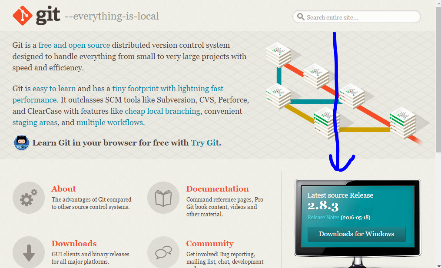
* Windows 10 como sistema operativo principal
* Visual Estudio
* Github
* Servidor en amazon
  + Windows Server 2012
  + Jenkins
  + Fling
* SmarterASP.com

**Visual Studio**

Instalaciòn Bàsica de Visual estudio.

Crear proyecto.

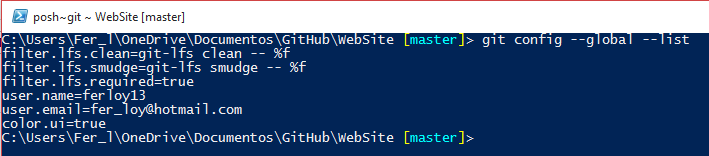
**Instalación de Git**

* <https://git-scm.com/> (para efectos usaremos la versión 2.8.3 64 bits)
* Luego una instalación típica de Microsoft Windows

**Configuración de Git Shell (desde bash)**

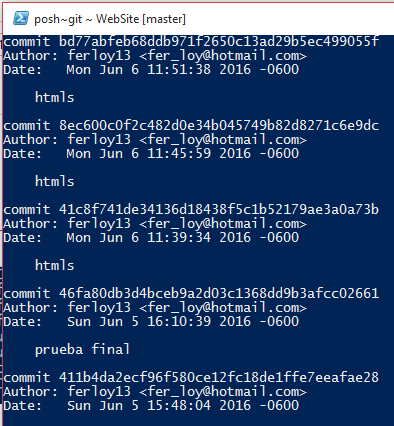
* Cd C:\Users\Fer\_l\OneDrive\Documentos\GitHub\WebSite

(Esto posicionara el cursor git en el lugar de la carpeta donde esta el archivo que vamos a versionar)

* git config --global user.name “ferloy13”
* git config --global user.email “[fer\_loy@hotmail.com](mailto:fer_loy@hotmail.com)”
* git config --global color.ui true
* git config --global --list

**Proyecto:**

* Git init (este comando marca el inicio de nuestro proyecto, aquí le decimos a git que empiece a monitorear todos los cambios
* Git status (este comando nos dirá cuál es el estado de nuestro proyecto)
* Git add nombrearchivo.txt (le decimos que archivos pasan a commit)
* Git add –A (le decimos que agregue todo)
* Git commit –m “mensaje” (Guardamos los cambios con un mensaje para identificarlos
* git log (nos da una lista de todos nuestros commits con su respectiva información)



* git checkout (Con este comando viajamos a través de nuestros commits o nuestras ramas se copia el código del commit y se agrega ejemplo
  + git checkout 8ec600c0f2c482d0e34b045749b82d8271c6e9dc
  + si queremos regresar al commit principal o al ultimo
  + git checkout master
* Git Reset sirve para regresar pero mata los commit subsiguientes
  + **Git reset --soft** Es el git mas reset mas simple y que no toca nuestro “Working Area”(No se mete con nuestro código)
  + **Git reset --mixed** Este git reset borra el “Staging Area”, sin tocar el “Working Area”
  + **Git reset --Hard** Este borra absolutamente todo lo que hay en el commit
* Git log
  + Git log > commits.txt (genera un archivo de commits)
* Git help / git status help \*\*\* toda la ayuda sobre los comandos de git.
* Git help (comando)

**Concepto HEAD**

El head es el commit de donde nos encontramos ahora, (el commit actual) con git log tenemos nuestro último commit es el head actual.

**Ramas**

Es una línea de tiempo en nuestro proyecto que nos sirve para arreglar errores, experimentar, hacer grandes cambios para luego pasar a la rama principal que seria nuestro final, cuando nosotros hacemos git init la rama principal que se crea se llama *master* y es donde comenzamos a trabajar, es la principal y estable de nuestro proyecto

* Git branch (nos muestra las ramas)
* Git branch nombre (crear nueva rama)
* Git checkout nombre (mueve de rama)

**GITHUB**

Git y Github no es lo mismo Github es una plataforma donde nosotros podemos guardar nuestro proyectos usando git para su gestión

<https://github.com/>

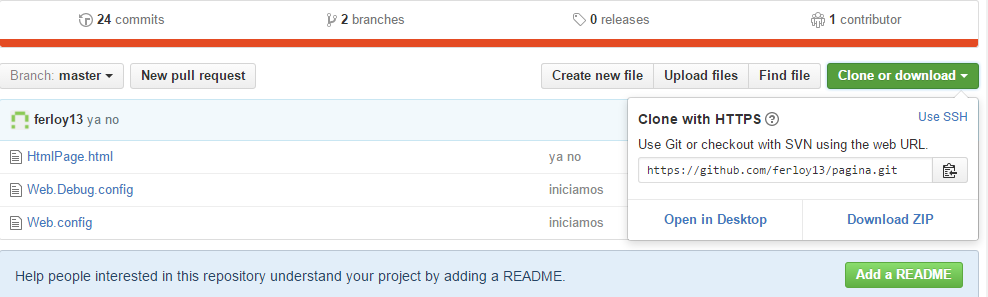
Donde se debe de hacer un registro común

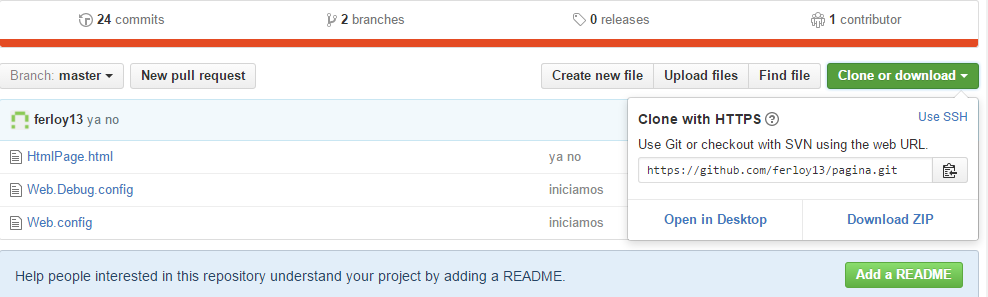
Debemos de crear un nuevo proyecto, y crear la rama principal

Los repositorios remotos son los que están en github y los repositorios locales son los que están en nuestro equipo.

Para subir nuestro proyecto

Usamos

Git remote add origin <https://github.com/ferloy13/pagina.git>



Comprobar

* Git remote –v

Para subir un commit

* Git push origin master

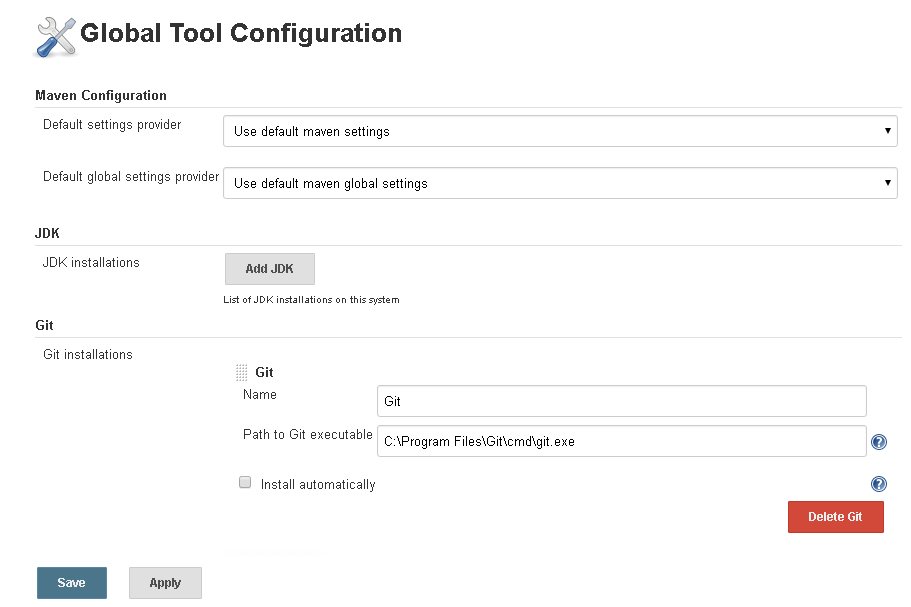
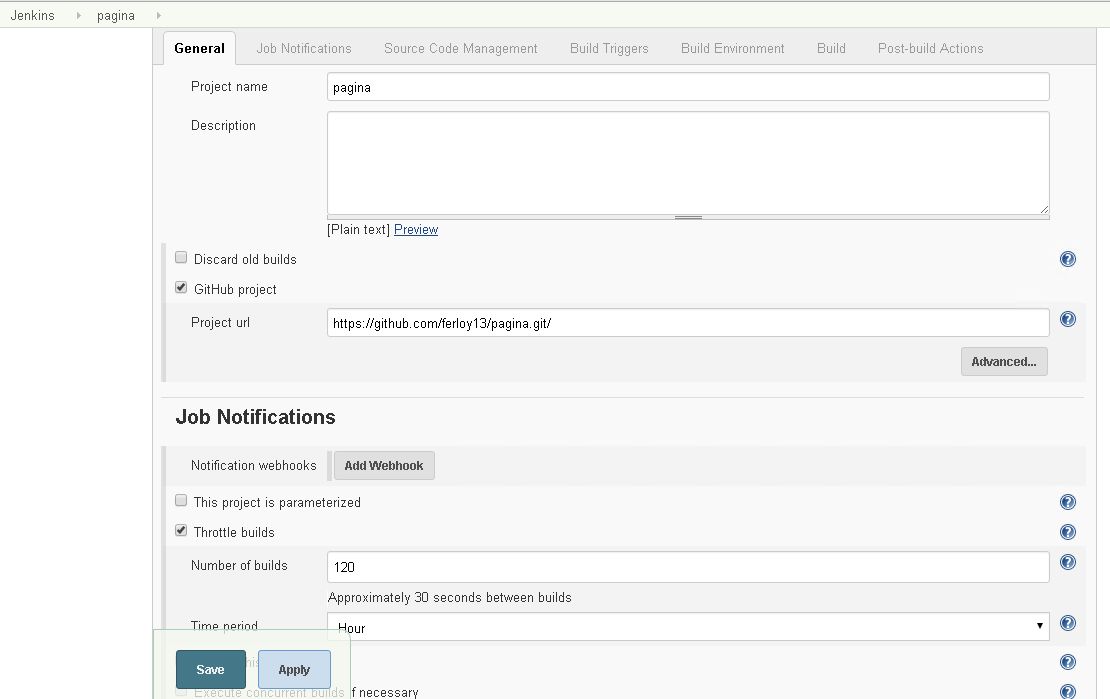
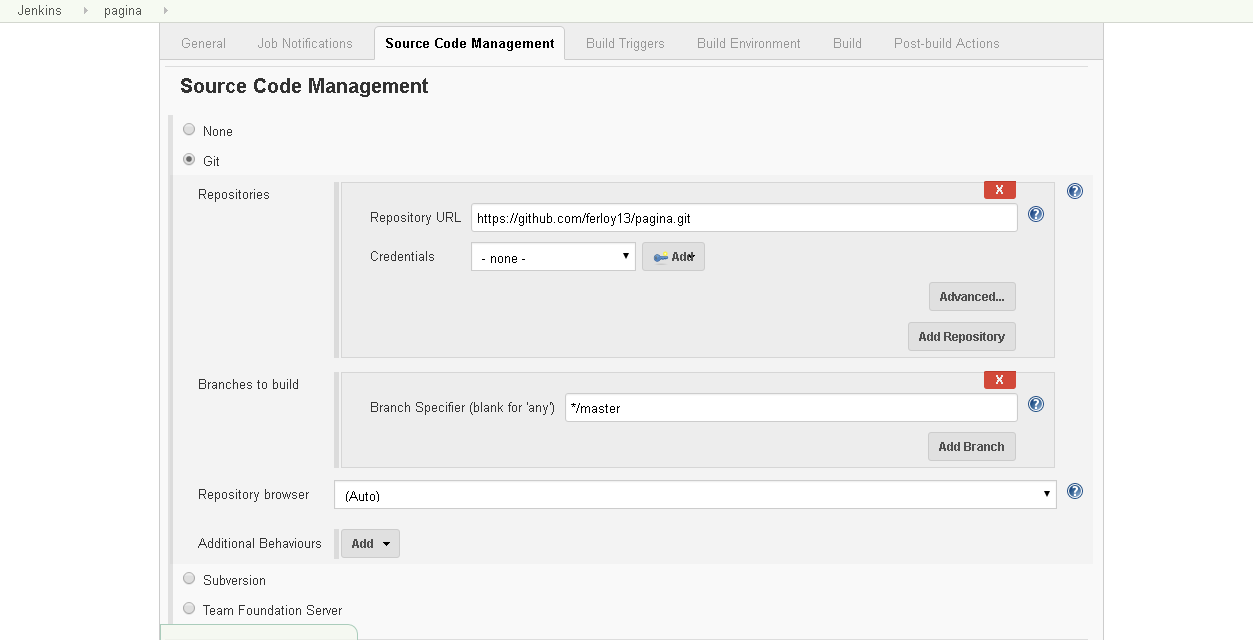
**En Resumen**

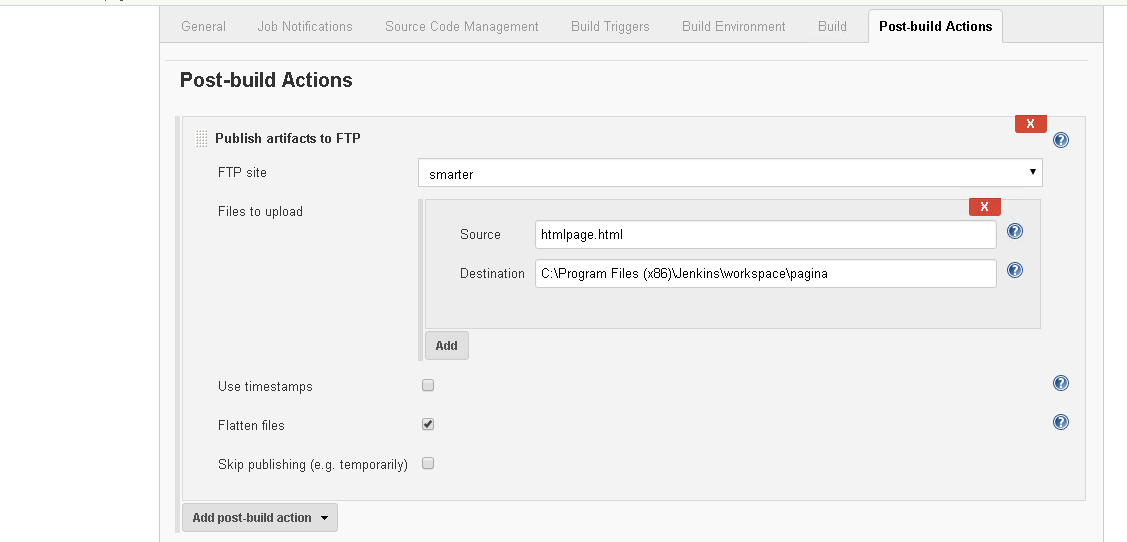
1. Establecerse en la carpeta donde esta el archivo
2. Git add –A (agregar todos los cambios)
3. Git commit –m “descripción” (Agregar el commit)
4. Git push origin master (subir el commit a github).

Jenkins

Instalar Jenkins

Configurar Jenkins

* Instalar plugins
  + Manage Jenkins
    - Manage plugins
    - Git plugins instalar
* 
* Crear nuevo proyecto
  + Freestyle Project
    - Ok
    - Configure
    - 
* 



SAVE

BUILD NOW

CONECTAR FTP