¿Qué es Git, GitHub?. Comandos, repositorio remoto, push

¿Qué es Nodejs?

¿Qué es TypeScript?

¿Qué es Angular?

SPA

Temáticas Angular: CLI, comandos ng, estilos, componentes, directivas.

¿Qué es Git, GitHub?

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando un sistema de versiones Git. El código que se almacena de una forma pública y también se puede hacer de forma privada, pero para esto debe crear una cuenta de paga

Comandos, repositorio remoto, push de Git, Github

## Comandos básicos de GIT

* **git config**  
  Uno de los comandos más usados en git es git config, que puede ser usado para establecer una configuración específica de usuario, como sería el caso del email, un algoritmo preferido para diff, nombre de usuario y tipo de formato, etc… Por ejemplo, el siguiente comando se usa para establecer un email:

git config --global user.email sam@google.com

* **git init**Este comando se usa para crear un nuevo repertorio GIT:

git init

* **git add**  
  Este comando puede ser usado para agregar archivos al index. Por ejemplo, el siguiente comando agrega un nombre de archivo temp.txt en el directorio local del index:

git add temp.txt

* **git clone**  
  Este comando se usa con el propósito de revisar repertorios. Si el repertorio está en un servidor remoto se tiene que usar el siguiente comando:

git clone alex@93.188.160.58:/path/to/repository

Pero si estás por crear una copia local funcional del repertorio, usa el comando:

git clone /path/to/repository

* **git commit**  
  El comando commit es usado para cambiar a la cabecera. Ten en cuenta que cualquier cambio comprometido no afectara al repertorio remoto. Usa el comando:
* git commit –m “Message to go with the commit here”
* **git status**  
  Este comando muestra la lista de los archivos que se han cambiado junto con los archivos que están por ser añadidos o comprometidos.

git status

* **git push**  
  Este es uno de los comandos más básicos. Un simple push envía los cambios que se han hecho en la rama principal de los repertorios remotos que están asociados con el directorio que está trabajando. Por ejemplo:

git push  origin master

* **git checkout**  
  El comando checkout se puede usar para crear ramas o cambiar entre ellas. Por ejemplo, el siguiente comando crea una nueva y se cambia a ella:

command git checkout -b <banch-name>

Para cambiar de una rama a otra solo usa:

git checkout <branch-name>

* **git remote**El comando git se usa para conectar a un repositorio remoto. El siguiente comando muestra los repositorios remotos que están configurados actualmente:

git remote -v

Este comando te permite conectar al usuario con el repositorio local a un servidor remoto:

git remote add origin <93.188.160.58>

* **git branch**  
  Este comando se usa para listar, crear o borrar ramas. Para listar todas las ramas se usa:
* git branch
* para borrar la rama:

git branch -d <branch-name>

* **git pull**  
  Para poder fusionar todos los cambios que se han hecho en el repositorio local trabajando, el comando que se usa es:

git pull

* **git merge**  
  Este comando se usa para fusionar una rama con otra rama activa:

git merge <branch-name>

* **git diff**Este comando se usa para hacer una lista de conflictos. Para poder ver conflictos con el archivo base usa:

git diff --base <file-name>

El siguiente comando se usa para ver los conflictos que hay entre ramas que están por ser fusionadas para poder fusionarlas sin problemas:

git diff <source-branch> <target-branch>

Para solo ver una lista de todos los conflictos presentes usa:

git diff

* **git tag**  
  Etiquetar se usa para marcar commits específicos con asas simples. Por ejemplo:

git tag 1.1.0 <instert-commitID-here>

* **git log**  
  Ejecutar este comando muestra una lista de commits en una rama junto con todos los detalles. Por ejemplo:

commit 15f4b6c44b3c8344caasdac9e4be13246e21sadw

Author: Alex Hunter <alexh@gmail.com>

Date:   Mon Oct 1 12:56:29 2016 -0600

* **git reset**  
  Para resetear el índex y el directorio que está trabajando al último estado comprometido se usa este comando:

git reset - -hard HEAD

* git rm  
  Este comando se puede usar para remover archivos del index y del directorio que está trabajando:

git rm filename.txt

* **git stash**  
  Este es uno de los comandos menos conocidos, pero ayuda a salvar cambios que no están por ser comprometidos inmediatamente, pero temporalmente:

git stash

* **git show**  
  Se usa para mostrar información sobre cualquier objeto git. Por ejemplo:

git show

* **git fetch**  
  Este comando le permite al usuario buscar todos los objetos de un repositorio remoto que actualmente no reside en el directorio local que está trabajando. Por ejemplo:

git fetch origin

* **git ls-tree**  
  Para ver un objeto de árbol junto con el nombre y modo de cada uno de ellos, y el valor blob´s SHA-1, se usa:

git ls-tree HEAD

* **git cat-file**  
  Usando el valor SHA-1, se puede ver el tipo de objeto usando este comando. Por ejemplo:

git cat-file –p d670460b4b4aece5915caf5c68d12f560a9fe3e4

* **git grep**  
  Este comando le permite al usuario buscar en los árboles de contenido cualquier frase o palabra. Por ejemplo, para buscar por www.tupaginaweb.com en todos los archivos se usaría:

git grep “www.tupaginaweb.com”

* **gitk**  
  Este es la interfaz gráfica para un repositorio local que puede invocar escribiendo y ejecutando:

gitk

* **git instaweb**  
  Con este comando un servidor web puede correr interconectado con el repositorio local. Un navegador web también está automáticamente dirigido a el:

git instaweb –http=webrick

* **git gc**  
  Para optimizar el repositorio por medio de una recolección de basura, que limpiara archivos innecesarios y los optimizara, usa:git hc
* **git archive**  
  Este comando le permite al usuario crear archivos Zip o Tar que contengan los constituyentes de un solo árbol de repositorio:git archive – -format=tar master
* **git prune**  
  Con este comando los objetos que no tengan ningún puntero entrante serán eliminados:git prune
* **git fsck**  
  Para poder hacer un chequeo de integridad del sistema de archivos git, usa este comando. Cualquier objeto corrompido será detectado:git fsck
* **git rebase**  
  Este comando se usa para la re aplicación de los compromisos en otra rama. Por ejemplo: git rebase master

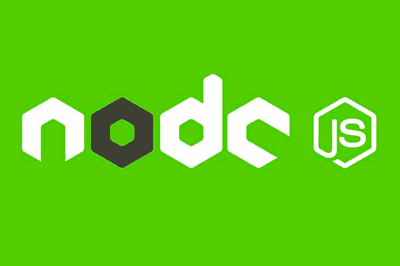


¿Qué es Nodejs?

Node.js es un entorno tiempo de ejecución multiplataforma de código abierto para la capa del servidor, pero non limitándose a ello. Basándose en código ECMAscript fue creado [C++](https://es.wikipedia.org/wiki/C%2B%2B) y [JavaScript](https://es.wikipedia.org/wiki/JavaScript).

Fue creado por el enfoque de ser de útil de una creación Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, [servidores web](https://es.wikipedia.org/wiki/Servidor_web).

Node.js es similar en su propósito a [Twisted](https://es.wikipedia.org/wiki/Twisted_(software)) o [Tornado](https://es.wikipedia.org/w/index.php?title=Tornado_(servidor_web)&action=edit&redlink=1) de [Python](https://es.wikipedia.org/wiki/Python), [Perl Object Environment](https://es.wikipedia.org/w/index.php?title=Perl_Object_Environment&action=edit&redlink=1) de [Perl](https://es.wikipedia.org/wiki/Perl), [libevent](https://es.wikipedia.org/w/index.php?title=Libevent&action=edit&redlink=1) o [libev](https://es.wikipedia.org/w/index.php?title=Libev&action=edit&redlink=1) de [C](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n)), [EventMachine](https://es.wikipedia.org/w/index.php?title=EventMachine&action=edit&redlink=1) de [Ruby](https://es.wikipedia.org/wiki/Ruby), [vibe.d](https://es.wikipedia.org/w/index.php?title=Vibe.d&action=edit&redlink=1) de [D](https://es.wikipedia.org/wiki/D_(lenguaje_de_programaci%C3%B3n)) y [Java EE](https://es.wikipedia.org/wiki/Java_EE).



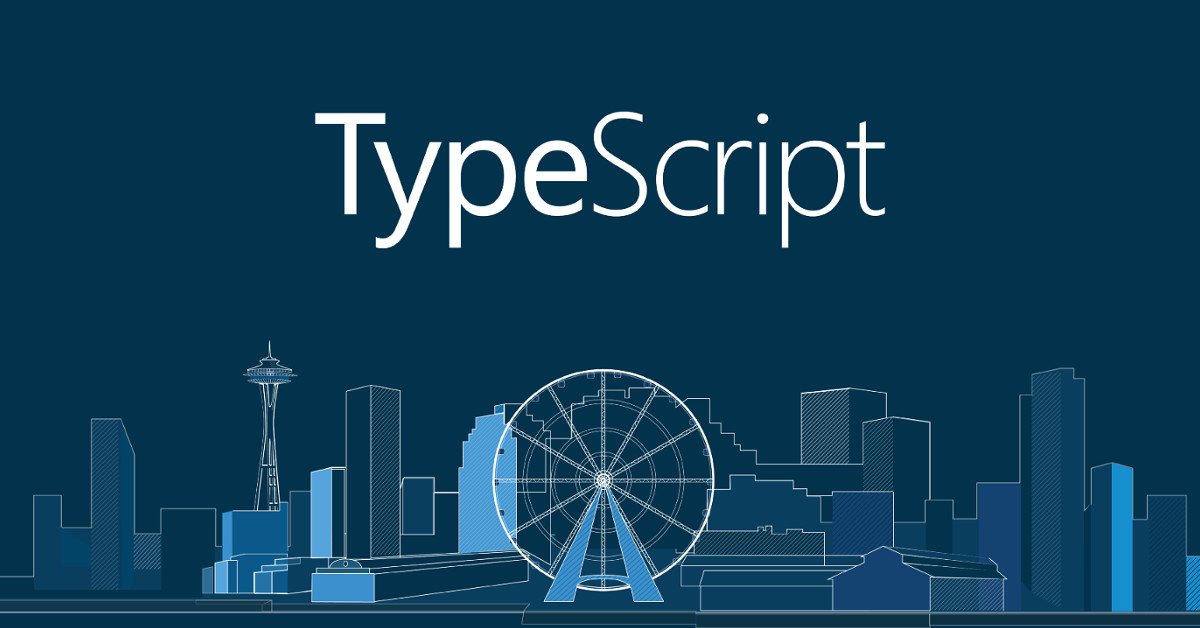
¿Qué es TypeScript?

TypeScript es un lenguaje de programación libre y de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto) desarrollado y mantenido por [Microsoft](https://es.wikipedia.org/wiki/Microsoft). Es un súper conjunto de [JavaScript](https://es.wikipedia.org/wiki/JavaScript), que esencialmente añade tipos estáticos y objetos basados en clases.

​ TypeScript puede ser usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor ([Node.js](https://es.wikipedia.org/wiki/Node.js)).

TypeScript extiende la sintaxis de JavaScript por tanto, cualquier código JavaScript existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen a código JavaScript.

TypeScript soporta ficheros de definición que contengan información sobre los tipos de librerías JavaScript. Esta permite a otros programas usar los valores definidos en los ficheros como si fueran entidades TypeScript de tipado estático.



¿Que es spa?

SPA significa single page aplication es un tipo de aplicación web donde todas las pantallas la muestra en una sola pestaña, sin recargar el navegador.

SPA es un sitio donde existe un único punto de entrada, generada como un archivo index.html. En la aplicación no hay otro archivo HTML que se podada acceder de forma separada. Que muestre contenido o parte de la aplicación todo este proceso se hace dentro del archivo llamado index.html.

SPA ofrece una experiencia que es tan agradable con el usuario al pesar muy poco los datos, mucho menos si estuvieran dentro de un complejo de código HTML y CSS.

LENGUAJES Y TECNOLOGIAS PARA PRODUCIR UNA SPA.

* AngularJS
* Angular 2
* React
* Polymer
* EmberJS