
Desarrollo del lado servidor con NodeJS, Express y MongoDB

Comenzando un proyecto con GIT

En esta lección veremos las herramientas necesarias para comenzar un proyecto. Empezaremos hablando de los sistemas de control de versiones (VCS) que nos permiten ir creando versiones a medida que avanzamos con nuestro proyecto. De esta manera, podremos volver a una versión anterior con total facilidad, si deseamos probar algún cambio sin afectar las versiones previas. Explicaremos conceptos teóricos de la herramienta y, luego, pasaremos a un tutorial paso a paso para que puedas crear tu propio ambiente desde cero.

¿Qué es un VCS?

VCS es un acrónimo para Version Control System o Sistema de Control de Versiones. Nos permite guardar registros de los cambios realizados a un archivo o a un sistema de archivos en un lapso de tiempo determinado. Así, podemos recurrir a versiones específicas en cualquier momento. Es importante destacar que este sistema puede ser utilizado para prácticamente cualquier tipo de archivo. Imagínese que usted es un diseñador gráfico y que quiere guardar todas las versiones del diseño que está haciendo. Con un VCS, usted podría revertir determinados archivos a un estado anterior, comparar versiones de archivos, ver quién hizo la última modificación a un archivo específico, entre otras cosas.

El VCS que utilizaremos en este curso es GIT.

¿Por qué usar GIT?

GIT es popularmente aceptado en la industria y, en los últimos años, se ha vuelto un estándar para el versionado de cualquier proyecto. Es muy simple de usar y admite una

gran cantidad de operaciones avanzadas para gestionar las versiones, logrando que, tanto usuarios principiantes como avanzados, puedan trabajar correctamente.

GIT fue creado por Linus Torvalds, el creador de Linux. No es parte de este curso profundizar sobre los VCS ni sobre GIT en cuestión, pero te daremos los comandos básicos que te permitan resguardar tu proyecto y compartirlo con quien quieras.

Flujo de trabajo en GIT

Lo primero que hacemos es configurar nuestro repositorio de GIT en nuestro directorio de proyecto. Con esta acción, todos los archivos que se encuentren dentro del directorio del proyecto pasarán a versionarse bajo GIT.

Cada archivo puede estar en 3 estados:

- **Modified:** se han realizado cambios en el archivo.
- **Staged:** el archivo va a pasar a formar parte de una nueva versión. Esto significa que podemos tener archivos que estén modificados, pero que no queramos que formen parte de la siguiente versión. Para distinguirlos dentro de los archivos modificados, lo que hacemos es pasarlos al estado de “staged”.
- **Committed:** el archivo se encuentra en el repositorio local de GIT.

Estos 3 estados de archivos nos dan lugar a hablar de 3 áreas:

- **Espacio de trabajo o *working directory*:** hace referencia al directorio raíz del proyecto que estará relacionado con el repositorio de GIT.
GIT va a monitorear todos los cambios que se encuentren dentro del espacio de trabajo.
- **Staging:** es el conjunto de archivos modificados y seleccionados para una próxima versión o commit.
- **Repositorio:** son todos los archivos ya versionados y contiene todo el historial de cambios.

Para pasar al repositorio un archivo, primero debe modificarse, luego pasarse a Staging y finalmente, realizar un commit que pase todos los archivos que están en Staging al repositorio.

De esta manera GIT guarda cada commit asociando su autor, la fecha y hora, un código autogenerated que identifica unívocamente al commit y, finalmente, los archivos en cuestión. En cualquier momento uno puede ver el historial de cambios y de commits.

Con estos conceptos básicos realizaremos un tutorial paso a paso para configurar tu proyecto. Puedes profundizar los conceptos siguiendo las lecturas del material complementario.

No copiar ni publicar