

## QUE SON LOS ARCHIVOS GITIGNORE

Un archivo .gitignore es un archivo de texto colocado en su repositorio de git que le dice a git que no realice un seguimiento de ciertos archivos y carpetas que no desea que se carguen en su repositorio principal. Tiene muchos usos y casi siempre tendrá que configurarlo si está configurando un nuevo repositorio.

### ¿CUAL ES EL PUNTO DE GITIGNORE?

En la mayoría de los proyectos, existe una diferencia entre code / config y *archivos generados* de ese código o configuración. Este último generalmente no es útil y generalmente se puede ignorar.

Por lo general, solo desea que el código y la configuración se rastreen a través de git. Esto se debe a que los archivos generados son efímeros y, si los elimina, simplemente puede volver a generarlos. A menudo, no tiene sentido acusarlos, ya que simplemente complicarán las cosas y causarán conflictos sindicales innecesarios.

### git add git commit git diff git stash .gitignore

Git ve cada archivo de tu copia de trabajo de una de las siguientes maneras:

1. Con seguimiento: Un archivo que se ha preparado o confirmado previamente.
2. Sin seguimiento: Un archivo que *no* se ha preparado o confirmado.
3. Ignorado: Un archivo que se le ha indicado explícitamente a Git que ignore.

Los archivos ignorados suelen ser artefactos de compilación y archivos generados por el equipo que pueden derivarse de tu fuente de repositorios o que no deberían confirmarse por algún otro motivo. Estos son algunos ejemplos habituales:

- Cachés de dependencias, como es el caso del contenido de `/node_modules` o `/packages`.
- Código compilado como, por ejemplo, los archivos `.o`, `.pyc` y `.class`.

- Directorios de salida de compilación, como es el caso de `/bin`, `/out` o `/target`.
- Archivos generados en tiempo de ejecución como, por ejemplo, `.log`, `.lock` o `.tmp`.
- Archivos ocultos del sistema, como es el caso de `.DS_Store` o `Thumbs.db`.
- Archivos personales de configuración de IDE como, por ejemplo, `.idea/workspace.xml`.

A los archivos ignorados se les hace un seguimiento en un archivo especial llamado `.gitignore` que se incorpora en la raíz de tu repositorio. En Git no hay ningún comando explícito para ignorar archivos: en su lugar, cuando tengas nuevos archivos que quieras ignorar, deberás editar y confirmar manualmente el archivo `.gitignore`. Los archivos `.gitignore` contienen patrones que establecen coincidencias con los nombres de archivo de tu repositorio para determinar si deberían ignorarse o no.

### **EJEMPLOS DE GITIGNORE:**

La notación es muy simple. Por ejemplo, si indicamos la línea

```
bower_components/
```

Estamos evitando que se procese en el control de versiones todo el contenido de la carpeta "bower\_components".

Si colocamos la siguiente línea:

```
*.DS_Store
```

Estaremos evitando que el sistema de control de versiones procese todos los archivos acabados de `.DS_Store`, que son ficheros de esos que crea el sistema operativo del Mac (OS X) automáticamente.

Hay muchos tipos de patrones aplicables a la hora de especificar grupos de ficheros, con comodines diversos, que puedes usar para poder indicar, de manera muy específica, lo que quieres que Git no procese al realizar el control de versiones. Puedes encontrar más información en la [documentación de gitignore](#), pero generalmente no lo necesitarás porque lo más cómodo es crear

el código de este archivo por medio de unas plantillas que ahora te explicaremos.