



# .gitignore

## 1. Introducción.

El archivo `.gitignore` es esencial en Git para especificar patrones de archivos o directorios que deben ser ignorados al realizar operaciones como `"git add"` o `"git commit"`, evitando así la inclusión de archivos no deseados y mejorando la calidad y consistencia del historial de versiones. Su uso facilita la colaboración al asegurar que todos los colaboradores del proyecto compartan un entorno de desarrollo más consistente y libre de archivos innecesarios.

Su utilidad principal radica en mantener el repositorio limpio y enfocado, excluyendo archivos prescindibles como configuraciones locales, archivos temporales y dependencias de proyectos.

Además, desempeña un papel clave al abordar limitaciones de espacio en repositorios remotos, ya que, al evitar la inclusión de archivos superfluos, como dependencias de terceros o archivos generados, se reduce significativamente el tamaño del repositorio. Esto mejora la eficiencia en la transferencia de datos entre repositorios locales y remotos, facilitando la colaboración y la gestión de proyectos distribuidos.

Esta última parte es importante porque como veremos más adelante, en algún momento nuestro repositorio deberá ser replicado en un servidor remoto, tal como GitLab, y debemos tener en cuenta que dichas plataformas de alojamiento no brindan capacidad de almacenamiento ilimitada. En el caso específico de GitLab, la asignación de espacio para cada repositorio se encuentra limitada a 10GB.

La sintaxis del archivo `.gitignore` permite definir reglas para excluir ciertos archivos o patrones de archivos, utilizando comodines y convenciones específicas. Por ejemplo, `*.log` excluye todos los archivos con extensión `".log"`, y `/logs/` excluye un directorio llamado `"logs"`. Estas reglas proporcionan flexibilidad para adaptarse a las necesidades específicas de cada proyecto. En la siguiente tabla se recogen algunos ejemplos:



### Patrones de git.

Patrón	Coincidencias de ejemplo	Descripción
*.log		Excluye todos los archivos con extensión ".log"
/logs/		Ignora el directorio "logs" y todo su contenido.
**/logs	logs/debug.log, logs/monday/foo.bar, build/logs/debug.log	Ignora todos los directorios llamados "logs", sin importar en qué nivel de la estructura del proyecto se encuentren.
**/logs/debug.log	logs/debug.log, pero no: logs/build/debug.log	Ignorar específicamente el archivo llamado "debug.log" que se encuentre en cualquier subdirectorio llamado "logs",
/debug.log	debug.log	Anteponiéndolo con una barra le indicaremos sólo el fichero que cuelgue de la raíz del proyecto.
debug.log	debug.log, log/debug.log	Indicando solamente el nombre del archivo entonces el patrón coincidirá con los archivos que tengan ese nombre en el directorio raíz del repositorio.
*~	documento.txt~	Ignora archivos que terminan con "~", típicamente utilizados como copias de respaldo temporales.
*.exe	programa.exe	Ignora archivos ejecutables de Windows.
.vscode/		Ignora el directorio de configuración de Visual Studio Code y sus archivos.



## 2. Metacaracteres.

En el contexto de Git y los archivos de configuración, los metacaracteres son caracteres especiales que no representan su valor literal, sino que se utilizan para definir patrones. Estos patrones se emplean comúnmente en archivos como `.gitignore` para especificar qué archivos o directorios se deben ignorar.

Aquí tienes una tabla con algunos ejemplos de metacaracteres, su descripción y ejemplos de uso:

Metacaracter	Descripción	Ejemplo
*	Coincide con cualquier secuencia de caracteres	*.log
?	Coincide con un solo carácter	temp?.txt
[ ]	Coincide con un solo carácter dentro del rango	[aeiou]
-	Rango de valores a comparar (Va acompañado del [ ])	[1-9][1-3] → Cadenas coincidentes: 32, 12, 82 y no coincidentes: 99
**	Coincide con cero o más directorios	logs/**/debug
{ }	Coincide con cualquiera de los patrones separados por coma	{*.jpg,*.png}
!	Niega el patrón, seleccionando todo excepto lo especificado	!important.txt
^	Coincide con el inicio de una línea cuando está al principio de la expresión. Cuando está entre paréntesis y seguido de caracteres, niega los caracteres que siguen.	^[xyz] → Empiezan por alguna de esas letras. [^xyz] → No contiene ninguna de esas letras.
\$	Coincide con el final de una línea	docs\$/



Colección de plantillas de `.gitignore` útiles.  
<https://github.com/github/gitignore>