

.gitignore

En muchas ocasiones habrá archivos en nuestro proyecto que no queremos que Git rastree.

Por ejemplo:

1. Archivos que contienen información sensible, como conexiones a base de datos (Nombre de Usuario/Contraseña).
2. Archivos temporales creados por editores.
3. Archivos compilados.

Beneficios obtenidas al ignorarlos:

Para el primer caso, reducimos los riesgos de filtrado de información sensible.

Para el segundo y tercer caso, reducimos el tamaño del repositorio.

# Utilizando .gitignore

Si se crea un archivo en el repositorio denominado .gitignore (Regularmente en la raíz), Git lo utilizará para determinar qué archivos y directorios debe ignorar, antes de hacer confirmaciones (commits).



Recuerda agregar el archivo .gitignore en alguna confirmación para compartir las normas de omisión con cualquier otro compañero que esté trabajando en el mismo proyecto.

Nombre



css



recursos



.gitignore



index.html

# Observación

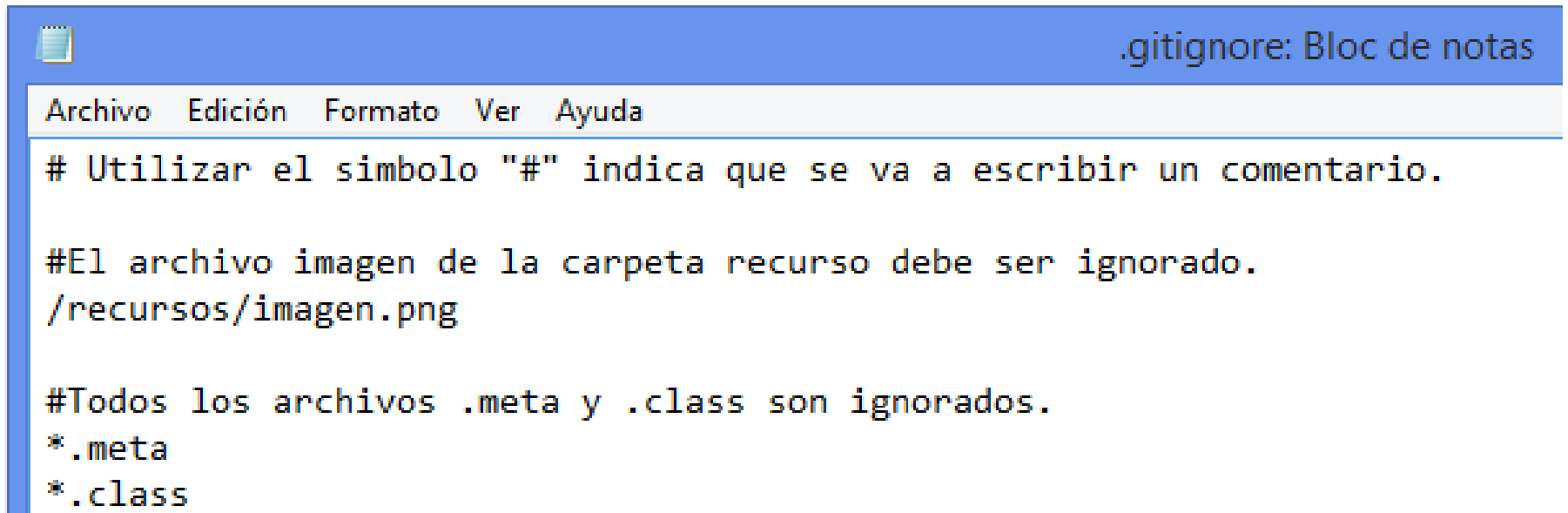
Para sistemas operativos como Mac OS o Linux, hará **invisible** archivos que no cuentan con nombre y solamente extensión.

**Se debe activar la opción de ver archivos ocultos.**



En Windows este problema no ocurre

# Ejemplo de uso y llenado



```
.gitignore: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
# Utilizar el simbolo "#" indica que se va a escribir un comentario.
#El archivo imagen de la carpeta recurso debe ser ignorado.
/recursos/imagen.png
#Todos los archivos .meta y .class son ignorados.
*.meta
*.class
```

# Ejemplos para indicar qué archivos ignorar

<b>*.log</b>	debug.log foo.log .log logs/debug.log	Un asterisco es un comodín que coincide con cero o más caracteres.
<b>/debug.log</b>	debug.log pero no logs/debug.log	Anteponer una barra inclinada coincide con los archivos solo en la raíz del repositorio.
<b>debug.log</b>	debug.log logs/debug.log	Por defecto, los patrones coinciden con los archivos en cualquier directorio

# Ejemplos para indicar qué archivos ignorar

**logs**

**logs**

**logs/debug.log**

**logs/latest/foo.bar**

**build/logs**

**build/logs/debug.log**

Si no agrega una barra, el patrón coincidirá con los archivos y el contenido de los directorios con ese nombre. En las coincidencias de ejemplo a la izquierda, se ignoran tanto los directorios como los archivos denominados registros.

**logs/**

**logs/debug.log**

**logs/latest/foo.bar**

**build/logs/foo.bar**

**build/logs/latest/debug.log**

Agregar una barra indica que el patrón es un directorio. Se ignorará todo el contenido de cualquier directorio en el repositorio que coincida con ese nombre, incluidos todos sus archivos y subdirectorios.

**logs/\*\*/debug.log**

**logs/debug.log**

**logs/monday/debug.log**

**logs/monday/pm/debug.log**

Un doble asterisco coincide con cero o más directorios.

# Ejemplos para indicar qué archivos ignorar

<b>/logs/debug.log</b>	logs/debug.log pero no debug.log build/logs/debug.log	Los patrones que especifican un archivo en un directorio particular son relativos a la raíz del repositorio. (Puede anteponer una barra diagonal si lo desea, pero no hace nada especial).
<b>/logs/*.log</b>  <b>!/logs/debug.log</b>	/logs/debug.log	La primera sentencia indica que cualquier archivo con extensión .log en la carpeta logs, deberá ser ignorado. La segunda sentencia con el símbolo “!” niega un archivo en específico para que vuelva a ser rastreado.



# Creando archivo

Si trabajas en Windows, posiblemente habrá problemas para crear un archivo sin nombre, por lo que se recomienda crear el archivo desde la consola de comandos.

Ejemplo de instrucción:

```
echo #Iniciando > .gitignore
```