



PATRONES GLOB

En la programación de computadoras , en particular, en Unix el término Glob se utiliza a veces para referirse a la coincidencia de patrones basados en caracteres comodín . El sustantivo Glob se usa para referirse a un patrón en particular, por ejemplo, al usar el patrón Glob para los archivos con extensión .log para que coincidan todos los archivos de tipo log. Es comúnmente usado porque su notación es más simple que las expresiones regulares y sin fuerza expresiva. Glob es una abreviatura de “Comando Global” para hacer referencia a que encierra y suministra todos los archivos del tipo que se quiera especificar.

GITIGNORE

En cualquier proyecto, existen ficheros que no deben incluirse en el repositorio. Por ejemplo:

- Ficheros de log
- Cachés
- Ficheros que contienen claves privadas (App keys y App secrets) para acceder a las API o servicios web
- Ficheros de configuración con contraseñas de bases de datos
- Ficheros binarios compilados
- Ficheros de configuración del entorno de desarrollo

Cuando usamos Git la manera más sencilla de evitar subir estos y otros ficheros no deseados al repositorio es crear un fichero .gitignore que hay que subir al repositorio o la otra manera es crear un fichero .git/info/exclude para poner las reglas locales. Por orden de prioridad, Git busca patrones para excluir ficheros en los siguientes lugares:

1. En la línea de comandos: ciertos comandos, como por ejemplo git ls-files o git read-tree, permiten excluir ficheros a través de argumentos. Estos patrones tendrán la máxima prioridad.
2. En los ficheros .gitignore
3. En el fichero \$GIT_DIR/info/exclude
4. En el fichero definido en la variable de configuración core.excludesfile

Todos los ficheros mencionados arriba incluyen en su interior un conjunto de patrones de ficheros que se van a ignorar. Cada patrón va en una línea. Por ejemplo, si incluimos la siguiente línea en alguno de estos ficheros:

*.jar (Git ignorará cualquier fichero con extensión .jar en cualquier directorio).



PLANTILLAS GITIGNORE

1. *.js - *.js.deps - *.js.map (Ignorar archivos y directorios creados por pub and dart2js) – arc.js
2. *.dart.js (Evita commit en archivos javascript generados) – por.dart.js
3. bg-?.jpg (Ignora los ficheros cuyos nombres empiezan por “bg-”, van seguidos exactamente de dos caracteres, y luego van seguidos de “.jpg”) – bg01.jpg
4. bg-*.jpg (Ignora todos los ficheros que empiecen por "bg-" y acaben en ".jpg") – bg-1.jpg
5. web/ (Ignorar el contenido de la carpeta web) – web/jaja.html
6. !*.php (Incluir expresamente en el repositorio todos los ficheros con extensión php) – !lok.php
7. *. [xn]ib (Ignora los ficheros .nib y .xib) – hola.nib y hola.xib
8. web/**/* .html (Ignora todos los ficheros .html que están en las subcarpetas de web pero no ignorar los ficheros .html que están en web) – web/prueba1/jaja.html
9. *.s#? (Ignora archivos de respaldo o backups) - back.s#
10. *.elc (Ignora archivos compilados) - com.elc
11. *.class (Ignora archivo tipo .class) – cla.class
12. *Db.properties (Ignora archivos generales de bases de datos HSQL) – jojo.Db.properties
13. !Build/ (Incluir expresamente los ficheros contenidos en la carpeta Build) – !Build/const.txt
14. .dox\$ (Ignorar documentos generados) – doc.dox\$
15. CATKIN_IGNORE (Ignorar archivos personalizados catkin)
16. *.LST (Ignorar archivos de lista) – List.LST
17. *.GDS (Excluir formatos de producción) – Product.GDS
18. *.HPGL (Excluir formatos de intercambio)- Exchange.HPGL
19. *.dll (Excluir bibliotecas dinámicas compiladas en C++) – c++.dll
20. tmp/**/* (Ignorar todos los subdirectorios y archivos en un directorio) – web/taller1/hola.txt
21. bin-debug/ (Ignora carpetas de construcción) – bin-debug/dbug.md
22. *.o (Excluye archivos de objeto) – object.o
23. *.ali (Ignora información de la librería Ada) – cosmos.ali
24. [Ll]ibrary/ (Ignora archivos dentro de la carpeta) – Library/ y library/
25. *.py[co] (Ignora ficheros que empiezan en .py y termina en c u o) – hola.pyc o hola.pyo
26. *. [ao] (Ignora los ficheros *.o y *.a) – fichero.a y fichero.o
27. !web/* .html (Ignorar todos los ficheros .html excepto en la carpeta web) – !web/hola.html
28. abc/** (Compara todos los archivos dentro del directorio “abc”) – archivo.txt
29. a/**/b (Compara todos los archivos del directorio a y b) – a/toro.txt y b/toro.txt



- 30. /*.c (Una barra inicial coincide con el comienzo de la ruta) - cat-file.c
- 31. /** (Compara todo los archivos de la carpeta abc) - abc/**
- 32. **/foo (Compara archivos o directorio “foo”) – foo/a.foo
- 33. foo/bar/[^_]*.txt (Ignora los archivos .txt pero no los _.txt) – hola.txt y hola_.txt
- 34. lib/model/om/* (Ignora todo lo que hay en el directorio) – lib/model/om/perro.html
- 35. *.html (Ignora todo los archivos htlm) – index.html
- 36. !foo.html (Ignora todos los html menos foo) - foo.html
- 37. *.DS_Store (Ignorar archivos basura en C#) – bage.DS_Store
- 38. [Tt]umbs.db (Excluye abos tipos de ficheros thumbs.db) – Thumbs.db y thumbs.db
- 39. ipch/(Ignora ficheros en la carpeta ipch en visualstudio) – ipch/jaja.ms
- 40. .svn (Excluye ficheros de subversión) – antes.svn
- 41. ~\$* (Ignora ficheros temporales de office) - ~\$per.docx
- 42. *.resharper (Ignora ficheros de herramientas visualstudio) – tool.resharper
- 43. [Dd]ebug*/ (Ignora ficheros en ambas rutas) – Debug/ y debug/
- 44. *.vspssc (Ignora archivos de visualstudio) – stu.vspssc
- 45. Ankh.NoLoad (Excluye archivos que coincidan en visualstudio) –
studio/Ankh.NoLoad