

Patrones Glob

Pulido, Andres
andrespc@hotmail.com

Fundacion Universitaria San Martin

I. TODOS LOS PATRONES QUE USA GIT

Hagamos un pequeño repaso de lo que ya sabemos:

- Las líneas en blanco no se consideran un patrón. Se usan para hacer los ficheros más legibles
- Si la línea comienza con #, se considera un comentario
- El prefijo ! sirve para negar un patrón
- Si el patrón termina en / sólo se aplicará a carpetas, es decir, un patrón “web/” ignorará la carpeta web/ pero no ignorará un fichero que se llame web

Hasta aquí lo que ya habíamos visto ¿qué nos falta?

- Si el patrón no termina en /, git lo interpreta como un “[shell glob pattern](#)”. Internamente git llama a la función [fnmatch](#) con el flag FNM_PATHNAME: esto significa que ni el * ni el ? incluyen el carácter /. En seguida veremos ejemplos.
- Una / al principio del patrón indica el inicio de una ruta relativa a la carpeta en la que se encuentre el fichero .gitignore. Una línea del tipo /config.php ignorará el fichero config.php pero no ignorará web/config.php

II. SHELL GLOBS EN GIT

Los ficheros .gitignore, .git/info/exclude y .gitignore_local permiten incluir patrones de tipo shell. Si habéis trabajado con cualquier shell de Unix o el CMD de windows, estaréis familiarizados con el funcionamiento de estos patrones. En caso contrario, aquí tenéis las reglas que se siguen.

En una expresión que contenga el símbolo “?”, éste se corresponde con uno y sólo un carácter

Ejemplos:

```
# Ignora los ficheros bg-01.jpg bg-1t.jpg
# # No ignora los ficheros bg-1.jpg y bg-001.jpg # bg-???.jpg
```

Este patrón ignorará los ficheros cuyos nombres empiezan por “bg-”, van seguidos exactamente de dos caracteres, y luego van seguidos de “.jpg”. Por eso el fichero bg-1.jpg no se ignora, porque entre “bg-” y “.jpg” no hay dos caracteres, sólo uno. **Este ejemplo ilustra que el símbolo “?” no indica un carácter opcional.**

En una expresión que contenga el símbolo “*”, éste se corresponde con cualquier número de caracteres

Ejemplos:

```
# Ignora todos los ficheros acabados en .nib # # Ignora tanto MainWindow.nib y builds/MainWindow.nib *.nib # Ignora todos los ficheros que empiecen por "bg-" y acaben en ".jpg" bg-*.jpg
```

En el primer ejemplo, estamos ignorando cualquier fichero que acabe en .nib, independientemente de la carpeta en la que se encuentre. En el segundo ejemplo, estamos ignorando cualquier fichero que empiece por “bg-” y acabe por “.jpg”. Una pregunta: ¿se ignorará el fichero “bg-.jpg” usando este patrón?. La respuesta la tenéis un poco más abajo.

En una expresión que contenga corchetes [], éstos se corresponden con uno y sólo un carácter de los que están incluidos en los corchetes.

Ejemplos:

```
# Ignora los ficheros *.o y *.a *. [ao] # Ignora los ficheros .nib y .xib *. [xn]ib
```

Estos patrones no ignoran los ficheros liberia.so o los ficheros mainwindow.ib. Es necesario que alguno de los caracteres que están entre corchetes aparezcan en el fichero.

Si se necesita utilizar los caracteres ? * [] en un patrón, estos deberán escaparse con \

Ejemplo:

```
# Ignorando ficheros que contienen [ y ] en el nombre # # Este patrón ignorará los ficheros # cache/cache-[hG7].cache y cache/cache-[009].cache cache/cache-\[???\].cache
```

III. SHELL GLOBS Y CARPETAS

¿Qué significa que git utilice el flag FNM_PATHNAME al llamar a la función [fnmatch](#)? **Que los símbolos * ? y [] no incluyen el caracter "/" de separación de directorios.**

```
# El uso de FNM_PATHNAME hace que el
siguiente patrón ignore los ficheros #
web/fichero.html, pero no ignore el
fichero web/admin/fichero.html # # Como
el " * " no incluye " / ", el fichero
web/admin/fichero.html no # encaja con
el patrón web/*.html
```

¿Qué ocurre si necesitamos justo el efecto contrario? es decir, queremos ignorar todos los ficheros .html que están en las subcarpetas de web pero no ignorar los ficheros .html que están en web. El patrón sería el siguiente:

```
# Ignora los ficheros .html que están en
subcarpetas de la carpeta web # No ignora
los ficheros web/*.html # # Con este
patrón, el fichero web/fichero.html se
incluira en el repositorio # mientras que
ficheros con web/admin/fichero.html o
web/images/index.html se # ignorarán
web/**/*.html
```

Otra pregunta: Este patrón ¿ignora el fichero web/admin/subcarpeta/fichero.txt?

IV. RUTAS RELATIVAS

Un patrón que comienza por "/" indica que para decidir si el fichero se ignora o no, se utiliza una ruta relativa al fichero .gitignore:

```
# Ignora los fichero *.c en la carpeta
raiz del proyecto: memory.c # No ignora
los ficheros en subcarpetas
como modules/cache.c /*.c
```

V. CONCLUSIONES

En esta entrega hemos visto cómo interpreta git los patrones glob shell y los patrones a rutas relativas. En la siguiente entrega veremos cómo podemos usar múltiples ficheros .gitignore. Estos nos ayudarán a salvar las limitaciones que tenemos a la hora de trabajar con múltiples subcarpetas.

REFERENCIAS

- [1] <http://aprendegit.com/ignorando-ficheros-en-git-parte-iv-mas-patrones/>
- [2] <http://git-scm.com/book/es/Fundamentos-de-Git-Guardando-cambios-en-el-repositorio>
- [3] <http://conocimientoabierto.es/primeros-pasos-configurar-git/223/>