

# Patrones Glob (Febrero 2014)

Lina Marcela Díaz Calderón 112501.

**Virgulilla** — La virgulilla generalmente es identificada como la tilde de la ñe, virgulilla de la ñe' (~) o rabo de choncho, aunque la Real Academia Española acepta también como ejemplos de virgulilla el apóstrofo (') , la cedilla (Ç) y el acento agudo ('). Las palabras «tilde» y «virgulilla» se pueden referir a cualquier trazo, sin embargo, el contexto puede indicar que se habla específicamente del signo ~.

## I. INTRODUCCIÓN

ESTE documento describe y especifica los patrones glob, relacionando ejemplos específicos de algunos de estos patrones con fin de evidenciar su funcionamiento.

## II. PATRONES GLOB

En los patrones Glob se encuentra todas las rutas que coincidan con un patrón específico, según las normas de la Shell de Unix. No se expande la virgulilla, pero sí los comodines \*, ? y los rangos expresados por []. Se realiza mediante las funciones `os.listdir()` y `fnmatch.fnmatch()`, sin invocar una sub-shell[1].

Por lo general encontramos estos tipos de patrones que se puede utilizar cuando se está usando asociación con patrones glob.

- La secuencia “[” se asocia con cualquier cadena, dentro de un único directorio.

```
$ hg add 'glob:[.py]
adding main.py
```

- La secuencia “[” se asocia con cualquier cadena, y cruza los límites de los directorios. No es un elemento estándar de los tokens de glob de Unix, pero es aceptado por varios intérpretes Unix populares, y es muy útil.

```
$ cd ..
$ hg status 'glob:[.py]
A examples/simple.py
A src/main.py
```

- La secuencia “?” se asocia con cualquier caracter sencillo.

```
$ hg status 'glob:[.py]'
```

- El caracter “[” marca el inicio de una clase de caracteres. Ella se asocia con cualquier carácter sencillo dentro de la clase. La clase se finaliza con un carácter “]”. Una clase puede contener múltiples rangos de la forma “a-f”, que en este caso es una abreviación para “abcdef”.

```
$ hg status 'glob:[.py]'
```

- Si el primer carácter en aparecer después de “[” en la clase de caracteres es un “!”, se niega la clase, haciendo que se asocie con cualquier carácter sencillo que no se encuentre en la clase.

Un “{” marca el inicio de un grupo de subpatrones, en donde todo el grupo es asociado si cualquier subpatrón en el grupo puede ser asociado. El carácter “,” separa los subpatrones, y el “}” finaliza el grupo.

```
$ hg status 'glob:[.py]'
```

A continuación podemos evidenciar un ejemplo básico del uso básico de glob://

```
<?php
// Ejemplo de uso de glob:// para encontrar archivos
// en un directorio y sus subdirectorios
// Se utiliza la función glob() para encontrar los archivos
$it = new DirectoryIterator("c:/xampp/htdocs/");
foreach($it as $f) {
    printf("Nombre: %s\n", $f->getFilename());
    printf("Tamaño: %s\n", $f->getSize()/1024);
}
//
```

```
tree,10K
findregx.php,0.6K
findfile.php,0.7K
dba_dump.php,0.9K
ncccdin.php,1.1K
plan_frus_dir.php,1.0K
lri_grop.php,0.9K
directorytree.php,0.9K
dba_array.php,1.1K
class_tree.php,1.0K
```

Los ficheros .gitignore, .git/info/exclude y .gitignore\_local permiten incluir patrones de tipo shell. Estos Globos tradicionales shell utilizan una sintaxis muy sencilla, que es menos expresivo que una Regular Expresión. Entrega final.

### III. EJEMPLOS DE PATRONES GLOB

#### A. Actionsript.gitignore - garantizar solo salto de línea final

```
# Build and Release Folders
bin/
bin-debug/
bin-release/

# Other files and folders
.settings/

# Project files, i.e. ".project", ".actionScriptProperties" and ".flex"
# should NOT be excluded as they contain compiler settings and other ir
# information for Eclipse / Flash Builder.
```

Fig. 1 Descripción Actionsript.gitignore

#### B. Agda.gitignore - Crear Agda.gitignore

```
*.agda.i
```

Fig. 2 Descripción Agda.gitignore

#### C. Go.gitignore

```
# Compiled Object files, Static and Dynamic: libs (Shared Objects)
*.o
*.a
*.so

# Folders
_obj
_test

# Architecture specific extensions/prefixes
*.{386|x86}
{586|x86}.out

*.cgo1.go
*.cgo2.c
*_cgo_defun.c
*_cgo_gotypes.go
*_cgo_export.*

*_testmain.go

*.exe
*.test
```

Fig. 3 Descripción Go.gitignore

#### D. CMake.gitignore -Added a .gitignore for files auto-generated by the CMake build-system

```
CMakeCache.txt
CMakeFiles
Makefile
cmake_install.cmake
install_manifest.txt
```

Fig. 4 Descripción CMake.gitignore

#### E. DM.gitignore -Alta. Gitignore para el lenguaje de DM.

```
*.dmb
*.rsc
*.int
*.lk
*.zip
```

Fig. 5 Descripción CMake.gitignore

#### F. Gradle.gitignore - tipo fijo en el nombre del archivo

```
.gradle
build/
```

Fig. 6 Descripción Gradle.gitignore

#### G. Jboss.gitignore - Archivo Desplegado ignorar

```
jboss/server/all/deploy/project.ext
jboss/server/default/deploy/project.ext
jboss/server/minimal/deploy/project.ext
jboss/server/all/log/*.log
jboss/server/all/tmp/*/*
jboss/server/all/data/*/*
jboss/server/all/work/*/*
jboss/server/default/log/*.log
jboss/server/default/tmp/*/*
jboss/server/default/data/*/*
jboss/server/default/work/*/*
jboss/server/minimal/log/*.log
jboss/server/minimal/tmp/*/*
jboss/server/minimal/data/*/*
jboss/server/minimal/work/*/*

# deployed package files #
*.DEPLOYED
```

Fig. 7 Descripción Gradle.gitignore

#### H. Processing.gitignore - Añadida el proyecto de procesamiento

```
.DS_Store
applet
application.linux32
application.linux64
application.windows32
application.windows64
application.macosx
```

Fig. 8 Descripción Gradle.gitignore

#### I. Perl.gitignore - ignorar más salidas de perl

```
/blib/
/.build/
_build/
cover_db/
inc/
Build
'Build/
Build.bat
.test_cover_stats
/Makefile
/Makefile.old
/MANIFEST.bak
/META.yml
/META.json
/MYMETA.*
nyptrprof.out
/pm_to_blib
*.o
*.bs
```

Fig. 9 Descripción Perl.gitignore

#### J. WordPress.gitignore - caso omiso de los datos de configuración sensibles

```
.htaccess
wp-content/uploads/
wp-content/blogs.dir/
wp-content/upgrade/
wp-content/backup-db/
wp-content/advanced-cache.php
wp-content/wp-cache-config.php
sitemap.xml
*.log
wp-content/cache/
wp-content/backups/
sitemap.xml.gz
wp-config.php
```

Fig. 10 Descripción WordPress.gitignore

K. *Gradle.gitignore* - tipo fijo en el nombre del archivo

```
.gradle
build/
```

**Fig. 11 Descripción Gradle.gitignore**

### L. *RubyMotion.gitignore*

```

+ .dat*
+ .repl_history
+ build
+ build/
+ tags

+ Editor files

```

**Fig. 12 Descripción RubyMotion.gitignore**

*M. WordPress.gitignore - también caso omiso de los datos de configuración sensibles*

```
.htaccess
wp-content/uploads/
wp-content/blogs.dir/
wp-content/upgrade/
wp-content/backups-db/
wp-content/advanced-cache.php
wp-content/wp-cache-config.php
sitemap.xml
=.log
wp-content/cache/
wp-content/backups/
sitemap.xml.gz
wp-config.php
```

**Fig. 13 Descripción WordPress.gitignore**

*N. Python.gitignore*

[illegible]

**Fig. 14 Descripción Python.gitignore**

*O. Objetivo-C.gitignore - Archivo ipa Alta al objetivo-C.gitignore*

[illegible]

**Fig. 15 Descripción Objetivo-C.gitignore**

*P. Scala.gitignore -No haga caso de la hoja de trabajo del directorio*

1. 在下列各题中，选择正确的答案。  
 2. 在下列各题中，选择正确的答案。  
 3. 在下列各题中，选择正确的答案。  
 4. 在下列各题中，选择正确的答案。  
 5. 在下列各题中，选择正确的答案。  
 6. 在下列各题中，选择正确的答案。  
 7. 在下列各题中，选择正确的答案。  
 8. 在下列各题中，选择正确的答案。  
 9. 在下列各题中，选择正确的答案。  
 10. 在下列各题中，选择正确的答案。

**Fig. 16 Descripción Scala.gitignore**

*Q. WWW.gitignore - comentarios añadidos*

```
# .php backup files
*.xml

# Dynamic plugins .dll
bin\
```

**Fig. 17 Descripción VVVV.gitignore**

R. gitignore / ZendFramework.gitignore

INTERNAL, EXTERNAL, PUBLIC, PRIVATE, GROUP, OTHER DOCUMENTATION

Fig. 18 Descripción gitignore / ZendFramework.gitignore

S. *Waf.gitignore* - añadido *Waf.gitignore*

\* for projects that use `Waf` for building: <http://code.google.com/p/waf/>  
`.waf.*`  
`.lock.*`

**Fig. 19 Descripción Waf.gitignore**

### T. Unity.gitignore - Información actualizada sobre el uso de la escritura boo generada por Unity3D

```
[Library]
[Temp]
[Output]

# Auto-generated version solution and project files
*.csproj
*.unityproj
*.asmx
*.pub
*.user
*.userprefs
*.pdb
*.disproj
```

**Fig. 20 Descripción Unity.gitignore**

#### IV. CONCLUSIÓN

Se concluye que los patrones que Git utiliza en los ficheros .gitignore, interpretan patrones Glob encuentra todas las rutas que concuerdan con un patrón específico.

## REFERENCIAS

- [1] Referencia de la Biblioteca de Python "6.19 glob", <http://pyspanishdoc.sourceforge.net/lib/module-glob.html>

<http://pyspanishdoc.sourceforge.net/lib/module-glob.html>  
Lina Marcela Díaz Calderón  
Lina Marcela Díaz Calderón 112501, Aplicaciones Web Avanzadas