

# Una sugerencia práctica de versionado semántico

20220506T1623

Una sugerencia práctica de numeros de versión semánticos: X.Y.p.c

(nota: está basada en el estandar <https://semver.org/lang/es/>, solo que usando la parte de "PARCHE" para desarrollo/pruebas)

Cara a los usuarios finales, solo existirán las dos primeras cifras X.Y

=> X es versión mayor: cambios no retrocompatibles

=> Y es versión menor: cambios compatibles con cualquier otra con el mismo X

También lo podemos ver como que:

=> Aumentamos X cuando se emite un lanzamiento completo a "producción".

=> Aumentamos Y cuando se emite una corrección sobre la última versión X. en producción

Cara al desarrollo, existirán cuatro cifras: X.Y.p.c

=> p.0 es una versión "para que alguien interno la pruebe", una versión que solo verán los usuarios internos de pruebas: X.Y.1.0, X.Y.2.0, X.Y.3.0, etc.

=> p.c es una versión "de compilacion", una versión que solo verán los programadores: X.Y.p.1, etc.

También lo podemos ver como que:

=> Aumentamos p cuando se emite una versión "para pruebas" internas.

=> Aumentamos c cuando empezamos a trabajar después de cualquier emisión.

Ejemplo práctico:

|                                       | DESARROLLO | PRODUCCIÓN | notas                     |
|---------------------------------------|------------|------------|---------------------------|
| trabajando                            | 1.0.0.1    |            |                           |
| emisión de una versión para probar    | 1.0.1.0    |            |                           |
| trabajando                            | 1.0.1.1    |            |                           |
| emisión de una versión para probar    | 1.0.2.0    |            |                           |
|                                       | etc        |            |                           |
| emisión de una versión para usuarios  |            | 1.1.0.0    |                           |
| trabajando                            | 2.0.0.1    |            |                           |
| emisión de una versión para probar    | 2.0.1.0    |            |                           |
| trabajando                            | 2.0.1.1    |            |                           |
| trabajando en la corrección de un bug |            | 1.1.0.1    | se parte desde la 1.1.0.0 |

|   |         |         |  |
|---|---------|---------|--|
| para probar la corrección                       |         | 1.1.1.0 |  |
| trabajando en la corrección de un bug           |         | 1.1.1.1 |  |
| para probar la corrección                       |         | 1.1.2.0 |  |
| emisión de una corrección para usuarios         |         | 1.2.0.0 |  |
| incorporar la corrección al desarrollo en curso | 2.0.1.1 |         | (*)  |
| trabajando                                      | 2.0.1.1 |         | (*)  |
| emisión de una versión para probar              | 2.0.2.0 |         |  |
|   | etc     |         |  |
| emisión de una versión para usuarios            |         | 2.1.0.0 | Se retira la 1.Y.0.0 que estuviera en ese momento. |
| trabajando                                      | 3.0.0.1 |         |  |
| emisión de una versión para probar              | 3.0.1.0 |         |  |
| trabajando                                      | 3.0.1.1 |         |  |
| emisión de una versión para probar              | 3.0.2.0 |         |  |
| trabajando                                      | 3.0.2.1 |         |  |
| emisión de una versión para probar              | 3.0.3.0 |         |  |
|   | etc     |         |  |
|   |         |         |  |

Nota: (\*) Incrementar .c más allá de .1 es opcional, ya que es algo solo para la persona o el equipo que está programando.

#### IMPORTANTE:

Una vez lanzada una versión X.Y (X.Y.0.0) o X.Y.p (X.Y.p.0),  
 JAMÁS de los jamases SACAR OTRA CON EL MISMO X.Y.p.0  
 => en la siguiente compilación:  
     incrementar .c (poner al menos X.Y.p.1)  
 => en el siguiente lanzamiento:  
     incrementar .p (si es de pruebas) o X.Y (si es de producción)

nota: No pasa nada por lanzar una X.Y para después, en muy poco tiempo, lanzar otra X.Y+1 o incluso un poco más tarde otra X.Y+2 (por ejemplo, si se detectan errores graves y se han de corregir con rapidez).

Tampoco pasa nada por llegar hasta versiones del estilo de 57.24.8 o similares...

EL VERSIONADO ES SOLO PARA IDENTIFICAR y saber con qué pieza ÚNICA de software estamos lidiando.

Conviene no mezclar aspectos "estéticos" o "marquetinianos" en el diseño del versionado. Por ejemplo: consideraciones acerca de qué número de versión queda más o menos "bonito" o cual es más o menos "vendible" o cual revela menos la cantidad de errores que se han tenido que corregir.

La mejor manera de mantener la unicidad es seguir reglas simples.