

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

(nota: Está basada en el estándar <https://semver.org/lang/es/>, solo que usando la parte de “parche” para “pruebas” de desarrollo.)

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

=> X es versión mayor: un lanzamiento completo “a bombo y platillo”.

=> Y es versión menor: un lanzamiento de correcciones, que es compatible 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.

nota: La emisión de una versión de producción X.Y va siempre con p0 y c0 → X.Y.p0.c0

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

=> N.0 es la versión N de pruebas,

una versión que solo verán los usuarios internos de pruebas: X.Y.p1.c0 , X.Y.p2.c0 , etc.

=> N.1 es una versión de trabajo, preparando la emisión de N,

una versión que solo verán los programadores: X.Y.p3.c1 , 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.

nota: La emisión de una versión de prueba siempre va con c0 → X.Y.pN.c0

## IMPORTANTE:

Una vez emitida una versión X.Y (X.Y.p0.c0) o X.Y.p (X.Y.pN.c0),

**JAMÁS** de los jamases SACAR OTRA CON EL MISMO X.Y.p

Para evitar repetir, la regla más simple puede ser:

=> Nada más emitir una versión, antes de seguir trabajando, aumentar la X o la Y o la p según corresponda, y poner c a uno.

=> Cuando llega el momento de emitir una nueva versión, justo antes de compilarla, revertir a .c0 o a .p0.c0 según corresponda.

Un ejemplo práctico:

	MAIN	RELEASE	notas
empezar el proyecto	1.1.p1.c1		
emisión de una versión para probar	<b>1.1.p1.c0</b>		
trabajando	1.1.p2.c1		
emisión de una versión para probar	<b>1.1.p2.c0</b>		
	etc		
emisión de una nueva versión a producción	<b>1.1.p0.c0</b>	<b>1.1.p0.c0</b>	
trabajando	2.1.p1.c1		
emisión de una versión para probar	<b>2.1.p1.c0</b>		
trabajando	2.1.p2.c1		
trabajando en la corrección de un bug		1.2.p1.c1	se parte desde la 1.1.p0.c0
para probar la corrección		<b>1.2.p1.c0</b>	
trabajando en la corrección de un bug		1.2.p2.c1	
para probar la corrección		<b>1.2.p2.c0</b>	
		etc	
emisión de la corrección a producción		<b>1.2.p0.c0</b>	
incorporar la corrección al desarrollo en curso	<b>2.1.p2.c0</b>		
trabajando	2.1.p3.c1		
emisión de una versión para probar	<b>2.1.p3.c0</b>		
	etc		
emisión de una nueva versión a producción	<b>2.1.p0.c0</b>	<b>2.1.p0.c0</b>	Se retira la versión 1.y.p0.c0
trabajando	3.1.p1.c1		
emisión de una versión para probar	<b>3.1.p1.c0</b>		
trabajando	3.1.p2.c1		
emisión de una versión para probar	<b>3.1.p2.c0</b>		
trabajando	3.1.p3.c1		
emisión de una versión para probar	<b>3.1.p3.c0</b>		

trabajando	3.1.p4.c1		
emisión de una nueva versión a producción	<b>3.1.p0.c0</b>	<b>3.1.p0.c0</b>	Se retira la versión 2.y.p0.c0
trabajando	4.1.p1.c1		
emisión de una versión para probar	<b>4.1.p1.c0</b>		
trabajando	4.1.p2.c1		
trabajando en la corrección de un bug		3.2.p1.c1	se parte desde la 3.1.p0.c0
para probar la corrección		<b>3.2.p1.c0</b>	
trabajando en la corrección de un bug		3.2.p2.c1	
para probar la corrección		<b>3.2.p2.c0</b>	
		etc	
emisión de la corrección a producción		<b>3.2.p0.c0</b>	
incorporar la corrección al desarrollo en curso	<b>4.1.p2.c0</b>		
trabajando	4.1.p3.c1		
emisión de una versión para probar	<b>4.1.p3.c0</b>		
trabajando	4.1.p4.c1		
	etc		
		etc	
	etc		

El versionado es SOLO PARA IDENTIFICAR exactamente con qué pieza única de software estamos lidiando. De ahí la importancia de que cada emisión sea realmente única.

Es decir, cualquier cambio, por mínimo que sea, ha de implicar si o si algún cambio de versión.

nota: Emitir una versión significa dejar una copia del software en manos de cualquier persona fuera del equipo que está programando.

nota: Nunca es buena idea lanzar otra emisión con la misma numeración. Si se descubre cualquier problema a pocos minutos u horas de haber emitido una versión. Tendremos que emitir otra versión con una numeración diferente.

nota: No pasa nada por emitir una versión para después, casi enseguida, emitir otra +1 e inmediatamente después otra +2, seguida de otra +3 al poco tiempo, etc.

nota: Tampoco pasa nada por llegar hasta versiones del estilo de **57.24.p8.c0** o similares...

Lo dicho, lo importante es que cada versión emitida sea única.

Al diseñar un esquema de versionado. Si se empiezan a considerar aspectos “estéticos” o “marquetinianos” tales como: qué número de versión queda más o menos “bonito”, o cual es más o menos “vendible”, o qué versión “conviene” emitir en ese momento, o cual puede “disimular” mejor la cantidad de errores que se han tenido que corregir, o...

Es muy posible que acabemos con un esquema de versionado subjetivo y difícil de gestionar. Un esquema donde tendremos discusiones y dudas continuas cada vez que vamos a emitir una nueva versión.

Para evitarlo, se ha de intentar poner reglas objetivas y simples. Reglas que permitan ir incrementando versiones sin mucho pensar. Como, por ejemplo, las anteriormente citadas:

- => Nada más emitir una versión, antes de seguir trabajando, aumentar la X o la Y o la p según corresponda, y poner c a uno.
- => Cuando llega el momento de emitir una nueva versión, justo antes de compilarla, revertir a .c0 o a .p0 .c0 según corresponda.