



**ANEP**



**UTU**

DIRECCIÓN GENERAL  
DE EDUCACIÓN  
TÉCNICO PROFESIONAL



Instituto Tecnológico Superior  
**UTU**



**Solicitante:**

**I.T.S. – Instituto Tecnológico Superior Arias - Balparda**

**Nombre de Fantasía del Proyecto: High-End**

**Grupo de Clase: 3ºBP**

**Turno: Matutino**

**Materia: Programación**

**Nombre de los Integrantes del Grupo: Jordi Hornos, Ignacio Fernández, Luka Fialho y Matías Suárez.**

**Fecha de entrega: 24/7/2020**

**Instituto Tecnológico Superior Arias Balparda**

*Gral. Flores 3591 esq. Bvar. José Batlle y Ordoñez - Montevideo*



**ANEP**



**UTU**

DIRECCIÓN GENERAL  
DE EDUCACIÓN  
TÉCNICO PROFESIONAL



Instituto Tecnológico Superior

**UTU**

---

## Indice

<b>1. Versión de Software</b>	<b>3</b>
<b>1.1. ¿Qué es una Versión?</b>	<b>3</b>
<b>1.2. ¿Cómo se gestionan las versiones?</b>	<b>3</b>
<b>1.3. Versión Pre-alfa</b>	<b>4</b>
<b>1.4. Versión Alfa</b>	<b>4</b>
<b>1.5. Versión Beta</b>	<b>4</b>
<b>1.6. Lanzamiento Final</b>	<b>5</b>
<b>1.7. ¿En qué momento se cambian los valores?</b>	<b>5</b>

**ANEP****UTU**DIRECCIÓN GENERAL  
DE EDUCACIÓN  
TÉCNICO PROFESIONAL

---

## **1. Versión de software**

### **1.1. ¿Qué es una versión?**

Una versión de software es un conjunto específico del servidor de archivos descargables (parches) que contiene el firmware, el software, los controladores de hardware, las herramientas y las utilidades disponibles para el servidor. Todos estos archivos se han probado en conjunto y se ha verificado que funcionan con el servidor.

Debe actualizar el firmware y el software del servidor lo antes posible una vez que esté disponible una nueva versión de software. Las versiones de software, a menudo, incluyen correcciones de errores, y la actualización garantiza que tendrá el firmware y el software más reciente.

El documento ReadMe (Léame) que se incluye con cada parche en una versión de software contiene información acerca del parche, por ejemplo, lo que ha cambiado o no ha cambiado respecto de la versión de software anterior, además de correcciones de errores en la versión actual.

Las notas del producto del servidor identifican las versiones de software admitidas en su servidor.

### **1.2. ¿Cómo se gestionan las versiones?**

Además de tener las versiones por números podemos agregar una clasificación por estabilidad del proyecto.

Las opciones que tenemos para esto son: Pre-alfa, Alfa, Beta, Versión final.



---

### **1.3. Versión Pre-alfa**

Es una vista previa muy temprana que aún no está destinada al público. Esta es una vista previa para desarrolladores en la que aún pueden aparecer errores elementales. En este momento, solo se encuentra disponible un marco aproximado del software, los bloqueos y las funciones e interfaces faltantes son más la regla que la excepción.

### **1.4. Versión Alfa**

En la versión alfa hay una primera compilación resumida, que todavía tiene muchos errores y solo tiene que ser parcialmente representativa para el software final. Normalmente, las versiones alfa solo están disponibles para grupos de usuarios muy limitados (principalmente otros ingenieros de software). Con los próximos grandes pasos hacia la versión beta, los grupos de prueba se ampliarán gradualmente

### **1.5. Versión Beta.**

La versión beta debería ser significativamente más funcional y estable que una alfa, pero aún puede estar muy incompleta y contener errores importantes. A menudo existe la posibilidad de que el software pase a una versión beta pública y, por lo tanto, lo pruebe un gran número de usuarios antes de su lanzamiento.

Este modelo no sólo proporciona a los desarrolladores abundante información, sino también valiosos comentarios para revisar su software. El modelo de lanzamiento de “acceso anticipado” también es una opción aquí, en el que los usuarios pueden comprar software antes del lanzamiento del software (generalmente a un precio reducido) y luego recibir acceso a una versión beta pública.

---

## 1.6. Lanzamiento Final

Derivado de la versión beta, que se seguirá desarrollando hasta el lanzamiento final, también se encuentra la “Beta perpetua” (beta eterna). Desde el desarrollo web, este término es una alusión al desarrollo y mejora constante de software como los sitios web. Así que, en última instancia, nunca se terminan del todo y se encuentran en una beta eterna.

El software “normal” se sigue desarrollando a partir de la versión beta para que estén disponibles una o más versiones candidatas. Estos candidatos de desarrollo están cerca de la calidad requerida para una versión de software; lo único que importa aquí es el nivel adecuado de garantía de calidad.

## 1.7. ¿En qué momento se cambian los valores?

Algo común es realizar el manejo de versiones mediante 3 número: X.Y.Z y cada uno indica una cosa diferente:

- El primero (X) se le conoce como **versión mayor** y nos indica la versión principal del software. Ejemplo: 1.0.0

- 
- El segundo (Y) se le conoce como **versión menor** y nos indica nuevas funcionalidades. Ejemplo: 1.2.0
  - El tercero (Z) se le conoce como **revisión** y nos indica que se hizo una revisión del código por algún fallo. Ejemplo: 1.2.2

Ahora que conocemos el significado de cada número, viene una pregunta importante: ¿Cómo sabemos cuándo cambiarlos y cuál cambiar?

- Versión mayor o X, cuando agreguemos nuevas funcionalidades importantes, puede ser como un nuevo módulo o característica clave para la funcionalidad.
- Versión menor o Y, cuando hacemos correcciones menores, cuando arreglamos un error y se agregan funcionalidades que no son cruciales para el proyecto.
- Revisión o Z, cada vez que entregamos el proyecto.