

# Cómo elegir la versión apropiada de FreeBSD

## El Proyecto de Documentación de FreeBSD

Copyright © 2005 El Proyecto de Documentación de FreeBSD  
\$FreeBSD: release/9.1.0/es\_ES.ISO8859-1/articles/version-guide/article.sgml  
38826 2012-05-17 19:12:14Z hrs \$

FreeBSD is a registered trademark of the FreeBSD Foundation.

Así que ha decidido instalar FreeBSD. ¡Bienvenido! el propósito de este documento es ayudarle seleccionar la versión apropiada.

*Traducción de Gábor Kövesdán <gabor@FreeBSD.org>.*

## Table of Contents

1 Introducción .....	1
2 El calendario de releases anteriores .....	2
3 Calendario de releases de aquí en adelante .....	4
4 ¿Cómo afectan estos factores a su decisión? .....	4
5 Conclusión .....	5

## 1 Introducción

Antes de decidir cuál de las versiones de FreeBSD quiere usar es importante que comprenda los conceptos relacionados con el desarrollo y el proceso de Ingeniería de Releases (RE).

FreeBSD se desarrolla gracias a un gran grupo de gente, casi siempre voluntarios. El código fuente del kernel, de las utilidades y de las bibliotecas más comunes están en un *sistema de gestión de código* del cual es posible descargarlo en cualquier momento. Aparte de esto, existen versiones (binarias) ya compiladas que se liberan cada poco tiempo. Una de estas versiones binarias cuidadosamente revisada será en su momento declarada *releases*.

### 1.1 Releases

El nombre de las *releases* contiene un *número mayor de release* y un *número menor de release*.

- El propósito de una release mayor es incluir nuevas funciones. Es inevitable que, al añadir nuevas funciones a FreeBSD o al quitarlas, sea necesario algunas veces perder la compatibilidad con versiones anteriores del sistema operativo.
- El propósito de una release menor es ante todo corregir errores y mejorar el rendimiento y la estabilidad. Es importante mantener la compatibilidad entre releases menores tanto cuando se trata de código como con los programas ejecutables. Si se da la ocasión, se añaden nuevas características a una release menor si estos principios se mantienen.

No obstante, tenga en cuenta que una “release” es solamente una instantánea del árbol de código en un momento dado, gracias a lo cual se le da una etiqueta o *tag*. Por ejemplo, la etiqueta que el grupo de ingeniería de releases dio a la release 5.4 fue `RELENG_5_4_0_RELEASE`. El desarrollo tiene lugar bajo la etiqueta `HEAD`.

## 1.2 Bifurcaciones

En el tiempo de sacar cada release, se crea una *rama*, por ejemplo `RELENG_5_4`. Aunque el código bajo `RELENG_5_4_0_RELEASE` no cambien, los que están bajo `RELENG_5_4` sí, al aplicar cambios en `HEAD` al corregir problemas de seguridad u otro tipo de fallo.

## 1.3 *STABLE* y *CURRENT*

Durante la vida de cada release mayor una rama individual puede convertirse en *STABLE*. Esto indica que el Proyecto FreeBSD cree que la rama ha demostrado suficiente calidad para que la mayoría de los usuarios puedan usarla. Las ramas que necesitan más pruebas antes de que pueda usarlas cualquiera reciben el nombre de *CURRENT*.

**Note:** El Proyecto FreeBSD no puede garantizar que el software que se distribuye sea todo lo *estable* que sea necesario para cualquier necesidad o uso. Es el usuario quien tiene la última palabra sobre esto. Por favor, tenga muy en cuenta que el proyecto lo forman voluntarios y no puede ofrecer ningún tipo de garantía.

## 1.4 *Ports* y *packages*

Aparte de los ficheros que se distribuyen del modo ya descrito antes, FreeBSD permite el uso de miles de aplicaciones fruto del trabajo de desarrolladores que no forman parte del proyecto. Podemos citar como ejemplos sistemas de ventanas, navegadores web, programas de correo electrónico, software ofimático, etc.) El proyecto en sí no desarrolla estos programas, solamente el “framework” que permite que éstos puedan instalarse; este “framework” recibe el nombre de *Colección de Ports*). Se pueden instalar aplicaciones desde el código fuente si su licencia permite este tipo de redistribución; es lo que en FreeBSD se llaman los *ports*)), o como software compilado si está permitido distribuirlos como tal, en cuyo caso reciben el nombre de *packages*.

## 2 El calendario de releases anteriores

Durante el desarrollo de la release 5.X de FreeBSD hubo que aprender en carne propia muchas lecciones que solamente pudieron verse con posterioridad. Los objetivos de la serie 5.X fueron muy ambiciosos. Veamos algunos:

- Ofrecer soporte para máquinas dotadas de multiproceso simétrico (Symmetric MultiProcessing, o SMP)
- Mejoras del rendimiento gracias a la adopción de una nueva estrategia de gestión de recursos en el kernel
- Añadir numerosas arquitecturas de procesador
- Introducción de un nuevo modelo de “threading”
- Introducción de un nuevo “scheduler”
- Añadir soporte de nuevas tecnologías como la gestión de energía (especialmente importante en máquinas portátiles), y sobre todo
- no declarar ninguna versión como `STABLE` hasta que estas tareas no se hubieran terminado

Esto llevó al problema de que había varios años de diferencia entre el momento en el que 4.X se declaró `STABLE` y el momento en el que 5.X se llegó a `STABLE`. Esta circunstancia tuvo diversos efectos no deseados:

- El número funciones cambiadas simultáneamente hizo muy difícil aislar esos cambios para hacerlos compatibles con las versiones anteriores a la creación de la rama `STABLE`.
- Eso significó que los usuarios que necesitaban imperiosamente una nueva función en particular (por ejemplo el poder ejecutar FreeBSD en hardware moderno) estaban obligados a usar (por ejemplo) FreeBSD 5.2.1 a pesar de que oficialmente era una release de uso exclusivo de desarrolladores, y sin tener en cuenta el hecho de que una release `CURRENT` no cumplía sus demandas.
- En los casos en los que se consiguió la compatibilidad con versiones anteriores los desarrolladores se encontraron con otro problema al intentar adaptar ciertas características a una versión que ellos mismos hacía tiempo que no usaban como su plataforma de desarrollo principal.
- El retraso también provocó que cuando 5.3 se declaró nueva release `STABLE` la cantidad acumulada de cambios hizo la actualización complicada.

A decir verdad, nadie estaba contento con el resultado.

Las lecciones que se aprendieron de todo esto fueron:

- Las nuevas releases mayores deben tener menos cambios estructurales importantes y deben publicarse con mayor frecuencia.
- Siempre que sea posible los cambios estructurales deben aislarse unos de otros. Esto obliga a que parte del desarrollo tengan lugar fuera del árbol principal y que se integren solamente cuando no afecten a otros procesos simultáneos de desarrollo.
- Las releases mayores deben tener fecha de salida propia no dependiente de la fecha de entrega asignada a un cambio estructural. Si un cambio estructural no está listo a tiempo se incluirá desactivado por omisión y será incluido en la siguiente release.

Al publicar grupos de cambios más pequeños y de una forma más habitual se intenta también dedicar menos tiempo y esfuerzo aplicando nuevas características de `HEAD` a la última versión `STABLE` (y poder así usar dichas nuevas características en más de una versión mayor); aún más, al estar los cambios más aislados el riesgo de provocar nuevos problemas de seguridad es mucho menor.

Además, el concentrarse en una fecha y no en la consecución de una característica lista para integrarse en el sistema, es más fácil planificar para el futuro tanto para los usuarios como a los desarrolladores de aplicaciones ajenas al proyecto y, cómo no, para los desarrolladores de FreeBSD.

Estas razones (y no el intentar ir a la par de las versiones mayores de otro sistema operativo) son el principal motivo del cambio en el calendario de liberación de versiones de FreeBSD.

### 3 Calendario de releases de aquí en adelante

Estos son los objetivos actuales del calendario del Proyecto:

- Sacar un release mayor cada 18 meses
- Sacar un release menor cada 4 meses
- Ofrecer paquetes compilados para la release menor más reciente de cada release mayor
- Ofrecer actualizaciones de seguridad y otras correcciones de fallos críticos para las versiones menores más recientes de cada versión mayor (que reciben el nombre de *ramas de seguridad*).

Dado el gran número de combinaciones de versiones instalables no es posible dar soporte a todas las releases. Esto es, en parte, debido a la cantidad limitada de máquinas de las que el Proyecto puede disponer, pero sobre todo a que la cantidad de voluntarios disponibles es limitada y su tiempo también.

Si quiere leer más sobre esto visite

<http://www.FreeBSD.org/releng/index.html#schedule>

Calendario de ingeniería de releases

<http://www.FreeBSD.org/security/security.html#supported-branches>

Calendario de ramas de seguridad

Estos documentos profundizan en los porqués de las decisiones tomadas sobre las ramas soportadas y el ciclo de vida de cada rama.

### 4 ¿Cómo afectan estos factores a su decisión?

Los principales factores que influyen en su decisión de qué versión instalar son, entre otros:

- ¿Qué grado de estabilidad necesita su sistema?
- ¿Cuántos trabajos quiere dedicar a la actualización?
- ¿Durante cuánto tiempo va a usar una versión dada entre una actualización y la que venga más adelante?
- ¿Cuánta importancia le da a la seguridad de su sistema?
- ¿Instalará desde código fuente o binarios?
- ¿Va a participar en el desarrollo de FreeBSD?

Aquí hay unas normas para ayudarle a tomar una decisión:

- Si sus necesidades son a corto plazo y quiere disfrutar del más alto grado posible de estabilidad (y no puede dedicar muchos recursos a la actualización) probablemente lo mejor sea instalar la release `STABLE` más reciente y

dejarla como está. Según sean sus requisitos de seguridad puede o no aplicar los parches de seguridad que vayan apareciendo.

- Si sus necesidades son a corto plazo y las nuevas características o la seguridad son muy importantes para usted (y está dispuesto a dedicar tiempo a las actualizaciones) debería seguir la rama `STABLE` más reciente.
- Si no va a poner la máquina en producción, va a dedicar tiempo a depurar unos cuantos problemas y en unos cuantos meses va a salir una nueva versión mayor, puede instalar esa rama y ayudar al Proyecto haciendo pruebas para hacer el sistema más estable y disponer de la mejor release posible a medio y largo plazo.
- Sólomente si quiere instalar desde código fuente y pasar tiempo depurando problemas del sistema base, enviar informes de fallos y utilizar las listas de correo dedicadas a esos fallos debe usar `HEAD`.

## 5 Conclusión

Esperamos que este artículo haya servido de ayuda para que comprender el modelo de desarrollo de FreeBSD y pueda decidir qué versión se ajusta más a sus necesidades.