



## RESPALDOS CON BACK IN TIME EN UBUNTU, LINUX MINT, ETC

Publicado el [27 de noviembre de 2017](#) por atareao 24.8K [Deja un comentario](#)

Este artículo se publicó hace **mas de tres años**. Ten en cuenta que con las nuevas versiones y las actualizaciones constantes de software, es fácil, que este **artículo** esté **desactualizado**. Si estás interesado en que lo **actualice**, envíame un correo a través del [formulario de contacto](#).



Una de las tareas de mantenimiento, prevención y seguridad, mas importantes que tenemos que realizar con cualquier dispositivo, ya sea ordenador o móvil, es la **copia de seguridad**. Sin lugar a dudas, es probablemente, la tarea que muchos posponemos. Y solo nos acordamos que no hicimos la copia de seguridad cuando surge el problema. Sin embargo, realizar una copia de seguridad es algo realmente sencillo, mecánico y que puede estar completamente desatendido. Así, una potente herramienta que te puede ayudar para crear respaldos de tu documentación es **Back In Time**.

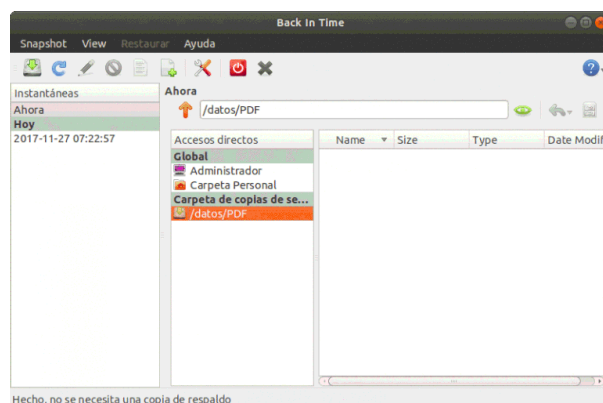
**Back In Time** no solo te ayuda a la hora de crear tus copias de seguridad en el momento, si no que te permite programarlas. De esta forma, no te tendrás que preocupar, tu mismo, de respaldar tu documentación, si no que **Back In Time** ya se encargará de ello.



## BACK IN TIME, CREACION Y MANTENIMIENTO DE COPIAS DE SEGURIDAD

**Back In Time** es una sencilla solución para la creación y mantenimiento de copias de seguridad en entornos Linux. Por supuesto, y como no podía ser de otra forma, se apoya en el todopoderoso **rsync**, sobre el que ya comenté en el artículo [“Sincronización a fondo con rsync”](#). Además utiliza enlaces duros para reducir el espacio de los archivos que no se modifican.

Se trata de una aplicación implementada en **Python 3** utilizando la librería gráfica **Qt 5**. Se desenvuelve perfectamente tanto en entornos de escritorio GNOME como KDE Plasma. Se trata de software libre, liberado bajo licencia GPL 2.



## SOBRE LAS COPIAS DE SEGURIDAD

Los respaldos se guardan en texto plano. Esto permite que podamos navegar dentro de nuestra copia de seguridad, ya sea desde nuestro explorador de archivos Nautilus o directamente desde el terminal. Y esto lo podemos hacer perfectamente, sin necesidad de utilizar *Back In Time*. Lo cual es un gran ventaja.

Por otro lado, tanto la propiedad como los permisos de los archivos se guarda en un archivo comprimido aparte. Todo esto para evitar conflictos con el dispositivo donde se guardan los respaldos. De esta manera, si el dispositivo donde guardamos la copia de seguridad no soporta permisos, *Back In Time*, utiliza el archivo para restaurar los

---

es posible que se pierdan los permisos.

## USO Y FUNCIONAMIENTO

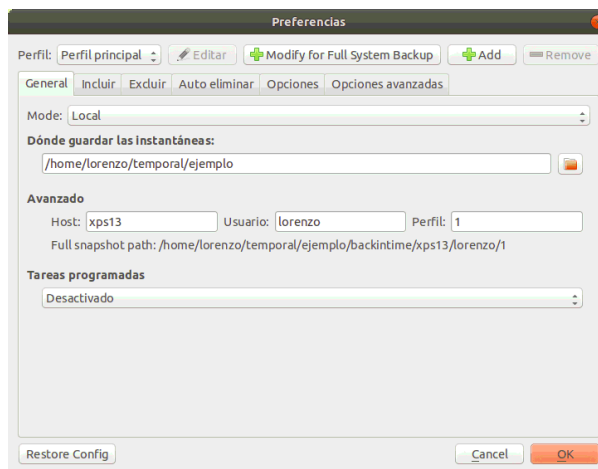
Como he comentado anteriormente, *Back In Time*, lo podemos utilizar tanto de forma manual como programada. En cualquier caso, y antes de utilizarlo, necesitamos definir unos parámetros mínimos. Básicamente debemos indicarle **que queremos respaldar** y **donde queremos respladarlo**.

### CONFIGURACIÓN DE RESPALDOS. EL DONDE.

Así, para indicarle donde queremos hacer el respaldo, *Back In Time*, nos ofrece cuatro modos posibles,

- *Local*. Las copias de seguridad locales se refieren tanto al almacenamiento en nuestro propio equipo, o en unidades montadas. Ahora bien, tanto el sistema de archivos de destino como el protocolo utilizado para montar la unidad deben soportar enlaces duros y simbólicos. Atención, porque **Samba** no soporta enlaces simbólicos por defecto. Es necesario activarlo. Por otro lado, **sshfs** no soporta enlaces duros.
- *Local cifrado*. El comportamiento es exactamente el mismo que en el caso de los respaldo locales, pero, estos se guardan cifrados con **EncFS**. El directorio cifrado se crea de forma automática en el directorio seleccionado.
- *SSH*. Este modo nos permite guardar nuestras copias de seguridad en un dispositivo remoto utilizando **SSH**. Para poder utilizar este modo, es necesario que el dispositivo remoto esté recogido en `known_hosts`. Además necesitarás las claves publica y privada guardadas en el dispositivo remoto, aunque desde la versión 1.2.0 de *Back In Time* esto ya lo hace **de forma automática**, si tienes una versión anterior, tendrás que configurarlo de forma manual.

esta forma, *Back In Time* montara una mística citrada del sistema de archivos local y los sincronizará utilizando `rsync`.



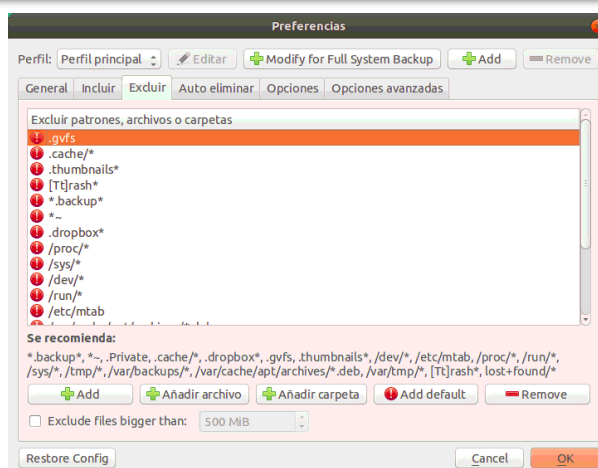
## EL CUANDO

Además de definir donde y en que modo queremos hacer los respaldos, si tenemos la intención **automatizarlos**, debemos indicar **cuando** queremos que estos se hagan. Así *Back In Time* nos da un amplio abanico de posibilidades. Esto se sustenta básicamente en **crontab**, en la mayor parte de los casos. Con lo que puedes ver la programación utilizando `crontab -l`, o incluso editarlo, utilizando `crontab -e`.

*Back In Time* nos permite programar los respaldos, no solo por tiempo, sino también cuando **conectemos un dispositivo**. Así, por ejemplo, podemos configurarlo para que cuando conectemos nuestro USB al ordenador se realice el respaldo de forma automática. Pero además podemos definir un tiempo mínimo, de forma que hasta que no haya transcurrido este intervalo no se realice el respaldo aunque conectemos el dispositivo.

## EL QUE

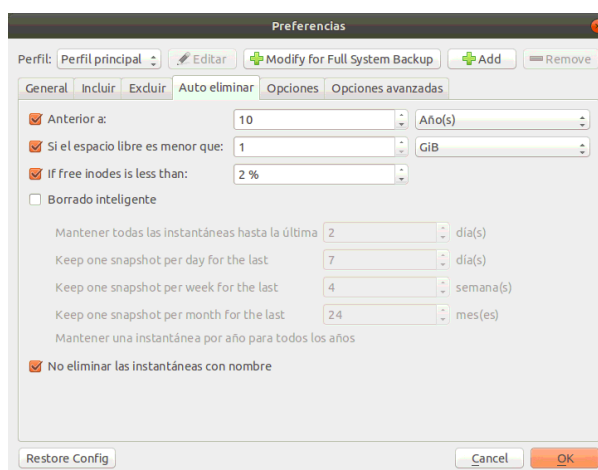
*Back In Time* te permite configurar que es lo que quieres respaldar, ya sean archivos o directorios, pero, también te permite configurar que es lo que no quieres respaldar. Es decir, te permite obviar copias



## LIMPIANDO

Además de todo esto, *Back In Time*, nos permite eliminar aquellos respaldos que cumplan una serie de condiciones,

- Respaldos que tengan una edad superior a un límite prefijado.
- Cuando el espacio libre sea inferior a un determinado límite.
- Cuando el nodo de inodos libres sea inferior a un valor prefijado.



## INSTALACIÓN

**Back In Time** no se encuentra en los repositorios oficiales de Ubuntu. Sin embargo, si utilizas [ppaurl](https://ppa.launchpad.net/ppaurl/ubuntu/+archive/primary), lo puedes instalar fácilmente haciendo clic en el siguiente enlace [Back In Time](#). En otro caso lo puedes instalar desde el terminal, ejecutando las siguientes ordenes para

---

```
sudo add-apt-repository ppa:bit-team/stable
sudo apt-get update
sudo apt-get install backintime-qt4
```

O bien, estas otras órdenes para la versión de prueba,

```
sudo add-apt-repository ppa:bit-team/testing
sudo apt-get update
sudo apt-get install backintime-qt
```

## OTRAS SOLUCIONES PARA REALIZAR TUS RESPALDOS O COPIAS DE SEGURIDAD

Además de *Back In Time*, en el ecosistema GNU/Linux, existe una amplia variedad de aplicaciones que nos permiten realizar copias de seguridad o respaldos de nuestra documentación. Entre ellos, cabe destacar los siguientes,

- [Deja Dup](#)
- [Cronopete](#)

Por supuesto, siempre puedes hacer uso de [rsync](#) combinado con [cron](#). Pero, si no estas ducho en el manejo de estas herramientas o del terminal, tanto *Back In Time* como *Deja Dup* seguro te van a ser de mucha utilidad.

## CONCLUSIONES

*Back In Time* es una herramienta sencilla y robusta, que te permitirá realizar y automatizar tus copias de seguridad. Los respaldos son una medida de seguridad fundamental para evitar problemas futuros, y sin lugar a dudas, **Back In Time**, es una potente herramienta que te simplificará considerablemente este tedioso trabajo.

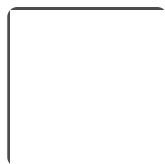
---

Más información,

- [Back In Time doc](#)
- [GitHub](#)

[BTFS un sistema de archivos BitTorrent para Ubuntu, Linux Mint, etc](#)

[Tizonia, un reproductor online para el terminal en Ubuntu, Linux Mint, etc](#)



### **BTFS UN SISTEMA DE ARCHIVOS BITTORRENT PARA UBUNTU, LINUX MINT, ETC**

BTFS es un sistema de archivos BitTorrent, que te permitirá ver una película con solo montar el archivo torrent o magnet link en tu Ubuntu, Linux Mint, etc



### **TIZONIA, UN REPRODUCTOR ONLINE PARA EL TERMINAL EN UBUNTU, LINUX MINT, ETC**




Tizonia es un reproductor online para el terminal. Una interesante alternativa a los reproductores tradicionales, con un reducido consumo de recursos.

Buscar



<p><b>Comunidad Linux con JF</b></p>	<p><b>Objetivo Rust</b></p>	<p><b>Navegación de vértigo con Raspberry OS</b></p>	<p><b>E</b> <b>o</b></p>
<p>hace 10 días • 1 comentario</p> <p>En esta nueva sección de Comunidad Linux la charla es con JF, un miembro ...</p>	<p>hace 2 meses • 1 comentario</p> <p>En este episodio del podcast te cuento sobre mi objetivo para este año ...</p>	<p>hace 2 meses • 1 comentario</p> <p>La Raspberry Pi Foundation está trabajando duro con su sistema operativo, y eso ...</p>	<p>ha</p> <p>Fi</p> <p>se</p> <p>er</p>

1 Comentario   El atareao    Política de privacidad de Disqus    1 Acceder ▾

 Recomendar 1    Tweet    Compartir   Ordenar por los mejores ▾



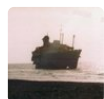
Únete a la conversación...

INICIAR SESIÓN CON

O REGISTRARSE CON DISQUS 



Nombre



**manuti** • hace 3 años • edited

Llevo años usándola <https://masmanuti.wordpress...>

Tengo todos las copias con este sistema desde hace años y es perfecto. He donado varias veces al proyecto porque no veo que lo hayan respaldado lo suficiente desde las distribuciones importantes. Y creo que para los mortales es mucho mejor solución de Déjà dup por ejemplo.

Ahora estoy con la versión testing porque he tenido problemas con las últimas versiones de librerías Qt y he tenido que huir hacia adelante. A ver como sigue en un futuro.

1 ^ | v • Responder • Compartir >

Dona para ayudarme con este proyecto



---

Lorenzo Carbonell  
Aviso legal

[YouTube](#)

[iVoox](#)

[Spotify](#)

[Apple podcast](#)

[Google podcast](#)

[Telegram](#)

[Twitter](#)

[Mastodon](#)

[GitHub](#)

[Facebook](#)

[Suscribir](#)

[Contactar](#)

[Quien soy](#)

---