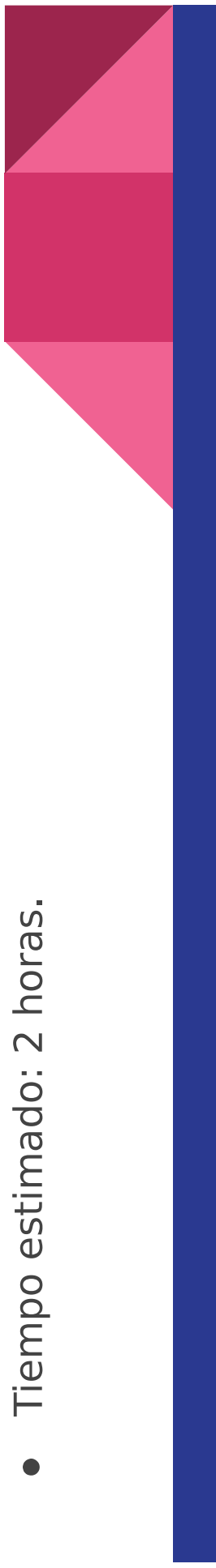




Aplicaciones y actualización

Aplicaciones y actualización. Objetivos.

- Instalación y actualización de aplicaciones.
- Actualización del sistema Ubuntu y Debian.
- Instalación con DEB. apt-get, aptitude, apt-cache, dpkg.
- Estructura interna de un fichero .deb.
- Conversor de paquetes alien.
- Tiempo estimado: 2 horas.



Aplicaciones y actualización. Conceptos básicos.

- En las distribuciones de GNU/Linux el software se instala mediante **paquetes**
 - Unidades de software para instalar aplicaciones
 - .DEB .RPM .TAR.GZ ... (**binarios** o **código fuente**)
 - Tienen dependencias entre ellos
 - En esta sección nos centramos en paquetes **binarios** .DEB
- Gestor de paquetes
 - Instalación, desinstalación, resolución de dependencias automáticamente
- Tipos de gestores de paquetes
 - Línea de comandos: **apt**, **aptitude**
 - Visuales: **Synaptic** (suelen basarse en los de línea de comandos)



Aplicaciones y actualización. Conceptos básicos.

- Veremos los paquetes .DEB:
 - Usados en la distribución Debian
 - Todas las distribuciones derivadas (Ubuntu) usan este tipo
- Existen herramientas visuales (Synaptic, GDebi)
 - En principio no se verán (solo línea de comandos)
 - Si queda tiempo, podemos verlas por encima



Aplicaciones y actualización. Gestor de paquetes: apt.

- Instalación:
 - \$ sudo apt-get install <paquete>
- Desinstalación:
 - \$ sudo apt-get remove <paquete>
- Purgar:
 - \$ sudo apt-get purge <paquete>
 - Purge elimina también la configuración de la aplicación
- Búsqueda de paquetes:
 - \$ sudo apt-cache search <texto_a_buscar>
- Info de paquete:
 - \$ sudo apt-cache policy <paquete>



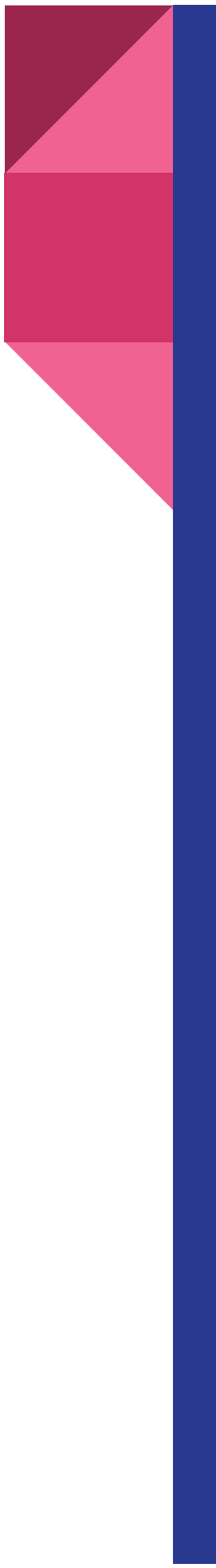
Aplicaciones y actualización. Gestor de paquetes: apt.

- Mini ejercicio:
 - Apt:
 - i. Comprobar si están instalados aptitude y geany
 - ii. Instalar con apt
 - iii. Comprobar que se han instalado
 - iv. Desinstalar geany
 - Aptitude:
 - i. Comprobar si están instalados apache y tomcat
 - ii. Instalar con aptitude (autocompletar)
 - iii. Comprobar que se han instalado
 - iv. Purgar tomcat



Aplicaciones y actualización. Gestor de paquetes: apt.

- Indicadores de estado:
 - p: nunca instalado/purgado
 - i: instalado
 - c: desinstalado
 - v: virtual
 - B: roto
 - u: desempaquetado
 - C: fallo en la configuración
 - H: fallo en la eliminación



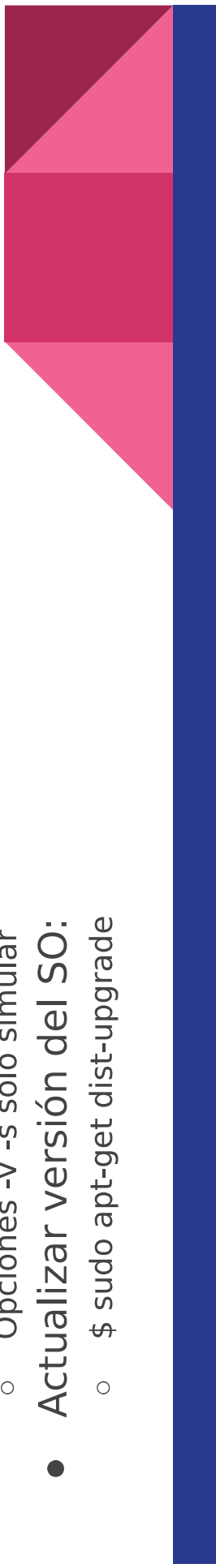
Aplicaciones y actualización. Gestor de paquetes: apt.

- `sudo aptitude -d install <paquete>` (solo descarga)
- `sudo aptitude -s install <paquete>` (simulación)
- `sudo aptitude versions <texto_a_buscar>`
 - `sudo apt-cache pkgnames <texto_a_buscar>`
- `sudo aptitude` (modo interactivo, para salir=q)
- `sudo aptitude search '~i' | less` (lista de instalados)



Aplicaciones y actualización. Actualizar y versiones de distro (Ubuntu).

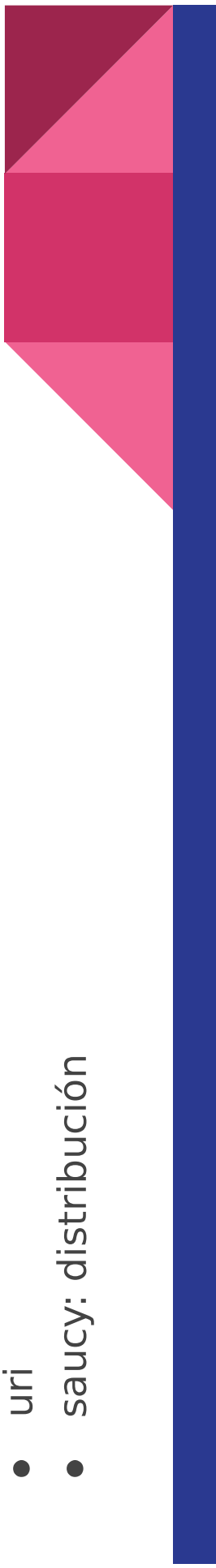
- `/etc/apt/sources.list`
 - Lista de repositorios (orígenes del software)
 - Por defecto se encuentran los repositorios de nuestra versión del SO
 - Se pueden incluir repositorios de otras versiones, u otro software
- Actualizar listado de paquetes:
 - `$ sudo apt-get update (consulta sources.list)`
- Actualizar software instalado:
 - `$ sudo apt-get upgrade`
 - Opciones `-V -s` solo simular
- Actualizar versión del SO:
 - `$ sudo apt-get dist-upgrade`



Aplicaciones y actualización. Actualizar y versiones de distro (Ubuntu).

- ```
sources.list
#deb cdrom:[Ubuntu 13.10 _Saucy Salamander_ - Release i386 (20131016.1)]/ saucy main restricted

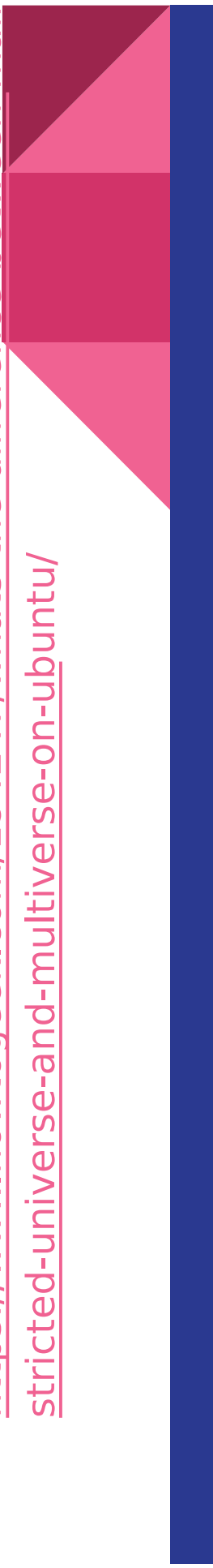
See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
newer versions of the distribution.
deb http://ch.archive.ubuntu.com/ubuntu/ saucy main restricted
deb-src http://ch.archive.ubuntu.com/ubuntu/ saucy main restricted
...
deb: precompilado
deb-src: fuentes
uri
saucy: distribución
```



# Aplicaciones y actualización. Actualizar y versiones de distro (Ubuntu).

- main: oficial de canonical, software libre abierto.
- restricted: oficial de canonical, software libre cerrado (drivers).
- universe: comunidad, software libre abierto.
- multiverse: no libre.

- <https://www.howtogeek.com/194247/whats-the-difference-between-main-restricted-universe-and-multiverse-on-ubuntu/>



# Aplicaciones y actualización. Actualizar y versiones de distro (Ubuntu).

- Versiones anteriores de software:

```
EOL upgrade sources.list
Required
deb http://old-releases.ubuntu.com/ubuntu/ CODENAME main restricted universe multiverse
deb http://old-releases.ubuntu.com/ubuntu/ CODENAME-updates main restricted universe multiverse
deb http://old-releases.ubuntu.com/ubuntu/ CODENAME-security main restricted universe multiverse
Optional
#deb http://old-releases.ubuntu.com/ubuntu/ CODENAME-backports main restricted universe multiverse
```

- Práctica no recomendable
  - Solo para software sin dependencias



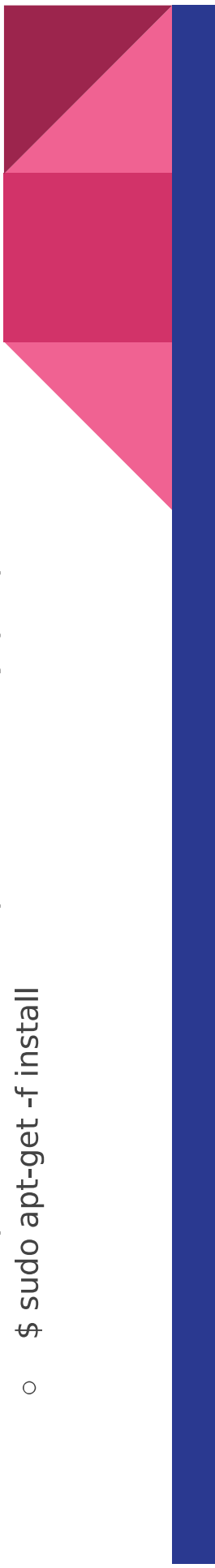
# Aplicaciones y actualización. Personal Package Archives (PPA).

- Sistema de paquetes para desarrolladores
- Los paquetes de sus desarrollos se depositan en repositorios PPA
- Añadir un repositorio PPA (Oracle Java por ejemplo):
  - \$ sudo add-apt-repository ppa:webupd8team/java && sudo apt-get update
  - Esto permitirá software actualizado directamente de los desarrolladores
- Eliminar un repositorio PPA:
  - \$ sudo ppa-purge ppa:webupd8team/java
  - En ocasiones el repositorio falla por quedarse obsoleto
- Ejemplo instalación wine



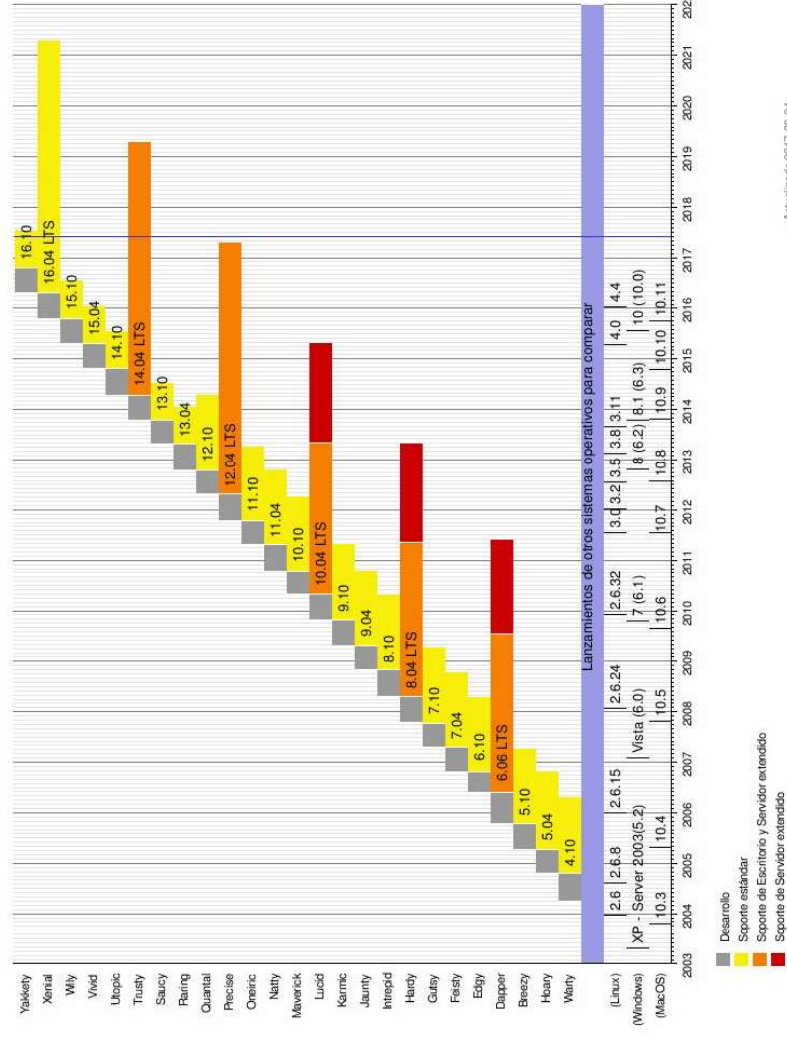
# Aplicaciones y actualización. Problemas al instalar o actualizar (Ubuntu).

- Paquetes rotos, fuentes no encontradas, paquetes antiguos, etc.
- Verificar la correcta resolución de dependencias:
  - `$ sudo apt-get check`
- Limpieza de paquetes antiguos:
  - `$ sudo apt-get autoclean` (clean borra todos)
- Eliminar paquetes huérfanos (dependencias de un paquete que ya se borró):
  - `$ sudo apt-get autoremove`
- Resolver problemas de dependencias y paquetes rotos:
  - `$ sudo apt-get -f install`



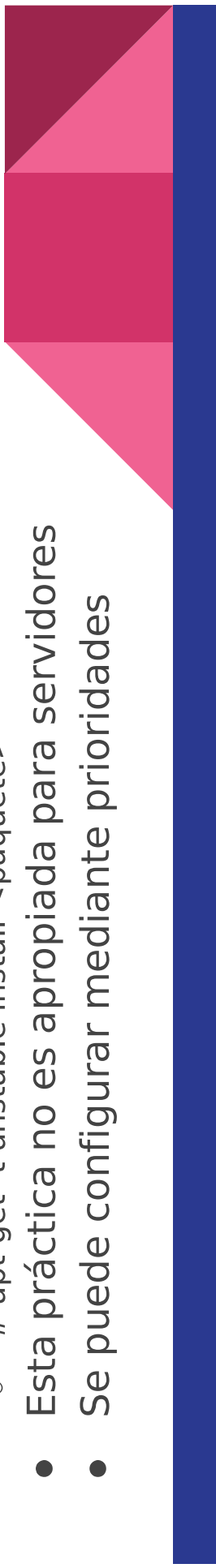
# Aplicaciones y actualización. Versiones Ubuntu.

Línea de tiempo de Ubuntu



# Aplicaciones y actualización. Actualización y versiones de distro (Debian).

- Ramas de desarrollo:
  - **stable** **testing** **unstable** **experimental**
  - Hacia “Stable”: software para uso, versiones poco recientes pero seguras.
  - Hacia “Unstable”: software en prueba, posibles errores, versiones más recientes.
  - Experimental: software todavía en desarrollo.
- Ejemplo para instalar un paquete en unstable:
  - (sources.list) deb <http://httpredir.debian.org/debian/unstable/main>
  - # apt-get update
  - # apt-get -t unstable install <paquete>
- Esta práctica no es apropiada para servidores
- Se puede configurar mediante prioridades





# Aplicaciones y actualización. Actualización y versiones de distro (Debian)

- Apt pinning:
  - Establecer prioridades de ramas en grupos de paquetes.
- Ejemplos:
  - `/etc/apt/preferences`
  - `Package: *`
    - Pin: release a=testing
    - Pin-Priority: 900
  - `Package: dpatch`
    - Pin: release o=Debian,a=experimental
    - Pin-Priority: 450
  - Otros ejemplos <https://wiki.debian.org/AptPreferences>



# Aplicaciones y actualización. Actualización y versiones de distro (Debian)

**Prioridad:  $P > 1000$ :** El paquete si instala aunque sea más antiguo que el paquete que tenemos instalado. (De este modo teóricamente es posible cambiar incluso de rama en Debian)

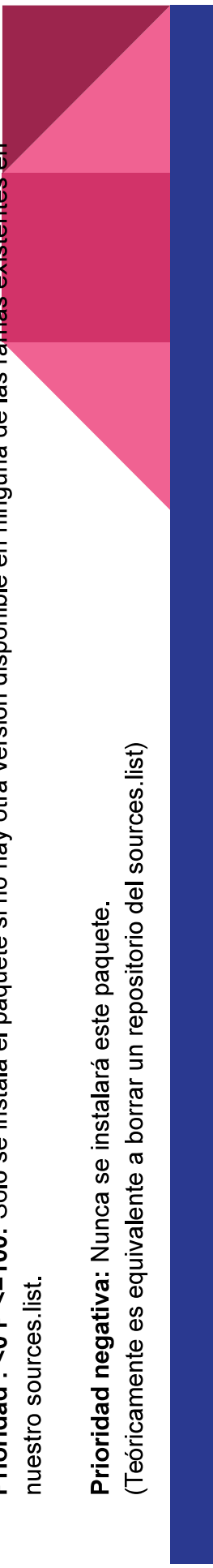
**Prioridad:  $< 990 \leq P \leq 1000$ :** El paquete se instala aunque no provenga de la rama principal. Si existe una versión más reciente del paquete previamente instalado el paquete no se instalará.

**Prioridad :  $< 500 \leq P \leq 990$ :** El paquete se instala siempre y cuando no exista una versión en la rama principal o tengamos una versión instalada de este paquete que sea más actual.

**Prioridad :  $< 100 \leq P \leq 500$ :** El paquete se instala siempre y cuando no exista el mismo paquete en cualquiera de las otras ramas. Tampoco se instala en caso de tener una versión del paquete instalada que sea más reciente.

**Prioridad :  $< 0 \leq P \leq 100$ :** Solo se instala el paquete si no hay otra versión disponible en ninguna de las ramas existentes en nuestro sources.list.

**Prioridad negativa:** Nunca se instalará este paquete.  
(Teóricamente es equivalente a borrar un repositorio del sources.list)



# Aplicaciones y actualización. Actualización y versiones de distro (Debian)

- Apt pinning, más ejemplos <https://geekland.eu/apt-pinning-en-debian/>
- sudo gedit /etc/apt/apt.conf

```
APT::Default-Release "testing";
APT::Cache-Limit 55000000;
Apt::Get::Purge;
APT::Clean-Installed;
APT::Get::Fix-Broken;
APT::Get::Fix-Missing;
APT::Get::Show-Upgraded "true";
```



# Aplicaciones y actualización. Actualización y versiones de distro (Debian)

- Apt pinning, más ejemplos <https://geekland.eu/apt-pinning-en-debian/>

- sudo gedit /etc/apt/preferences

Package: \*

Pin: release a=testing

Pin-Priority: 900

Package: \*

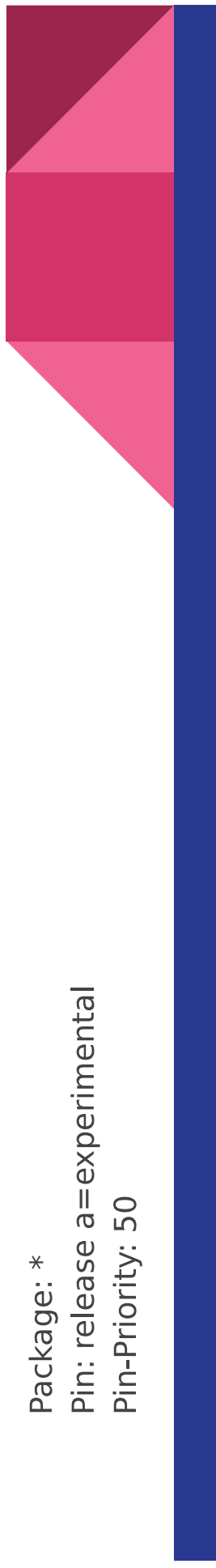
Pin: release a=unstable

Pin-Priority: 600

Package: \*

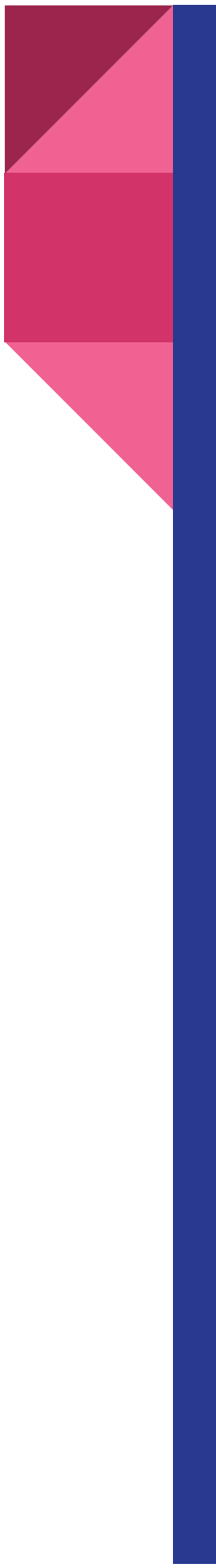
Pin: release a=experimental

Pin-Priority: 50



# Aplicaciones y actualización. Paquete .DEB.

- Formato de Debian y derivados (Ubuntu, Mint, ...).
- Es binario, por tanto previamente compilado
- Estructura interna:
  - **debian-binary**: número de versión del formato deb. Este es "2.0" para las versiones actuales de Debian.
  - **control.tar.gz**: toda la meta-información del paquete.
  - **data.tar.<\*>**: - los archivos a instalar.
- Comandos para extraer el contenido:
  - \$ sudo dpkg -e chrome.deb chromedeb (control)
  - \$ sudo dpkg -x chrome.deb chromedeb (archivos)



# Aplicaciones y actualización. Paquete .DEB: dpkg.

- Sirve para instalar paquetes .DEB individualmente:
  - `$ sudo dpkg -i paquete.deb`
- El problema es que no resuelve dependencias
  - Pero podemos ayudarnos de apt: `sudo apt-get install -f`
- Listar los paquetes instalados:
  - `$ sudo dpkg -l [<nombre_paquete>]`
  - `$ sudo dpkg -l | grep <nombre_aproximado>`
- Desinstalar paquete:
  - `$ dpkg -r <nombre_en_listado>` (mantiene configuración)
  - `$ dpkg -p <nombre_en_listado>` (borra configuración)
- Ejemplo instalación mendeley



# Aplicaciones y actualización. Paquete .DEB: dpkg.

- Información de paquete:
  - `$ sudo dpkg -s paquete.deb`
- Ficheros contenidos en un paquete:
  - `$ sudo dpkg --contents package.deb`
- Lista de ficheros/directorios instalados en el sistema:
  - `$ sudo dpkg -L <nombre_paquete>`
- A qué paquete(s) pertenece un fichero/directorio:
  - `$ sudo dpkg -S /usr/lib/tar` (pertenece al paquete tar)



# Aplicaciones y actualización. Paquete .DEB: dpkg.

- Retener paquetes:
  - En ocasiones no queremos que se actualice un programa, por motivos de seguridad, configuración o estabilidad.
  - Con dpkg se puede
- Retener:
  - `sudo echo geany hold | sudo dpkg --set-selections`
- Comprobar:
  - `sudo dpkg --get-selections geany`
- Quitar retención:
  - `echo geany install | sudo dpkg --set-selections`
- Con Synaptic no funciona, tiene su propio método





# Aplicaciones y actualización. Conversión de paquetes: Alien

- Convierte entre diferentes formatos (por ejemplo .RPM -> .DEB)
  - `sudo alien -k paquete.rpm`
  - `sudo alien -r paquete.deb`
- Después se instalaría con `dpkg`
- En ocasiones, pueden surgir problemas al instalarlo
- Ejemplo mendeley en rpm

