

Dotfiles

Restaura tu ambiente de trabajo en minutos

Santos Gallegos - @stsewd

Si funciona, no lo toques!





STOP RE-INSTALLING YOUR OS



- YOU KNOW WHAT? I'M GONNA
RE-INSTALL IT EVEN HARDER.

\$ whoami

- Web developer 
- Ethical hacker
- @stsewd - <https://stsewd.dev>
- Core developer



Read the Docs

Mi primera distro



Mi distro actual



Mi distro actual



Definición

dot files

- archivos

~/.bashrc
.gitignore

¿Dónde están los dotfiles?

\$HOME

- `~/.bashrc`
- `~/.vimrc`
- `~/.ssh/config`

Xdg specification (`~/.config/`)

- `~/.config/nvim/init.lua`
- `~/.config/kitty/kitty.conf`

<https://specifications.freedesktop.org/basedir-spec/latest/>

Dotfiles

- Archivos de texto plano
- Legibles y editables por humanos (*)
- Fáciles de versionar usando Git



Aplicando las configuraciones

```
$ git clone https://github.com/usuario/dotfiles ~/dotfiles  
$ ls -a ~/dotfiles  
.bashrc
```

Aplicando las configuraciones

```
$ cp ~/dotfiles/.bashrc ~/ .bashrc
```



Aplicando las configuraciones

```
$ ln -s ~/dotfiles/.bashrc ~/ .bashrc
```



Automatizando el proceso

Automatizando el proceso

symlinks.sh

```
#!/bin/bash  
  
ln -s ~/dotfiles/.bashrc ~/.bashrc
```

Configurando Git

- `~/.gitconfig`
- `~/.config/git/config`

Organizando los archivos en el repositorio

- .config/git/config
- config/git/config
- gitconfig

Organizando los archivos en el repositorio

- bashrc
- config/git/config
- symlinks.sh

Organizando los archivos en el repositorio

symlinks.sh

```
#!/bin/bash  
  
ln -s ~/dotfiles/bashrc ~/.bashrc  
ln -s ~/dotfiles/config/git/ ~/.config/
```

Restaurando nuestro sistema

```
$ git clone https://github.com/usuario/dotfiles.git  
$ cd ~/dotfiles  
$ ./symlinks.sh
```

symlinks.sh

- No funciona si el script es ejecutado múltiples veces
- Estamos asumiendo que el directorio `~/.config` ya existe
- Estamos asumiendo que el repositorio está en `~/dotfiles`

symlinks.sh

```
#!/bin/bash

ln -sf `pwd`/bashrc ~/ .bashrc
mkdir -p ~/ .config
ln -sf `pwd`/config/git/ ~/ .config/
```

Instalación de programas

Tenemos nuestras configuraciones y un script para aplicarlas. Para que estas configuraciones tengan efecto, necesitamos las aplicaciones que las usen.



Flatpak



install.sh

```
#!/bin/bash  
  
sudo apt update  
  
sudo apt install -y git curl vim
```

Makefile

- Build system (c, c++)
- make
- Recetas
- Targets
- Dependencias

Makefile syntax

dependencia:

```
@echo Creando directorio  
mkdir -p ~/directorio
```

target: dependencia

```
@echo Creando archivo  
echo "UbuCon Latam!" > ~/directorio/file.txt
```

Ejecutando un Makefile

- make
- make -f Makefile
- make [nombre del target] (make target)

Makefile

```
install:
```

```
    sudo apt update
```

```
    sudo apt install -y git curl vim
```

```
symlinks:
```

```
    ln -sf `pwd`/bashrc ~/.bashrc
```

```
    mkdir -p ~/.config
```

```
    ln -sf `pwd`/config/git/ ~/.config/
```

```
all: install symlinks
```

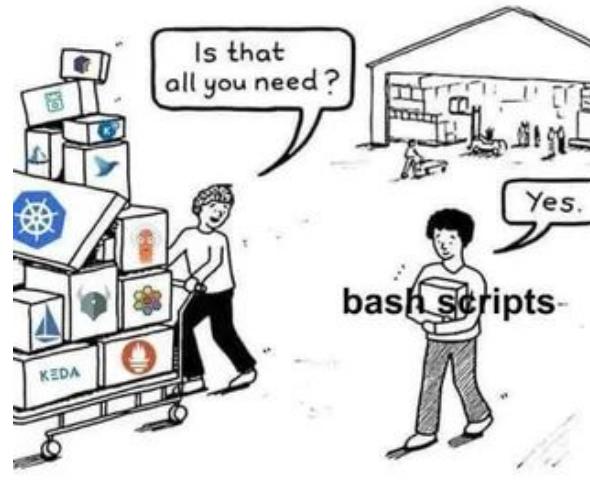
```
.PHONY: install symlinks all
```

Recomendaciones

- Evitar ejecutar el script/Makefile con sudo
- No pongas información sensible en tus dotfiles (contraseñas, tokens)
- Asegurar que tu script se pueda ejecutar varias veces
- Agrupa los scripts/targets
- Imprime el progreso
- Evitar en lo posible la interacción con el usuario
- Está bien no automatizar todo, pero es importante al menos documentar el proceso

Alternativas (overkilling?)

- GNU Stow
- Nix
- Terraform



Aprende y comparte

- Busca en GitHub por el topic dotfiles: <https://github.com/topics/dotfiles>
- Toma inspiración
- Hazlos tuyos
- Publica tus dotfiles

Mis dotfiles

<https://github.com/stsewd/dotfiles>



Aprende y comparte tus dotfiles



<https://stsewd.dev/charla-dotfiles>

Santos Gallegos - @stsewd

