

# ZSH

- [ZSH install](#)
- <http://zsh.sourceforge.net/>
- introducción:
  - <https://linuxconfig.org/learn-the-basics-of-the-zsh-shell>
  - <http://reasoniamhere.com/2014/01/11/outrageously-useful-tips-to-master-your-z-shell/>

## + características

- [ZSH features](#)
- [oh-my-zsh](#)
- [ZSH TAB Expansion](#)
- [ZSH Globbing \(extended\)](#)
- [ZSH expansion modifiers](#)
- [ZSH Parameter Expansion Flags](#)
- [ZSH zmv](#)
- [ZSH modules](#)
- [ZSH prompt](#)

## curiosidades

- **src**: recarga entorno (plugin)
- **take** : función crea directorio(s) y te lleva dentro
- **zsh\_stats** : función de comandos más usados
- **kill <caracteres><TAB>** : muestra los procesos que empiezan por <caracteres> para seleccionar el proceso a matar
- **expandir ayudas** : puedes ver todas las opciones de un comando tecleando el comando y el parámetro, por ejemplo: `ls -B<TAB>`
- **dirs [-v]**: muestra últimos directorios donde hemos estado
  - con `~#` podemos saltar a cualquiera de ellos
- **fc** permite editar en nuestro editor favorito el último comando, para hacer los arreglos necesarios
- **bindkey -v + plugin vi-mode**: trabajar en modo VIM en vez de modo EMACS
  - <https://dougblack.io/words/zsh-vi-mode.html>

```
• typeset -l name
  name="John Doe"
  echo $name # john doe
```

- **ESC,Q**: [push-line](#) → guarda la línea en curso en buffer, permite ejecutar otro comando y la reestablece
  - <https://unix.stackexchange.com/questions/74184/how-to-save-current-command-on-zsh>
  - [push-input](#): versión multilínea, sin keybinding
- ZSH Line Editor: `zle`, `bindkey`, `vared`
- **zshaddhistory**: controlar por función (hook) que comandos se guardan o no
  - <https://unix.stackexchange.com/questions/312528/zsh-history-comment-out-dangerous-commands>
  - <https://superuser.com/questions/352788/how-to-prevent-a-command-in-the-zshell-from-being-save-d-into-history>
  - <https://unix.stackexchange.com/questions/6094/is-there-any-way-to-keep-a-command-from-being-added-to-your-history>
  - <http://zsh.sourceforge.net/Doc/Release/Options.html#History>
  - <https://medium.com/@prasincs/hiding-secret-keys-from-shell-history-part-1-5875eb5556cc>
  - `export HISTORY_IGNORE=«(ls|cat|AWS|SECRET)»`

- afectaría a la sugerencia de comandos recientes mientras trabajas? → si

## peculiaridades

- read (input variable)

- `read -r "option?Quina opció vols" # retorna en option el resultat`

- `vared -p 'What would you like to do?: ' -c tmp`

- -q: espera resposta y/n

- -A: array? → <https://stackoverflow.com/questions/36453146/why-does-read-a-fail-in-zsh>

- ```
() {
  local compcontext='yn:yes or no:(y n)'
  vared -cp 'erasing all directories (y/n) ? ' ans
}
```

- bucle inline:

```
while true; do read -sk1 "?repetir (n surt): "; if [[ $REPLY != "n" ]]; then
echo "comanda a executar"; else break;fi; done
```

## mirar

- <https://code.joejag.com/2014/why-zsh.html> ✓
- <https://github.com/sorin-ionescu/prezto> ✓
- <http://grml.org/zsh/zsh-lovers.html> ✓
- [http://www.bash2zsh.com/zsh\\_refcard/refcard.pdf](http://www.bash2zsh.com/zsh_refcard/refcard.pdf) ✓

## cheatsheet

- zsh-refcard.pdf

/via [http://www.bash2zsh.com/zsh\\_refcard/refcard.pdf](http://www.bash2zsh.com/zsh_refcard/refcard.pdf)

## themes

- personalizar un theme : añadir los cambios a `*${SZH_CUSTOM}/themes*`
- personalizar agnoster : <https://github.com/agnoster/agnoster-zsh-theme/issues/65>
- pentesters: <https://deephacking.tech/oh-my-pentest-report-tema-de-oh-my-zsh-para-pentesters/>

## functions

- anonymous functions : <http://zsh.sourceforge.net/Doc/Release/Functions.html>

## tools

- shell state options: uso de % para ciertas funciones en el prompt
  - <http://zsh.sourceforge.net/Doc/Release/Prompt-Expansion.html>
  - [https://wiki.archlinux.org/index.php/Zsh\\_\(Espa%C3%B1ol\)](https://wiki.archlinux.org/index.php/Zsh_(Espa%C3%B1ol))

From:

<https://miguelangel.torresegea.es/wiki/> - **miguel angel torres egea**

Permanent link:

<https://miguelangel.torresegea.es/wiki/linux:zsh:start>

Last update: **22/09/2025 08:36**

