

# Instalación de zsh

[ATRÁS](#)   [TERMINAL](#)   [INSTALACIÓN DE ZSH](#)

Como mencionamos en un apartado anterior, el **intérprete de comandos** ( *también llamado **shell*** ) es el software o programa que gestiona los comandos que introduce el usuario, para intentar comprenderlos lo mejor posible y comunicarselos a la máquina o sistema de la mejor forma posible.

En GNU/Linux, tenemos multitud de intérpretes de comandos, siendo quizás los más conocidos **bash** y **zsh**. Sin embargo, existen multitud de ellos:

Nombre	Año	Abreviatura	Descripción
Bourne Shell	1977	<b>sh</b>	Intérprete ligero y sencillo, considerado el estándar.
C Shell	1978	<b>cs</b> <b>h</b>	Intérprete basado en el lenguaje de programación C.
Korn Shell	1983	<b>ksh</b>	Intérprete de AT&T de Bourne Shell.
Almquist Shell	1989	<b>ash</b>	Intérprete ligero y rápido, popularizado en máquinas de bajo coste.
<a href="#">Bourne Again Shell</a>	1989	<b>bash</b>	Popular intérprete de la mayoría de los GNU/Linux.
Z Shell	1990	<b>zsh</b>	Intérprete shell mejorado, con añadidos de otras shells.
<a href="#">Fish Shell</a>	2005	<b>f</b> <b>ish</b>	Intérprete enfocado en la sencillez y una interfaz agradable.
<a href="#">Ion Shell</a>	2015	<b>ion</b>	Shell del sistema operativo <a href="#">Redox</a>

En este artículo veremos como instalar y configurar **zsh** para tener una terminal moderna, atractiva y con funcionalidades interesantes como autocompletado de comandos basado en el historial, resaltado y detección de comandos existentes o completado avanzado de comandos.

## Instalar Z Shell (zsh) ⚓

El intérprete **Z Shell** ( *más conocido como **zsh*** ) es uno de los shells más conocidos de GNU/Linux, adoptado hace poco por Apple como su intérprete de terminal por defecto desde 2019. Hoy en día, quizás el más extendido es **bash**, sin embargo, **zsh** gana cada día más adeptos debido a su capacidad y potencia de personalización y su activa comunidad. Es una excelente **alternativa a bash**.



Si tenemos un sistema recién instalado ( *o por defecto* ), lo más probable es que tengamos un **bash** o un **sh** ( *más simple* ). Lo primero que necesitamos tener instalado en nuestro sistema es el intérprete de **zsh**. Para instalarlo utilizaremos **apt**, que se encargará de buscar los paquetes necesarios y sus dependencias:

```
sudo apt update && sudo apt install zsh
```

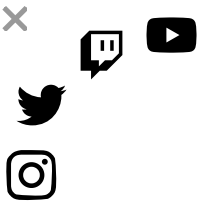
Con esto tendríamos el **intérprete** instalado, pero se trata de un «zsh pelado», sin configuración. De hecho, de momento ni siquiera se va a iniciar por si solo y seguiremos utilizando el **interprete** que teníamos antes. Si escribimos **zsh** podemos obligar al sistema a abrir **zsh** y ver si se ha instalado correctamente. Nos debería aparecer una pantalla similar a la siguiente:

Hablo de esto en...



Tabla de contenidos

- Instalar Z Shell (zsh)
- Utilizar zsh por defecto
- Framework Oh-My-Zsh
- Cambiar el tema de zsh
- Error 12: character not in range
- Autocompletado y sugerencias





```
zsh.newuser-install.  
You are seeing this message because you have no zsh startup files  
(the files .zshenv, .zprofile, .zshrc, .zlogin in the directory  
~). This function can help you with a few settings that should  
make your use of the shell easier.  
  
You can:  
  
(q)  Quit and do nothing.  The function will be run again next time.  
  
(0)  Exit, creating the file ~/.zshrc containing just a comment.  
     That will prevent this function being run again.  
  
(1)  Continue to the main menu.  
  
(2)  Populate your ~/.zshrc with the configuration recommended  
     by the system administrator and exit (you will need to edit  
     the file by hand, if so desired).  
  
--- Type one of the keys in parentheses ---
```

Esta es la pantalla de bienvenida de **zsh**, que nos da varias opciones para configurarlo manualmente, dejar su configuración vacía, entre otras. Marcamos la opción **0**, que simplemente creará nuestro fichero de configuración **~/.zshrc** para que no nos vuelva a salir este mensaje de bienvenida cada vez que arranquemos **zsh**.

Si todo ha ido bien, debería aparecernos algo parecido a esto ( *con el nombre de nuestro sistema* ):

```
DESKTOP-0HQ7ANH% _
```

Escribimos **exit** para salir de esta sesión y volver a la anterior ( *donde aún no usabamos **zsh*** ).

## Utilizar zsh por defecto

Habrás comprobado que si cierras la terminal y la vuelves a abrir, parece no haber cambiado nada y no inicia **zsh**. Esto ocurre porque aunque está instalada en el sistema, aún estás usando otro intérprete de comandos ( *probablemente, el anterior que tenías* ). Para saber que intérprete de comandos o **shell** estamos utilizando, puedes escribir **echo \$0** o **echo \$SHELL**, el cuál debería mostrarte cuál estas utilizando en el momento actual.

Por otro lado, escribiendo **cat /etc/shells** podemos ver que intérpretes tenemos instalados en nuestro sistema y podemos utilizar. Si queremos cambiar el shell con el que inicia un usuario ( *por ejemplo, manz* ), solo tenemos que utilizar el comando **chsh** ( *change shell* ):

```
chsh manz -s /bin/zsh
```

Perfecto. Ahora cuando volvamos a abrir una terminal, nos arrancará la shell **zsh**. Ahora que ya lo tenemos todo listo, toca configurarlo para que haga **magia**.

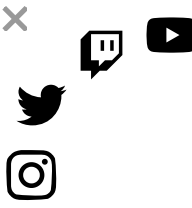
## Framework Oh-My-Zsh

Uno de los frameworks más populares de **zsh** es [Oh-My-Zsh!](#) , el cuál podremos instalar para dotar a nuestro sistema de funcionalidades muy interesantes que nos harán más cómodo su uso. Para instalarlo necesitaremos **curl** y **git**, por lo que instalamos dichos requisitos previamente:

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

Una vez los tengamos instalados, escribimos el comando siguiente, el cuál descargará el script oficial de instalación de **Oh-my-zsh!**:

```
$ sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"  
  
Cloning Oh My Zsh...  
Cloning into '/home/manz/.oh-my-zsh'...
```



Es posible que al finalizar la instalación nos pregunte si queremos sobrescribir la configuración de `~/ .zshrc` o si queremos establecer **zsh** como shell por defecto, si corresponde. Una vez terminado el proceso, debería aparecernos algo como lo siguiente:

Podemos ver que el **prompt** de la shell ya ha cambiado a algo más bonito y amigable, haciendo uso de colores. Sin embargo, puede resultarnos demasiado simple. Veamos como podemos cambiarlo por otro **theme**.

## Cambiar el tema de zsh

Una de las características más llamativas de **zsh** es la posibilidad de cambiar el **theme** de la terminal, a uno más vistoso y llamativo. La shell **zsh** incorpora de serie varios [themes para la terminal](#) que podemos utilizar según nuestros gustos.

Por ejemplo, uno de mis preferidos es **agnoster**. Para seleccionar este tema, simplemente tenemos que editar el archivo de configuración `~/.zshrc` y buscar la línea siguiente:

```
ZSH_THEME="robbyrussell"
```

El tema [robbyrussell](#) es el que viene por defecto, pero podemos cambiarlo por [agnoster](#) o por cualquier otro de la lista de [themes de zsh](#) . De hecho, incluso podemos indicar [random](#) como tema, y descomentar una línea que aparece más abajo, que nos activa un tema aleatorio de esa lista, de forma aleatoria cada vez que abramos una terminal:

```
ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" "xiong-chiamiov" )
```

Si todo ha ido bien al cambiar el tema por **agnoster**, al guardar el fichero de configuración y abrir una nueva terminal ya debería aparecernos la nueva interfaz del intérprete de comandos:

manz@DESKTOP-0IQD6JV /mnt/c/Users/Manz

Ten en cuenta que es posible que algunos caracteres se nos muestren incorrectamente ( *como cuadrados o similar*). Esto suele ocurrir cuando usamos una tipografía que no es capaz de mostrar los caracteres especiales que necesitamos. Para solucionarlo, te recomendamos instalar y utilizar una [tipografía Powerline](#) .



## Error 12: character not in range

Es muy común, que instalando los plugins, gestionando los themes y personalizando los apartados anteriores, te puedas encontrar de repente con que al iniciar la terminal, nos aparece un error similar al siguiente:






```
(anon):12: character not in range
```


Este problema ocurre por conflictos en la lectura de caracteres Unicode, ya que es posible que esté intentando leer alguno que no exista en el conjunto de caracteres instalados ( *es muy común en Docker, donde no tienes los locales preinstalados* ). Para resolverlo, simplemente instala los paquetes de localización escribiendo **sudo apt install locales-all** y reinicia la sesión de usuario.

## Autocompletado y sugerencias

Existen una serie de plugins creados por la comunidad de **zsh** muy interesantes, que podemos instalar en nuestra terminal para ampliar sus capacidades y hacerla mucho más cómoda.

Plugin	Descripción
<a href="#">zsh-syntax-highlighting</a>	Resaltado de sintaxis para comandos, basado en <b>fish</b> .
<a href="#">zsh-autosuggestions</a>	Sugerencias basadas en el historial, basado en <b>fish</b> .
<a href="#">zsh-completions</a>	Autocompletado mejorado para comandos.

El primero de ellos, **zsh-syntax-highlighting**, es el **resaltado de color de comandos**, el cuál viene de una característica que incorporaba inicialmente el intérprete de comandos **fish**. Al escribir un comando en la terminal nos aparecerá en **rojo** si no está instalado en el sistema o en **verde** si está disponible para utilizar. De esta forma sabremos, a medida que vamos escribiendo los comandos, si los estamos escribiendo bien ( *o por lo contrario, tenemos erratas* ) o si no se encuentra instalado un comando, por ejemplo, incluso antes de pulsar la tecla .

El segundo, **zsh-autosuggestions**, es un plugin que a medida que escribimos un comando, nos sugiere y recuerda comandos escritos anteriormente ( *que residen en el historial de la terminal* ) en color gris de fondo, para que sea más sencillo para nosotros identificarlos. Pulsando  le decimos a zsh que lo recupere íntegro, lo que puede ser útil para comandos largos.



Por último, el plugin **zsh-completions** nos permite activar ciertas características interesantes de autocompletado en **zsh**, como por ejemplo atajos al pulsar  en **git**, **composer** de PHP, **nano**, el firewall **ufw**, **node** y muchos otros.

Para instalar estos plugins, lo que tendremos que hacer es simplemente copiar y pegar los siguientes comandos. En ellos, lo que estamos haciendo es descargarnos ( *clonar el repositorio del plugin de git* ) y guardándolo en nuestra carpeta personal, en un apartado de plugins personalizados de oh-my-zsh. Podemos hacerlo todo con estos tres comandos:

```
git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-autosuggestions
git clone https://github.com/zsh-users/zsh-syntax-highlighting.git ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-syntax-highlighting
git clone https://github.com/zsh-users/zsh-completions ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-completions
```

Una vez hecho, editamos el fichero **~/.zshrc** y bajamos hasta la linea donde se cargan los plugins, generalmente con esta configuración **plugins=(git)**. Ampliamos esa lista de plugins, escribiendo lo siguiente:

```
plugins=(git zsh-autosuggestions zsh-syntax-highlighting zsh-completions)
autoload -U compinit && compinit
```

Con esto, cargaremos esos 4 plugins y tendremos listo estas funcionalidades adicionales desde que volvamos a abrir una nueva terminal.

Es posible que el plugin de autocompletado no aplica el color adecuado. Puedes solucionarlo editando el archivo **~/.zshrc** y añadiendo la linea **export ZSH\_AUTOSUGGEST\_HIGHLIGHT\_STYLE="fg=5"**. Esto permitirá cambiar el color del autocompletado desde el historial, donde el **5** es el color ( *en este caso **color morado*** ).





Docente, divulgador informático y [streamer de código](#) . Amante de **CSS**, **Javascript** y de la plataforma web. Ha trabajado como profesor en la Universidad de La Laguna y es director del curso de Programación web FullStack y FrontEnd de EOI desde Tenerife. En sus ratos libres, busca GIF de gatos en Internet.

2 comentarios

¡Accede a Discord y cuéntanos!

## Terminal de Linux

Documentación sobre la terminal de GNU/Linux y las mejores herramientas y comandos para ser más productivo.

Creado y mantenido por [@Manz](#) con ❤️

