

# Ejemplo de Configuración del Módulo CLI de Cisco IOS Shell

## Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Configurar](#)

[Tuberías y redirección](#)

[Applets integrados](#)

[Condiciones](#)

[Loops](#)

[Operaciones de cadena](#)

[Ejemplo de configuración](#)

[Verificación](#)

[Troubleshoot](#)

[Información Relacionada](#)

## Introducción

Este documento describe el módulo CLI de Cisco IOS<sup>®</sup> Shell (IOS.sh).

## Prerequisites

## Requirements

Cisco recomienda que tenga conocimiento de Cisco IOS Release 15.1(4)M, 15.1(2)S y versiones posteriores. Cisco IOS.sh se debe configurar y habilitar para utilizar las funciones y funciones de Cisco IOS.sh en su router.

## Componentes Utilizados

La información que contiene este documento se basa en estas versiones de software y hardware.

- Switches Cisco Catalyst serie 6500

- Cisco IOS versión 15.1(1)SY

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

## Antecedentes

Cisco IOS.sh es un módulo que le permite utilizar el script de shell en la CLI de Cisco IOS. Este módulo ayuda a automatizar las acciones rutinarias y simplifica la administración de dispositivos. Se puede utilizar junto con otras herramientas de administración, como Cisco IOS Embedded Event Manager (EEM).

## Configurar

**Nota:** Use la [Command Lookup Tool \(clientes registrados solamente\)](#) para obtener más información sobre los comandos usados en esta sección.

Para habilitar la CLI de Shell en el dispositivo, ingrese:

```
Switch#terminal shell
```

Para inhabilitar la CLI de Shell, ingrese:

```
Switch#terminal no shell
```

Esta función admite la variable PATH, que contiene una lista de directorios donde shell puede buscar funciones/applets. Además, la CLI de Shell admite estas características:

## Tuberías y redirección

```
Switch#show version | grep 15.1
```

```
Cisco IOS Software, s2t54 Software (s2t54-ADVENTERPRISEK9-M),  
Version 15.1(1)SY, RELEASE SOFTWARE (fc2)
```

```
Switch#show version | grep 15 > bootdisk:version.txt
```

```
Switch#dir bootdisk:
```

```
Directory of bootdisk:/
```

```
14 -rw- 280 Nov 15 2012 23:25:32 +00:00 version.txt
```

```
1024557056 bytes total (577273856 bytes free)
```

```
Switch#cat bootdisk:version.txt
```

```
Cisco IOS Software, s2t54 Software (s2t54-ADVENTERPRISEK9-M),  
Version 15.1(1)SY, RELEASE SOFTWARE (fc2)
```

## Applets integrados

Cisco IOS.sh contiene varios subprogramas integrados que ayudan a crear aplicaciones y filtros para las funciones de usuario:

```
[[          evaluate a logical test expression
cat        output data from a pipe or file to the terminal
cut        edit piped output
echo       echo arguments to the terminal
false      return false in while or if expressions, and set the result
fetch      return values from the configuration database
grep       search for regular expressions in piped output or files
head       print the first lines in the input
interface  print interfaces that match the argument
let        evaluate a numeric expression, and set the result
man        print information for built-ins
more       page piped output to the terminal
nl         number the lines in the input
null       ignore the input
printf     output formatted data to the terminal
read       read input into variables
set_oper   set operational values
sleep      pause execution of the terminal
sort       sort the input
tail       print the tail of the input
true       return true in while or if expressions, and set the result
uname      print system information
wc         count lines, words, and characters
```

## Condiciones

Estos son los comandos para el **if.....de otro modo**.

```
Switch#x=5
Switch#if [[ $x != 5 ]];
then.else.fi>then
then.else.fi>echo false;
then.else.fi>else
else..fi>echo true;
else..fi>fi;
true
Switch#
```

Estos operadores de comparación se utilizan para trabajar con valores enteros:

### Operadores Descripción

-eq	Arg1-eq Arg2. True si Arg1 es igual a Arg2
-ne	Arg1-ne Arg2. True si Arg1 no es igual a Arg2
-lt	Arg1-lt Arg2. True si Arg1 es menor que Arg2
-gt	Arg1-gt Arg2. True si Arg1 es mayor que Arg2
-ge	Arg1-ge Arg2. True si Arg1 es mayor o igual que Arg2
-le	Arg1-le Arg2. True si Arg1 es menor o igual que Arg2

Estas condiciones están disponibles para trabajar con archivos:

### Operadores Descripción

-a o -e	True si existe un archivo
-d	True si existe un archivo y es un directorio
-f	True si existe un archivo y es un archivo normal

- r True si existe un archivo y es legible
- s True si existe un archivo y tiene un tamaño mayor que cero
- w True si existe un archivo y es ejecutable
- nt Prueba si el archivo 1 es más reciente que el archivo 2. La fecha de modificación del archivo utiliza para esta comparación
- ot Prueba si el archivo 1 es anterior al archivo 2

Para utilizar varias condiciones dentro de un operador de comparación, puede utilizar estos operadores lógicos:

```
&& Logical AND
|| Logical OR
```

## Loops

Los loops son los mismos que en el shell de Linux. Hay dos operadores disponibles.

Estos son los comandos para el operador **do**:

```
Switch#for x in 11
do..done>do
do..done>ping 192.0.2.$x
do..done>done
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Switch#
```

Estos son los comandos para el operador **While**:

```
Switch#for i in 1 2 3
do..done>do
do..done>while [ $i -lt 3 ]
do..done>do
do..done>echo $i
do..done>done
do..done>i=i+1
do..done>done
Switch#
```

## Operaciones de cadena

Para las comparaciones de cadenas, el **if.....other** se puede utilizar con estas modificaciones:

Operadores Descripción

- == Cadena 1 == cadena2, True si las cadenas son iguales.
- != String1 != string2 , True si las cadenas no son iguales
- < String1 < string2, True si string1 es lexicográficamente menor que string2
- > String1 > string2, True si string1 es lexicográficamente mayor que string2

# Ejemplo de configuración

```
Switch#function add_desc() {
{..} >show running-config interface Gi$1
{..} >configuration terminal
{..} >interface Gi$1
{..} >description $2
{..} >end
{..} >show running-config interface Gi$1
{..} >}
Switch#add_desc 5/1 SHELL_TEST
Building configuration...
```

Current configuration : 74 bytes

```
!
interface GigabitEthernet5/1
ip address 192.0.2.10 255.255.255.0
end
```

Enter configuration commands, one per line. End with CNTL/Z.  
Building configuration...

Current configuration : 98 bytes

```
!
interface GigabitEthernet5/1
description SHELL_TEST
ip address 192.0.2.10 255.255.255.0
end
```

Switch#

## Verificación

Utilize esta sección para confirmar que su configuración funcione correctamente.

Ingrese el comando **show terminal** para mostrar la configuración de terminal actual:

```
Switch#show terminal
Line 2, Location: "", Type: "XTERM-COLOR"
Length: 44 lines, Width: 155 columns
Baud rate (TX/RX) is 9600/9600
Status: PSI Enabled, Ready, Active, No Exit Banner, Ctrl-c Enabled
.....
Preferred transport is lat.
Shell: enabled
Shell trace: off
```

Ingrese el comando **show shell Functions** para mostrar todas las funciones disponibles en el módulo, que incluyen funciones definidas por el usuario:

```
Switch#show shell functions
#User defined functions:

Function namespace: DEFAULT
function add_desc()
{
```

```
show running-config interface Gi$1
configure terminal
interface Gi$1
description $2
end
show running-config interface Gi$1
}
```

## Troubleshoot

Actualmente, no hay información específica de troubleshooting disponible para esta configuración.

## Información Relacionada

- [Guías de Configuración de Catalyst 6500](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)