

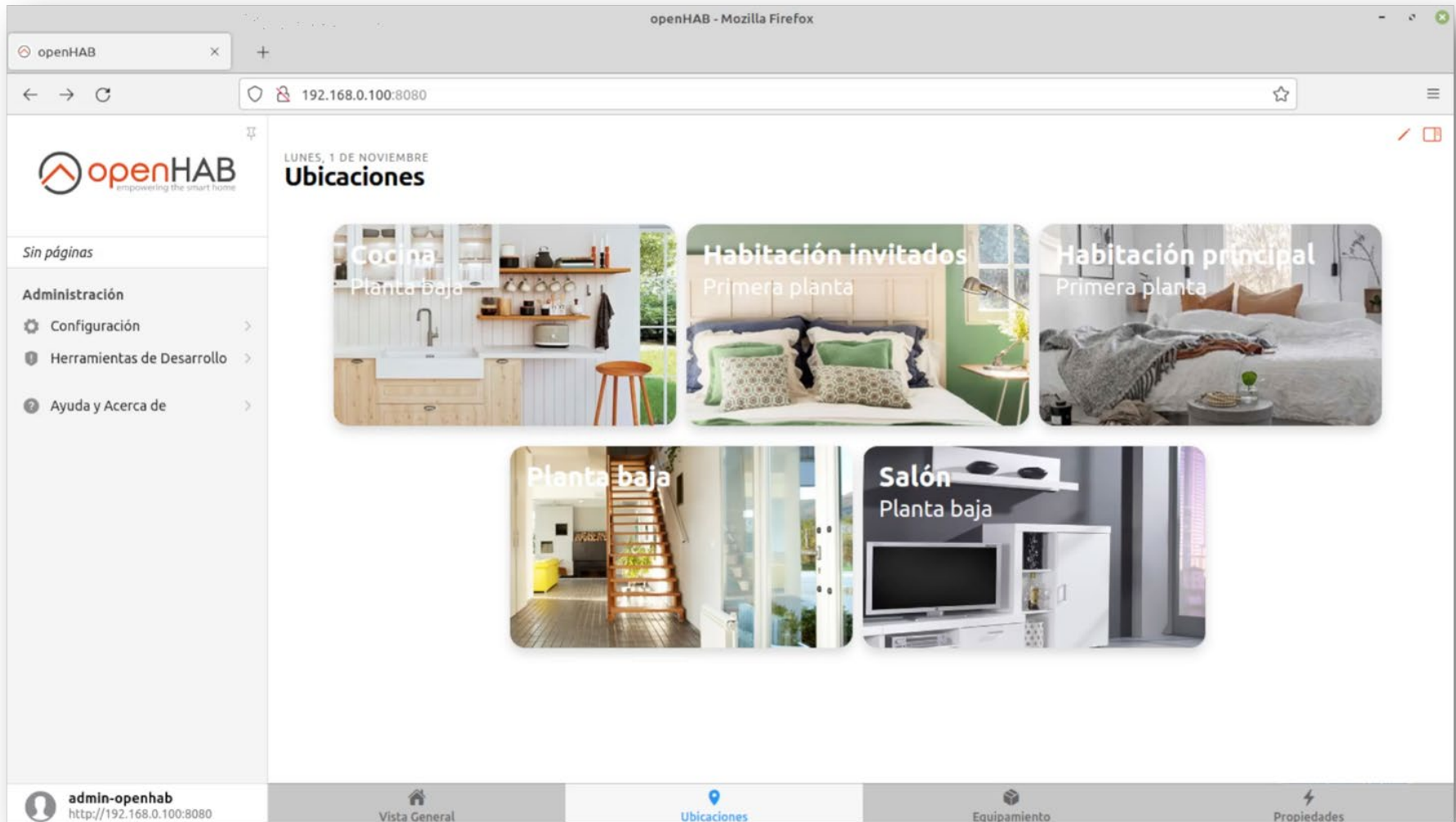
Universidad Politécnica de Madrid

Uso del IoT para construir tú mismo un hogar digital

Creación de tareas de automatización e inteligencia

Javier Malagón

# Controlar los dispositivos del hogar



openHAB - Mozilla Firefox

openHAB

192.168.0.100:8080

**openHAB**  
empowering the smart home

LUNES, 1 DE NOVIEMBRE  
**Ubicaciones**

Sin páginas

Administración

- Configuración
- Herramientas de Desarrollo
- Ayuda y Acerca de

Cocina  
Planta baja

Habitación invitados  
Primera planta

Habitación principal  
Primera planta

Planta baja

Salón  
Planta baja

admin-openhab  
http://192.168.0.100:8080

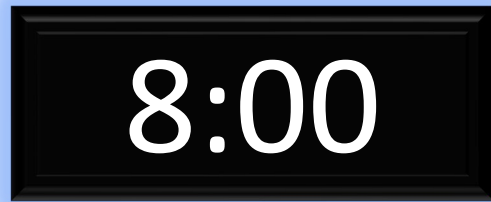
Vista General

Ubicaciones

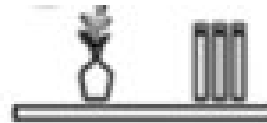
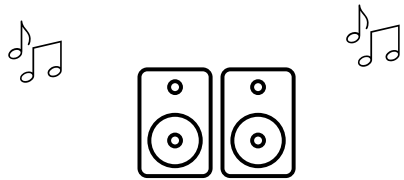
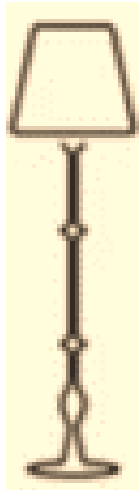
Equipamiento

Propiedades

# Ejemplos de automatización



# Ejemplos de automatización



# ¿Qué es una regla?

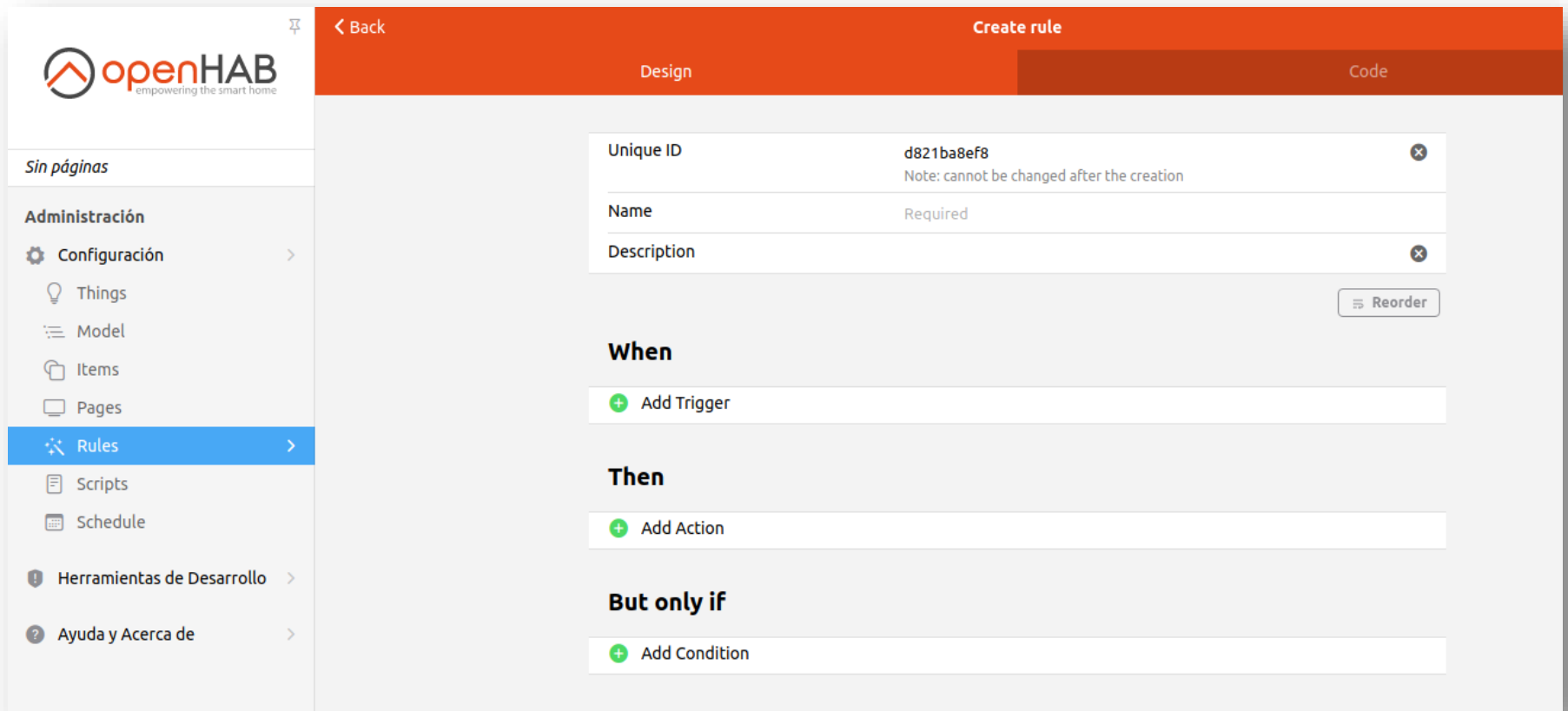
Cuando:

Evento 1,  
Evento 2,  
.....

Acciones:

Acción 1,  
Acción 2,  
.....

# Cómo se hace una regla



The screenshot shows the OpenHAB web interface. On the left is a sidebar with the OpenHAB logo and a navigation menu. The main content area is titled 'Create rule' and has two tabs: 'Design' (selected) and 'Code'. Under the 'Design' tab, there are three input fields: 'Unique ID' (with value 'd821ba8ef8' and a note 'Note: cannot be changed after the creation'), 'Name' (with value 'Required'), and 'Description'. A 'Reorder' button is located to the right of these fields. Below the input fields are three sections: 'When' with an 'Add Trigger' button, 'Then' with an 'Add Action' button, and 'But only if' with an 'Add Condition' button.

**openHAB**  
empowering the smart home

Sin páginas

Administración

- Configuración
- Things
- Model
- Items
- Pages
- Rules**
- Scripts
- Schedule

Herramientas de Desarrollo

Ayuda y Acerca de

< Back

Create rule

Design Code

Unique ID: d821ba8ef8  
Note: cannot be changed after the creation

Name: Required

Description

Reorder

**When**

+ Add Trigger

**Then**

+ Add Action

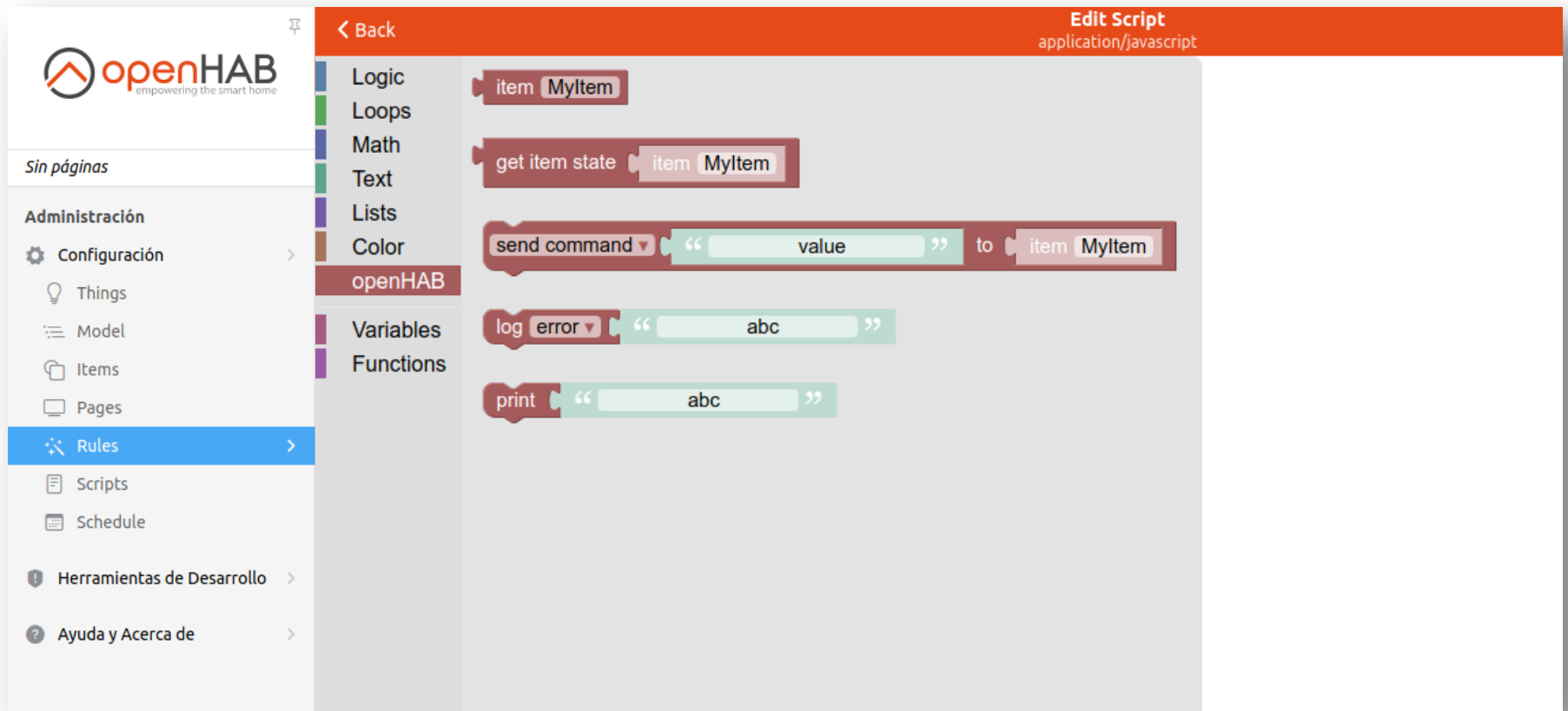
**But only if**

+ Add Condition



Item  
Action

# Cómo se hace una regla



The screenshot displays the openHAB web interface in the 'Edit Script' mode for an application/javascript rule. The left sidebar contains the navigation menu with 'Rules' selected. The main workspace shows a sequence of five script blocks:

- `item MyItem`
- `get item state` block with a sub-block `item MyItem`
- `send command` block with a sub-block `value` and a target `item MyItem`
- `log error` block with a sub-block `abc`
- `print` block with a sub-block `abc`

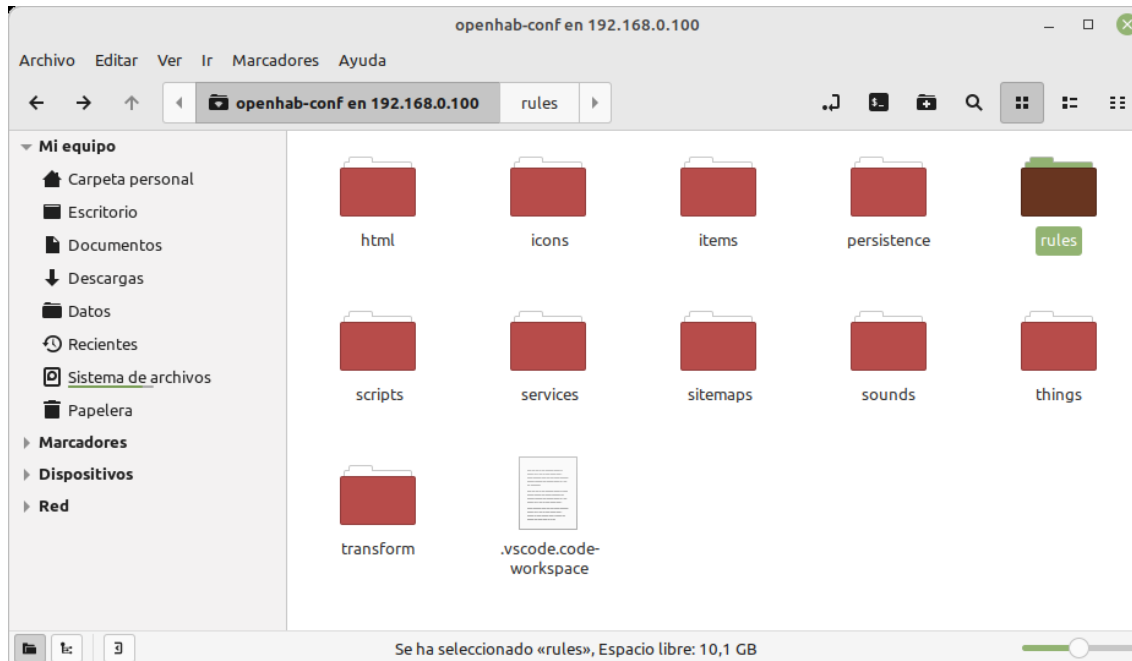
# Cómo se hace una regla



The screenshot displays the OpenHAB web interface. On the left is a navigation sidebar with the OpenHAB logo and a menu including 'Sin páginas', 'Administración', 'Configuración', 'Things', 'Model', 'Items', 'Pages', 'Rules' (highlighted), 'Scripts', 'Schedule', 'Herramientas de Desarrollo', and 'Ayuda y Acerca de'. The main area is titled 'Edit Script' and shows a rule script for 'application/vnd.openhab.dsl.rule'. The script is as follows:

```
1 Thread::sleep(100)
2 val power = triggeringItem as SwitchItem
3 switch power {
4     case MonoPrice_Z11_Power: zone = "1"
5     case MonoPrice_Z12_Power: zone = "2"
6     case MonoPrice_Z13_Power: zone = "3"
7     case MonoPrice_Z14_Power: zone = "4"
8     case MonoPrice_Z15_Power: zone = "5"
9     case MonoPrice_Z16_Power: zone = "6"
10 }
11 if(receivedCommand==ON){
12     AudioMTX.sendCommand("<1" + zone + "PR01\r")
13     logInfo("MonoPrice", "Zone " + zone + " Power ON")
14 }
15 else if(receivedCommand==OFF){
16     AudioMTX.sendCommand("<1" + zone + "PR00\r")
17     logInfo("MonoPrice", "Zone " + zone + " Power OFF")
18 }
19 AudioMTX.sendCommand("?1" + zone + "\r")
20 Thread::sleep(300)
21 MonoPrice_Status.postUpdate(AudioMTX.state.toString.trim)
```

# Cómo se hace una regla



```
9 rule "Alle Lichten aan/uit"
10
11 when
12   Item Lichten_aan_uit received update
13 then
14   if (Lichten_aan_uit.state == ON) {
15     sendCommand(FF_Bureau_Light, ON)
16     sendCommand(GF_Toilet_Light, ON)
17     sendCommand(GF_Berging_Light, ON)
18     sendCommand(GF_Kitchen_Light, ON)
19     sendCommand(G_TrapGarageKeuken_Light, ON)
20   }
21   if (Lichten_aan_uit.state == OFF) {
22     sendCommand(FF_Bureau_Light, OFF)
23     sendCommand(GF_Toilet_Light, OFF)
24     sendCommand(GF_Berging_Light, OFF)
25     sendCommand(GF_Kitchen_Light, OFF)
26     sendCommand(G_TrapGarageKeuken_Light, OFF)
27   }
28 end
```

# Creación de una regla sencilla

openHAB - Mozilla Firefox

openHAB

192.168.0.100:8080/settings/rules/66c9232c66

openHAB empowering the smart home

Sin páginas

Administración

- Configuración
- Things
- Model
- Items
- Pages
- Rules**
- Scripts
- Schedule

Herramientas de Desarrollo

Ayuda y Acerca de

admin-openhab  
http://192.168.0.100:8080

Levantarse

Design Code

Status: **IDLE**

Unique ID

Name

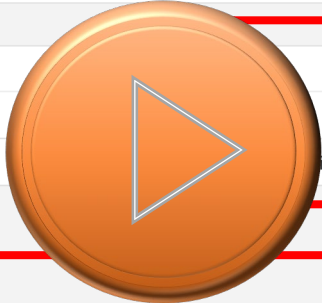
Description

When

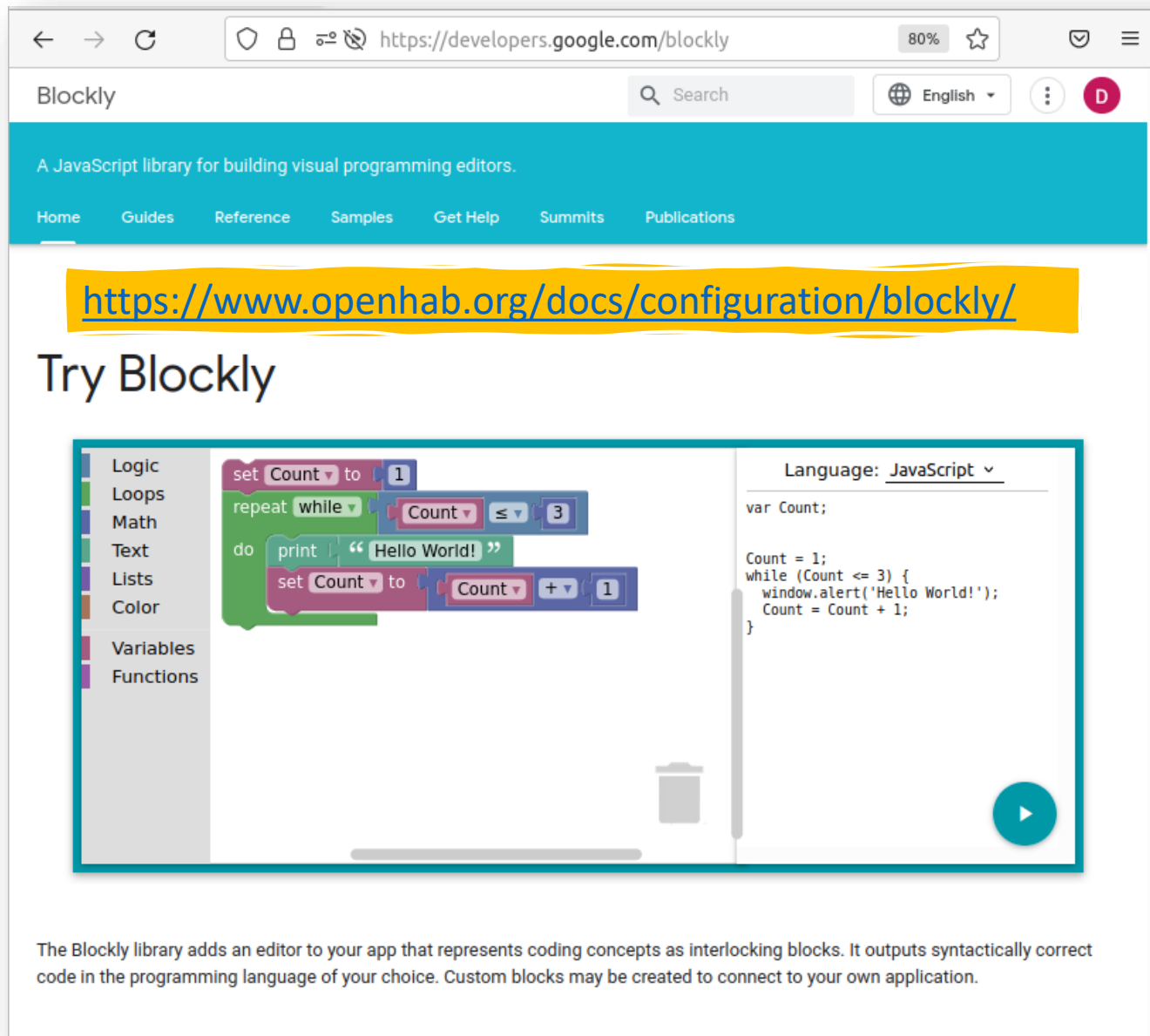
- When the time is 07:00**  
Triggers at a specified time
- Add Trigger

Then

- Send command ON to MultiActuadorMQTT\_Enchufe**  
Sends a command to a specified item.
- Send command UP to MultiActuadorHTTP\_Persiana**  
Sends a command to a specified item.
- Add Action



# Creación de regla usando un programa hecho con Blockly



The screenshot shows a web browser window at <https://developers.google.com/blockly>. The page title is "Blockly" and the subtitle is "A JavaScript library for building visual programming editors." A navigation menu includes "Home", "Guides", "Reference", "Samples", "Get Help", "Summits", and "Publications". A yellow highlight box contains the URL <https://www.openhab.org/docs/configuration/blockly/>. Below this is the heading "Try Blockly".

The main content area displays a Blockly code editor. On the left is a category menu with "Logic", "Loops", "Math", "Text", "Lists", "Color", "Variables", and "Functions". The central workspace shows a program with the following blocks:

- set Count to 1
- repeat while loop containing:
  - Count ≤ 3
  - do block containing:
    - print "Hello World!"
    - set Count to Count + 1

On the right, the "Language: JavaScript" dropdown is selected, and the corresponding code is shown:

```
var Count;

Count = 1;
while (Count <= 3) {
  window.alert('Hello World!');
  Count = Count + 1;
}
```

At the bottom of the editor is a play button icon.

The text below the editor reads: "The Blockly library adds an editor to your app that represents coding concepts as interlocking blocks. It outputs syntactically correct code in the programming language of your choice. Custom blocks may be created to connect to your own application."

# Creación de regla usando un programa hecho con Blockly

**openHAB**  
empowering the smart home

Sin páginas

Administración

- Configuración
- Things
- Model
- Items
- Pages
- Rules**
- Scripts
- Schedule

Herramientas de Desarrollo

Ayuda y Acerca de

**admin-openhab**  
http://192.168.0.100:8080

Si la temperatura es < 20° encender el actuador de enchufe  
application/javascript Save (Ctrl-S)

Logic  
Loops  
Math  
Text  
Lists  
Color  
openHAB  
Variables  
Functions

```
log info " Empezar la regla encender "
```

```
set Temperatura to get item state item SensorTemperaturaSalon
```

```
if Temperatura ≤ 20
```

```
do
```

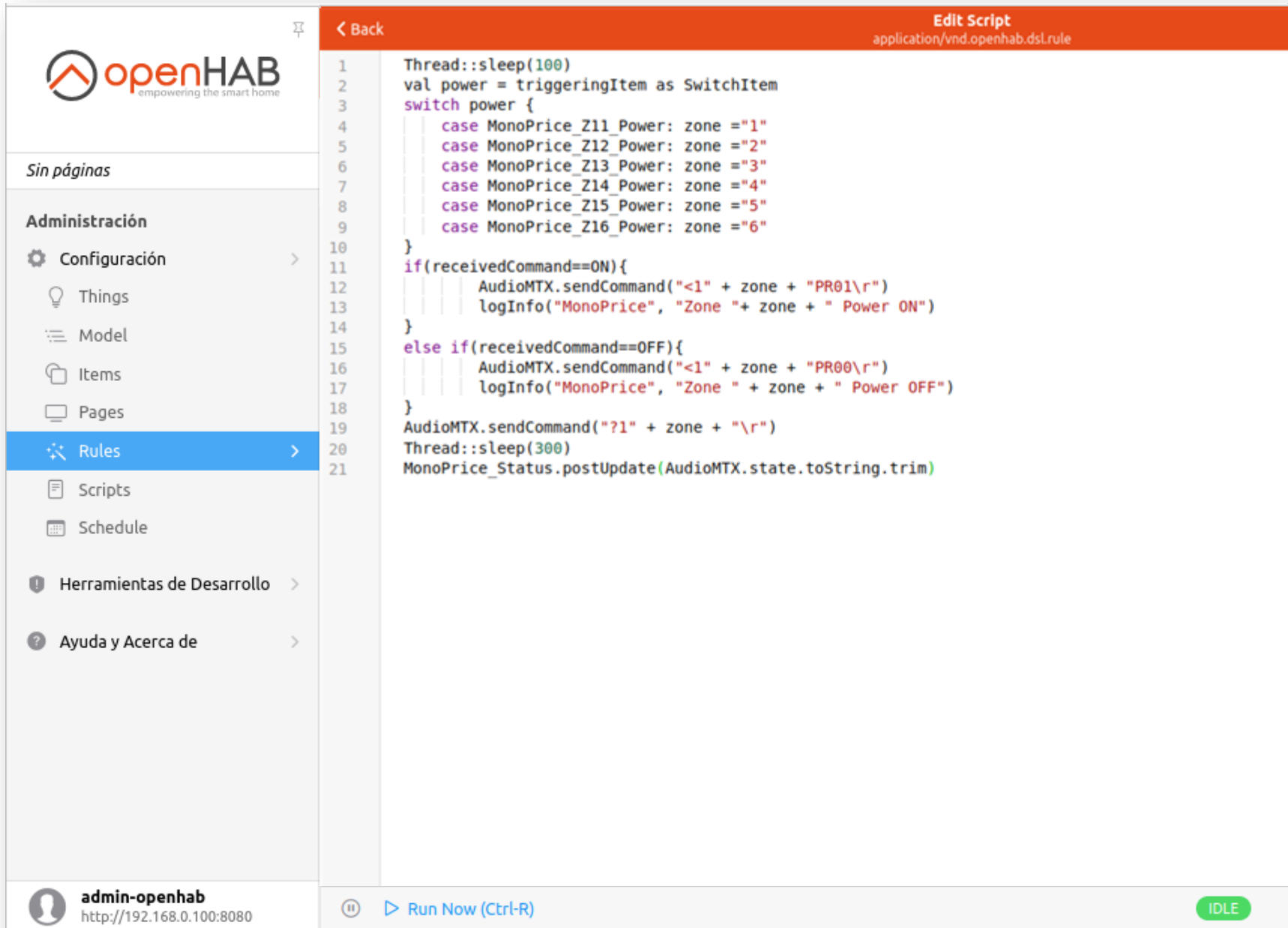
```
log info " Se ha bajado de 20°. Encender la estufa "
```

```
send command " ON " to item MultiActuadorHTTP_Enchufe
```

```
log info " Terminar la regla encender "
```

Run Now (Ctrl-R) IDLE

# Creación de regla usando un programa en la interfaz web



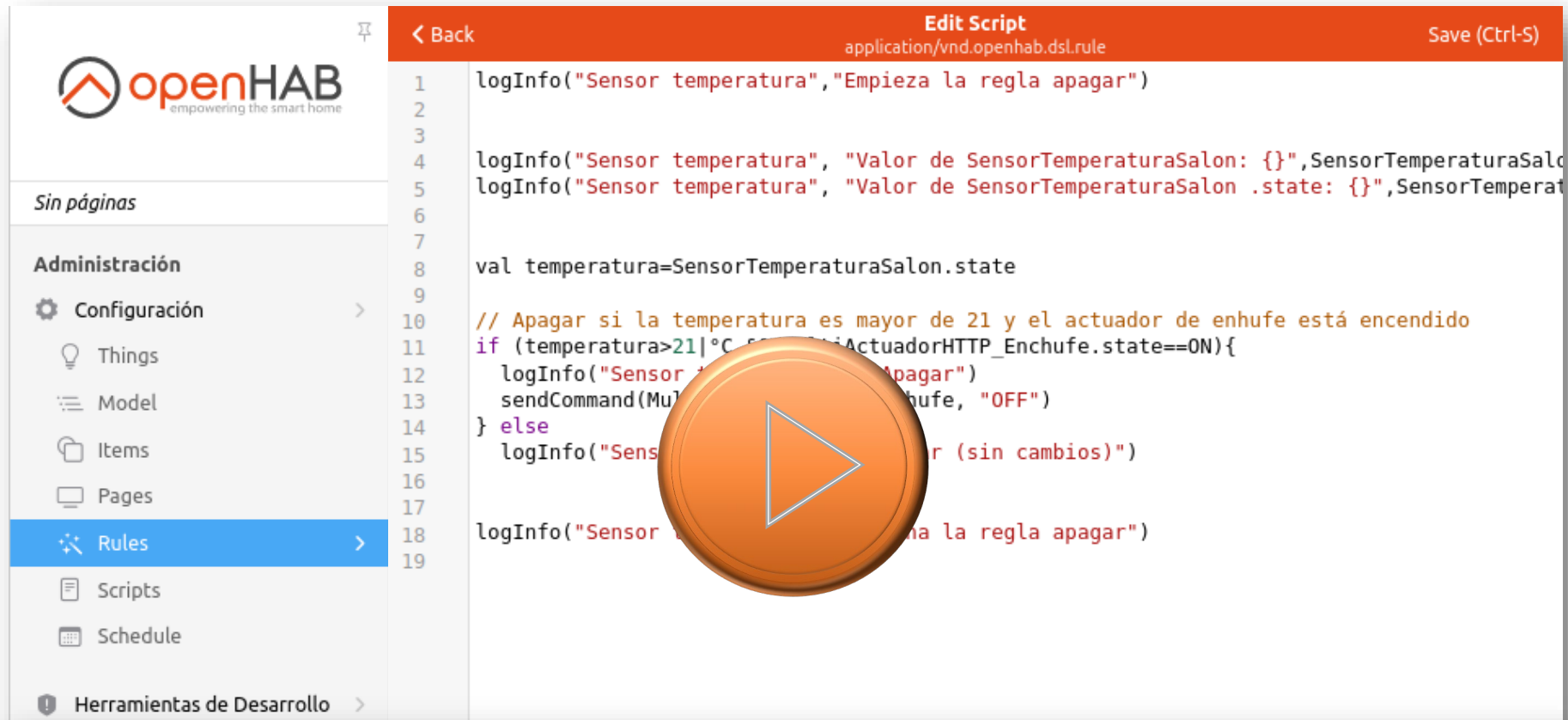
The screenshot displays the OpenHAB web interface. On the left is a navigation sidebar with the OpenHAB logo and a menu containing: Sin páginas, Administración (with sub-items: Configuración, Things, Model, Items, Pages, Rules, Scripts, Schedule), Herramientas de Desarrollo, and Ayuda y Acerca de. The main area is titled 'Edit Script' and shows a DSL rule for controlling a power switch based on zone. The script includes a sleep(100) at the start, a switch statement for zones 1-6, and logic to send commands to an AudioMTX device and log the state. It also includes a sleep(300) and a postUpdate call. At the bottom, there is a status bar with 'admin-openhab http://192.168.0.100:8080', a 'Run Now (Ctrl-R)' button, and an 'IDLE' indicator.

```
1 Thread::sleep(100)
2 val power = triggeringItem as SwitchItem
3 switch power {
4     case MonoPrice_Z11_Power: zone ="1"
5     case MonoPrice_Z12_Power: zone ="2"
6     case MonoPrice_Z13_Power: zone ="3"
7     case MonoPrice_Z14_Power: zone ="4"
8     case MonoPrice_Z15_Power: zone ="5"
9     case MonoPrice_Z16_Power: zone ="6"
10 }
11 if(receivedCommand==ON){
12     AudioMTX.sendCommand("<1" + zone + "PR01\r")
13     logInfo("MonoPrice", "Zone " + zone + " Power ON")
14 }
15 else if(receivedCommand==OFF){
16     AudioMTX.sendCommand("<1" + zone + "PR00\r")
17     logInfo("MonoPrice", "Zone " + zone + " Power OFF")
18 }
19 AudioMTX.sendCommand("?1" + zone + "\r")
20 Thread::sleep(300)
21 MonoPrice_Status.postUpdate(AudioMTX.state.toString.trim)
```

# xtend

[https://www.eclipse.org/xtend/documentation/203\\_xtend\\_expressions.html](https://www.eclipse.org/xtend/documentation/203_xtend_expressions.html)

# Creación de regla usando un programa en la interfaz web



**openHAB**  
empowering the smart home

Sin páginas

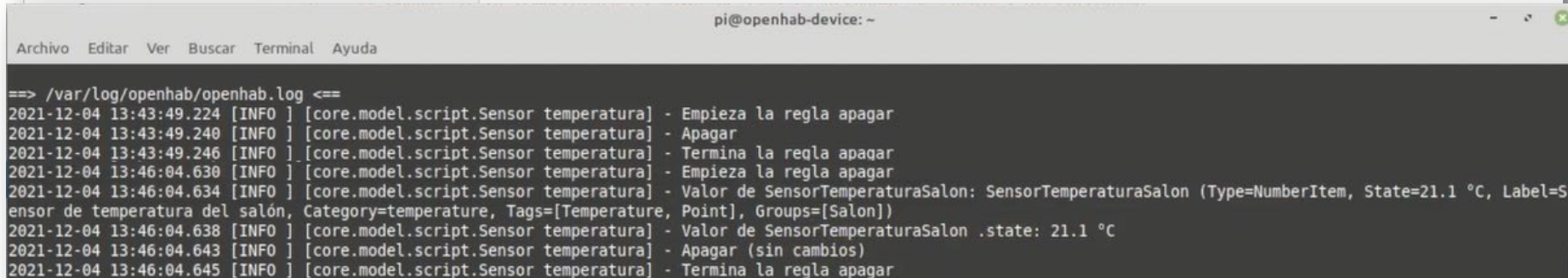
Administración

- Configuración
- Things
- Model
- Items
- Pages
- Rules**
- Scripts
- Schedule

Herramientas de Desarrollo

**Edit Script**  
application/vnd.openhab.dsl.rule Save (Ctrl-S)

```
1 logInfo("Sensor temperatura","Empieza la regla apagar")
2
3
4 logInfo("Sensor temperatura", "Valor de SensorTemperaturaSalon: {}","SensorTemperaturaSalon")
5 logInfo("Sensor temperatura", "Valor de SensorTemperaturaSalon .state: {}","SensorTemperaturaSalon")
6
7
8 val temperatura=SensorTemperaturaSalon.state
9
10 // Apagar si la temperatura es mayor de 21 y el actuador de enchufe está encendido
11 if (temperatura>21|°C == ON) {
12   logInfo("Sensor temperatura","Apagar")
13   sendCommand(MultiChannelActuatorHTTP_Enchufe, "OFF")
14 } else
15   logInfo("Sensor temperatura","Apagar (sin cambios)")
16
17
18 logInfo("Sensor temperatura","Termina la regla apagar")
19
```



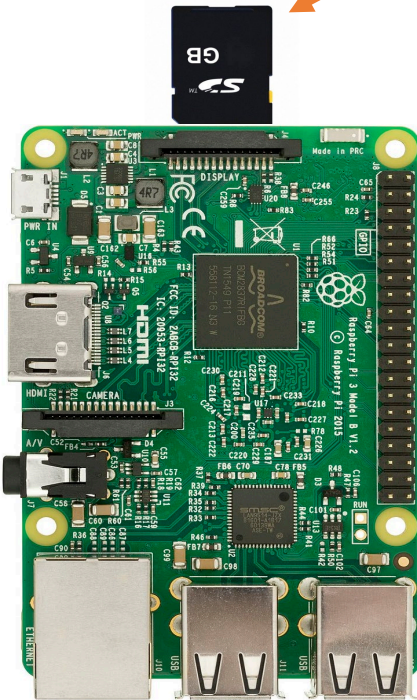
pi@openhab-device: ~

Archivo Editar Ver Buscar Terminal Ayuda

```
==> /var/log/openhab/openhab.log <==
2021-12-04 13:43:49.224 [INFO ] [core.model.script.Sensor temperatura] - Empieza la regla apagar
2021-12-04 13:43:49.240 [INFO ] [core.model.script.Sensor temperatura] - Apagar
2021-12-04 13:43:49.246 [INFO ] [core.model.script.Sensor temperatura] - Termina la regla apagar
2021-12-04 13:46:04.630 [INFO ] [core.model.script.Sensor temperatura] - Empieza la regla apagar
2021-12-04 13:46:04.634 [INFO ] [core.model.script.Sensor temperatura] - Valor de SensorTemperaturaSalon: SensorTemperaturaSalon (Type=NumberItem, State=21.1 °C, Label=Sensor de temperatura del salón, Category=temperature, Tags=[Temperature, Point], Groups=[Salon])
2021-12-04 13:46:04.638 [INFO ] [core.model.script.Sensor temperatura] - Valor de SensorTemperaturaSalon .state: 21.1 °C
2021-12-04 13:46:04.643 [INFO ] [core.model.script.Sensor temperatura] - Apagar (sin cambios)
2021-12-04 13:46:04.645 [INFO ] [core.model.script.Sensor temperatura] - Termina la regla apagar
```

# Creación de regla usando un fichero

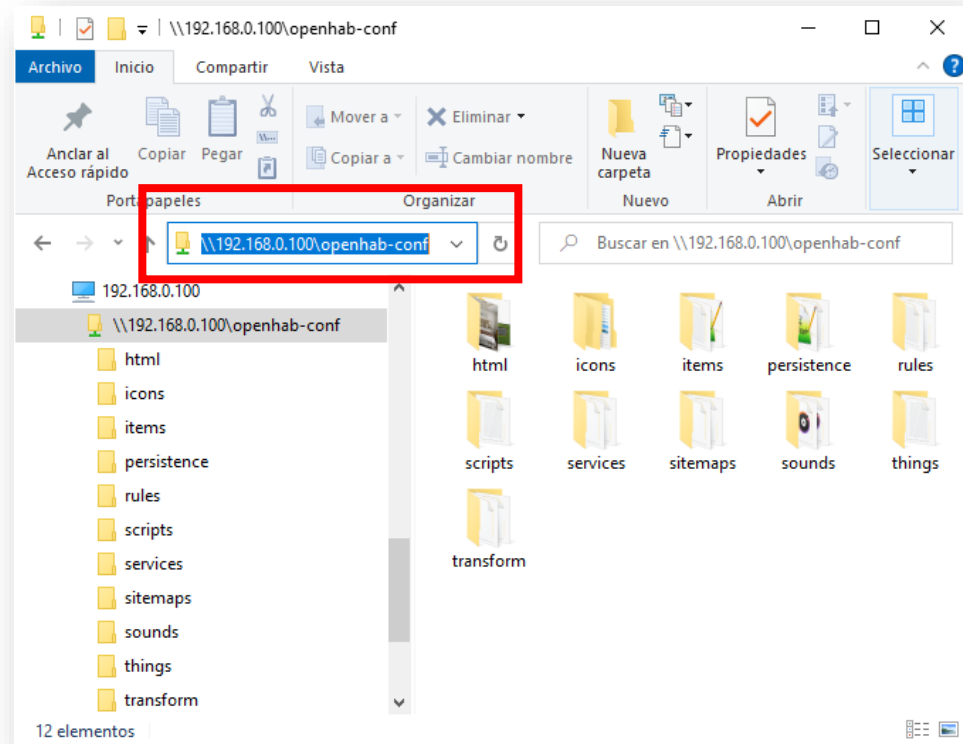
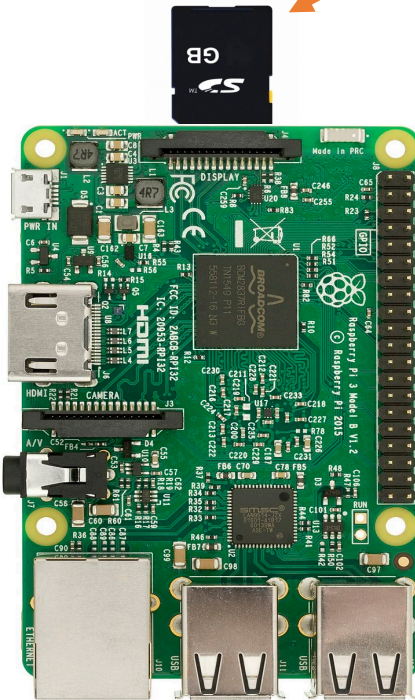
/etc/openhab/rules



# Creación de regla usando un fichero

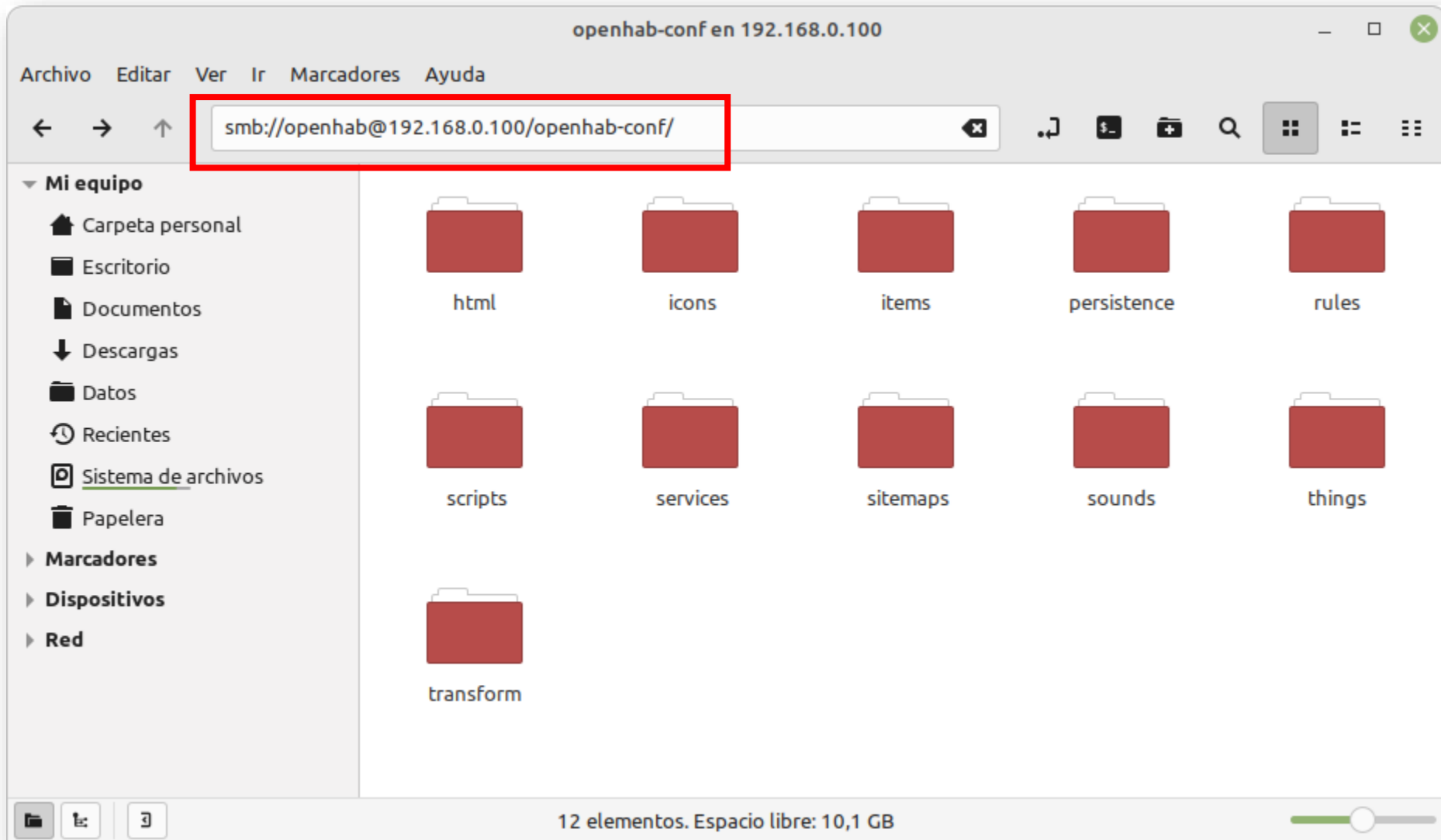
/etc/openhab/rules

\\IP-raspberry\openhab-conf  
Usuario: openhab. Clave: habopen

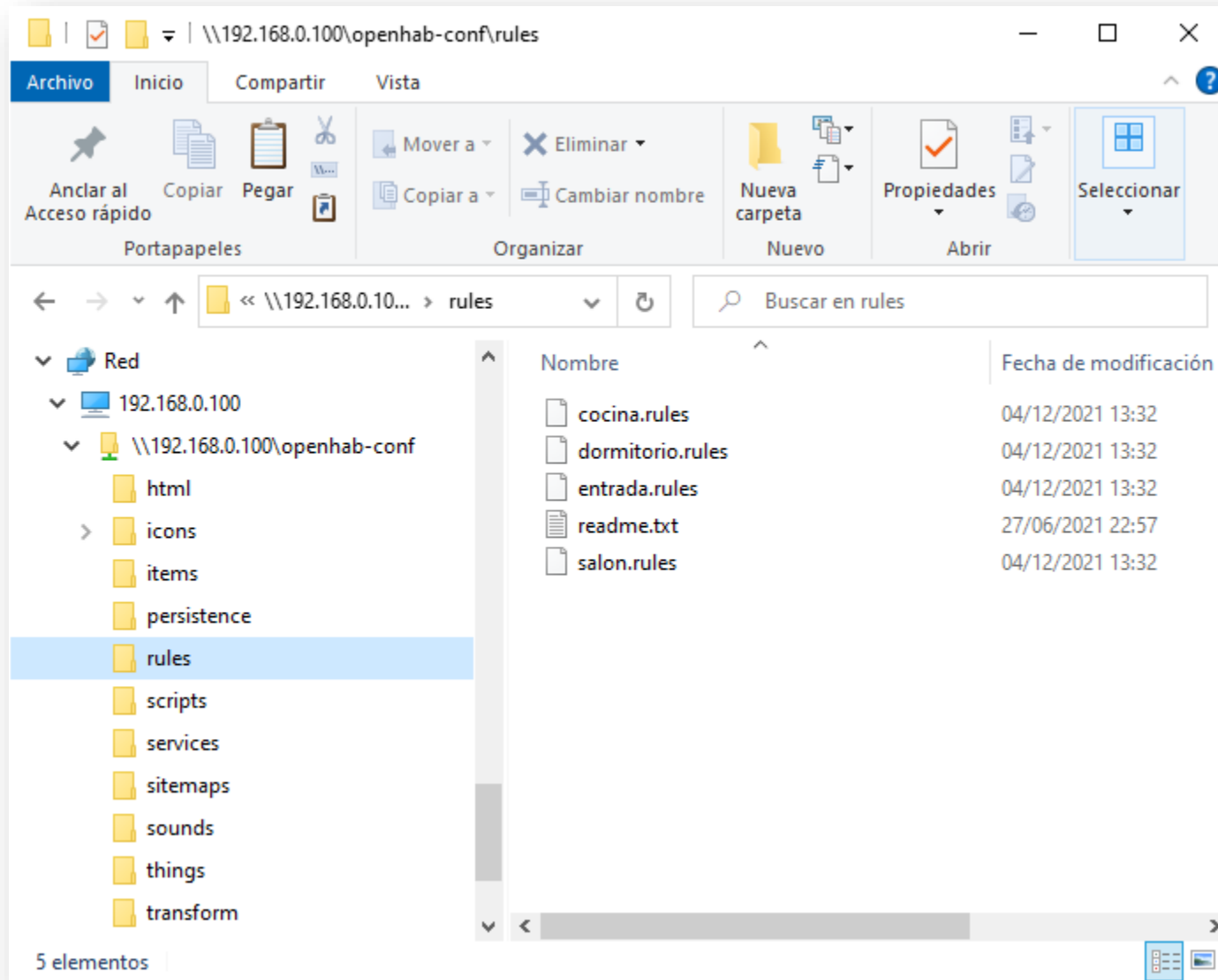


# Creación de regla usando un fichero

smb://openhhab@IP-Rasperrry/openhab-conf/



# Creación de regla usando un fichero



```
rule "<Nombre de la regla>"
```

```
when
```

```
    <condición desencadentante>
```

```
    [or <condición desencadentante>
```

```
    [or ...]]
```

```
then
```

```
    <Bloque de instrucciones>
```

```
end
```

- Un ítem recibe un comando, se actualiza o cambia de estado:  
`Item <item> received command [comando]`  
`Item <item> received update [estado]`  
`Item <item> changed [from <estado>] [to < estado>]`  
Ej: `Item SensorTemperaturaSalon chaged`

- En un tiempo determinado:


`Time is midnight`

`Time cron "<cron expression>"`

Ej: `Time cron "0 0 8 ? * MON,TUE,WED,THU,FRI"`

<https://www.freeformatter.com/cron-expression-generator-quartz.html>

```
val temperatura=SensorTemperaturaSalon.state  
  
// Apagar si la temperatura es mayor que 21°C y el actuador de enchufe está encendido  
if (temperatura>21|°C && MultiActuadorEnchufe.state==ON){  
    logInfo("Sensor temperatura", "Apagar (sin cambios)")  
    sendCommand(MultiActuadorHTTP_Encendido, "OFF")  
} else  
    logInfo("Sensor temperatura", "Apagar (sin cambios)")
```





¿Cómo mejorar las reglas anteriores

- ✓ Funcionar solo en invierno
- ✓ Encenderla solo si hay alguien en casa
- ✓ ....

<https://github.com/openhab/openhab-distro/blob/main/distributions/openhab-demo/src/main/resources/conf/rules/demo.rules>

# Ejemplo de temporizador

```
30 rule "Timer Demo"
31 when
32     Item Light_GF_Corridor_Ceiling received command
33 then
34     if (receivedCommand == ON) {
35         if (timer === null) {
36             // first ON command, so create a timer to turn the light off again
37             timer = createTimer(now.plusSeconds(10)) [|
38                 sendCommand(Light_GF_Corridor_Ceiling, OFF)
39             ]
40         } else {
41             // subsequent ON command, so reschedule the existing timer
42             timer.reschedule(now.plusSeconds(10))
43         }
44     } else if (receivedCommand == OFF) {
45         // remove any previously scheduled timer
46         if (timer !== null) {
47             timer.cancel
48             timer = null
49         }
50     }
51 end
```




## Escena = conjunto de acciones

Escena buenas noches:

- ✓ Apagar todas las luces de la casa
- ✓ Bajar todas las persianas

Settings Items Select



escena

	<b>Escena buenas noches</b> Switch · Point EscenaBuenasNoches	OFF >
	<b>Escena buenos días</b> Switch · Point EscenaBuenosDias	OFF >
	<b>Escena ver la tele</b> Switch · Point EscenaVerTele	OFF >

Settings Rules Select

escena

E


<b>Escena buenas noches</b> 	IDLE >
Escenas-1	
<b>Escena buenos días</b>	IDLE >
b252490dc7	
<b>Escena ver la tele</b> 	IDLE >
Escenas-2	

```
rule "Escena: Buenas noches"  
when  
    Item EscenaBuenasNoches received command ON  
then  
    sendCommand(Salon_LuzTecho, OFF)  
    sendCommand(Salon_Persiana, DOWN)  
    EscenaBuenasNoches.postUpdate(OFF)  
end
```




```
rule "Escena: Ver la tele"  
when  
    Item EscenaVerTele received command ON  
then  
    sendCommand(Salon_LuzTecho, 20)  
    sendCommand(Salon_Tele, ON)  
    sendCommand(Salon_Homecinema, ON)  
    EscenaVerTele.postUpdate(OFF)  
end
```

- ▼ Casa  
House
- ▼ Escenas  
Equipment
  - ⚡ Escena buenas noches  
Point
  - ⚡ Escena buenos dias  
Point
  - ⚡ Escena ver la tele  
Point

## Casa



### Escenas

	Escena buenas noches	<input type="checkbox"/>
	Escena buenos dias	<input type="checkbox"/>
	Escena ver la tele	<input type="checkbox"/>

Cerrar



Universidad Politécnica de Madrid

Uso del IoT para construir tú mismo un hogar digital

Creación de tareas de automatización e inteligencia

Javier Malagón