

# ómo compilar un dispositivo Matter

## ACERCA DE ESTE CODELAB



77 minutos



Última actualización: 2 de octubre de 2023



Escrito por Jonathan Donald, Doug Ferraz

## . Introducción

Matter es un protocolo de conectividad que ofrece oportunidades emocionantes para el desarrollo de dispositivos inteligentes. En este codelab, compilarás tu primer dispositivo Matter. Para obtener más información sobre Matter, visita el [Developer Center de Google Home: Matter](https://developers.home.google.com/matter?hl=es-419) (https://developers.home.google.com/matter?hl=es-419) o el [sitio web de Connectivity Standards Alliance](http://www.csa-iot.org) (http://www.csa-iot.org).

## ué aprenderás

- Cómo integrar un dispositivo físico con Matter
- Cómo encargar y controlar tu dispositivo Matter con Google Home

## ué necesitarás

- Una unidad central, que es cualquier [dispositivo Google Nest compatible con Matter](https://developers.home.google.com/matter/supported-devices?hl=es-419#hubs) (https://developers.home.google.com/matter/supported-devices?hl=es-419#hubs), como el Nest Hub (2a generación)
- Una placa de evaluación de Matter [ESP32-DevkitC](https://www.espressif.com/en/products/hardware/esp32-devkitc/overview) (https://www.espressif.com/en/products/hardware/esp32-devkitc/overview) de Espressif Systems o [ESP32 M5STACK](https://www.espressif.com/en/products/modules/esp32) (https://www.espressif.com/en/products/modules/esp32)

- Una PC con Linux
- Conocimientos básicos de Linux

## . Configure su entorno

### identifica el dispositivo en serie

El primer paso para configurar tu entorno de desarrollo es determinar a qué puerto en serie está conectado tu dispositivo. Esta información te permitirá programar e interactuar con tu panel de desarrolladores.

1. Conecta la placa para desarrolladores a la computadora con un cable USB.
2. Busca en el sistema de archivos `/dev` para encontrar el dispositivo de la placa del desarrollador. Para limitar la búsqueda, especifica el prefijo de dispositivo de la placa del desarrollador. El ESP32 de Espressif usa `/dev/ttyUSBx`:

```
user@host $ ls /dev/ttyUSB*  
/dev/ttyUSB0
```

### onfigura tu concentrador

Configura tu concentrador **con la misma Cuenta de Google que vas a usar en este codelab.**

## ómo configurar tu entorno de desarrollo

### equisitos previos

Estas instrucciones se probaron en Debian Linux y deberían funcionar en la mayoría de las distribuciones de Linux basadas en Debian, incluido Ubuntu. Si trabajas con una distribución de Linux diferente, el procedimiento de configuración de dependencias puede variar de lo que se indica a continuación.

### nstala dependencias

Ejecuta el siguiente comando para instalar los objetos binarios del paquete de Linux que quizás no estén instalados:

```
sudo apt-get install git gcc g++ pkg-config libssl-dev libdbus-1-dev \
libglib2.0-dev libavahi-client-dev ninja-build python3-venv python3-dev \
python3-pip unzip libgirepository1.0-dev libcairo2-dev libreadline-dev screen
```

## Configura el SDK

Para continuar con este codelab, necesitarás el SDK de Espressif (el Espressif IoT Development Framework o “ESP-IDF”).

1. Crea un directorio que contenga el ESP-IDF:

```
$ mkdir ~/esp-idf_tools
```

2. Clona el ESP-IDF de GitHub en este directorio:

```
$ cd ~/esp-idf_tools
$ git clone -b v4.4.3 --recursive https://github.com/espressif/esp-idf.git
```

3. Completa la instalación de la cadena de herramientas:

```
$ cd ./esp-idf
$ ./install.sh
$ cd ~/
```

## Configura el SDK de Matter

1. Clona el repositorio de Matter de código abierto:

```
$ git clone https://github.com/project-chip/connectedhomeip.git
$ cd ./connectedhomeip
$ git fetch origin v1.0-branch
$ git checkout FETCH_HEAD
```

2. Recupera los submódulos del repositorio:

```
$ ./scripts/checkout_submodules.py --shallow --platform esp32
```

### 3. Inicia el entorno de desarrollo de Matter:

```
$ source ./scripts/bootstrap.sh
```

## . Consola para desarrolladores de Google Home

La consola para desarrolladores de Google Home

(<https://partnerdash.google.com/apps/nestdeveloperconsole?hl=es-419>) es la aplicación web en la que puedes administrar tus integraciones de Matter con Google Home.

Cualquier dispositivo Matter que haya aprobado la certificación de Matter de la Connectivity Standards Alliance (Alliance) funciona en el ecosistema de Google Home. Los dispositivos en desarrollo que no están certificados se pueden encarar en el ecosistema de Google Home bajo ciertas condiciones. Consulta Restricciones de vinculación

([https://developers.home.google.com/matter/integration/pair?hl=es-419#pairing\\_restrictions](https://developers.home.google.com/matter/integration/pair?hl=es-419#pairing_restrictions)) para obtener más información.

## ómo crear un proyecto de desarrollador

En primer lugar, ve a la consola para desarrolladores de Google Home:

[Ir a la consola para desarrolladores de Google Home](https://console.home.google.com/projects?hl=es-419)

(<https://console.home.google.com/projects?hl=es-419>)

1. Haz clic en **Crear proyecto**.
2. Ingresa un nombre de proyecto único y haz clic en **Crear proyecto**.

★ **Importante:** Al diseñar el nombre de un proyecto, es recomendable que lo hagas significativo y distinto. Considere incluir el nombre de su empresa y el tipo de dispositivo o la Acción que implementará. Por ejemplo, "encom-smartlight-switch" usa ambos elementos.

# Create a project

Create a project to get started.

Project name

Project name\*

30 characters max

Create project

Already have a Works with Assistant, cloud, or Local Home SDK project? [Import existing project.](#)

3. Haz clic en **+ Add integration**, que te llevará a la pantalla **Matter resources**, en la que podrás ver la documentación de desarrollo de Matter y leer sobre algunas herramientas.
4. Cuando estés listo para continuar, haz clic en **Next: Develop**, que muestra la página de la **lista de tareas de Matter**.
5. Haz clic en **Siguiente: Configuración**.
6. En la página **Configuración**, ingresa el **Nombre del producto**.
7. Haz clic en **Select device type** y selecciona el tipo de dispositivo en el menú desplegable (en este caso, **Light**).
8. En el ID del proveedor (VID), selecciona **Test VID** y elige 0xFFFF1 en el menú desplegable Test VID. En ID del producto (PID), ingresa 0x8000, haz clic en **Guardar y continuar** y, luego, en **Guardar** en la siguiente página. Usa estos valores exactos de VID/PID; los pasos posteriores del codelab dependerán de ellos.

★ **Nota:** Los VID 0xFFFF1, 0xFFFF2, 0xFFFF3 y 0xFFFF4 están reservados para pruebas. Antes de lanzar un producto, deberás solicitar un VID en [Connected Standards Alliance](https://csa-iot.org/) (<https://csa-iot.org/>). Además, los PID 0x8000 a 0x801F tienen DAC de prueba (certificados de certificación de dispositivos) (<https://github.com/project-chip/connectedhomeip/blob/080bb5730aeed4019fdde46401208d74a546f5bb/src/credentials/examples/ExampleDACs.h>)

que se pueden usar para ejemplos. A menos que planees configurar tu propio DAC, usa un PID que esté dentro de este rango.

## Device information \*

\* Required

Product name \*

ACME Lightbulb

14/24

Device type \*

Light

## Vendor ID (VID) \*

Use a CSA-issued vendor ID or test vendor ID with this integration.



CSA-issued vendor ID (to certify)



Test VID

Select the test VID that matches the VID used in your device build.

Test VID \*

0XFFF1



Note: A test VID can be used in development but a CSA-issued vendor ID is required to certify your integration.

## Product ID (PID) \*

Enter a unique PID for your product in hex value. Make sure it starts with 0x.

Product ID \*

0x8000

Use any generic number in hexadecimal format prior to production

9. Ahora verás tu integración en **Integraciones de Matter**.
10. Reinicia tu concentrador para asegurarte de que reciba la configuración más reciente del proyecto de integración de Matter. Si tienes que cambiar el VID o PID más adelante, también deberás reiniciar el dispositivo después de guardar el proyecto para que se aplique el cambio. Consulta [Cómo reiniciar los dispositivos Google Nest o Google Wifi](https://support.google.com/googlenest/answer/6272001?hl=es-419) (<https://support.google.com/googlenest/answer/6272001?hl=es-419>) para obtener instrucciones de reinicio paso a paso.

## . Compila el dispositivo

Todos los ejemplos de Matter se encuentran en la [carpeta de ejemplos del repositorio de GitHub](https://github.com/project-chip/connectedhomeip/tree/master/examples) (<https://github.com/project-chip/connectedhomeip/tree/master/examples>). Hay varios ejemplos disponibles, pero nuestro enfoque en este codelab es la [lighting-app](https://github.com/project-chip/connectedhomeip/tree/master/examples/lighting-app) (<https://github.com/project-chip/connectedhomeip/tree/master/examples/lighting-app>).

Este ejemplo es un dispositivo simple que aparece en Google Home como una luz de encendido y apagado, que responde a los comandos de encendido y apagado. Hacer que controle una luz eléctrica real está fuera del alcance de este codelab.

## onfigura la compilación

1. Configura el SDK de Matter y activa el entorno de compilación de Matter:

```
$ cd ~/esp-idf_tools/esp-idf
$ source export.sh
$ cd ~/connectedhomeip
$ source ./scripts/activate.sh
```

2. Habilita [Ccache](https://ccache.dev/) (<https://ccache.dev/>), que acelera el proceso de compilación:

```
$ export IDF_CCACHE_ENABLE=1
```

3. Ve al directorio de compilación del ESP32 **lighting-app** y configura la arquitectura de destino:

```
$ cd ./examples/lighting-app/esp32
```



```
$ idf.py set-target esp32
```

a. Ejecuta la utilidad de configuración:

```
$ idf.py menuconfig
```

b. Selecciona **Demo** -> **Device Type** y establece **Device Type** en **ESP32-DevKitC**.

c. Presiona la flecha hacia la izquierda para volver al menú de nivel superior.

d. Selecciona **Component config** --->.

e. Selecciona **CHIP Device Layer** --->.

f. Selecciona **Device Identification Options** --->.

g. Establece **Vendor ID** en el VID asignado por la Alianza o en un VID de prueba.

h. Establece **Product ID** en el PID que estableciste en la integración de Matter en la consola para desarrolladores de Google Home.

i. Presiona **S** para guardar.

j. Presione **Intro** para aceptar la ruta de acceso predeterminada en la que se guardará la configuración.

k. Presiona **Intro** para descartar el diálogo de confirmación de guardado.

l. Presiona **Q** para salir de la utilidad de configuración.

## Ejecuta la compilación

Invoca la secuencia de comandos de compilación:

```
idf.py build
```

La compilación se debería completar sin errores.

## Programa el dispositivo

1. Conecta la placa del desarrollador a la computadora con un cable USB.

2. Borra el firmware anterior del dispositivo (si solo tienes una placa del desarrollador conectada a la computadora, puedes omitir la opción `-p {device}`, ya que el dispositivo debería detectarse automáticamente):

```
idf.py -p {device} erase-flash
```

3. Copia tu nueva aplicación en la pizarra del desarrollador con lo siguiente:

```
idf.py -p {device} flash
```

Puedes encontrar más información sobre las opciones de escritura en la memoria flash en la [página de documentación de esptool.py](https://docs.espressif.com/projects/esptool/en/latest/esp8266/esptool/flash-modes.html) de Espressif (<https://docs.espressif.com/projects/esptool/en/latest/esp8266/esptool/flash-modes.html>).

## . Conéctate al dispositivo

1. Abre una ventana de la terminal.
2. Toma nota del directorio en el que te encuentras y, luego, conéctate a tu nuevo dispositivo Matter usando GNU `screen`:

```
$ screen -L {device} 115200
```

3. Si ves una consola en blanco, presiona el botón **RESTABLECER** para iniciar el proceso de inicio del dispositivo.

★ **Importante:** Cuando te conectas al dispositivo con `screen`, la consola se llena rápidamente de resultados. Si necesitas desplazarte hacia atrás y ver el resultado anterior, y tu emulador de terminal no tiene un búfer desplazable, necesitarás otra forma de verlo.

La solución alternativa es ver el archivo `screen0.log`, que es un archivo de registro que contiene el resultado de `screen` de GNU y se encuentra en el mismo directorio donde ejecutaste `screen`. Puedes hacerlo con un editor de texto, o en otra ventana de la terminal con `pager`, `more` o cualquier utilidad de paginación que esté instalada en tu sistema Linux.

## . Solicita el dispositivo

**Nota:** Este paso solo se realizará correctamente si ya configuraste tu proyecto en la consola para desarrolladores de Google Home.

### est Hub

Se requiere un concentrador para poner tu dispositivo en la estructura Matter. Este es un dispositivo Google Nest, como el Nest Hub (2a generación), que es compatible con Matter y que funciona como router de borde para dispositivos compatibles con Thread y como ruta de entrega local para enrutar intents de casa inteligente.

Consulta [esta lista](https://developers.home.google.com/matter/supported-devices?hl=es-419#hubs) (<https://developers.home.google.com/matter/supported-devices?hl=es-419#hubs>) para ver qué concentradores son compatibles con Matter.

Antes de comenzar el proceso de comisión, asegúrate de que se cumpla lo siguiente:

- Tu concentrador está vinculado con la misma Cuenta de Google que usaste para acceder a la consola de Google Home.
- El concentrador está conectado a la misma red Wi-Fi que la computadora que usas para ejecutar tu dispositivo virtual de Matter.
- La unidad central está en la misma [estructura](https://developers.home.google.com/cloud-to-cloud/primer/home-graph?hl=es-419) (<https://developers.home.google.com/cloud-to-cloud/primer/home-graph?hl=es-419>) que usas en la app de Google Home (la "casa" en [Google Home Graph](https://developers.home.google.com/cloud-to-cloud/primer/home-graph?hl=es-419) (<https://developers.home.google.com/cloud-to-cloud/primer/home-graph?hl=es-419>) representa la estructura).

## incula el dispositivo

Sigue las [instrucciones de vinculación del ESP32](https://developers.home.google.com/matter/integration/pair?hl=es-419)

(<https://developers.home.google.com/matter/integration/pair?hl=es-419>) para vincular tu dispositivo.

**Nota:** Si utilizas un dispositivo M5STACK, ten en cuenta que la pantalla permanecerá en blanco después de la instalación, por lo que tendrás que ver el código QR con la URL que aparece en la consola. También puedes escribir el código de vinculación manual.

Ejemplo de resultado de la consola que muestra la URL del código QR:

```
(1926) chip[DL]: Done driving station state, nothing else to do...
(1936) chip[SVR]: SetupQRCode: [MT:X.XXXXXXXXXXXXXXXXXX]
```

(1936) chip[SVR]: Copy/paste the below URL in a browser to see the QR Code:  
(1946) chip[SVR]: <https://project-chip.github.io/connectedhomeip/qrcode.html?dat>  
(1956) chip[SVR]: Manual pairing code: [XXXXXXXXXXXX]

## olución de problemas

### a comisión falla

- Asegúrate de haber creado un proyecto con la combinación correcta de VID/PID en la consola de Google Home (#2).
- Asegúrate de ingresar el VID y el PID correctos en la configuración de compilación (#3).
- Asegúrate de seleccionar el Device Type correcto en la configuración de compilación (#3).

Para obtener más sugerencias sobre cómo solucionar problemas, consulta la página de solución de problemas de Matter (<https://developers.home.google.com/matter/troubleshooting?hl=es-419>).

## . Cómo controlar el dispositivo

Una vez que tu dispositivo compatible con Matter se ponga en marcha correctamente y aparezca en la app de Google Home como una bombilla, puedes intentar controlarlo con Asistente de Google, la app de Google Home o el simulador del Asistente de Google en la extensión de Google Home para VS Code.

## sistente de Google

Usa Asistente de Google en tu teléfono o concentrador para activar o desactivar el estado del dispositivo a partir de comandos por voz, como decir "Hey Google, enciende las luces".

Para ver más ejemplos de comandos, consulta la sección *Cómo controlar los dispositivos de casa inteligente con comandos por voz* de Cómo controlar los dispositivos de casa inteligente que se agregaron a la app de Google Home

(<https://support.google.com/googlenest/answer/7073578?hl=es-419>).

## pp de Google Home

Puedes presionar las etiquetas **Activar** y **Desactivar** junto al ícono de la bombilla que se muestra en la app de Google Home.

Consulta [Cómo controlar dispositivos con la app de Google Home](https://support.google.com/googlenest/answer/7073578?hl=es-419#HomeAppControl)

(<https://support.google.com/googlenest/answer/7073578?hl=es-419#HomeAppControl>) para obtener más información.

## Simulador de Asistente de Google

En la extensión de Google Home para VS Code, con el [simulador de Asistente de Google](https://developers.home.google.com/matter/tools/vs-code?hl=es-419#google_assistant_simulator)

([https://developers.home.google.com/matter/tools/vs-code?hl=es-419#google\\_assistant\\_simulator](https://developers.home.google.com/matter/tools/vs-code?hl=es-419#google_assistant_simulator)), puedes emitir declaraciones a tu dispositivo con una interfaz similar a un chat.

## . Felicitaciones.

Creaste y pusiste en marcha correctamente tu primer dispositivo Matter. ¡Genial!

En este codelab aprendiste a hacer lo siguiente:

- Instalar un entorno de desarrollo de Matter desde los requisitos hasta un estado de trabajo
- Compila y ejecuta un dispositivo Matter.
- Comisión y controla tu dispositivo desde Google Home.

Para obtener más información sobre Matter, explora estas referencias:

- [Primer de Matter](https://developers.home.google.com/matter/primer?hl=es-419) (<https://developers.home.google.com/matter/primer?hl=es-419>) de Google Home, en el que aprenderás los conceptos y principios importantes del protocolo de Matter.
- La Especificación de Matter, la Biblioteca de Dispositivo de Matter y la Biblioteca de Clústeres de Aplicaciones de Matter, publicadas por la [Connected Standard Alliance](http://www.csa-iot.org) (<http://www.csa-iot.org>).
- El [repositorio de Matter en GitHub](https://github.com/project-chip/connectedhomeip/) (<https://github.com/project-chip/connectedhomeip/>).

Salvo que se indique lo contrario, el contenido de esta página está sujeto a la [licencia Atribución 4.0 de Creative Commons](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), y los ejemplos de código están sujetos a la [licencia Apache 2.0](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). Para obtener más información, consulta las [políticas del sitio de Google Developers](https://developers.google.com/site-policies?hl=es-419) (<https://developers.google.com/site-policies?hl=es-419>). Java es una marca registrada de Oracle o sus afiliados.