Contenido

[Introducción 1](#_Toc137044011)

[Configuración 1](#_Toc137044012)

[Proyecto “Hello World” 1](#_Toc137044013)

[Tarea: Configurar un entorno virtual de Python 2](#_Toc137044014)

[Tarea: Escribir el código 3](#_Toc137044015)

[Tarea: Conectar el “Hardware” 4](#_Toc137044016)

# Introducción

Trabaje con la guía correspondiente para configurar su dispositivo y completar un proyecto “Hello World”. Este será el primer paso para crear una “Lampará IoT” en las 4 lecciones de introducción.

En lugar de comprar un dispositivo IoT, junto con sensores y actuadores, puede usar su computadora para simular hardware IoT. El proyecto [CounterFit](https://github.com/CounterFit-IoT/CounterFit) le permite ejecutar una aplicación localmente que simula hardware IoT, como sensores y actuadores, y acceder a los sensores y actuadores desde el código Python local que está escrito de la misma manera que el código que escribiría en una Raspberry Pi usando hardware físico.

# Configuración

Para usar CounterFit, deberá instalar algún software gratuito en su computadora.

Instale el software necesario.

1. Instala [Python](https://www.python.org/downloads/).
2. Instale el código de Visual Studio ([VS Code](https://code.visualstudio.com/?WT.mc_id=academic-17441-jabenn)).
3. Instale la extensión VS Code [Pylance](https://marketplace.visualstudio.com/items?WT.mc_id=academic-17441-jabenn&itemName=ms-python.vscode-pylance).

Las instrucciones para instalar y configurar la aplicación CounterFit se proporcionarán en el momento pertinente en las instrucciones de asignación, ya que se instala por proyecto.

# Proyecto “Hello World”

Cuando se comienza con un nuevo lenguaje de programación o tecnología, es tradicional crear una aplicación para mostrar que todas las herramientas están configuradas correctamente.

La aplicación “Hello World” para el hardware de IoT virtual se asegurará de que tenga el código de Python y Visual Studio instalado correctamente. También se conectará a CounterFit para los sensores y actuadores virtuales de IoT. No usará ningún hardware, solo se conectará para demostrar que todo funciona.

Esta aplicación estará en una carpeta llamada nightlight y se reutilizará con un código diferente en partes posteriores de esta tarea para crear la aplicación de luz nocturna.

Una de las características poderosas de Python es la capacidad de instalar [paquetes pip](https://pypi.org/). Usarás pip para instalar unas librerías requeridas para interactuar con CounterFit.

De manera predeterminada, cuando instala un paquete, está disponible en todas partes de su computadora, y esto puede generar problemas con las versiones del paquete, como una aplicación depende de una versión de un paquete, la dependencia se puede romper cuando instala una nueva versión para una aplicación diferente. Para solucionar este problema, puede usar un [entorno virtual de Python](https://docs.python.org/3/library/venv.html), esencialmente una copia de Python en una carpeta dedicada, y cuando instala paquetes de pip, se instalan solo en esa carpeta.

## Tarea: Configurar un entorno virtual de Python

Configure un entorno virtual Python e instale los paquetes pip para CounterFit.

1. Desde su terminal o línea de comando (modo administrador), ejecute lo siguiente en una ubicación de su elección para crear y navegar a un nuevo directorio:

Directorio de trabajo: ..\Downloads\00\_dIoT\_2023\_labs\lab01\

|  |
| --- |
| PS > mkdir nightlight  PS > cd nightlight |

1. Ahora ejecute lo siguiente para crear un entorno virtual en la carpeta .venv

|  |
| --- |
| PS > python -m venv .venv |

1. Activar el entorno virtual en Windows utilizando PowerShell

|  |
| --- |
| PS > .\.venv\Scripts\Activate.ps1 |

1. Una vez que se haya activado el entorno virtual, el comando python ejecutará la versión de Python que se utilizó para crear el entorno virtual. Ejecute lo siguiente para obtener la versión:

|  |
| --- |
| PS > python --version |

La salida debe contener los siguiente:

|  |
| --- |
| (.venv) -> nightlight python –version  Python 3.9.11 |

1. Ejecute los siguientes comandos para instalar los paquetes de Pip para CounterFit

Estos paquetes pip solo se instalarán en el entorno virtual y no estarán disponibles fuera de este.

|  |
| --- |
| PS > pip install CounterFit  PS > pip install counterfit-connection  PS > pip install counterfit-shims-grove |

## Tarea: Escribir el código

Una vez que el entorno virtual de Python esté listo, puede escribir el código para la aplicación 'Hello World'. Cree una aplicación de Python para imprimir "Hello World" en la consola.

1. Desde su terminal o línea de comando, ejecute lo siguiente dentro del entorno virtual para crear un archivo de Python llamado app.py:

|  |
| --- |
| PS > type nul > app.py |

1. Abra la carpeta actual en VS Code:

|  |
| --- |
| PS > code . |

1. Cuando se inicia VS Code, activará el entorno virtual de Python. El entorno virtual seleccionado aparecerá en la barra de estado inferior:

Código VS que muestra el entorno virtual seleccionado

1. Si VS Code Terminal ya se está ejecutando cuando se inicia VS Code, no tendrá activado el entorno virtual. Lo más fácil de hacer es eliminar el terminal usando el botón **Eliminar la instancia del terminal activo**:

VS Code Mata el botón de instancia de terminal activo

Puede saber si el terminal tiene activado el entorno virtual, ya que el nombre del entorno virtual será un prefijo en el aviso del terminal. Por ejemplo, podría ser:

|  |
| --- |
| PS > (.venv) -> nighlight |

Si no tiene .venv un prefijo en el indicador, el entorno virtual no está activo en la terminal.

1. Inicie una nueva Terminal VS Code seleccionando \*Terminal -> Nueva Terminal. El nuevo terminal cargará el entorno virtual, y en el terminal aparecerá la llamada para activarlo. El indicador también tendrá el nombre del entorno virtual (. venv):

|  |
| --- |
| PS > -> nighlight source .venv/bin/activate  (.venv) -> nighlight |

1. Abra el archivo app.py desde el explorador de VS Code y agregue el siguiente código:

|  |
| --- |
| print (‘Hello World!’) |

La función print imprime todo lo que se le pasa a la consola.

1. Desde la terminal de VS Code, ejecute lo siguiente para ejecutar su aplicación de Python:

|  |
| --- |
| PS > python app.py |

Lo siguiente estará en la salida:

|  |
| --- |
| PS > (.venv) -> nighlight  Hello world! |

😀 ¡Su programa 'Hello World' fue un éxito!

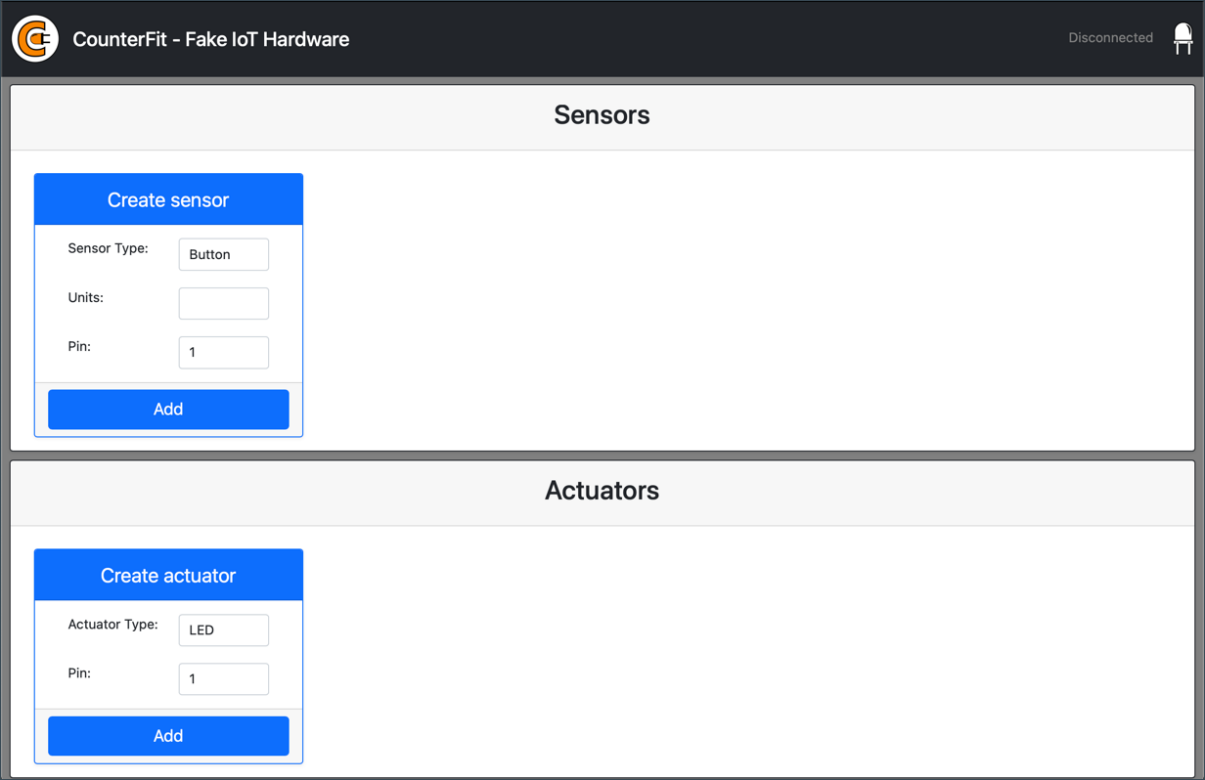
## Tarea: Conectar el “Hardware”

Como segundo paso de 'Hello World', ejecutará la aplicación CounterFit y conectará su código a ella. Este es el equivalente virtual de conectar algún hardware IoT a un kit de desarrollo.

1. Desde la terminal de VS Code, inicie la aplicación CounterFit con el siguiente comando:

|  |  |
| --- | --- |
| PS > counterfit |  |

La aplicación comenzará a ejecutarse y se abrirá en su navegador web “**127.0.0.1:5000**”:



Se marcará como Desconectado, con el LED en la esquina superior derecha apagado.

1. Agregue el siguiente código en la parte superior de app.py:

|  |
| --- |
| from counterfit\_connection import CounterFitConnection  CounterFitConnection.init('127.0.0.1', 5000) |

1. Deberá iniciar una nueva terminal VS Code seleccionando el botón Crear una nueva terminal integrada. Esto se debe a que la aplicación CounterFit se está ejecutando en el terminal actual.



1. En esta nueva terminal, ejecute el archivo app.py como antes. El estado de CounterFit cambiará a Conectado y el LED se encenderá.



Puede encontrar el código en la carpeta [code/virtual-device.](https://github.com/microsoft/IoT-For-Beginners/tree/main/1-getting-started/lessons/1-introduction-to-iot/code/virtual-device)

😀¡Su conexión con el hardware fue un éxito!