



# Programación Sin Código para Biología

## Guía para Principiantes

Dra. Stephanie Norwood, University of Cambridge

Liliana Artega Dorantes

Prof. Mario Artega Vazquez, Universidad Veracruzana







# Programación Sin Código para Biología

## Guía para Principiantes

### Aprenderás cómo:

- Entender y controlar una placa Arduino
- Programar sin utilizar código
- Utilizar componentes electrónicos simples como pantallas y sensores
- Construir tus propios dispositivos para la investigación biológica

## Iniciativa OpenPlant Biomaker

Biomaker fue establecido en 2014 como un programa interdisciplinario de aprendizaje en proyectos e innovación, dirigido por el Centro de Investigación Interdisciplinario de Biología Sintética de la Universidad de Cambridge (SynBio IRC) y OpenPlant, uno de los seis Centros Nacionales de Investigación de Biología Sintética (SBRCs) del Reino Unido. Es financiado primeramente por los consejos de investigación BBSRC y EPSRC del Reino Unido, y una beca de colaboración NERC/NFS entre el Reino Unido y los Estados Unidos de América.

Biomaker facilita el financiamiento de proyectos interdisciplinarios en equipos desarrollados en la intersección de la electrónica, impresión en 3D, tecnología de sensores, instrumentación DIY (del inglés 'Do it yourself' o hazlo tu mismo) y biología de bajo costo, así como talleres y eventos de divulgación. El proyecto busca construir tecnologías abiertas y promover el desarrollo de habilidades de investigación y colaboraciones. Además aprovecha los estándares abiertos existentes y un rico ecosistema de recursos para microcontroladores, simplificando la programación y la informática física para diseñadores, artistas y científicos. Estas herramientas permiten a los biólogos programar y desarrollar herramientas auténticas de laboratorio. El proyecto Biomaker también proporciona una ruta directa a ingenieros y científicos en el campo de la física para obtener experiencias prácticas con sistemas biológicos.

Aspiramos a romper las barreras que impiden el trabajo interdisciplinario y promocionar el tipo de formación que es útil para construir instrumentos y dispositivos para experimentos biológicos en el laboratorio y el campo. Desarrollamos kits para principiantes en programación sin código que permiten a biólogos construir prototipos de bioinstrumentación para la medición y control de sistemas biológicos. Éstos tienen una amplia gama de aplicaciones incluyendo la instrumentación, la microscopía, la microfluídica, la impresión 3D, los dispositivos biomédicos, el diseño de ADN, las ciencias de plantas, y divulgación y participación pública. Puedes encontrar ejemplos de proyectos pasados Biomaker que utilizan estas herramientas en la página web Biomaker [www.biomaker.org](http://www.biomaker.org).

Un aspecto importante de Biomaker es el uso de herramientas y hardware de código abierto y bajo costo, lo cual permite un acceso equitativo a la tecnología y al conocimiento básico, fomentando un ambiente colaborativo, apoyando el establecimiento de una bioeconomía abierta, sustentable y equitativa.







# Bienvenido a la Guía para Principiantes en Programación Sin Código para Biología

Esta guía para principiantes ha sido elaborada por el equipo Biomaker para ayudarte a entender las bases de 'biomaking' y la construcción de instrumentación personalizada para la investigación biológica.

Esta guía está diseñada para aquellos con poca o nula experiencia en programación o hardware, esta guía hace uso de software gratuito de código abierto y hardware de bajo costo para introducirte a los principios detrás de la elaboración de tus propios instrumentos.

Trabajar con esta guía puede ser útil como base para aquellos que tengan en mente unos retos o tareas en específicos, así como para aquellos que simplemente buscan expandir sus habilidades en el campo biológico.

Si bien nos enfocaremos en el aprendizaje de aspectos útiles para la investigación biológica, la información de esta guía también puede ser utilizada para una amplia gama de aplicaciones de programación sin código, y esperamos que estas habilidades puedan ser aplicables sea cual sea tu área de interés.

La guía te enseñará cómo utilizar el software gratuito de acceso abierto XOD para programación sin código, así como el uso de algunos dispositivos de hardware sencillos de bajo costo, tal como LEDs, sensores y pantallas.

Esta guía se encuentra diseñada para acompañar al 'Grove All-In-One Beginner Kit' para la placa de desarrollo Arduino, diseñada por Seeed Studio. Versiones alternativas de la placa Arduino Uno y componentes acompañantes también pueden ser usadas, pero tendrás que conectar o cablear estos componentes antes de comenzar.

Hemos elegido este kit 'todo en uno' que te permitirá comenzar sin tener que conectar o cablear componentes. En el momento de redactarse esta guía, este kit está disponible a través de distribuidores internacionales por aproximadamente \$24 USD. Más información está disponible en la página web de Seeed Studio [www.seeedstudio.com/distributors.html](http://www.seeedstudio.com/distributors.html).

## AUTORES

Stephanie Norwood  
Jim Haseloff

## TRADUCTORES

Liliana Artega Dorantes  
Mario Artega Vazquez  
Fernando Guzman Chavez

## IMÁGENES

Stephanie Norwood  
SparkFun Electronics  
Adafruit Industries  
Participantes del Biomaker Challenge

## CREADORES DE LA LIBRERÍA XOD

Matt Wayland  
Marco Aita  
Cesar Sosa  
Usuario XOD: gst  
Usuario XOD: e  
Usuario XOD: gweimer

## CONTACTOS

Stephanie Norwood  
coordinator@synbio.cam.ac.uk

Jim Haseloff  
jh295@cam.ac.uk



Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

# Contenidos

---

|  |           |
|--|-----------|
| <b>Lección 1: Introducción</b>                   | <b>01</b> |
| <b>La Guía</b>                                   | <b>04</b> |
| <b>El Kit de Inicio</b>                          | <b>06</b> |
| <b>El Microcontrolador</b>                       | <b>08</b> |
| <b>El XOD IDE</b>                                | <b>12</b> |
| <b>Lección 2: Primeros Pasos</b>                 | <b>17</b> |
| <b>Preparar Tu Placa</b>                         | <b>19</b> |
| <b>Tarea 1: Probando Tu Placa</b>                | <b>20</b> |
| <b>Tarea 2: Dispositivos de Entrada y Salida</b> | <b>26</b> |
| <b>Lección 3: Explora XOD</b>                    | <b>31</b> |
| <b>Tarea 3: Nodos Tweak y Watch</b>              | <b>33</b> |
| <b>Tarea 4: Nodos Flip, Clock y Count</b>        | <b>36</b> |

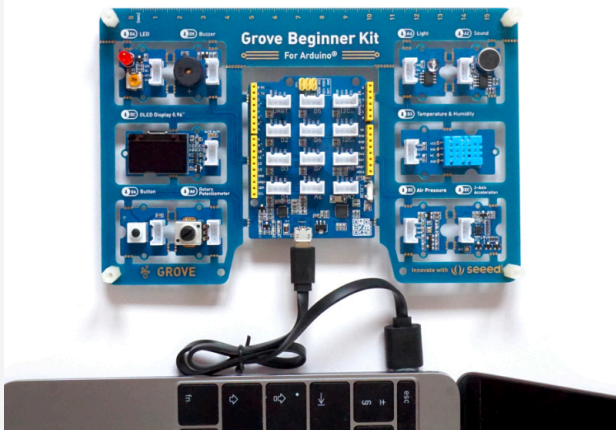
---

|  |           |
|--|-----------|
| <b>Tarea 5: Nodos Concat, Join y Format-Number</b> | <b>42</b> |
| <b>Lección 4: Construir Dispositivos</b>           | <b>47</b> |
| <b>Tarea 6: Crear Nuevos Nodos</b>                 | <b>49</b> |
| <b>Tarea 7: Utilizar Buses</b>                     | <b>56</b> |
| <b>Tarea 8: Programas Lógicos</b>                  | <b>60</b> |
| <b>Tarea 9: Secuencias y Bucles</b>                | <b>64</b> |
| <b>Lección 5: Sigüientes Pasos</b>                 | <b>73</b> |
| <b>Ampliando Tu Capacidad</b>                      | <b>75</b> |
| <b>Estudios de Caso</b>                            | <b>80</b> |
| <b>Información Adicional</b>                       | <b>84</b> |
| <b>Glosario</b>                                    | <b>90</b> |
| <b>Índice</b>                                      | <b>94</b> |

# Programación Sin Código para Biología

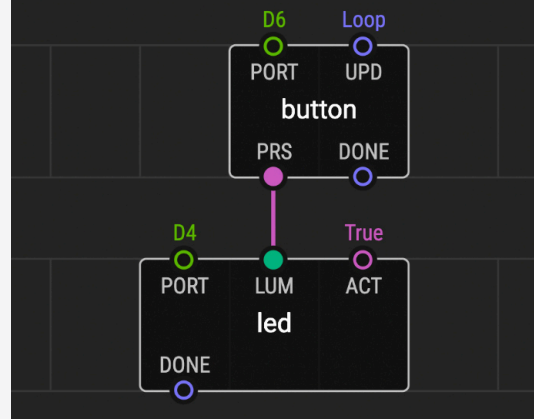
## Guía para Principiantes

### Lección 1: Introducción

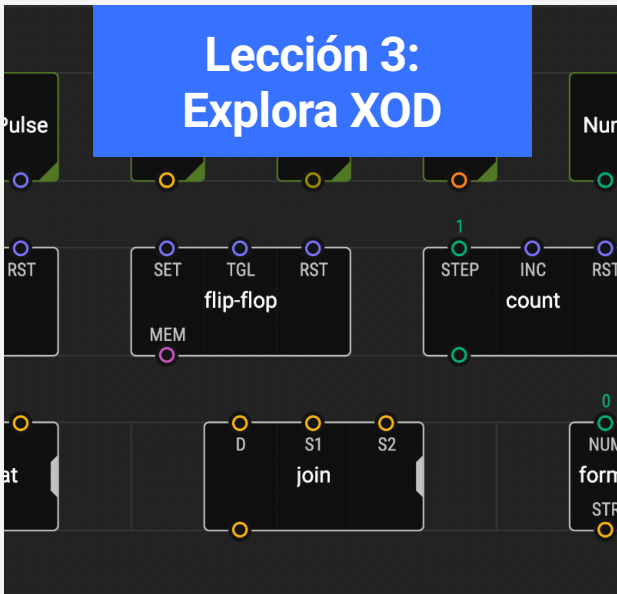


LA GUÍA  
EL KIT DE INICIO  
EL MICROCONTROLADOR  
EL XOD IDE

### Lección 2: Primeros Pasos



PREPARAR TU PLACA  
TAREA 1: PROBANDO TU PLACA  
TAREA 2: DISPOSITIVOS DE ENTRADA Y SALIDA



TAREA 3: NODOS TWEAK Y WATCH  
 TAREA 4: NODOS FLIP, CLOCK Y COUNT  
 TAREA 5: NODOS CONCAT, JOIN Y FORMAT-NUMBER

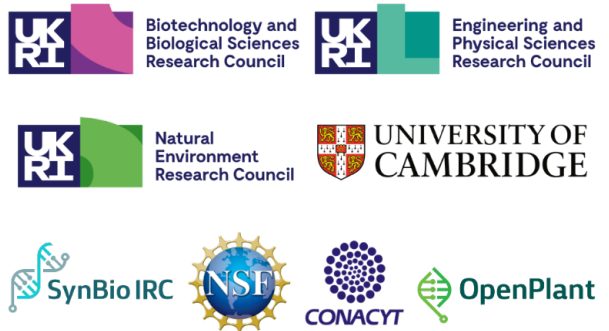


TAREA 6: CREAR NUEVOS NODOS  
 TAREA 7: UTILIZAR BUSES  
 TAREA 8: PROGRAMAS LÓGICOS  
 TAREA 9: SECUENCIAS Y BUCLES

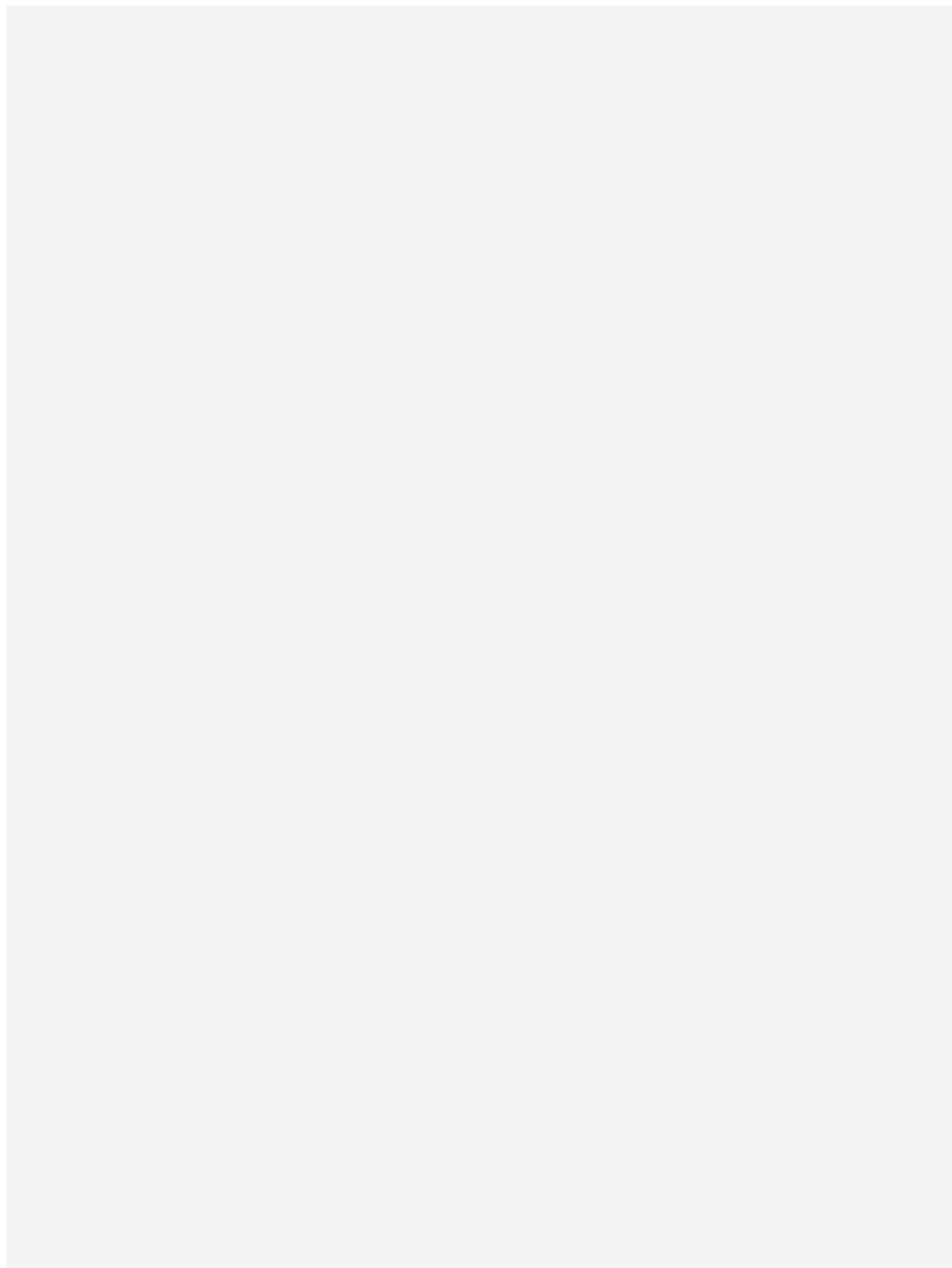


AMPLIANDO TU CAPACIDAD  
 ESTUDIOS DE CASO  
 INFORMACIÓN ADICIONAL  
 GLOSARIO

## Financiadores y Patrocinadores



BBSRC EPSRC NERC CONACYT  
 NSF OPENPLANT SYNBIO IRC  
 UNIVERSITY OF CAMBRIDGE





# Lección 1: Introducción



**La Guía**



**El Kit de Inicio**



**El Microcontrolador**



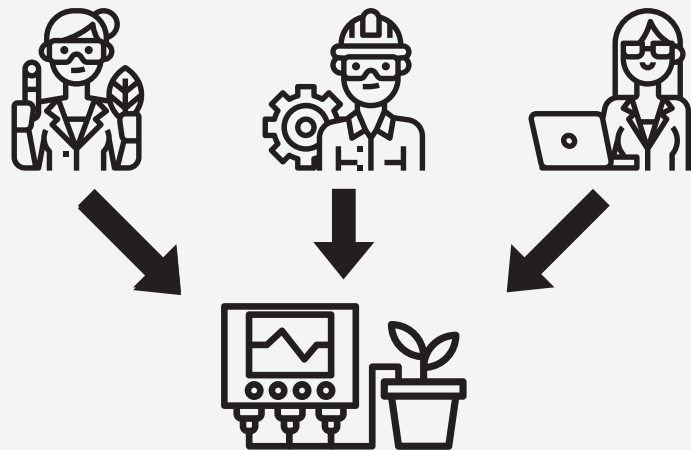
**El XOD IDE**

# Introducción

## Biomaker y programación sin código para biología

El equipo Biomaker elaboró esta guía para introducir a biólogos u otros científicos con poco entrenamiento formal en programación, a la información básica de Biomaker, incluyendo:

- Uso de microcontroladores basados en Arduino
- Uso de sensores, displays y actuadores
- Uso de programación visual XOD



Estas nuevas habilidades nos capacitan de muchas maneras. Los científicos pueden adquirir experiencia y nuevas formas de pensar para aplicarlas en su trabajo. Además, los componentes de este tipo de instrumentación suelen ser muy baratos, sobre todo si los comparamos con las soluciones comerciales disponibles. El uso de recursos sencillos de hardware y software permiten una modificación, ampliar y reparar fácilmente los instrumentos personalizados, y el uso de componentes y sistemas de código libre promueven el intercambio de información y el establecimiento de proyectos colaborativos. Esto crea un conjunto de recursos en continuo crecimiento que la comunidad puede utilizar y aprovechar.

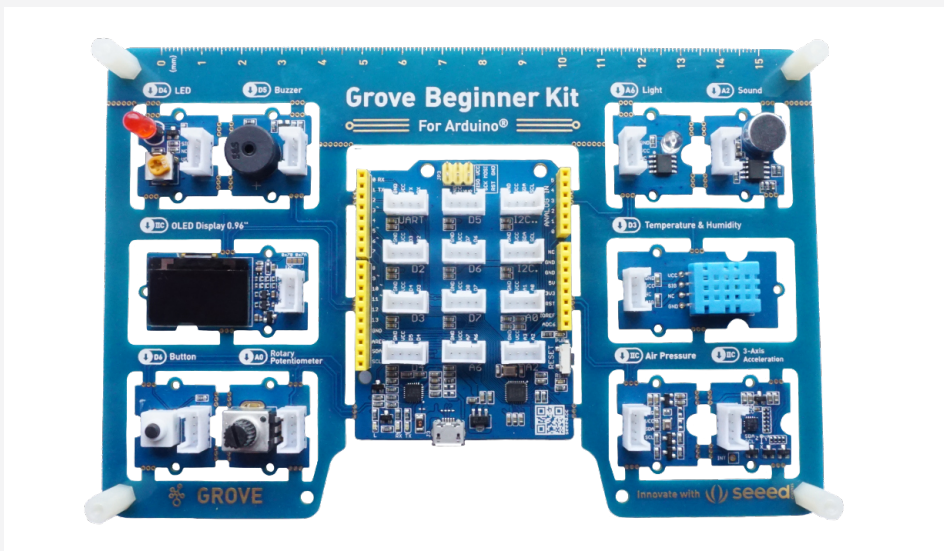
Esperamos que disfrutes participando en este curso online, que aprendas algo nuevo y que te resulte útil para tu futuro profesional. Sobre todo, animamos a los que estén interesados en profundizar sus habilidades a suscribirse al Biomaker Challenge, donde podrán unirse a un equipo de científicos e ingenieros con ideas afines para construir bio-instrumentos para aplicaciones del mundo real.



En esta primera lección cubriremos algunos de los antecedentes básicos que requerirás saber antes de empezar a programar.

En primer lugar, echaremos un vistazo a esta guía y a cómo utilizarla. Luego veremos la placa Grove y exploraremos cada uno de sus dispositivos incorporados, incluyendo cómo pueden ser utilizados. Después, hablaremos brevemente de los microcontroladores y de cómo programarlos, y finalmente te presentaremos el software XOD IDE y parte de su terminología.

Estas bases te ayudarán a familiarizarte con las herramientas que utilizaremos a lo largo de esta guía. Muchos de estos conceptos se volverán a tratar cuando apliquemos estos conocimientos para realizar tareas prácticas más adelante.



El Kit para Principiantes: 'Grove All-in-One Beginner Kit for Arduino'

## OBJETIVOS

Al final de este capítulo deberás ser capaz de:

- Nombrar las diferentes partes del tablero de Grove y dar ejemplos de cómo podrían utilizarse.
- Describir el concepto básico de un microcontrolador.
- Describir los pasos involucrados en la programación de la placa Arduino y cómo la información es transmitida en este sistema.
- Nombrar tres de los tipos más comunes de comunicación electrónica y explicar la diferencia entre ellos.
- Enumerar las conexiones de los pines (puertos) de cada uno de los componentes.
- Recordar las diferentes partes del software XOD IDE y describir para qué se utilizan.
- Recordar los tres términos clave utilizados en la programación de XOD y su significado.
- Enumerar los tipos de datos utilizados en XOD y dar ejemplos de cada uno.

# La Guía

La guía está dividida en cuatro lecciones principales, cada una de las cuales se describe a continuación. Estas lecciones están diseñadas para ser trabajadas en orden, y comienzan explorando el kit de inicio de Biomaker y el entorno de desarrollo integrado (IDE) de XOD. El resto de la guía, te llevará a través de una serie de tareas diseñadas para introducirte a tu placa, así como algunos aspectos clave de la programación en XOD.

El capítulo final ofrece información adicional de utilidad, así como algunos detalles sobre cómo ampliar tus habilidades y empezar a diseñar tus propios dispositivos.

Además de la información contenida en esta guía, la página web de XOD también ofrece algunos tutoriales útiles y un foro comunitario donde puedes encontrar ayuda en [www.xod.io](http://www.xod.io).



## Introducción

En este capítulo te daremos una breve introducción al kit de inicio de Biomaker, incluyendo la placa Grove, los microcontroladores, cómo controlarlos y cómo utilizar el software XOD.



## Primeros Pasos

Este capítulo te llevará a través de algunas tareas sencillas para empezar a usar tu placa y el XOD IDE. Aprenderás a utilizar los dispositivos de LED, buzzer y botones.



## Explora XOD

Este capítulo explorará algunas de las funciones más útiles de XOD. Entender cómo usar estas funciones te dará una gran base a partir de la cual trabajar en programación de XOD.



## Construir Dispositivos

Este capítulo profundizará en algunas funciones más complejas en XOD ¡Al final deberás ser capaz de empezar a desarrollar tus propias ideas, programas y dispositivos!

## Recursos de Programación sin Código para Biología

Además de esta guía para principiantes, el equipo de OpenPlant Biomaker también ha creado una gama de recursos útiles para principiantes y fabricantes avanzados que están disponibles en la página web de Biomaker. Estos recursos están diseñados para ayudarte a aprender más sobre las posibilidades de Biomaking y a ampliar tu capacidad para empezar a construir tus propios dispositivos. Incluyen tutoriales adicionales, videos e información sobre hardware y dispositivos comunes.

Todos los recursos Biomaker y Programación sin Código para Biología están disponibles para su descarga en la página web Biomaker en [www.biomaker.org/resources](http://www.biomaker.org/resources).



## Herramientas de Acompañamiento de la Guía

### GROVE BEGINNER KIT FOR ARDUINO

El kit de inicio Biomaker se compone de esta guía para principiantes, y del 'Grove Beginner Kit for Arduino' (Kit de Iniciación para Arduino de Grove). Este kit está fabricado por la empresa de hardware de código abierto Seeed Studio y se basa en un sencillo microcontrolador Arduino. El kit viene como una placa PCB integrada con varios dispositivos útiles de entrada y salida ya conectados y listos para funcionar. ¡No tener que soldar, cablear o conectar componentes, significa que es perfecto para iniciar con el hardware!

La sección del Kit de Inicio (p6-7) ofrece un rápido resumen de cada parte de la placa y muestra para qué se puede utilizar, mientras que la sección del Microcontrolador (p8-11) da algunos antecedentes sobre la placa Arduino.

### XOD IDE

El entorno de desarrollo integrado (IDE) XOD es un software gratuito de código abierto que permite programar dispositivos basados en microcontroladores, como las placas Grove o Arduino, utilizando 'nodos' visuales en lugar de código escrito. Los nodos pueden representar dispositivos o funciones, y al vincularlos de diferentes maneras puedes crear una amplia variedad de programas diferentes. Programar de forma visual puede ahorrar parte del tiempo y la energía necesarios para aprender un nuevo lenguaje así como grandes cantidades de sintaxis.

La sección XOD IDE (p12-15) proporciona un resumen rápido de las diferentes partes del XOD IDE y de lo que verás cuando cargues el software por primera vez, así como terminología útil utilizada en la programación XOD.

### PÁGINA WEB XOD

La página web de XOD ([www.xod.io](http://www.xod.io)) ofrece mucha información útil para los principiantes, incluyendo tutoriales y una guía de usuario en la pestaña 'Documentation' (Documentación), una base de datos de librerías en la pestaña 'Libraries' (Librerías) y un foro muy útil en la pestaña 'Community' (Comunidad).

### PÁGINA WEB BIOMAKER

El sitio web de Biomaker ([www.biomaker.org](http://www.biomaker.org)) alberga diversos materiales útiles, incluyendo descargas digitales de esta guía, el archivo tutorial que la acompaña y una serie de otros tutoriales y manuales de Biomaker, en la pestaña 'Resources' (Recursos).

También puedes encontrar ejemplos de proyectos Biomaker anteriores, en la pestaña 'Projects' (Proyectos). Con más de 180 proyectos hasta ahora, hay mucha inspiración para Biomaker en ciernes. Los proyectos también están documentados en el Biomaker Hackster Hub ([www.hackster.io/biomaker](http://www.hackster.io/biomaker)).

# El Kit de Inicio

El kit de inicio Biomaker se compone de esta guía para principiantes, y del Kit Grove Beginner para Arduino. Este kit está elaborado por la empresa de hardware de código abierto Seeed Studio y se basa en un sencillo microcontrolador Arduino.

1

## LED

Un diodo luminoso (LED) rojo. Esta luz puede utilizarse como señal de notificación o advertencia en dispositivos.

2

## BUZZER

Buzzer (zumbador) piezoeléctrico incorporado. Puede ser programado para que emita sonidos en diferentes frecuencias.

3

## PANTALLA OLED

Pantalla OLED, la cual puede utilizarse para mostrar tanto imágenes como texto. Se trata de una matriz de 64x128 píxeles que puede mostrar el contenido deseado en blanco y negro.

4

## BOTÓN

Un simple botón que responde a las pulsaciones. Puede ser utilizado como interruptor de encendido/apagado o como disparador.

5

## POTENCIÓMETRO ROTATORIO

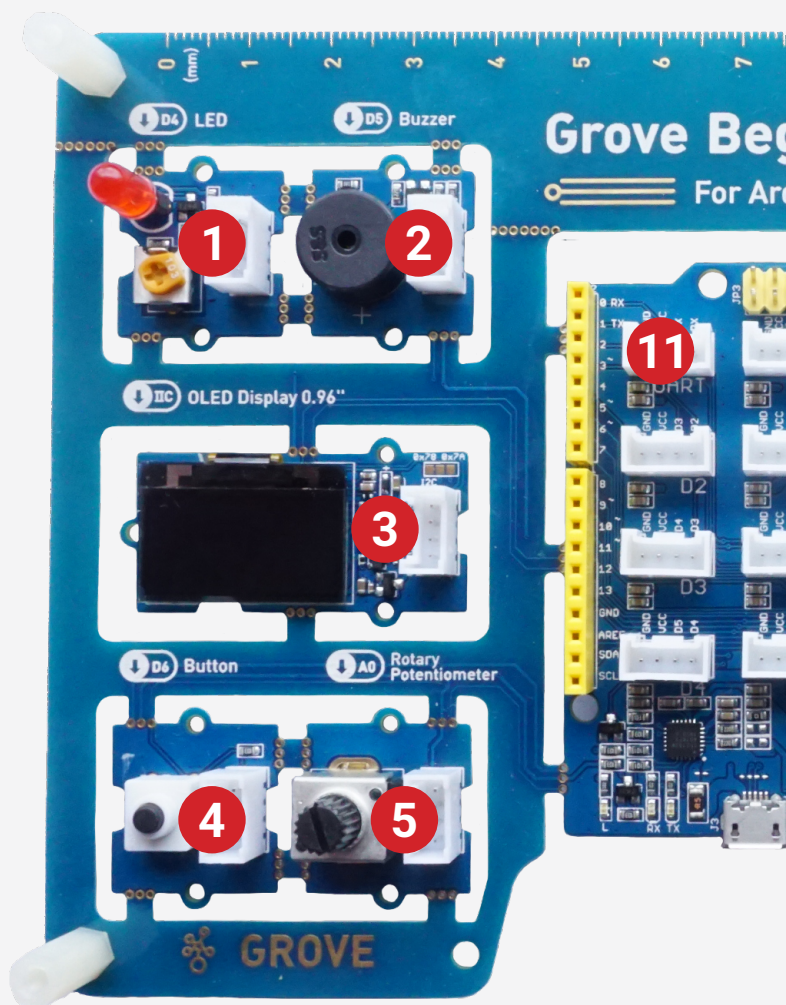
También conocido como 'knob sensor' (sensor de perilla), ya que detecta el ángulo de rotación de la perilla. Puede utilizarse como dial para cambiar el volumen o el brillo.

11

## PLACA DE DESARROLLO DE MICROCONTROLADORES

Basada en la placa de desarrollo Arduino Uno y Seeeduino Lotus, este módulo es el cerebro de la placa.

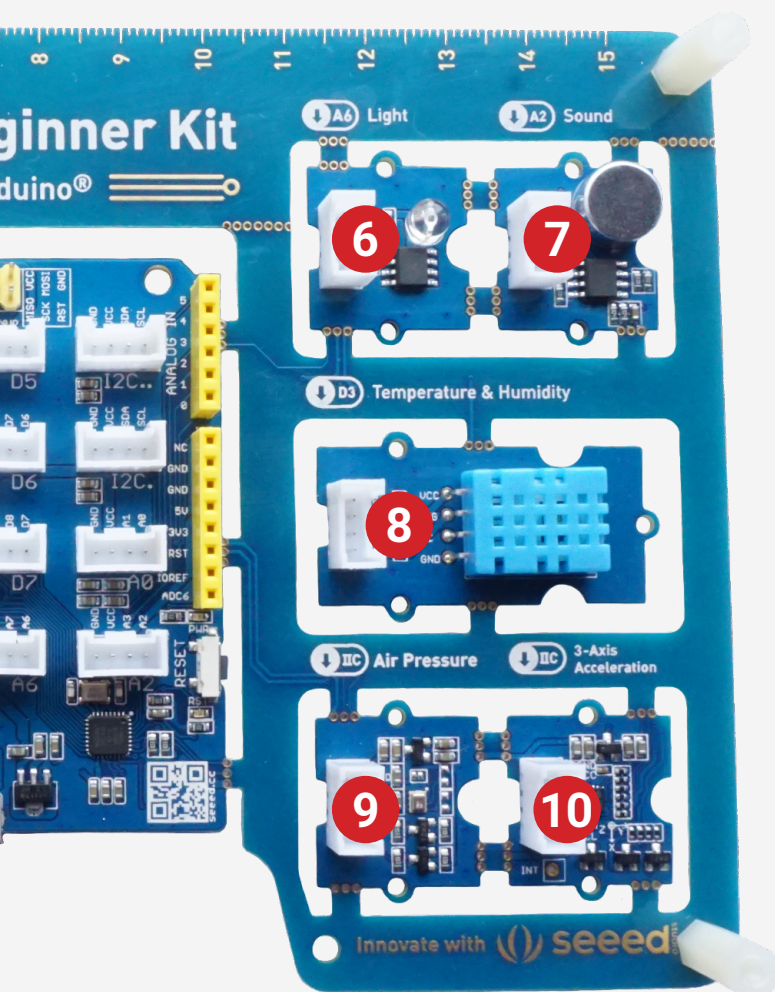
Un chip microcontrolador ATmega328P se encuentra en el núcleo, actuando como una pequeña computadora de baja potencia que puede ser reprogramada para crear cualquier dispositivo que desees.





El kit viene como una placa PCB con varios dispositivos útiles de entrada y salida ya conectados y listos para funcionar. ¡No tener que soldar, cablear o conectar componentes, significa que es perfecto para iniciar con el hardware!

A continuación se presenta un resumen rápido de cada parte de la placa y para qué se pueden utilizar.



# 6

## SENSOR DE LUZ

Una fotorresistencia que puede detectar la intensidad de la luz del entorno.

# 7

## SENSOR DE SONIDO

Un micrófono simple que puede detectar la intensidad del sonido del entorno.

# 8

## SENSOR DE TEMPERATURA Y HUMEDAD

También conocido como higrómetro. Es un sensor digital precalibrado que puede medir la temperatura y la humedad del ambiente. No funciona por debajo de 0°C.

# 9

## SENSOR DE PRESIÓN DEL AIRE

También conocido como barómetro. Un sensor digital de alta precisión que puede medir tanto la presión del aire como la temperatura. También puede ser utilizado para medir la altitud.

# 10

## SENSOR DE ACELERACIÓN DE 3 EJES

También conocido como acelerómetro. Detecta el movimiento de la placa. Puede ser utilizado para medir la orientación, la inclinación, el movimiento o los gestos.

Los enchufes blancos del centro y los enchufes amarillos de los bordes pueden utilizarse para conectar componentes adicionales.

Este módulo también tiene un botón de reinicio para reiniciar el programa en cualquier momento, y un puerto micro USB, para conectar la placa a tu computadora.

En el compartimento derecho de la caja Grove se incluye un cable USB, y en el compartimento izquierdo de la caja Grove se incluyen cables Grove (para conectar los componentes a las tomas blancas).

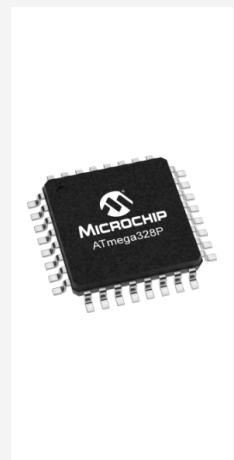


# El Microcontrolador

## ¿Qué es un Microcontrolador?

Un microcontrolador es una pequeña computadora de bajo consumo encapsulada en un diminuto chip electrónico. A diferencia de un ordenador de uso general, como un portátil o un PC, los microcontroladores suelen estar diseñados para realizar una tarea y ejecutar un programa específico. Son de bajo costo y sólo requieren pequeñas cantidades de energía, por lo que suelen utilizarse en dispositivos electrónicos sencillos como en aparatos de cocina, dispositivos médicos implantables y herramientas eléctricas.

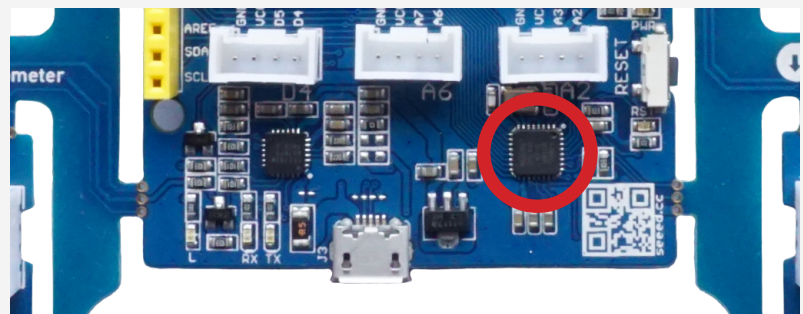
Como cualquier otra computadora, un microcontrolador tiene un procesador para procesar la información (CPU), una memoria 'a largo plazo' para almacenar datos (ROM) y una memoria 'a corto plazo' para acceder a los datos (RAM). Se comunica con el mundo exterior a través de una serie de pines metálicos que pueden enviar (salida) o recibir (entrada) información.



*El microcontrolador ATmega328P utilizado en la placa Arduino*

Una placa de desarrollo de microcontroladores, como la placa Grove Arduino, aloja un chip microcontrolador en una pequeña placa PCB junto con algunas piezas y conexiones adicionales, lo que facilita que cualquiera pueda programar y conectar componentes a un microcontrolador en casa.

Las placas de desarrollo están pensadas para ser baratas y de fácil acceso, y suelen utilizarse para desarrollar prototipos e instrumentos personalizados. Para hacerse una idea de la amplia gama de proyectos que es posible lograr con una placa de desarrollo Arduino, echa un vistazo a la plataforma de documentación de proyectos Hackster en [www.hackster.io/Arduino](http://www.hackster.io/Arduino).

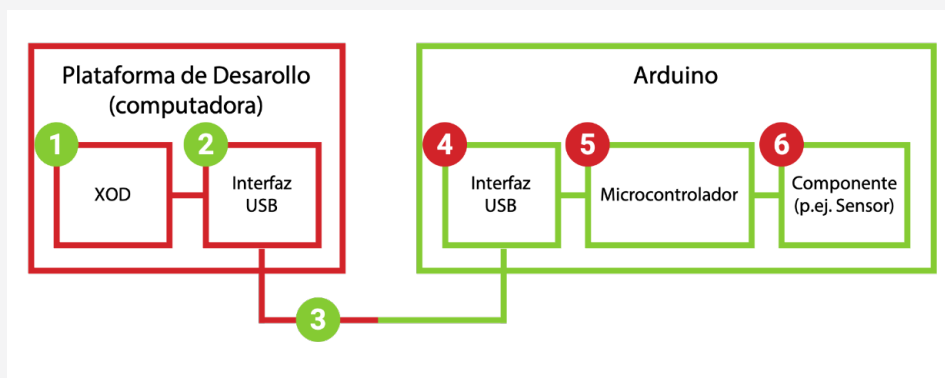


*Ubicación del microcontrolador ATmega328P en la placa Grove (círculo rojo)*



## Controla tu Placa Arduino

Para decirle a la placa qué hacer, la tendrás que conectar a una computadora. Esto se conoce como la 'plataforma de desarrollo' (development host), ya que es donde escribirás y desarrollarás el programa que quieres instalar. El siguiente diagrama explica cómo es transferida la información de tu computadora a la placa. Una vez que el programa ha sido transferido, la placa puede ser desconectada y será capaz de ejecutar el programa deseado independientemente de su ordenador, aunque necesitará una fuente de alimentación alternativa. La placa puede ser programada para realizar una multitud de tareas diferentes dependiendo de los componentes que desee utilizar y de los programas que instale.



Flujo de trabajo para programar tu placa Arduino

- 1** Escribe tu programa utilizando el software XOD.
- 2** Carga tu programa.
- 3** La información se envía de la computadora a la placa Grove a través de un cable USB.
- 4** La información es recibida por la placa Grove.
- 5** El programa se escribe en la memoria del chip microcontrolador. Esto permite que la placa actúe como su propia computadora independiente, llevando a cabo el programa específico que has cargado.
- 6** La información se envía a los componentes integrados en la placa a través de los pines microcontroladores. El programa almacenado en la memoria del microcontrolador indicará a los componentes lo que deben hacer, por ejemplo, encender el buzzer a un determinado tono.

# El Microcontrolador

## Tipos de Comunicación

Hay varias formas diferentes para que la placa se comunique con sus componentes. Se conocen como protocolos de comunicación, y son las diferentes formas en las que se pueden transferir datos entre dispositivos. Qué pin se conecta a qué dispositivo depende del tipo de protocolo de comunicación que se utilice, y eso depende del tipo de dispositivo. A continuación describimos los tres tipos de comunicación que se utilizan en la placa Grove.

Es importante conocer estos diferentes tipos de comunicación, ya que determinarán la conexión de pines de la placa (ver página siguiente), así como la forma de conectar cualquier componente adicional que quieras añadir.

### ANALÓGICA

Los sockets analógicos se utilizan para conectar dispositivos de entrada y salida analógicos. Pueden transmitir señales continuas (lo que significa que pueden tener una cantidad infinita de valores dentro de un rango determinado). La mayoría de los sensores son analógicos.

### DIGITAL

A diferencia de los dispositivos analógicos, los dispositivos digitales no pueden tomar un rango de valores, sólo pueden comunicarse cambiando entre dos estados: encendido y apagado. Estos sockets digitales se utilizan para la mayoría de los componentes no sensoriales.

### I2C

Para los dispositivos que manejan tanto entradas como salidas necesitamos protocolos de comunicación dúplex, que transmiten datos en ambas direcciones. El protocolo utilizado en la placa Grove es I2C.

Los pines de circuito Inter-integrated (I2C) proporcionan una forma de comunicarse con varios dispositivos a la vez. En este caso, múltiples dispositivos son conectados al mismo pin, y a cada dispositivo se le da un 'nombre' (conocido como dirección). Las direcciones se escriben como XXh, siendo XX un código de dos dígitos de números y letras. Por ejemplo, 19h ó 3Ch.

Estos son los tres tipos de pines utilizados en la placa Grove. Las placas Arduino también son capaces de conectarse a dispositivos utilizando otros dos protocolos de comunicación, conocidos como UART y SPI. No utilizaremos estos tipos de comunicación en esta guía, pero si quieres aprender más puedes encontrar un excelente tutorial que compara los protocolos I2C, UART y SPI en el sitio web de SparkFun en [www.learn.sparkfun.com/tutorials/i2c](http://www.learn.sparkfun.com/tutorials/i2c).





## Conexiones de Pines

Una vez cargado el programa en el chip del microcontrolador, el chip debe comunicarse con los componentes de la placa. La información puede enviarse como una salida del microcontrolador a los componentes, o ser recibida por el microcontrolador como una entrada de un componente.

Cada dispositivo de la placa está conectado tanto a la fuente de alimentación como a uno o varios pines (también conocidos como 'puertos') del chip del microcontrolador. Esto permite al microcontrolador comunicarse con los componentes. Es importante saber qué componente está conectado a cada pin, ya que necesitaremos utilizar esta información cuando programemos.

La siguiente tabla indica qué dispositivos de la placa se conectan a qué pines.

| <b>PIN</b> | <b>DISPOSITIVO</b>              | <b>DISPOSITIVO<br/>(NOMBRE INGLÉS)</b> |
|------------|---------------------------------|--|
| A0         | Potenciómetro Rotativo          | Rotary Potentiometer                   |
| A2         | Sensor de Sonido                | Sound Sensor                           |
| A6         | Sensor de Luz                   | Light Sensor                           |
| D3         | Sensor de Temperatura y Humedad | Temperature and Humidity Sensor        |
| D4         | LED                             | LED                                    |
| D5         | Buzzer (Zumbador)               | Buzzer                                 |
| D6         | Botón                           | Button                                 |
| I2C (19h)  | Acelerador de Tres Ejes         | Three-Axis Accelerator                 |
| I2C (77h)  | Sensor de Presión del Aire      | Air Pressure Sensor                    |
| I2C (3Ch)  | Pantalla OLED                   | OLED Screen                            |

# EL XOD IDE

El entorno de desarrollo integrado (IDE) XOD es un software gratuito de código abierto que te permite programar dispositivos basados en microcontroladores, como las placas Grove o Arduino, utilizando 'nodos' visuales en lugar de código escrito.

## 1 TU PATCH

La parte central de tu pantalla muestra el 'patch' del XOD que tienes abierto en ese momento. Un patch es un espacio para crear tu programa. Es como un archivo en otro programa, y puede ser usado para crear un pequeño programa, o una sección de un programa más grande. Puedes crear múltiples 'patches' y almacenarlos juntos en un proyecto.

Cuando abras por primera vez el XOD IDE verás un proyecto llamado 'welcome-to-xod'. Puedes crear un nuevo proyecto navegando a 'File > New Project' (Archivo > Nuevo Proyecto) en la barra de menú.

## 2

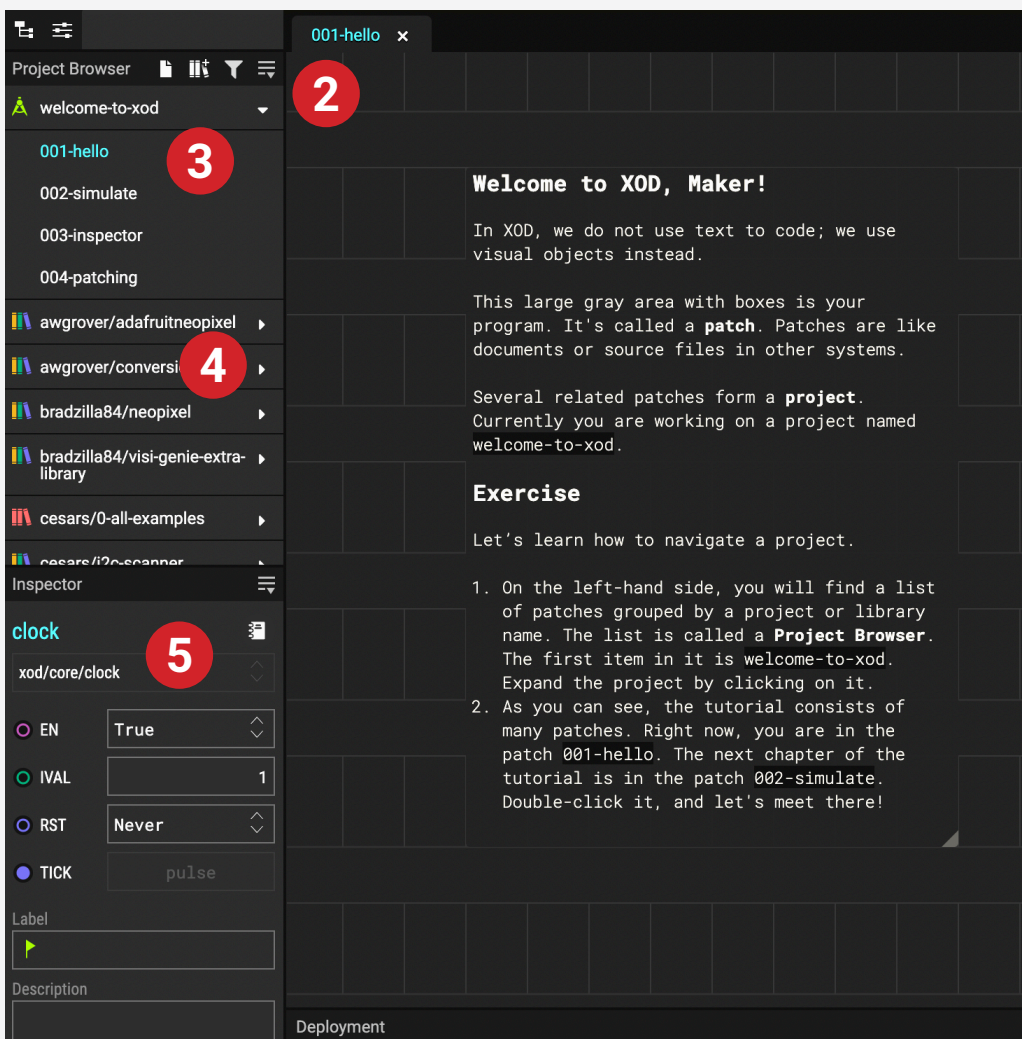


Los cuatro botones a la izquierda del navegador de proyectos representan: **AÑADIR PATCH (ADD PATCH)** Te permite añadir nuevos patches a tu proyecto.

**AÑADIR LIBRERÍA (ADD LIBRARY)** Te permite instalar nuevas librerías. Puedes descargar las librerías que otros usuarios han hecho para ampliar el número de nodos disponibles.

**FILTRO (FILTER)** Te permite filtrar qué bibliotecas y nodos puedes ver.

**MENÚ PROJECT BROWSER (NAVEGADOR DE PROYECTOS)** Te permite minimizar o mover el panel del 'Project Browser' (navegador de proyectos).



## 6 QUICK HELP (AYUDA RÁPIDA)

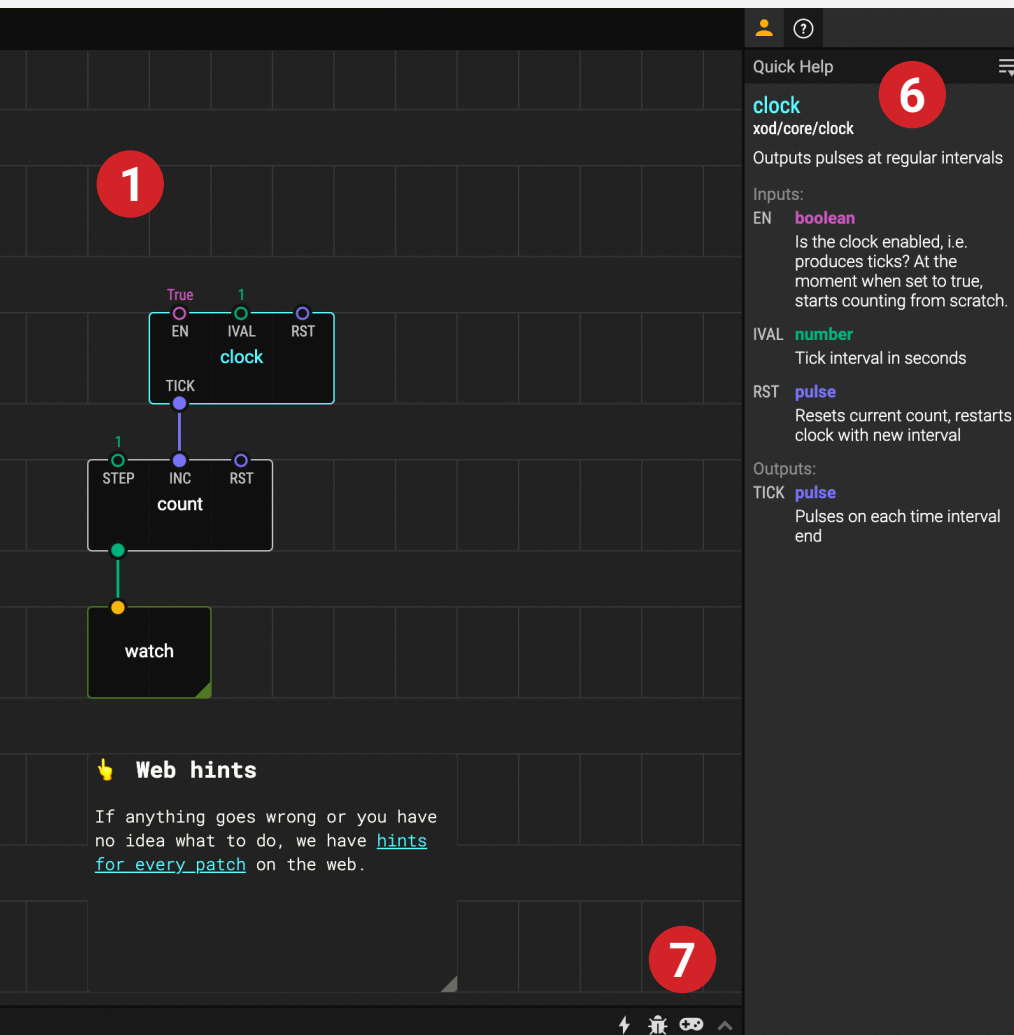
El panel de 'Quick Help' (ayuda rápida) proporciona información sobre cualquier nodo que hayas seleccionado en ese momento. Haz clic en un nodo para ver información sobre lo que hace y lo que hace cada una de sus entradas y salidas ('pines').

Puedes activar y desactivar este panel utilizando el botón de signo de interrogación situado en la parte superior derecha de la pantalla.



Los nodos pueden representar dispositivos o funciones, y al vincularlos entre sí de distintas maneras puedes crear una gran variedad de programas diferentes. Programar de forma visual puede ahorrar parte del tiempo y la energía necesarios para aprender un nuevo lenguaje y grandes cantidades de sintaxis.

A continuación se presenta un resumen rápido de las diferentes partes del XOD IDE, y lo que verás cuando cargues el software por primera vez.



### 3 PROJECT BROWSER (NAVEGADOR DE PROYECTOS): PATCHES DEL PROYECTO

La mitad superior del panel del 'Project Browser' muestra el proyecto que tienes abierto. Si haces click en la flecha desplegable junto al nombre del proyecto, podrás ver todos los patches dentro del proyecto.

### 4 PROJECT BROWSER (NAVEGADOR DE PROYECTOS): LIBRERÍAS

La mitad inferior del panel del 'Project Browser' muestra todas las librerías que has instalado. Cuando utilices XOD por primera vez, éstas se limitarán a las librerías básicas del XOD (p.ej. xod/core). Las librerías con iconos rojos tienen un error en alguna parte de sus patches.

### 5 INSPECTOR

El panel del 'Inspector' muestra información sobre el nodo o el patch que tienes seleccionado en ese momento. Aquí es donde introducirás información y cambiarás las propiedades de tus nodos.

Este panel también contiene las casillas de 'Label' (Etiqueta) y 'Description' (Descripción), que puedes utilizar para ayudar a documentar tus programas.

### 7



Los cuatro botones de la parte inferior del patch representan:

**UPLOAD TO ARDUINO (SUBIR A ARDUINO)**  
Sube tu patch a la placa.

**UPLOAD AND DEBUG (SUBIR Y DEPURAR)**  
Sube el patch y observa lo que ocurre en tu

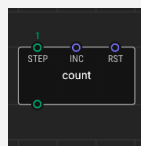
pantalla al mismo tiempo. Útil para probar programas y depurar.

**SIMULATE (SIMULAR)**  
Simula tu programa sin hardware. Útil para comenzar.

**TOGGLE DEPLOYMENT PANE (ACTIVAR PANEL DE DESPLIEGUE)**  
Activa y desactiva el panel de despliegue para ver cómo está funcionando la carga.

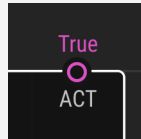
# EL XOD IDE

## Terminología XOD



### Nodos (Nodos)

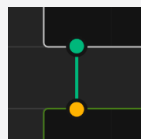
Los nodos son los bloques de construcción de un programa en XOD. Representados como un rectángulo negro con borde blanco, mostrarán su nombre al centro y tendrán un número de pequeñas entradas y salidas circulares en la parte superior e inferior, conocidas como pines. Los nodos pueden representar cualquier cosa, desde hardware (como un LED o un sensor) hasta operaciones matemáticas o lógicas (como sumar, restar, y, o, si, etc.).



### Pins (Pines)

En XOD, 'pin' se refiere a las entradas y salidas de un nodo, los cuales se representan como pequeños círculos. Los pines de entrada se encuentran en el borde superior de un nodo, y los de salida en el borde inferior del nodo. Los pines pueden tener diferentes tipos de datos (véase la página siguiente) y se colorean respectivamente. El nombre de un pin aparecerá debajo del círculo, y el valor actual de un pin aparecerá encima de él.

Nota: para evitar confusiones con los 'pines' del microcontrolador, XOD se refiere a ellos como 'ports' (puertos), es decir, lo que Grove denomina 'pin A0', XOD lo denomina 'port A0'.



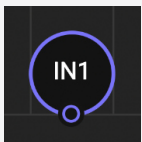
### Links (Enlaces)

Los enlaces se utilizan para conectar nodos. Para crear un enlace, haz clic en un pin de salida de un nodo y luego en un pin de entrada de otro nodo (o viceversa). Una vez que un pin está enlazado, el círculo cambiará a un color sólido. Los pines de un tipo de datos sólo pueden conectarse a otros tipos determinados (por ejemplo, un pin numérico puede conectarse a otro pin numérico, a un pin de cadena o a un pin booleano, pero no a un pin de byte, puerto o pulso). XOD no le permitirá conectar tipos de pines que no funcionen juntos.



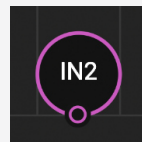
## Tipos de Datos XOD

Los pines en XOD pueden tener diferentes tipos de datos dependiendo de lo que representen. Cada tipo de dato está representado por un color distinto. A continuación se describen los seis tipos de datos principales en XOD. También hay tipos de datos personalizados, pero no los discutiremos aquí. Una explicación completa de cada tipo de datos XOD y cómo interactúan puede encontrarse en [www.xod.io/docs/guide/data-types](http://www.xod.io/docs/guide/data-types) y [www.xod.io/docs/reference/data-types](http://www.xod.io/docs/reference/data-types) respectivamente.



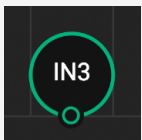
### Pulse (Pulso)

Los pines de pulso son como disparadores. No representan ningún tipo de dato específico, sino que se utilizan para indicar cuándo ha sucedido algo, o para desencadenar que algo suceda en un momento determinado.



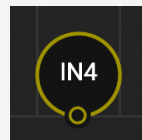
### Boolean (Booleano)

Los pines booleanos representan información lógica, es decir, sólo pueden estar en dos estados: Verdadero (encendido) o Falso (apagado). Pueden usarse como un interruptor, y suelen utilizarse cuando se trabaja con nodos lógicos como 'y', 'o', etc.



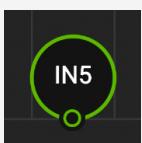
### Number (Número)

Los pines de número son muy sencillos: representan números. Los números pueden ser enteros o fracciones en el rango de  $\pm 16$  millones, y pueden mostrar hasta 6 dígitos significativos.



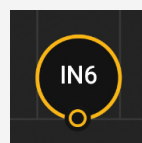
### Byte

Los pines Byte representan bytes, un tipo de datos fundamental en la informática. Pueden escribirse de varias formas, pero nosotros utilizaremos los hexadecimales, que contienen dos dígitos (0-9,A-F) seguidos del sufijo h (por ejemplo, 3Ch).



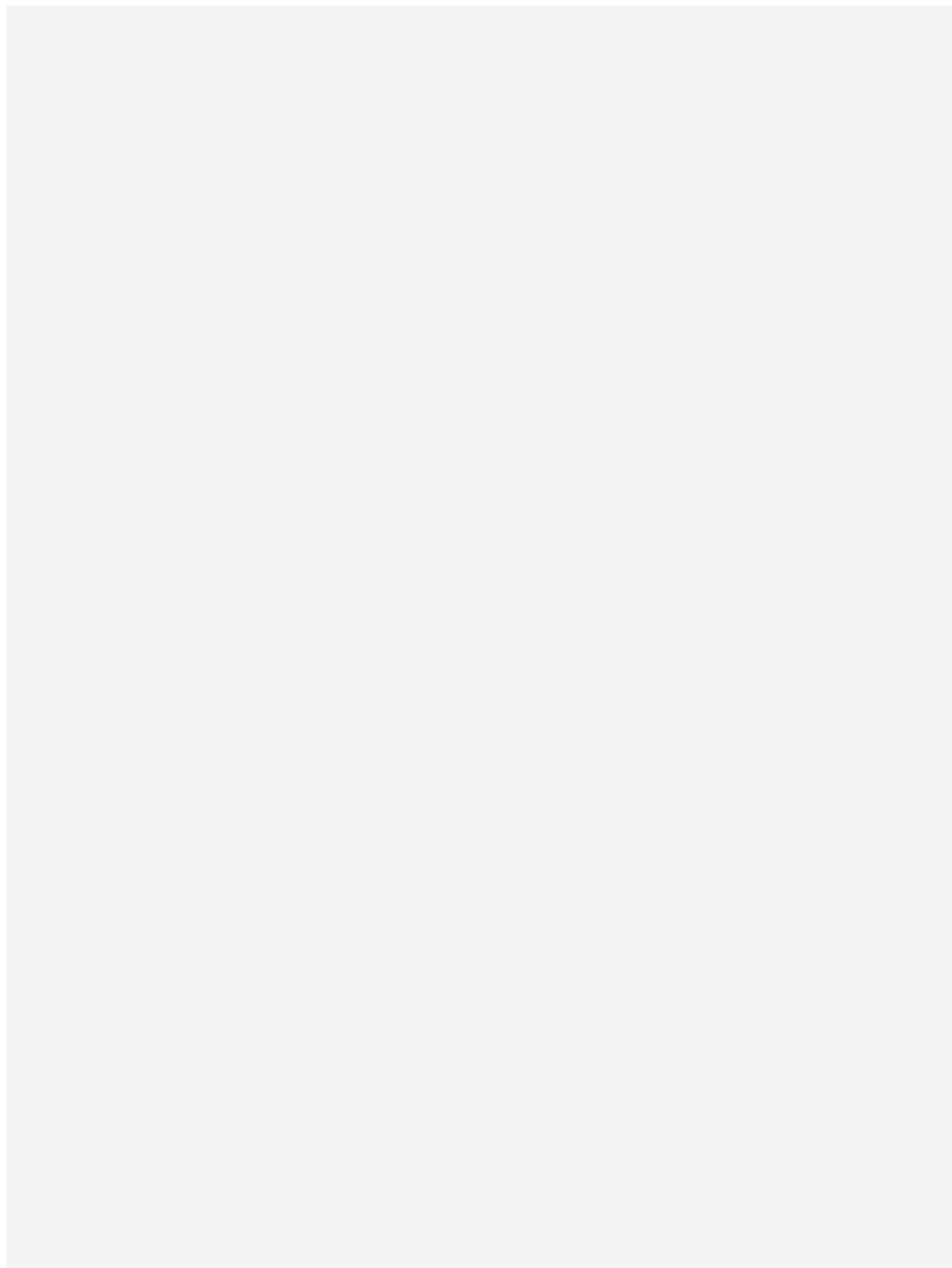
### Port (Puerto)

Los pines de puerto representan los diferentes 'puertos' (pines del microcontrolador) de la placa, es decir, A0- A6 y D0-D13. Utilizaremos estos pines para indicar a los nodos de hardware a qué pin/puerto está conectado el hardware.



### String (Cadena)

Los pines de cadenas representan cadenas de texto. Se utilizan para introducir o leer texto. Por lo general, se trata de un texto introducido por el usuario o que debe ser leído por él, por ejemplo, un texto que debe mostrarse en una pantalla.





# Lección 2: Primeros Pasos



**Prepara tu Placa**



**Probando tu Placa**



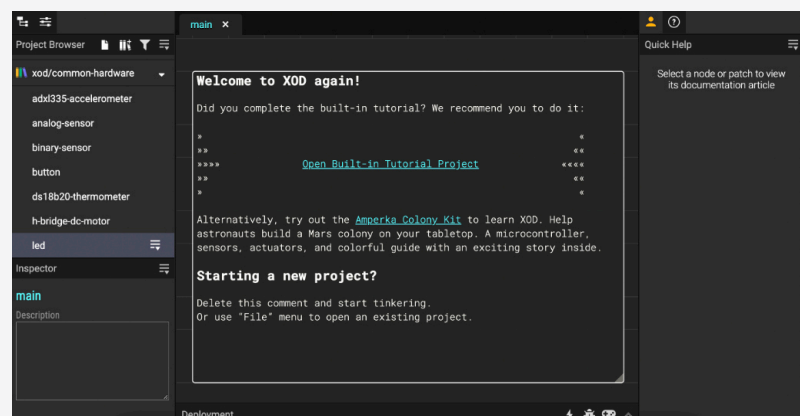
**Dispositivos de Entrada y Salida**

# Primeros Pasos

Este capítulo trata todo sobre cómo tomar los primeros pasos hacia la programación sin código y sobre cómo usar tu placa.

Comenzando por 'Preparando tu placa', aprenderás a cómo preparar tu computadora, incluyendo cómo descargar el XOD IDE, drivers USB y archivos de tutoriales Biomaker.

Lo anterior está seguido de dos tareas: 'Probando tu Placa' y 'Dispositivos de Entrada y Salida'. En la Tarea 1 usarás el XOD IDE para probar tu conexión y programar un solo LED a tu placa. En la Tarea 2 aprenderás a construir un dispositivo sencillo utilizando los módulos de botón y buzzer en tu placa, así como expandir tu conocimiento sobre cómo utilizar XOD.



La pantalla de inicio de XOD

## OBJECTIVOS

Al final de este capítulo deberás poder:

- Preparar tu kit de inicio Biomaker descargando el software y drivers relevantes, conectando tu placa y abriendo el XOD IDE.
- Nombrar las secciones del XOD IDE y comprender para qué se usan.
- Aplicar tu conocimiento sobre el XOD IDE para realizar las siguientes tareas simples: agregar un nodo, cambiar los ajustes de un pin, conectar nodos, agregar una librería.
- Usar el XOD IDE para cargar programas a tu placa.
- Usar el XOD IDE para borrar programas de tu placa.
- Usar tres de los componentes integrados a la placa Grove: el LED, el buzzer y el botón.
- Entender cómo diagnosticar y solucionar problemas de tu programa y localizar ayuda adicional.
- Entender cómo pueden ser utilizados juntos los dispositivos de entrada y salida para construir dispositivos sencillos.



# Prepara tu Placa

## Descargas

### XOD

Para descargar el software gratis XOD, simplemente visita [www.xod.io](http://www.xod.io) y descarga el IDE para escritorio de la página principal de XOD. Necesitarás descargar el IDE correcto para tu sistema operativo (Windows, MacOS, Linux etc.).

Ten en cuenta que el IDE basado en navegador también está disponible, pero no admite hardware, por lo que no es apto para usar con esta guía.

### DRIVER USB

Para que tu computadora se comunique con la placa Arduino necesitará tener el driver correcto instalado, en este caso un driver CP210. La mayoría de los sistemas operativos ya tienen los drivers correctos instalados, incluyendo: Windows 7, MacOS X v10.10.5 (Yosemite) a v10.15.5 (Catalina), Linux y Ubuntu v18.04.2, 64-bit.

Si estás usando uno de los sistemas mencionados anteriormente sugerimos que ignores este paso y continúes con la guía.

Si estás usando un sistema operativo diferente, o estás teniendo problemas para conectarte a tu placa, tal vez tengas que descargar un driver CP210. Las descargas para los sistemas operativos más comunes están disponibles desde Silabs en: [www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers](http://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers).

### TUTORIALES

Para acompañar esta guía, el equipo Biomaker ha creado un archivo tutorial XOD. Este archivo te permitirá trabajar con las tareas en esta guía dentro del entorno XOD.

Puedes decidir trabajar con las tareas utilizando las instrucciones paso a paso en este libro, utilizando el archivo tutorial XOD, o utilizando una combinación de ambos.

Si decides utilizar el archivo tutorial XOD, te sugerimos que te tomes el tiempo de leer la introducción y la información al inicio de cada capítulo en esta guía, de lo contrario puedes perderte de información útil.

Puedes descargar el archivo tutorial XOD en la página web Biomaker en: [www.biomaker.org/nocode-programming-for-biology-handbook](http://www.biomaker.org/nocode-programming-for-biology-handbook).



# Probando tu Placa

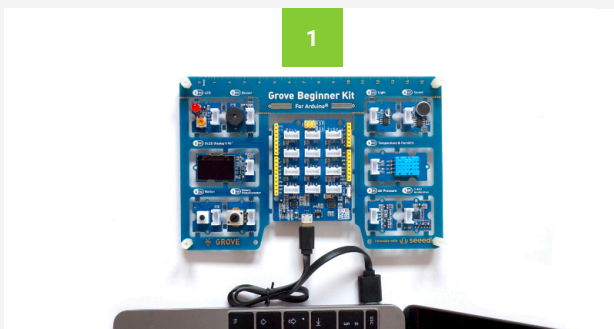
## Tarea 1: Requisitos

- Computadora con MacOS, Windows o Linux (software XOD y driver USB instalados)
- Grove Beginner Kit for Arduino (módulos LED y de botón)
- Cable USB-A a micro USB

Ahora que has descargado el software, ¡Ya estás listo para ponerte manos a la obra!

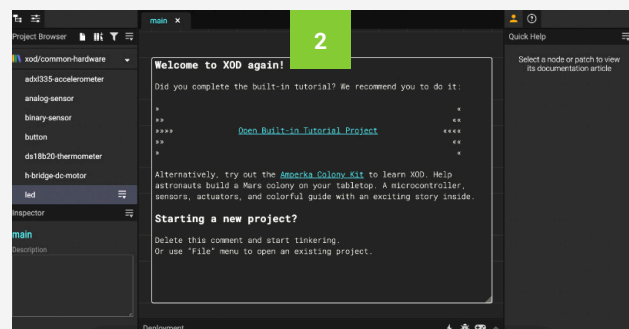
Esta tarea te guiará sobre cómo conectar tu placa a la computadora y cargar tu primer programa utilizando XOD. Utilizaremos la luz LED de la esquina superior izquierda de la placa para probar nuestra conexión.

Puedes notar que cuando conectas la placa por primera vez se enciende la pantalla OLED en la parte central izquierda. Esto es parte del modo de demostración incorporado en la placa. Puedes aprender más sobre esto en el Manual de Usuario de Grove.



### CONECTA TU PLACA

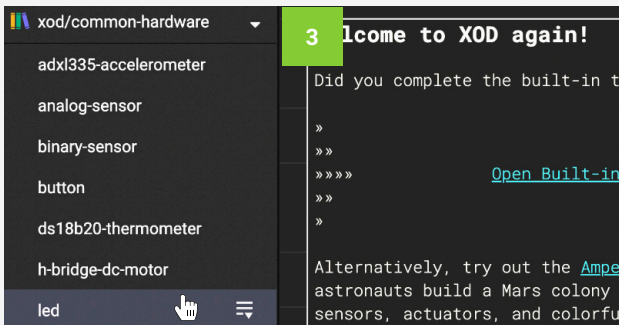
Utiliza el cable USB incluido para conectar tu placa a la computadora. Este cable se encuentra en el compartimento derecho de la caja de Grove. Enchufa el extremo micro USB en el socket situado en la parte inferior de la sección central de la placa, y el extremo USB estándar (USB-A) a tu computadora.



### ABRE EL IDE XOD

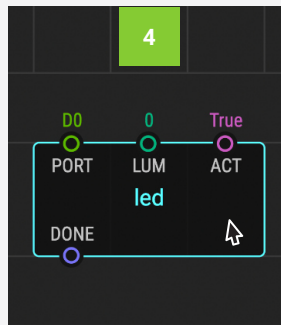
Abre el software XOD en tu computadora.

Si estás utilizando el archivo tutorial proporcionado, abre este archivo en XOD. Puedes seguir las instrucciones en los patches tuto101-tuto114 para completar esta tarea.



### AÑADE UN NODO LED

Usando el 'Project Browser' en la parte superior izquierda, encuentra la librería *xod/common-hardware*, haz clic en el menú desplegable y encuentra el nodo llamado 'led'. Haz clic en este nodo y arrástralo al patch. Hay varias maneras de añadir un nuevo nodo. Consulta el cuadro abajo para obtener más información.



### SELECCIONA EL NODO

Haz clic en el nodo *led*. El borde se volverá azul y aparecerá más información en el panel de Inspector (inferior izquierda).

## Añadir Nodos a tu Patch

Hay varias formas de añadir un nodo a tu patch XOD, y la que utilices ¡Depende completamente de ti!

#### 1. ARRASTRAR DESDE LAS LIBRERÍAS

Como se ha descrito en el **Paso 3**. Si conoces la librería en la que se encuentra el nodo, puedes encontrar la biblioteca en el Project Browser, hacer clic en el menú desplegable, hacer clic en el nodo, arrastrarlo al patch y soltarlo.

#### 2. DOBLE CLIC EN EL PATCH

Si conoces el nombre del nodo que deseas, o quieres buscar un nodo, puedes utilizar la barra de búsqueda. Haz doble clic en cualquier lugar del patch y aparecerá la barra de búsqueda. Comienza a escribir el nombre del nodo y aparecerán las opciones. Haz clic en el nodo correcto y se insertará en tu patch.

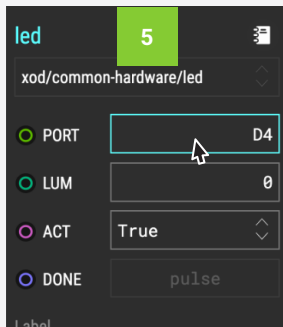
#### 3. MÉTODO ABREVIADO DE TECLADO

Es similar a hacer doble clic en el patch. Haz clic en cualquier parte del patch y pulsa 'i' en tu teclado. Esto hará que aparezca la misma barra de búsqueda que la anterior.

#### 4. BARRA DE MENÚ

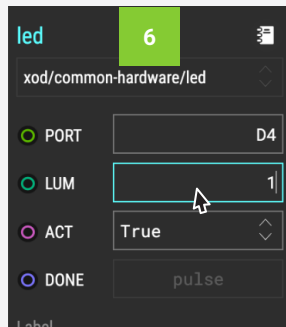
Esta es una tercera forma de hacer aparecer la barra de búsqueda. Selecciona 'Edit > Insert Node...' (Edición > Insertar nodo...) en la barra de menú.

# Probando tu Placa



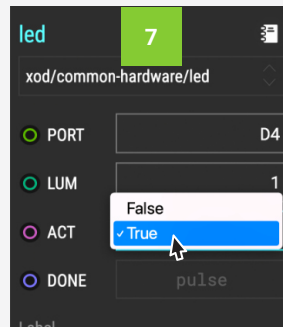
## COLOCA EL PORT PIN

El LED de la placa está conectado al puerto D4, así que haz clic en el cuadro de texto junto a PORT y ponlo en D4 escribiendo 'D4'.



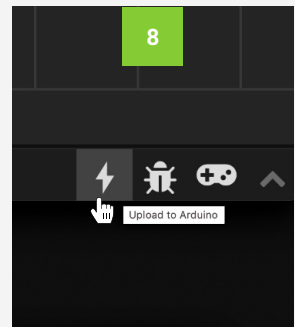
## COLOCA EL LUM PIN

LUM significa luminosidad, es decir, cuán brillante es el LED en una escala de 0-1. Escribir '1' (nivel más brillante).



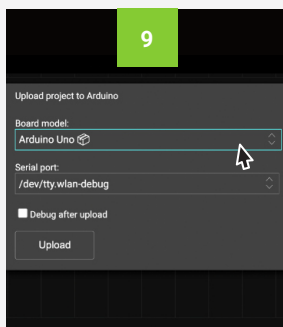
## COLOCA EL ACT PIN

ACT es un pin booleano que sólo puede ser verdadero o falso. Selecciona 'True'. Esto asegura que el LED responda.



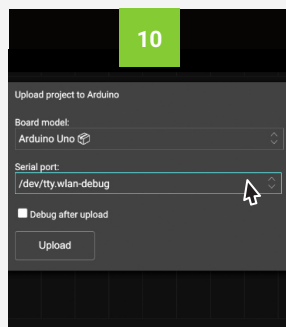
## CARGAR

Haz clic en el pequeño ícono del rayo en la parte inferior derecha, o selecciona 'Deploy > Upload to Arduino...' en la barra de menú.



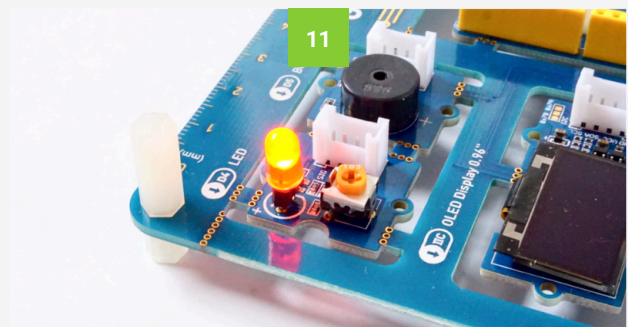
## ESTABLECE EL MODELO DE PLACA

Utiliza el menú desplegable para seleccionar 'Arduino Uno' o 'Arduino/Genuino Uno'.



## ESTABLECE PUERTO SERIE

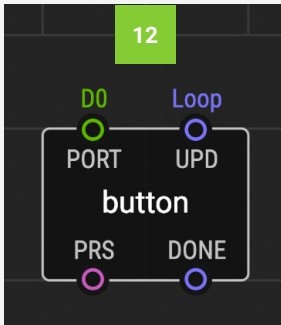
Utiliza el menú desplegable para seleccionar la opción que termine con '(Silicon Labs)'.



## ¡VIGILA TU LED!

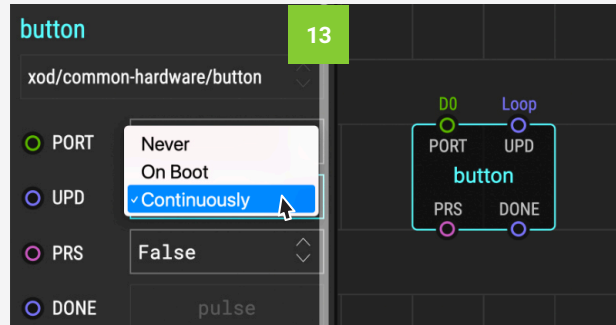
Haz clic en cargar y observa el LED de tu placa. ¡Debería iluminarse!

Si no es así, ¡no te preocupes! Echa un vistazo al cuadro de diagnóstico y resolución de problemas en la página siguiente (p25).



### AÑADE UN NODO BUTTON (BOTÓN)

Ahora añadimos otro nodo. Usando una de las formas en la p21, añade un nodo 'button' de la librería *xod/common-hardware*.



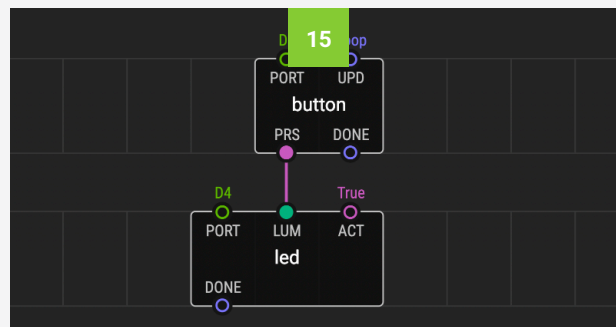
### AJUSTA PINS BUTTON: UPD PIN

El pin UPD especifica la frecuencia con la que se actualiza el programa. Puede establecerse en 'Never' (Nunca), 'On Boot' (Al Arrancar) o 'Continuously' (Continuamente). Selecciona 'Continuously', para que cada vez que pulsemos el botón se lea instantáneamente.



### AJUSTA PINS BUTTON: PORT PIN

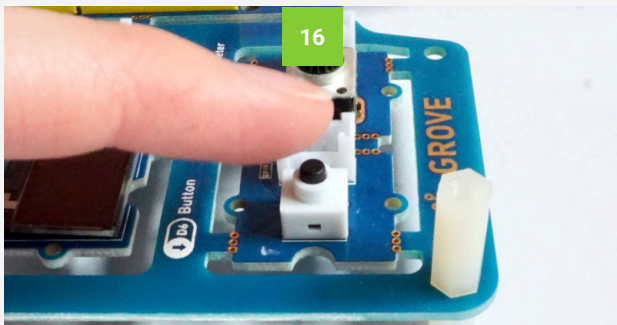
Al igual que con el nodo *led*, PORT especifica a qué puerto está conectado el botón. Ajústalo a 'D6'.



### CONECTA LOS NODOS

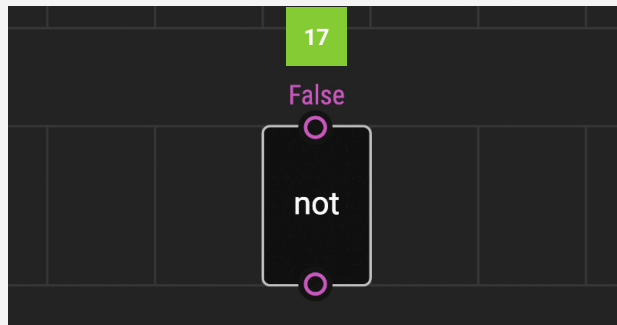
Queremos que el LED se encienda cada vez que pulsemos el botón. Conectamos el pin de salida de *button* PRS ('press', pulsar) al pin de entrada de *led* LUM haciendo clic en PRS y luego en LUM. Ahora cuando seleccionas el nodo *led* no podrás configurar el pin LUM, porque su valor está determinado por el nodo *button*.

# Probando to Placa



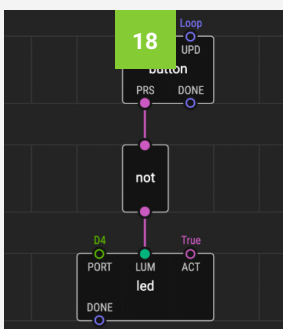
## CARGA Y PRUEBA

Carga el patch y ve lo que sucede. Notarás que el programa está al revés. El LED está encendido y se apaga cuando lo pulsas. Esto se debe a que los botones de la placa están configurados para estar encendidos por defecto, y se apagan al pulsarlos. Podemos arreglar este programa con un nodo lógico.



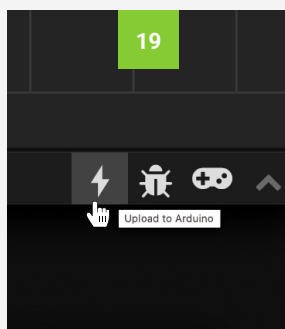
## AÑADE UN NODO NOT (NO)

Para invertir la señal del botón podemos utilizar un tipo diferente de nodo que represente una función lógica, en lugar de una pieza de hardware. Inserta un nodo 'not' (no) de la librería *xod/core*.



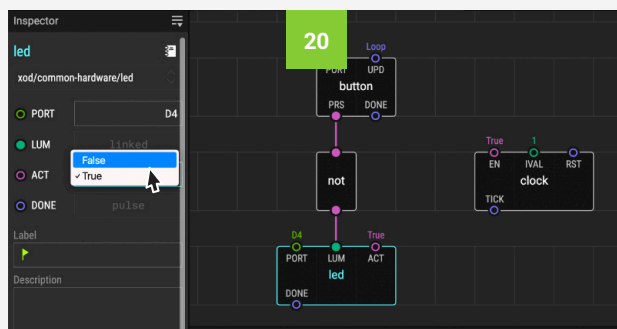
## RECABLEA EL PATCH

Elimina el enlace entre *button* y *led*. Conecta *PRS* al pin de entrada de *not*, y el pin de salida de *not* a LUM.



## CARGA Y PRUEBA

Ahora intenta cargar tu programa de nuevo. Esta vez debería funcionar como estaba previsto.

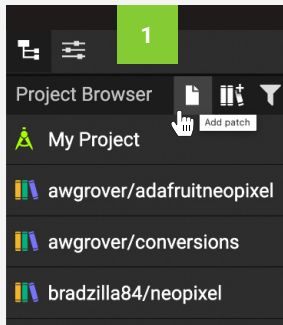


## ¡EXPERIMENTA!

¡Felicidades!, ya has hecho un programa sencillo que utiliza una entrada (el botón) y una salida (el led) para afectar al cambio. ¿Por qué no intentas experimentar con este patch? Juega con algunos pines. Por ejemplo, cambia el pin ACT del led, o enlaza un nodo 'clock' (reloj) al pin UPD del botón y ve qué sucede.

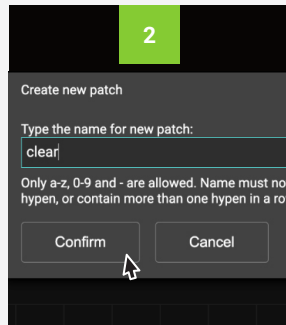
## Borrar el Programa de la Placa

Una vez cargado un programa en tu placa, permanecerá allí y se reiniciará cada vez que enciendas la placa. Cada vez que cargues un nuevo programa se escribirá sobre el programa anterior. No es necesario borrar el programa de la placa antes de cargar un nuevo, pero si deseas reiniciar tu placa puedes hacerlo manualmente cargando un patch en blanco en XOD.



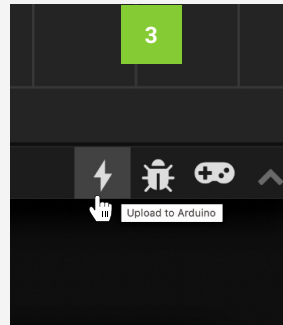
### HAZ UN NUEVO PATCH

Añade un nuevo patch. Haz clic en 'Add patch' (Añadir patch) en el Project Browser o seleccionar 'File > New Patch...' en el menú.



### NOMBRA EL PATCH

Escribe un nombre para tu nuevo patch, por ejemplo, 'clear' (borrar) y haz clic en confirmar o pulsa la tecla Enter.



### CARGAR

Carga el patch vacío haciendo clic en el botón de carga como antes. Esto apagará el LED y borrará el programa de la placa.



## Diagnóstico y Resolución de Problemas

Si el LED no se enciende de inmediato, hay que comprobar algunas cosas:

- 1. ¿LA PLACA ESTÁ CONECTADA CORRECTAMENTE?**  
Si tu placa está conectada correctamente, la luz de encendido situada en el lado derecho del módulo Seeeduno debería encenderse. Si no es así, asegúrate de que el cable USB está completamente conectado.
- 2. ¿HAS CONFIGURADO CORRECTAMENTE LOS PARÁMETROS DE TU NODO?**  
Configurar los parámetros incorrectos es un error común. En este caso debes asegurarte de que los pines están configurados de la siguiente manera: `PORT=D4 LUM=1 ACT=True`
- 3. ¿HAS CARGADO UTILIZANDO EL MODELO DE PLACA Y EL PUERTO SERIE CORRECTOS?**  
Después de hacer clic en el botón de carga, asegurar que tienes el modelo de placa y el puerto serie correctos seleccionados. Usa los menús desplegables para seleccionar 'Arduino Uno'/'Arduino/Genuino Uno' y '...(Silicon Labs)'.
- 4. ¿ES NECESARIO INSTALAR UN CONTROLADOR USB?**  
Si el XOD no reconoce tu tarjeta, es posible que tengas que instalar un controlador USB CP210 (véase la p19).

¿Aún necesitas ayuda? XOD proporciona nodos *watch* y *tweak* que son útiles para solucionar problemas y depurar tu programa. Lee más sobre ellos en la página 33 de esta guía, o echa un vistazo a la guía de depuración de XOD en [www.xod.io/docs/guide/debugging](http://www.xod.io/docs/guide/debugging). Para más ayuda, siempre puedes contactar al equipo de Biomaker en [coordinator@synbio.cam.ac.uk](mailto:coordinator@synbio.cam.ac.uk).

# Dispositivos de Entrada y Salida

## Tarea 2: Requisitos

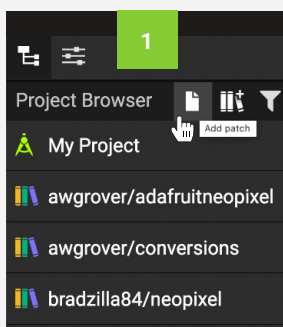
- Computadora con MacOS, Windows o Linux (software XOD y driver USB instalados)
- Grove Beginner Kit for Arduino (módulos de buzzer, botón y potenciómetro rotativo)
- Cable USB-A a micro USB

¡Muy bien! Ahora entiendes los principios básicos del uso del XOD, y puedes programar tu placa para controlar dos de los dispositivos de la placa: el LED y el botón.

Al igual que el botón y el LED, la mayoría de los dispositivos que utilizarás pueden agruparse en dos categorías generales: entradas y salidas.

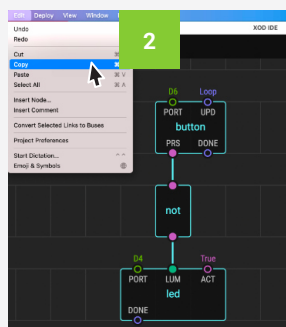
Entender cómo controlar las diferentes entradas y salidas, y cómo combinarlas, es clave para fabricar dispositivos útiles.

En esta tarea nos basaremos en nuestros conocimientos para añadir dos nuevos dispositivos a nuestra caja de herramientas. El buzzer y el potenciómetro rotativo (también conocido como sensor de perilla).



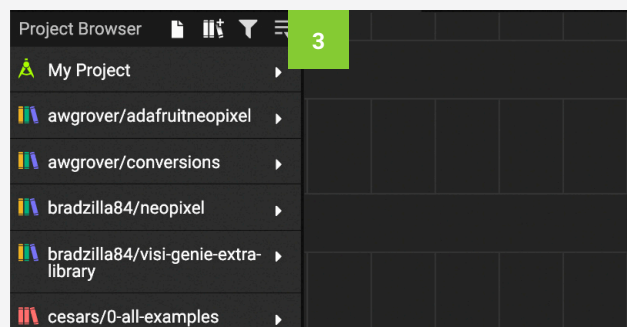
### HAZ UN NUEVO PATCH

Sigue las instrucciones de la página 25 para abrir un nuevo patch. (Si estás utilizando el archivo del tutorial, pasa al tuto201).



### COPIA TU PATCH

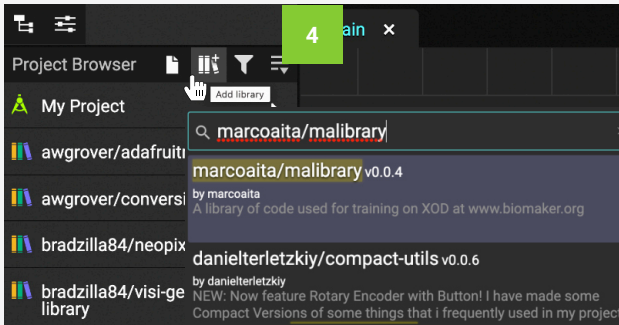
Arrastra un recuadro alrededor de los tres nodos del último patch. Copia y pega en tu nuevo patch ¡Esto nos ahorrará trabajo!



### ENCONTRAR UN NODO BUZZER

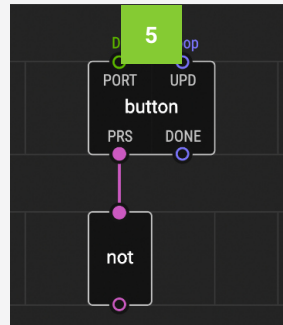
No hay ningún nodo preinstalado para representar el buzzer, pero varios han sido creados por miembros de la comunidad XOD. Usaremos uno de la librería llamada *marcoaita/malibrary*.





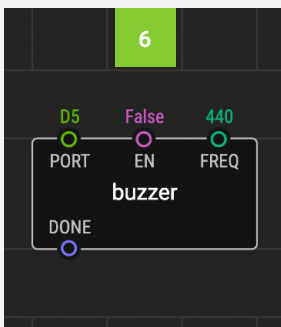
### AÑADE UNA LIBRERÍA

Añade esta librería haciendo clic en el botón 'Add library' (Añadir librería) del Project Browser o navegando hasta 'File > Add Library...' (Archivo > Añadir Librería...). Escribe el nombre completo de la librería y, cuando aparezca, haz clic en ella para instalarla. Si aparece un mensaje de error pidiendo la instalación de dependencias, acéptalo.



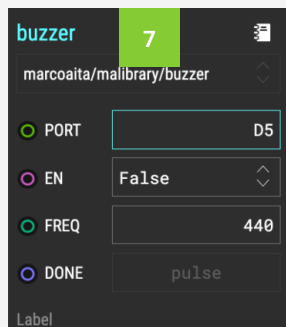
### BORRAR NODO LED

Esta vez queremos utilizar el buzzer como salida en lugar del LED. Haz clic en el nodo *led* y elimínalo.



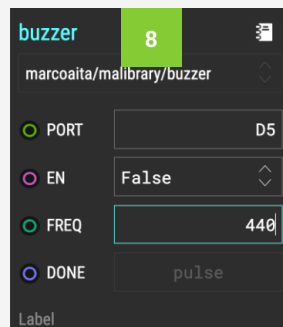
### AÑADIR UN NODO BUZZER

Ahora que la librería está instalada puedes buscar el nodo *marcoaita/malibrary/buzzer* y añadirlo como siempre.



### AJUSTAR LOS PINES BUZZER: PORT

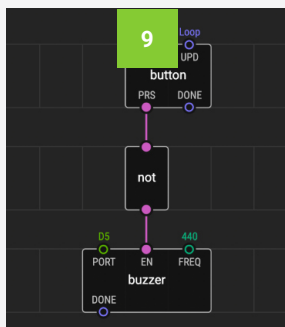
El buzzer está conectado al puerto D5, así que asegúrate de que el pin PORT está ajustado a 'D5'.



### AJUSTAR LOS PINES BUZZER: FREQ

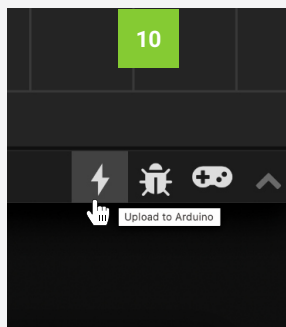
El pin FREQ establece la frecuencia y el tono del buzzer. Puedes dejarlo como 440, o cambiarlo para ver qué pasa.

# Dispositivos de Entrada y Salida



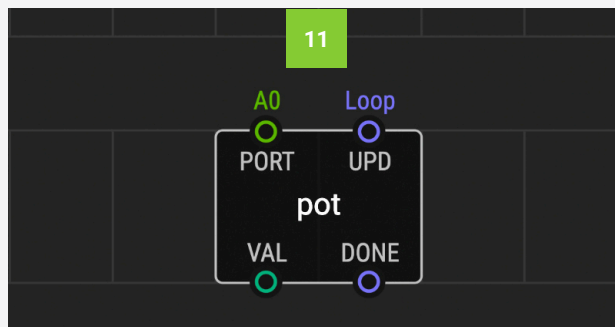
### VOLVER A CONECTAR LOS NODOS

Conecta el pin de salida de *not* al pin EN ('enable', habilitado) de *buzzer*. Esto habilitará al buzzer cuando se pulse el botón.



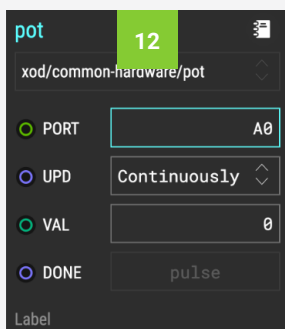
### CARGAR Y PROBAR

Ahora intenta cargar tu programa. El buzzer debería encenderse cuando se pulsa el botón.



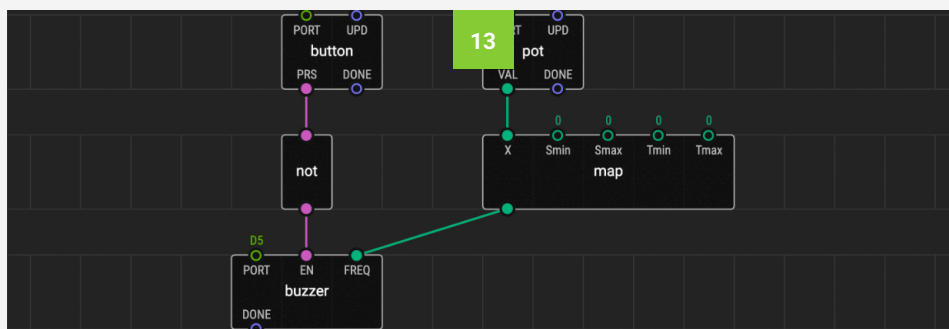
### AÑADE UN NODO POT

Ahora vamos a añadir una segunda entrada. Podemos usar el potenciómetro rotativo incorporado (perilla) para ajustar la frecuencia del sonido del buzzer. Para representar el potenciómetro podemos utilizar el nodo *pot* de la librería *xod/common-hardware*. Añade un nodo *pot* al patch.



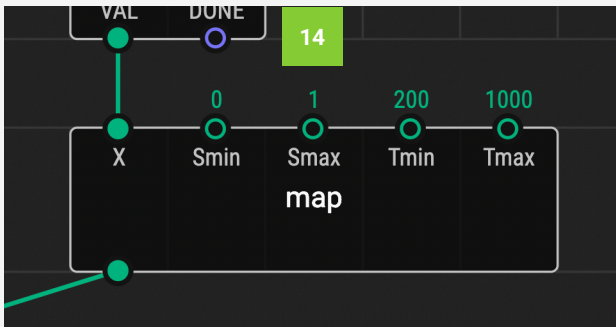
### AJUSTAR PINES POT

El potenciómetro está conectado al puerto A0, por lo tanto, ajusta PORT a A0. Ajusta UPD a 'Continuously'.



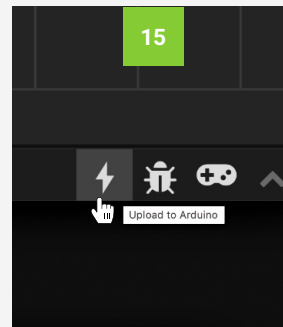
### VALORES DE MAP: AÑADIR Y CONECTAR UN NODO MAP

Podríamos conectar el pin de salida VAL (valor) directamente a FREQ. Sin embargo, esto no funcionaría bien, ya que la salida VAL oscila entre 0 y 1, y las frecuencias emitidas por el buzzer son más altas. Para evitar esto, podemos añadir un nodo *map*. Esto mapeará tu rango de entrada a un nuevo rango de salida, así podemos cambiar la escala 0-1 del potenciómetro a una escala mayor que el buzzer pueda utilizar. Añade un nodo *map* desde *xod/math*. Conecta el pin de *pot* VAL al pin de *map* X y el pin de salida de *map* al pin FREQ.



#### AJUSTAR PINS MAP

Smin y Smax establecen el rango de origen, mientras que Tmin y Tmax establecen el rango de destino. Ajusta Smin a '0' y Smax a '1'. Ajusta Tmin a '200' y Tmax a '1000'.



#### CARGAR Y PROBAR

Ahora intenta cargar tu programa. Utiliza el botón para encender y apagar el buzzer, y el potenciómetro para ajustar la frecuencia.

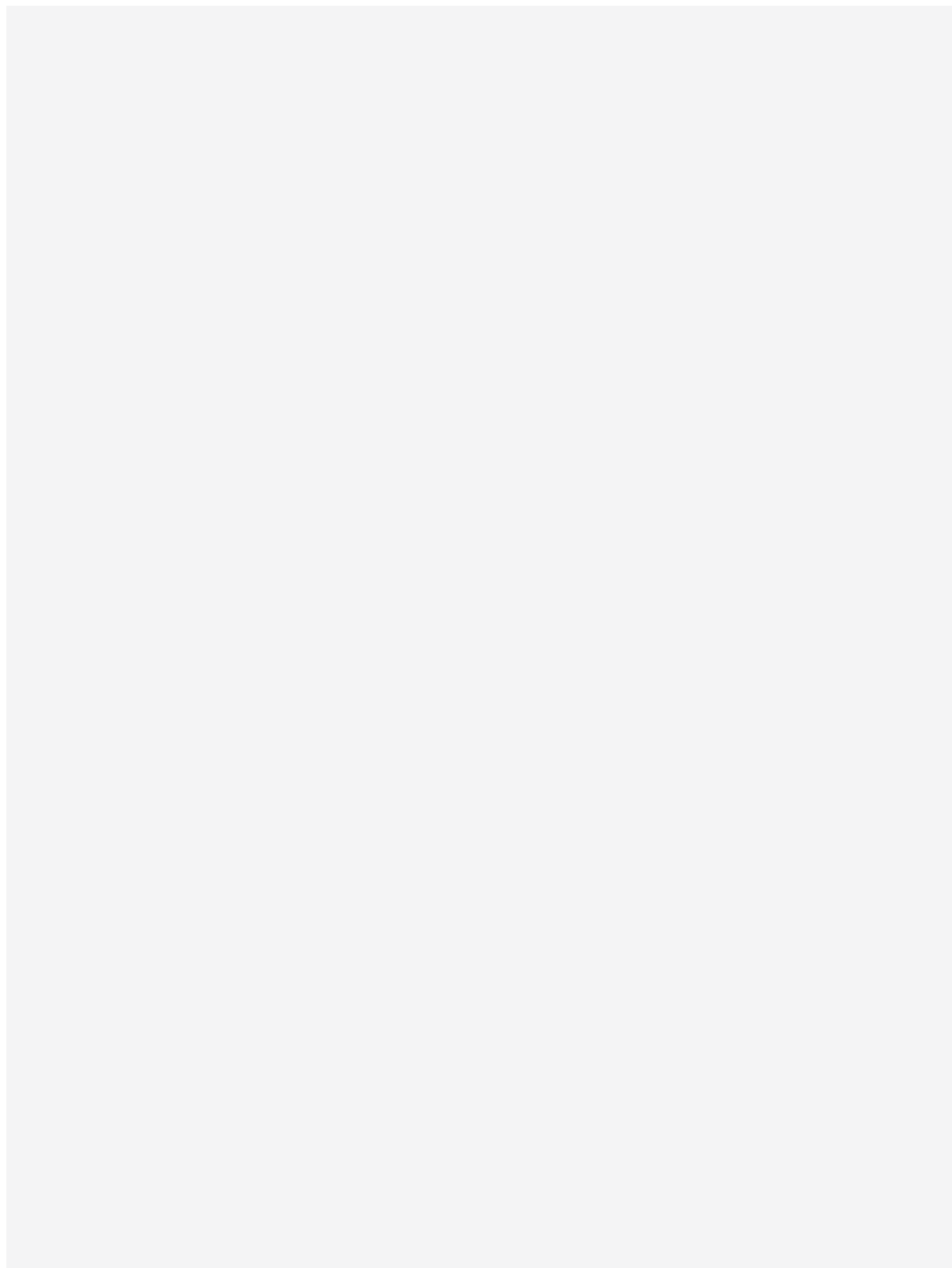
## Construyendo Dispositivos Complejos

Cuando construyas dispositivos biológicos, tendrás que combinar una serie de entradas y salidas para crear un programa y un dispositivo funcionales. A menudo recibirás entradas, por ejemplo, de un botón de encendido/apagado o de un sensor, y luego utilizarás estas entradas para crear la salida deseada, por ejemplo, mostrar una lectura, enviar datos a un computadora o mover un motor.

El ejemplo del botón buzzer y el del potenciómetro utilizado aquí es un ejemplo muy simple, pero el principio se aplica también en sistemas más complejos. Utilizar XOD permite visualizar este flujo de información desde la entrada hasta la salida, lo que puede ser útil y a veces más intuitivo que la codificación tradicional basada en texto.

En la próxima lección comprenderemos mejor lo que es posible en XOD explorando una variedad de nodos y procesos útiles utilizando una gama de dispositivos incorporados en la placa.

Cuando estés listo para explorar más allá de las capacidades del kit de inicio, la página web 'Resources' (Recursos) de Biomaker explora una variedad de dispositivos de entrada y salida comunes que son útiles para la construcción de dispositivos biológicos.



# Lección 3: Explora XOD



**Nodos Tweak y Watch**



**Nodos Flip, Clock y Count**



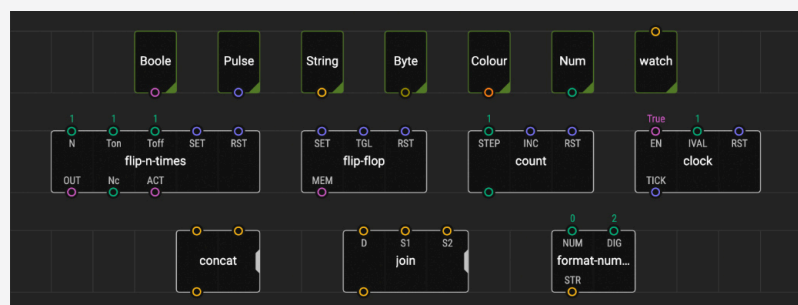
**Nodos Concat, Join y Format-Number**

# Explora XOD

Este capítulo explorará algunos de los nodos más útiles que ofrece XOD. Estos nodos se utilizan muy comúnmente cuando se construyen instrumentos simples, y te dará una buena base para comenzar a explorar dispositivos más complejos.

Las secciones de este capítulo se dividen en tres tareas. En primer lugar, como continuación de la Tarea 2 del capítulo anterior, la Tarea 3 explora los 'Nodos Tweak y Watch', que son útiles para simular y la solución de problemas. En segundo lugar, la Tarea 4 examina los 'Nodos Flip, Clock y Count', que son útiles para asegurar la correcta sincronización de los programas. Por último, la Tarea 5 examina los 'Nodos Concat, Join y Format- Number' que son útiles para usar y formatear texto en XOD.

Este capítulo también te invita a experimentar con el uso de XOD. Ofrece sugerencias sobre cómo ampliar las tareas, y te anima a empezar a pensar en las diferentes formas en las que puedes lograr un resultado deseado utilizando la programación sin código.



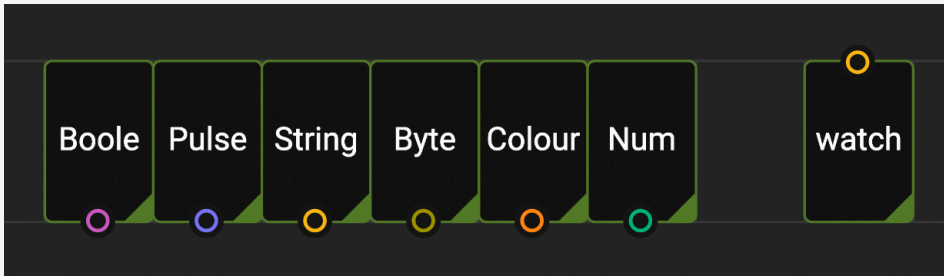
Nodos XOD útiles

## OBJETIVOS

Al final de este capítulo deberás ser capaz de:

- Explicar la función de los siguientes nodos XOD: tweak, watch, flip-n-times, flip-flop, clock, count, concat, join, format-number.
- Aplicar tus conocimientos sobre estos nodos para empezar a construir programas sencillos.
- Utilizar el XOD IDE para crear y guardar un nuevo proyecto.
- Utilizar el XOD IDE para 'cargar y depurar' programas, permitiéndote ver y editar tu programa en tiempo real.
- Utilizar dos más de los componentes incorporados en la placa Grove: el sensor de temperatura y humedad y el sensor de presión atmosférica.
- Construir y comparar diferentes versiones de un programa para lograr diferentes funciones y resultados.
- Experimentar con los programas que has construido cambiando los parámetros y explorando nuevos nodos.
- Saber dónde encontrar más información sobre los nodos básicos disponibles en XOD.

# Nodos Tweak y Watch



## NODOS TWEAK

Los nodos 'tweak' (de ajuste) proporcionados en XOD son una gran manera de editar tus programas mientras se están ejecutando. Se utilizan junto con las funciones 'Simulate' (Simular) y 'Upload and Debug' (Cargar y Depurar) de XOD para editar los programas en tiempo real, lo que significa que no tienes que volver a cargar el programa cada vez que quieras hacer un pequeño cambio, como alterar el valor de un pin.

Hay múltiples nodos tweak disponibles dependiendo del tipo de pin que quieras cambiar, y tendrás que usar el nodo tweak correspondiente al tipo de pin; es decir, *tweak-boolean*, *tweak-pulse*, *tweak-byte*, *tweak-color* y *tweak-number*. También hay varios nodos *tweak-string* dependiendo del tamaño de la cadena que quieras introducir, por ejemplo, *tweak-string-16* te permite introducir una cadena de hasta 16 caracteres, mientras que *tweak-string-128* te permite introducir hasta 128 caracteres.

Para utilizar los nodos tweak necesitarás usar el botón 'Upload and Debug' en lugar del botón 'Upload to Arduino', ya que éste abre la función 'Debugger' (depurador) de XOD, que te permite editar los nodos en vivo. Para editar un nodo tweak en el Debugger, haz clic en el nodo y podrás hacer cambios en el Inspector mientras el programa se está ejecutando.

## NODO WATCH

El nodo 'watch' (visualizar) es lo contrario de un nodo tweak. En lugar de permitirte introducir datos, te permite ver los datos de salida mientras el programa se está ejecutando. Conectando un nodo *watch* a cualquier pin de salida te permite ver el valor actual de ese pin. Esto es útil para poder visualizar lo que el programa está haciendo, y dónde están ocurriendo cualquier problema.

Un nodo *watch* puede conectarse a cualquier salida, excepto a una salida de 'pulse' (pulso). Para visualizar la salida de un pin de pulso, puedes utilizar un nodo *count* (de conteo) en combinación con un nodo *watch*. Esto se discute más adelante en la página 40.

Al igual que los nodos *tweak*, el nodo *watch* deben utilizarse junto con la función Debugger. Cuando el Debugger se abre, el nodo *watch* se vuelve verde y muestra el último valor recibido del pin conectado.



# Tweak and Watch Nodes

## Tarea 3: Requisitos

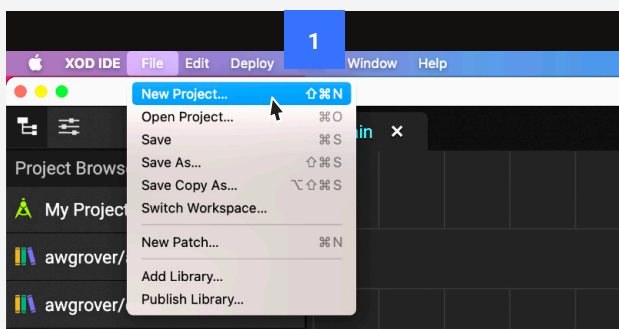
- Computadora con MacOS, Windows o Linux (software XOD y driver USB instalados)
- Grove Beginner Kit for Arduino (módulo de sensor de temperatura y humedad)
- Cable USB-A a micro USB

En esta tarea veremos cómo podemos utilizar los nodos *tweak* y *watch* para tomar lecturas de otro de los dispositivos integrados: el sensor de temperatura y humedad.

Utilizaremos un nodo *tweak-pulse* para que actúe como un botón y tome una lectura cada vez que lo pulsemos (hacemos un 'tweak'), y nodos *watch* para mostrar nuestras lecturas en la pantalla de la computadora.

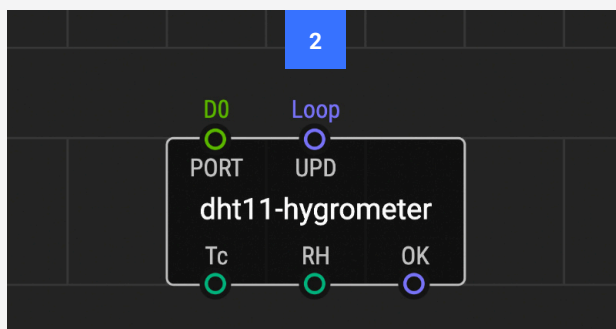
También utilizaremos el modo 'Simulate'/'Debug' (Simular/Depurar) de XOD, que nos permite observar y realizar cambios mientras el código se está ejecutando.

Este es un ejemplo de cómo los nodos *tweak* y *watch* pueden ser utilizados para probar rápida y fácilmente un patch. Son muy útiles para probar y depurar patches, así que deberías intentar acostumbrarte a usarlos mientras construyes.



### HAZ UN NUEVO PROYECTO

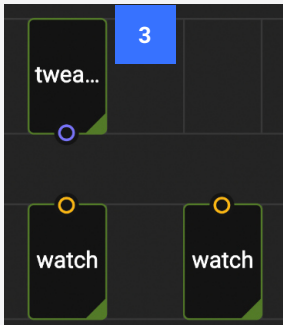
En primer lugar, guarda tu antiguo proyecto (no puedes tener dos proyectos abiertos a la vez). A continuación, ve a 'File > New Project...' (Archivo > Nuevo Proyecto...) en la barra de menús. Por último, nombra tu proyecto, y ya puedes empezar. (Si estás utilizando el archivo del tutorial XOD, pasa al tuto301).



### AÑADE Y CONFIGURA EL NODO DEL HIGRÓMETRO

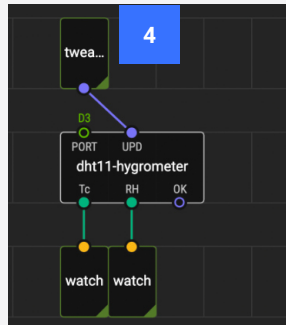
El sensor de temperatura y humedad integrado se conoce técnicamente como higrómetro DHT11. Hay un nodo XOD preinstalado para este dispositivo llamado 'dht11-hygrometer'. Añade este nodo a tu patch desde la librería *xod-dev/dht*. Ajusta el pin del puerto a 'D3'.





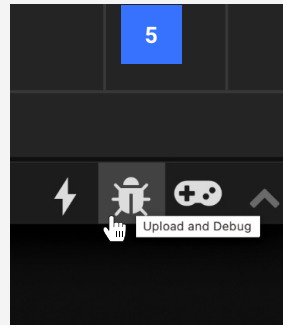
### AÑADE LOS NODOS TWEAK Y WATCH

Añade un *tweak-pulse* y dos nodos *watch* de la librería *xod/debug*.



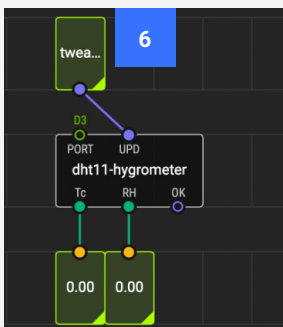
### CONECTA LOS NODOS

Conecta el nodo de *tweak-pulse* al pin UPD y un nodo *watch* a cada uno de los pines Tc (temperatura °C) y RH (humedad relativa).



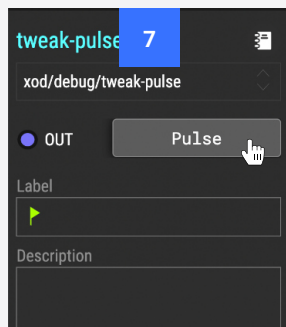
### CARGAR Y DEPURAR

Haz clic en el botón 'Upload and Debug' (mariposa) o utiliza el botón de carga y marca la casilla 'Debug after upload'.



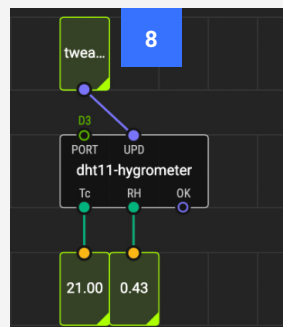
### MIRA

Mira el nodo *watch*. No mostrará una lectura porque el pin UPD está configurado para que sólo se actualice cuando lo ajustemos.



### PULSE EL NODO

Mientras se está ejecutando, haga clic en el nodo *tweak*. En el Inspector haz clic en el botón al lado de OUT que dice 'pulse'.

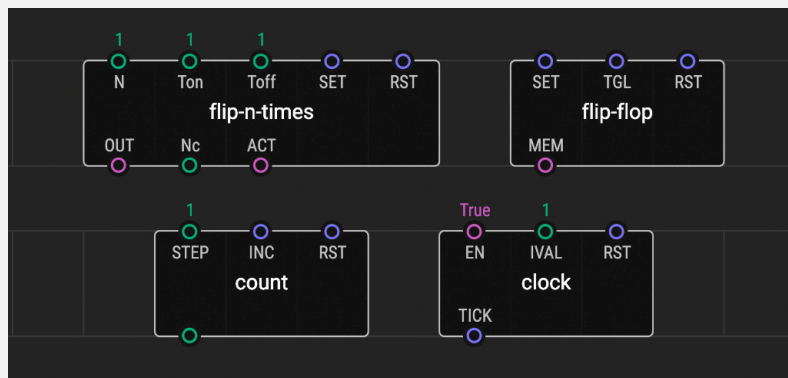


### MIRA OTRA VEZ

Vuelve a mirar los nodos *watch*. Esta vez deberían mostrar la temperatura y la humedad actuales.



# Nodos Flip, Clock y Count



## NODOS FLIP

Los nodos *flip* son nodos lógicos booleanos que cambian (o hacen 'flip') entre dos estados: 'Verdadero' y 'Falso'. Hay dos nodos flip útiles en XOD: *flip-n-times* y *flip-flop*.

El primer nodo, *flip-n-times*, cambiará entre 'Verdadero' y 'Falso' un número determinado de veces (N). Puedes determinar el tiempo que pasa en cada estado usando los pines Ton y Toff, y toda la secuencia será iniciada por un pulso al pin SET. Este nodo es útil para crear secuencias y patrones.

El segundo nodo, *flip-flop*, cambiará entre los estados 'Verdadero' y 'Falso' cada vez que el pin toggle (TGL) reciba un pulso. Este es un nodo particularmente útil que puede actuar como una palanca o interruptor en muchas situaciones.

## NODO CLOCK

Al igual que los nodos *flip*, el nodo *clock* (reloj) también es útil para controlar el tiempo de tus programas. Sin embargo, en lugar de dar una salida booleana 'Verdadero'/'Falso', el nodo clock envía pulsos en un intervalo de tiempo específico. Este nodo crea un 'ticking' o pulsaciones regulares, que puede ser utilizado para controlar tu programa.

## NODO COUNT

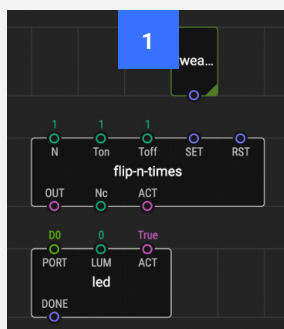
El nodo *count* (de cuenta) es un complemento de los nodos *flip* y *clock*, y actúa como una medida de cuántas veces se ha enviado una señal de pulso o booleana 'True'. Esto es útil para mantener un seguimiento de tu programa y su progreso. El nodo *clock* también es muy útil, cuando se utiliza junto con un nodo *watch*, para visualizar la salida de un pin de pulso (ver [Tarea 4, Pasos 15-16](#) para un ejemplo de este uso).

## Tarea 4: Requisitos

- Computadora con MacOS, Windows o Linux (software XOD y driver USB instalados)
- Grove Beginner Kit for Arduino (módulo LED)
- Cable USB-A a micro USB

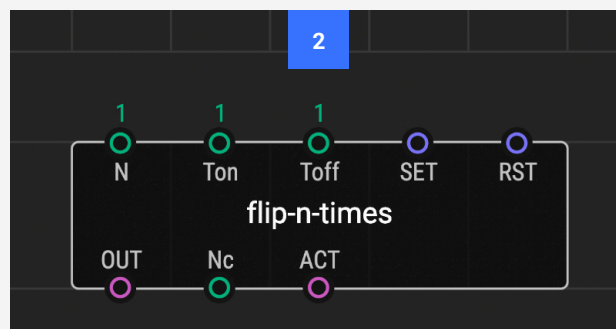
En esta tarea experimentaremos con los nodos *flip*, *clock* y *count* para controlar el comportamiento del LED incorporado, haciéndolo parpadear.

Los nodos *flip* y *clock* pueden ser útiles para modificar y cronometrar el comportamiento de los nodos, mientras que el nodo *count* puede ser útil para monitorizar estos comportamientos. En el contexto de los dispositivos biológicos, estos nodos son muy útiles para el ajuste fino de los dispositivos y para la construcción de programas más amplios.



### NUEVO PATCH Y AÑADIR NODOS

Abre un patch (o ve a tuto401). Añade estos nodos: *tweak-pulse* (*xod/debug*), *flip-n-times* (*xod/core*) y *led* (*xod/common-hardware*).

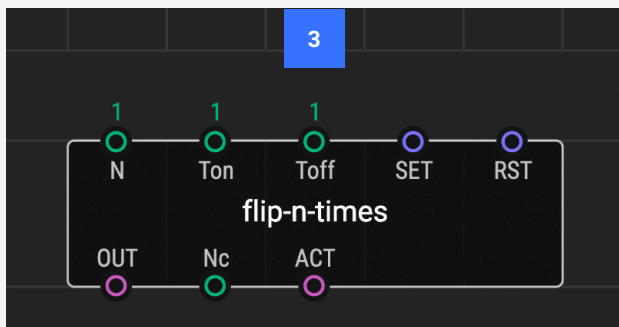


### ENTRADAS DEL NODO FLIP-N-TIMES

Este nodo tiene 5 pines de entrada: SET, RST (reiniciar), N (número), Ton (tiempo de encendido) y Toff (tiempo de apagado). Este nodo define una secuencia que cambiará entre verdadero y falso N número de veces. Ton y Toff definen la duración de cada estado. Un pulso a SET iniciará la secuencia, y un pulso a RST la reiniciará.

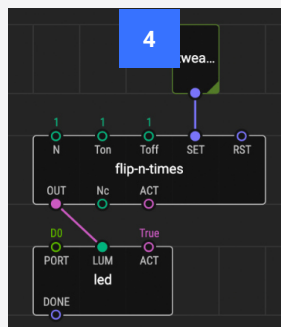


# Nodos Flip, Clock y Count



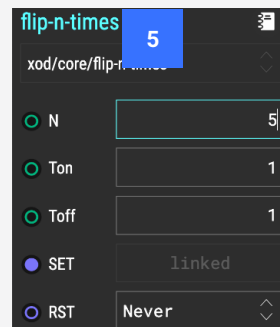
### SALIDAS DEL NODO FLIP-N-TIMES

El nodo *flip-n-times* tiene tres salidas. OUT lee el estado actual del nodo (verdadero/falso). Nc lee el número de veces que se ha realizado el ciclo. ACT lee si la secuencia se está ejecutando o no. Si quieres tener una mejor idea de cómo funcionan estas salidas, siempre puedes añadir nodos *watch* y ver lo que está pasando.



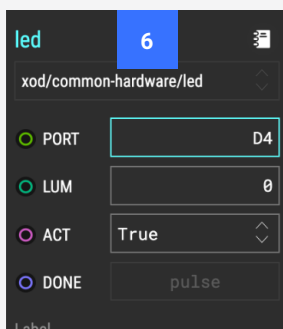
### CONECTA LOS NODOS

Conecta el nodo *tweak-pulse* al pin SET de *flip-n-times*, y el pin OUT de *flip-n-times* al pin LUM de *led*.



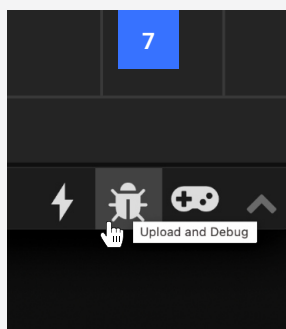
### AJUSTA LOS PINES DE FLIP-N-TIMES

Ajusta N a '5', Ton y Toff a '1', y RST a 'Never'. Puedes añadir un nodo *tweak-pulse* al pin RST para probar cómo funciona.



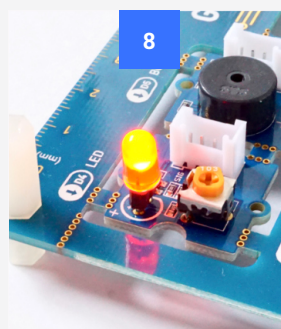
### AJUSTA LOS PINES DE LED

Al igual que en la [Tarea 1 Pasos 5-7](#) (p22), ajusta el pin PORT a 'D4' y el pin ACT a 'True'.



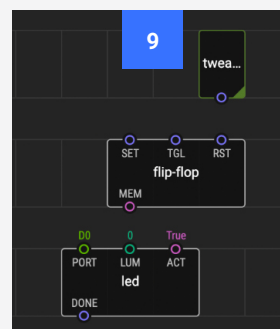
### CARGAR Y DEPURAR

Carga y depura el programa con el botón de la mariquita.



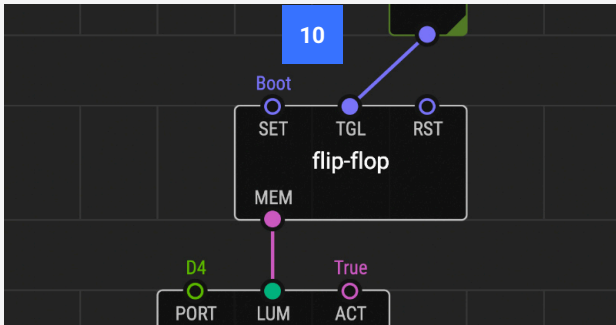
### AJUSTA Y VE

Pulsa el nodo *tweak-pulse* y observa lo que ocurre con el LED. Cada vez que pulses, la luz debería parpadear 5 veces.



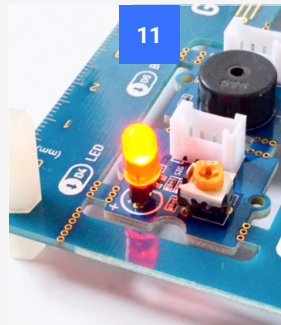
### NODO FLIP-FLOP

Ahora vamos a probar un nodo *flip* diferente. Elimina el nodo *flip-n-times* y añade un nodo *flip-flop* (*xod/core*).



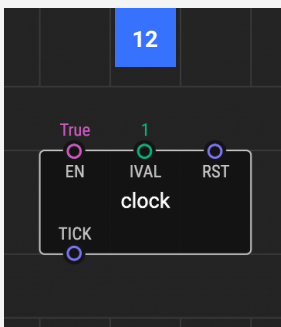
### CONECTA Y AJUSTA LOS PINES FLIP-FLOP

El nodo *flip-flop* tiene 3 pines de entrada: SET, RST y TGL ('toggle', alternar). TGL cambia el nodo entre verdadero y falso cada vez que recibe un pulso. Conecta el nodo *tweak-pulse* a TGL. Ajusta SET a 'On Boot' y RST a 'Never'. El pin de salida MEM (memoria) lee el último estado del nodo. Conéctalo al pin LUM de *led*.



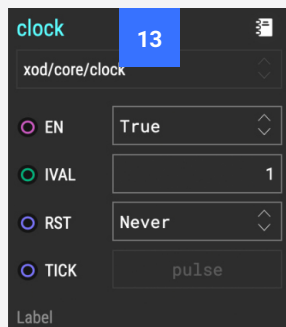
### PRUEBA EL PATCH

Sube el programa y pulsa *tweak-pulse*. El LED debería cambiar entre encendido y apagado cada vez que lo pulses.



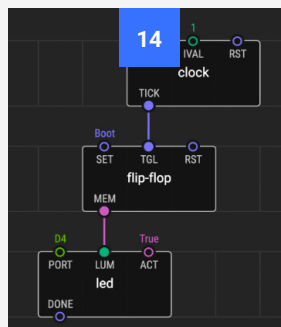
### NODO CLOCK

El nodo *flip-flop* también puede hacer parpadear el LED cuando se combina con *clock*. Añade un nodo *clock* (*xod/core*).



### AJUSTA LOS PINES DE CLOCK

Ajusta EN (habilitado) a 'True' y RST a 'Never'. IVAL determina la frecuencia del reloj (en segundos). Ajústalo a '1'.

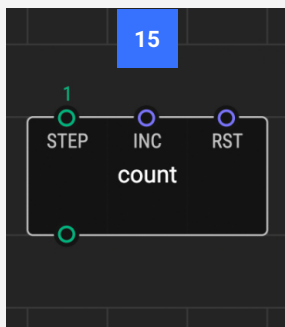


### RECONECTAR

Elimina el nodo *tweak-pulse* y conecta el nodo *clock* en su lugar enlazando el pin TICK con el pin TGL.

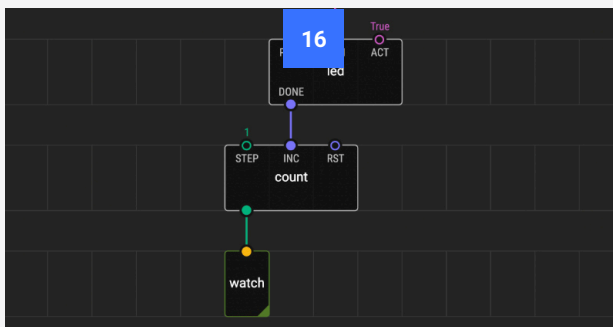


# Nodos Flip, Clock y Count



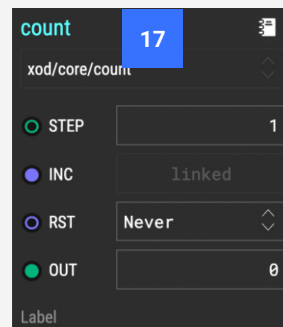
## NODO COUNT

También añadiremos un nodo *count* (*xod/core*) a este patch para poder ver el número de veces que parpadea el LED.



## CONECTA EL NODO COUNT

Conecta el pin *DONE* de *led* al pin *INC* (incremento) de *count*. El pin *DONE* pulsa cada vez que el LED se enciende o se apaga, y el pin *INC* incrementa la cuenta cada vez que se pulsa. Conecta el pin de salida de *count* a un nodo *clock* para que veamos la cuenta. Esto nos permitirá ver en la pantalla cada vez que el LED pulsa.



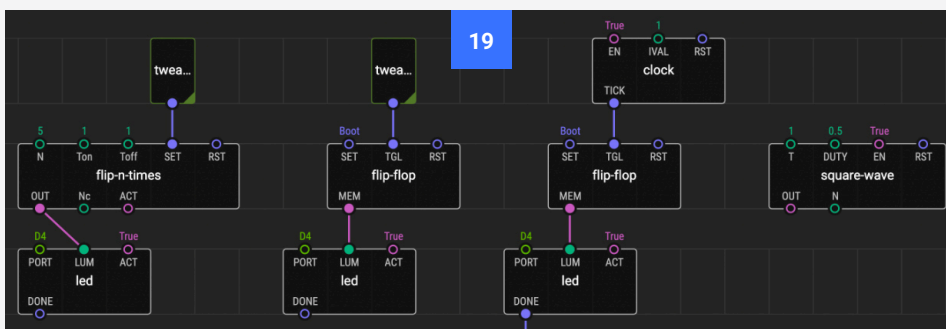
## AJUSTA LOS PINS DE COUNT

Ajusta *RST* a 'Never'. *STEP* determina cuánto aumenta el recuento con cada pulso. Ajústalo a '1'.



## PRUEBA EL PATCH

Carga el programa. Observa el LED y el nodo *watch*. El LED debe parpadear y la cuenta aumentará con cada parpadeo.



## ¡EXPERIMENTA!

En programación, siempre hay más de una manera de lograr un resultado similar, y diferentes métodos pueden adaptarse a diferentes aplicaciones. Aquí hemos probado dos formas diferentes de hacer que el LED parpadee, pero hay muchas otras formas con las que puedes experimentar. ¿Por qué no pruebas a utilizar un nodo *square-wave* para hacer que el LED parpadee? Comprueba si puedes resolverlo utilizando el panel de ayuda y la página web de XOD. Intenta experimentar con diferentes tiempos y patrones de parpadeo.



## Vigilando los Pins de Pulse: Combinación de Nodos Count y Watch

La combinación de nodos *count* y *watch* es una forma muy útil de visualizar la salida de un pin de pulso (pulso). Desafortunadamente, una salida de pulso no puede conectarse directamente a un nodo *watch* en XOD, ya que los tipos de datos (pulso y cadena) chocan.

Podemos evitar esto conectando una salida de pulso al pin INC de un nodo *count*, y luego conectando un nodo *watch* al pin de salida de *count* (como hicimos en los **Pasos 15-16** de esta tarea). En esta configuración, el nodo *count* aumenta con cada pulso enviado, y podemos visualizar esto en el Debugger con el nodo *watch*.

## Descubriendo Nuevos Nodos

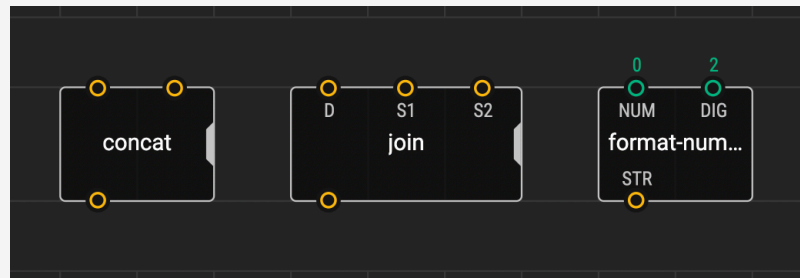
El último paso de esta tarea te invita a probar un nuevo nodo: el nodo *square-wave*. No hemos proporcionado aquí información específica sobre este nodo, pero es útil para practicar el descubrimiento de nuevos nodos por ti mismo. Para saber cómo funciona un nuevo nodo, lo mejor es empezar por entender qué hacen sus pines. En la mayoría de los casos, el panel de Quick Help (Ayuda Rápida) ofrece una breve explicación del nodo y de lo que hace cada uno de sus pines. Esta información suele ser suficiente para entender la función del nodo y cómo utilizarlo.

También puedes encontrar documentos adicionales sobre cada nodo en la página web de XOD. Puedes acceder a ella haciendo clic en el botón de documento junto al nombre del nodo en el panel Inspector. O visitando [www.xod.io/libs](http://www.xod.io/libs) y utilizando la función de búsqueda.

Cuando los documentos de un nodo no son suficientes, por lo general se puede averiguar su función añadiendo nodos *tweak* a sus entradas y nodos *watch* a sus salidas. A continuación, simula o 'Upload and Debug' tu patch. Intenta editar cada una de las entradas y observa cómo afecta a las salidas. De esta manera, a menudo puedes determinar la función de un nodo de forma experimental.

Si sigues teniendo problemas, siempre puedes encontrar ayuda en el foro del XOD en [www.forum.xod.io](http://www.forum.xod.io).

# Nodos Concat, Join y Format-Number



## NODO CONCAT

El nodo *concat* ('concatenate', concatenar) permite unir dos o más conjuntos de cadenas. Esto es útil para combinar diferentes entradas para su visualización o almacenamiento. Por ejemplo, combinar la lectura de un número de un sensor con el símbolo de sus unidades. *concat* unirá las entradas directamente, así que si necesitas un espacio entre ellas tendrás que introducirlo en tu cadena.

*concat* es un nodo 'variadic' (variádico). Esto significa que puedes cambiar el número de entradas. Puedes hacerlo arrastrando la pestaña blanca en el lado derecho del nodo.

## NODO JOIN

El nodo *join* (unir) es similar al nodo *concat*, pero tiene la característica adicional de permitirte elegir cómo se unen las diferentes entradas de cadena. El pin D ('delimiter', delimitador) determina qué carácter se utiliza para separar las entradas, por ejemplo, un espacio, una coma o dos puntos, etc. Esto es especialmente útil para almacenar lecturas de datos, ya que puedes separar los valores con una coma o un tabulador para crear archivos separados por comas (.csv) o por tabuladores (.tsv). Al igual que *concat*, el nodo *join* es variádico y puede tomar tantas entradas como sea necesario.

## NODO FORMAT-NUMBER

Como su nombre indica, el nodo *format-number* te permite dar formato a las salidas numéricas. Con este nodo, puedes determinar la cantidad de decimales mostrados, lo que puede ser muy útil si quieres mostrar la lectura de un sensor, por ejemplo. Este nodo también convierte el formato de la entrada de un número a una cadena. Otros nodos útiles para formatear números son *number-split-to-digit*, de la librería *gst/number-split-to-digit*, *dec-to-2digits* y *dec-to-4digits*, ambos de la librería *cesars/utls*.

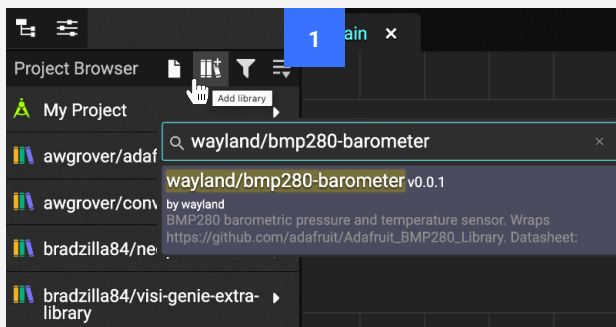


## Tarea 5: Requisitos

- Computadora con MacOS, Windows o Linux (software XOD y driver USB instalados)
- Grove Beginner Kit for Arduino (módulo de sensor de presión de aire)
- Cable USB-A a micro USB

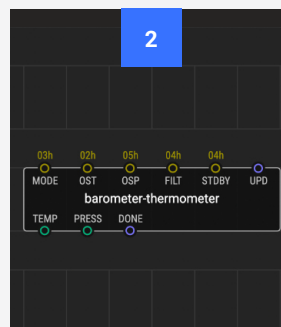
En esta tarea veremos el uso de los nodos *concat*, *join* y *format-number*. Estos nodos son especialmente útiles cuando se trabaja con datos y visualizaciones.

Los nodos *concat* y *join* se utilizan para combinar información en forma de cadenas (texto). El nodo *format-number* se utiliza para establecer el número de puntos decimales que se muestran en un número. En combinación, estos nodos son útiles para de formato a las salidas de los módulos de sensores, tanto para el almacenamiento de datos, como para su visualización en una pantalla.



### NUEVO PATCH Y AÑADIR LIBRERÍA

Abre un nuevo patch (o pasa al tuto501). Para trabajar con el sensor de presión atmosférica (también conocido como barómetro) necesitarás instalar la librería *wayland/bmp280-barometer*.

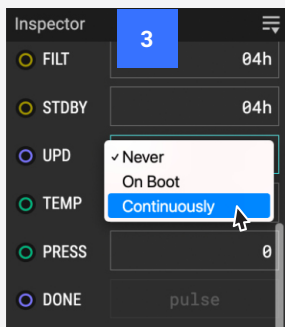


### AÑADE BAROMETER-THERMOMETER

Añade un nodo *barometer-thermometer* (barómetro-termómetro) (*wayland/bmp280-barometer*) al patch.

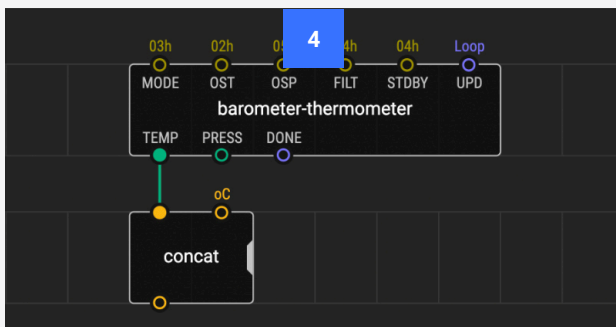


# Nodos Concat, Join y Format-Number



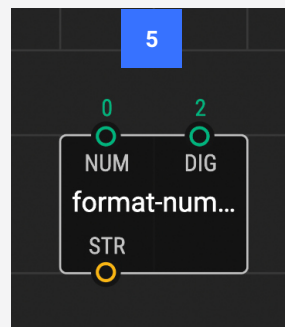
## AJUSTA BAROMETER-THERMOMETER

Pon UPD en 'Continuously'. Deja las demás entradas como están. Las salidas son TEM (temperatura) PRESS (presión).



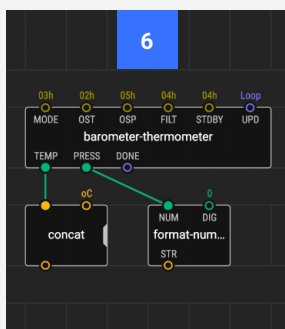
## AÑADE Y AJUSTA EL NODO CONCAT

Añade un nodo *concat* (xod/core). Este nodo combina múltiples cadenas en una sola salida. Además, es variádico, es decir, puedes arrastrar la pestaña blanca en el lado derecho del nodo para cambiar el número de entradas. Conecta la primera entrada al pin TEMP. Ajusta la segunda entrada a 'oC' (grados centígrados).



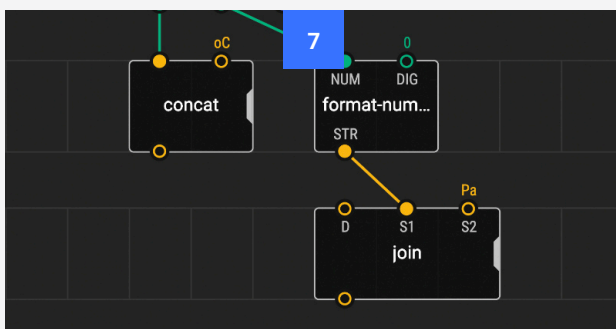
## AÑADE NODO FORMAT-NUMBER

Añade un nodo *format-number* (xod/core). Este nodo permite cambiar el número de decimales de un número.



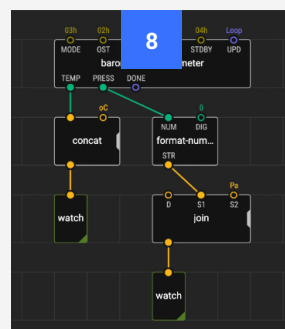
## AJUSTA EL NODO FORMAT-NUMBER

Vincula el pin PRESS de *barometer-thermometer* al pin NUM (número) de *format-number*. Ajusta DIG (dígitos) a '0'.



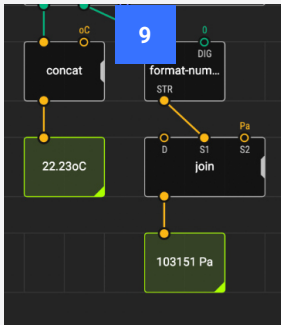
## AÑADE Y AJUSTA EL NODO JOIN

Añade un nodo *join* (xod/core). Es similar a *concat*, pero tiene un pin D (delimitador). D determina cómo se unen las entradas. Está automáticamente ajustado para ser un espacio. Déjalo así. Conecta la primera entrada (S1) al pin STR (cadena) de *format-number*. Ajusta la segunda entrada a 'Pa' (pascales).



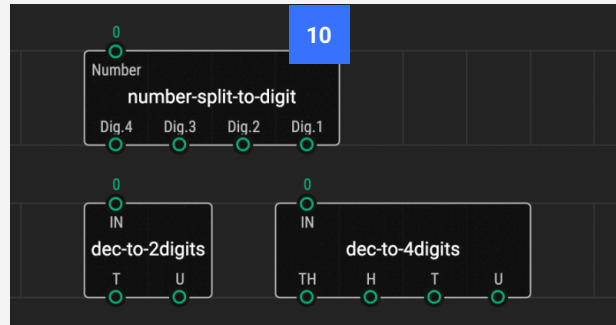
## AÑADE NODOS WATCH

Añade dos nodos *watch* (xod/debug). Vincula uno a la salida del nodo *concat*, y otro a la salida del nodo *join*.



### PRUEBA EL PATCH

Carga y depura. Mira los nodos *watch*. Puedes ampliar los nodos *watch* tirando de la esquina inferior derecha.



### ¡EXPERIMENTA!

Juega con los nodos de tu patch para ver cómo puedes dar formato a la salida del sensor de diferentes maneras. Intenta explorar otros nodos disponibles para formatear números en XOD. Por ejemplo, *number-split-to-digit* (*gst/number-split-to-digit*), *dec-to-2digits* o *dec-to-4digits* (*cesars/utls*).



## Más Información Sobre los Nodos Básicos

Esta sección ha cubierto un número de nodos básicos que hemos encontrado útiles en la construcción de dispositivos biológicos simples. Sin embargo, hay muchos otros nodos útiles, tanto preinstalados en XOD, como creados por usuarios de XOD como tú.

En el resto de esta guía seguiremos explorando nodos y técnicas útiles en XOD, pero si quieres explorar por ti mismo, aquí tienes unos cuantos recursos útiles para familiarizarte con XOD:

### TUTORIAL XOD

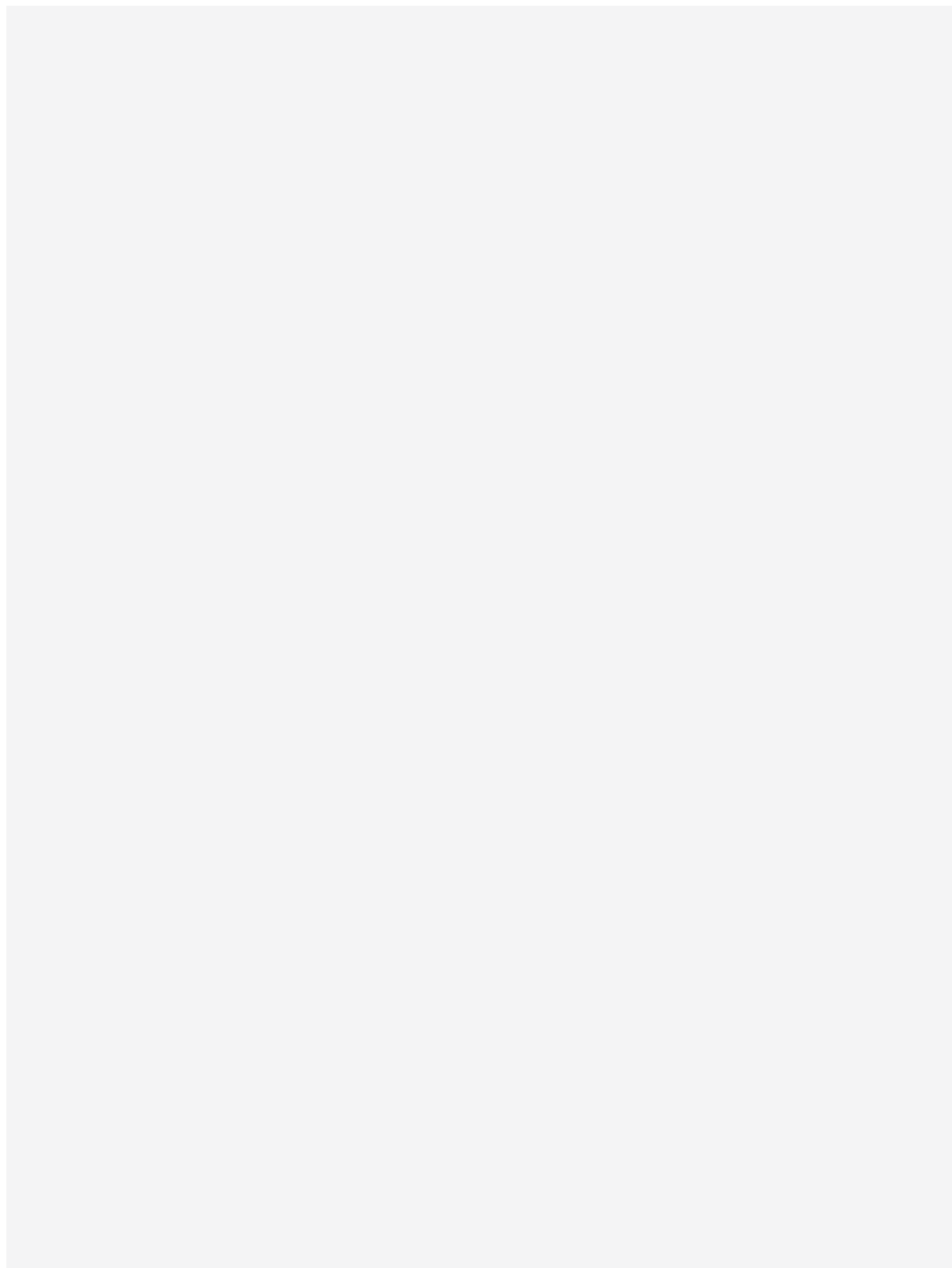
Cada vez que abras el XOD te ofrecerá la opción de seguir su tutorial incorporado. Trabajar con él es una buena manera de aprender más sobre lo que puede hacer XOD. También está disponible en línea en [www.xod.io/docs/tutorial](http://www.xod.io/docs/tutorial).

### GUÍA XOD

La guía del usuario del XOD ofrece consejos sobre algunos conceptos más complejos, así como algunos casos de estudio para trabajar. Está disponible en línea en [www.xod.io/docs/guide](http://www.xod.io/docs/guide).

### LIBRERÍA XOD/CORE

Echar un vistazo a los nodos de la librería del núcleo de XOD te ayudará a entender los nodos más básicos disponibles y lo que hacen. Encuentra *xod/core* en el Project Browser.



# Lección 4: Construir Dispositivos



**Crear Nuevos Nodos**



**Utilizar Buses**



**Programas Lógicos**



**Secuencias y Bucles**

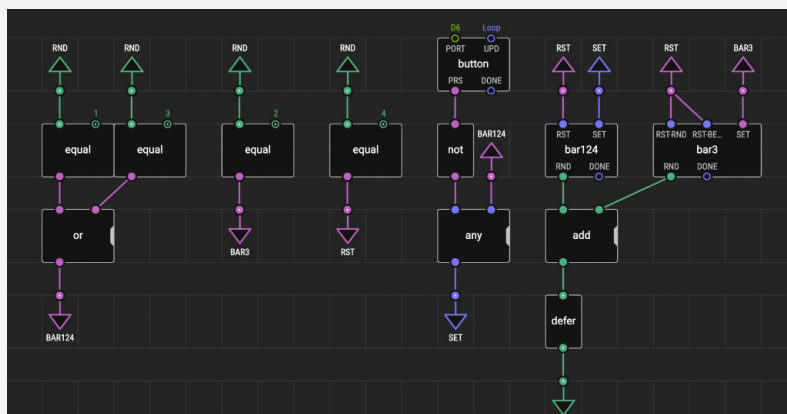


# Construir Dispositivos

Hasta ahora en esta guía hemos explorado cómo utilizar algunos nodos útiles para realizar algunas tareas simples, como ver las lecturas de un sensor, o hacer parpadear un LED. Sin embargo, a menudo queremos realizar tareas más complejas, como la lectura y el almacenamiento de datos, la visualización de información en una pantalla o la creación de programas lógicos complejos.

Esta lección se basará en lo que ya hemos aprendido, y explorará algunos conceptos más complejos en XOD. Esto te ayudará a construir programas y dispositivos más grandes y complejos de una manera ordenada y eficiente.

Las dos primeras tareas de esta lección, la Tarea 6 y la Tarea 7, tratarán sobre cómo crear nuevos nodos y cómo utilizar buses, respectivamente. Estas habilidades son útiles para crear programas ordenados y compartimentados. La Tarea 8 y la Tarea 9 explorarán cómo podemos utilizar estas habilidades para construir programas basados en la lógica, y cómo podemos introducir secuencias y bucles, que son útiles para los dispositivos biológicos.



Patch XOD de la Tarea 9: Secuencias y Bucles

## OBJETIVOS

Al final de este capítulo deberás ser capaz de:

- Describir la función de los nodos terminales de XOD y cómo se usan.
- Crear nuevos nodos combinando nodos actuales y nodos terminales.
- Probar y usar los nuevos nodos que has creado en tus programas.
- Describir la función de los buses, sus ventajas y cómo usarlos.
- Utilizar nodos matemáticos y lógicos para crear programas lógicos.
- Recordar al menos dos métodos diferentes de crear secuencias en XOD.
- Implementar bucles de programación en XOD usando el nodo *defer*.
- Usar los dispositivos restantes de la placa Grove: el sensor de sonido, el sensor de luz, el sensor de aceleración de 3 ejes y la pantalla OLED.
- Programar la pantalla OLED para mostrar gráficos como texto y formas.
- Recordar cómo documentar tus nodos correctamente, incluyendo cómo describir el nodo y sus pines y cómo añadir cajas de comentarios.
- Recordar cómo publicar nodos o grupos de patches en una librería.

# Crear Nuevos Nodos

## ¿Por qué Crear Nuevos Nodos?

Si queremos construir programas capaces de realizar funciones más complejas que solo producir una simple entrada y salida, tendremos que añadir algunas habilidades más a nuestro repertorio. Los programas más complejos suelen requerir más nodos, y la multitud de nodos y enlaces puede resultar rápidamente confusa. La buena noticia es que XOD ofrece varias formas de reducir la complejidad de tus patches y mantener tus programas limpios y ordenados. Esto es una buena práctica para que puedas mantenerte al tanto de lo que estás haciendo, y también para aquellos que requieran entender tu programa.

Una forma de simplificar un programa complejo es crear tus propios nodos. Esto significa que puedes encapsular funciones específicas dentro de tu programa en pequeños paquetes ordenados que pueden ser fácilmente conectados entre sí. También tiene la ventaja de que pueden ser fácilmente compartidos con la comunidad más amplia de XOD, haciendo que nuevos nodos útiles estén disponibles para todos.



Nodos terminales de XOD

Crear tus propios nodos es mucho más fácil de lo que parece. Es esencialmente lo mismo que crear cualquier otro patch, pero tenemos que añadir nodos especiales llamados 'terminales' (terminales) para permitir que nuestro nuevo nodo se comunique con otros nodos.

Hay dos tipos de terminales: 'inputs' (entradas) y 'outputs' (salidas). Al igual que los nodos *tweak*, vienen en diferentes tipos basados en su tipo de datos. Los nodos de arriba, de izquierda a derecha y de arriba a abajo, son: *input-boolean* (entrada booleano), *input-byte* (entrada byte), *input-number* (entrada número), *input-port* (entrada puerto), *input-pulse* (entrada pulso), *input-string* (entrada cadena), *input-t1* (tipo de entrada personalizado), *output-boolean* (salida booleano), *output-byte* (salida byte), *output-number* (salida número), *output-port* (salida puerto), *output-pulse* (salida pulso), *output-string* (salida cadena), *output-t1* (tipo de salida personalizado).

En la siguiente tarea exploraremos con más detalle cómo utilizar estas terminales para crear tus propios nodos, utilizando como ejemplo la pantalla OLED incorporada a tu placa.

El sitio web de XOD también ofrece excelente información sobre cómo hacer tus propios nodos en [www.xod.io/docs/guide/nodes-for-xod-in-xod](http://www.xod.io/docs/guide/nodes-for-xod-in-xod).



# Crear Nuevos Nodos

## Tarea 6: Requisitos

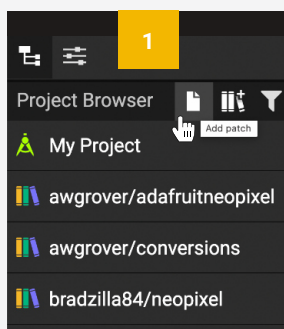
- Computadora con MacOS, Windows o Linux (software XOD y driver USB instalados)
- Grove Beginner Kit for Arduino (módulos de sensor de sonido y pantalla OLED)
- Cable USB-A a micro USB

En esta tarea aprenderemos a hacer un nuevo nodo que nos permitirá escribir texto en nuestra pantalla OLED.

Instruir a la pantalla OLED para que muestre el texto es una tarea un poco más compleja que la que hemos hecho hasta ahora, e implica varios nodos para representar la pantalla en lugar de uno.

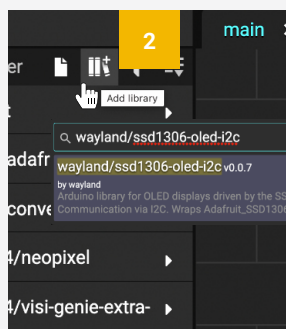
En esta tarea combinaremos estos múltiples nodos en uno solo, que luego usaremos para mostrar las lecturas de nuestro sensor integrado de sonido.

La creación de nodos de este tipo es útil, ya que ayuda a simplificar el patch, y también podemos guardar los nuevos nodos para utilizarlos posteriormente.



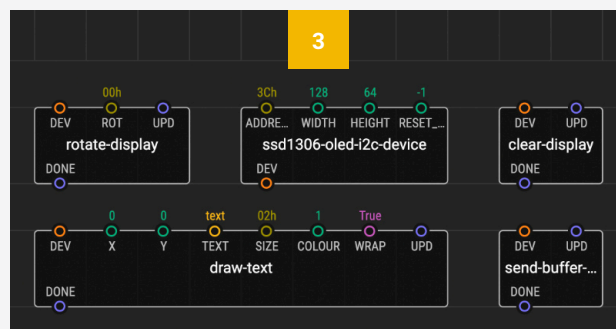
### NUEVO PROYECTO Y NUEVO PATCH

Guarda tu proyecto y crea uno nuevo (o pasa al tuto601). Añade un patch y nómbralo 'write-text-to-oled' (escribir texto en oled).



### AÑADE LIBRERÍA

Para trabajar con la pantalla OLED necesitarás instalar la librería wayland/ssd1306-oled-i2c.



### AÑADE NODOS

Desde la librería wayland/ssd1306-oled-i2c, añade los siguientes nodos a tu patch: ssd1306-oled-i2c-device (dispositivo i2c ssd1306 oled), rotate-display (dar vuelta a la pantalla), clear-display (borrar la pantalla), draw-text (escribir texto), send-buffer-to-display (pasar información a la pantalla).



## CONFIGURANDO LA PANTALLA OLED

La configuración de la pantalla OLED requiere varias conexiones entre estos nodos. Vamos a ir paso a paso.

### A

#### SSD1306-OLED-I2C-DEVICE

Este nodo representa el dispositivo OLED. WIDTH y HEIGHT eligen las dimensiones de la pantalla en píxeles. Déjalos como '128' y '64'. ADDRESS identifica el puerto, déjalo como '3Ch'. RESET representa el pin de reset de la pantalla. Déjalo como '-1' ya que nuestra placa no tiene un pin de reset de pantalla dedicado. **La salida de este nodo, DEV (dispositivo) necesita ser conectada a cada uno de los pines de entrada DEV de los otros nodos.**

### B

#### ROTATE-DISPLAY

Puedes cambiar la orientación de la pantalla usando este nodo. Pon ROT en '02h', que es el correcto para nuestra pantalla. Ajusta UPD en 'On Boot' para que la pantalla se actualice cuando se inicie el programa. Este nodo inicia una secuencia que nos permite mostrar elementos en la pantalla. **En esta secuencia, cada pin DONE se conecta al siguiente pin UPD.** Conecta el pin DONE de *rotate-display* al pin UPD de *clear-display*.

### C

#### CLEAR-DISPLAY

Este nodo debe utilizarse antes de mostrar algo en la pantalla. Conecta el pin DONE al pin UPD de *draw-text*.

### D

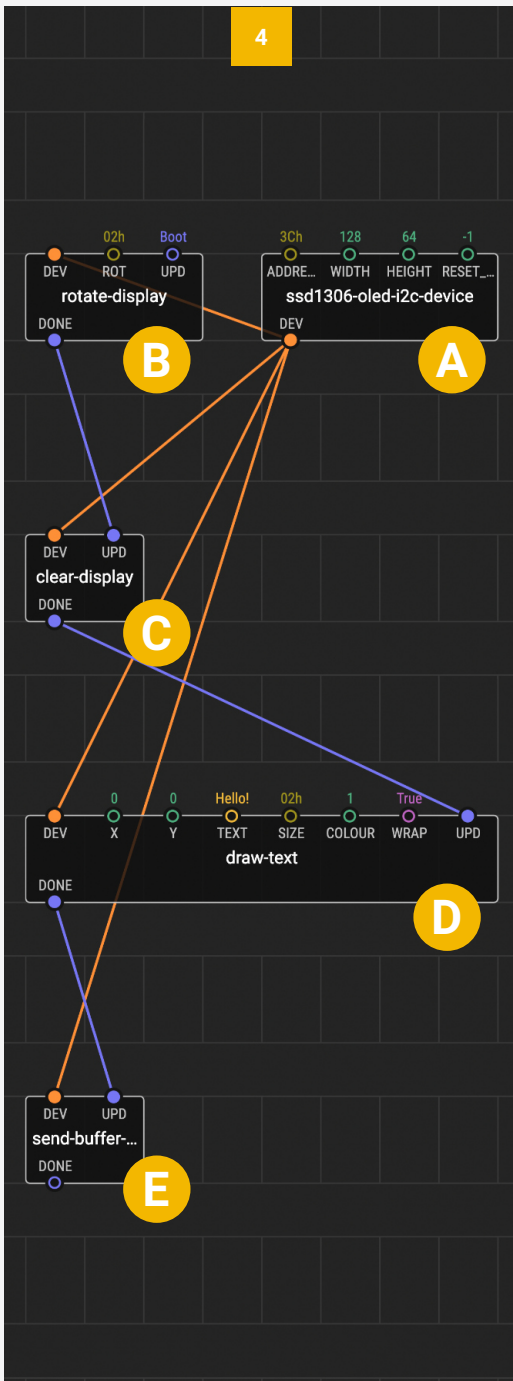
#### DRAW-TEXT

Este nodo introduce el texto que queremos mostrar. X y Y determinan la posición del texto por coordenadas. Déjalas como '0', '0'. TEXT es donde se introduce el texto. Utilice 'Hello!' (¡Hola!) como prueba. SIZE determina el tamaño del texto. Déjalo como '02h'. COLOUR determina el color del texto (blanco o negro). Déjalo en '1'. WRAP determina si el texto se envuelve dentro de los límites de la pantalla. Déjalo como 'True'. Conecta el pin DONE al pin UPD de *send-buffer-to-display*.

### E

#### SEND-BUFFER-TO-DISPLAY

Hasta ahora hemos escrito información en la memoria del microchip, pero no la hemos enviado a la pantalla. Este nodo es el paso final que envía estos datos. Es necesario utilizarlo para mostrar algo en la pantalla.

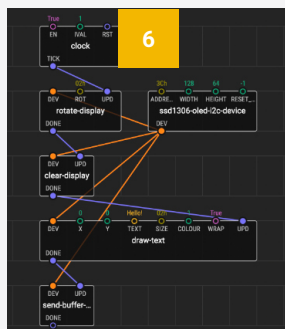


# Crear Nuevos Nodos



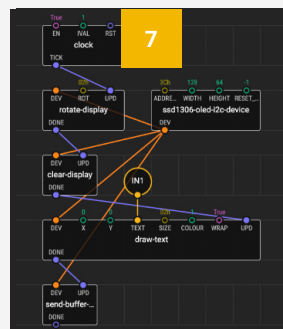
## PRUEBA EL PATCH

Carga el patch y observa tu pantalla OLED. Debería aparecer un texto blanco de la pantalla (superior izquierda).



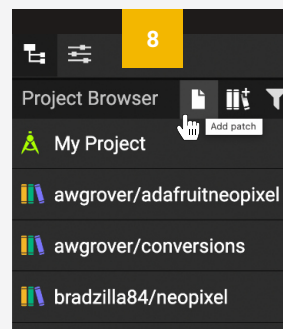
## AÑADE UN NODO CLOCK

Añade un nodo *clock* (*xod/core*) y enlaza el pin TICK con UDP de *rotate-display*. Esto actualizará tu pantalla una vez por segundo.



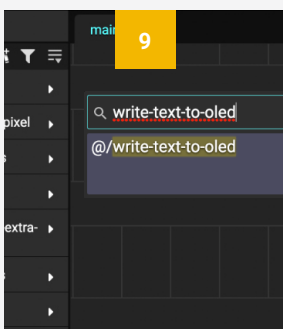
## AÑADE UN NODO INPUT-STRING

Añade un nodo *input-string* (entrada cadena) (*xod/patch-nodes*) y conéctalo al pin TEXT de *draw-text*.



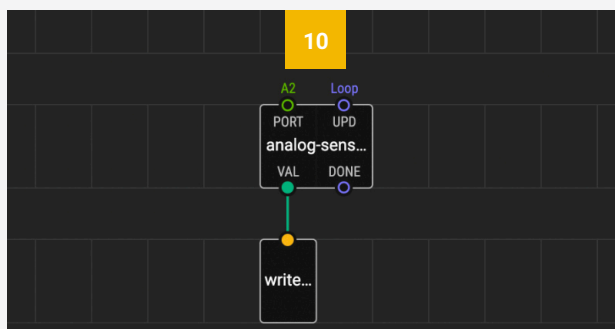
## HAZ UN NUEVO PATCH

Ahora vamos a añadir nuestro nuevo nodo a un patch. Añade un patch y nómbralo 'sound-sensor' (sensor de sonido).



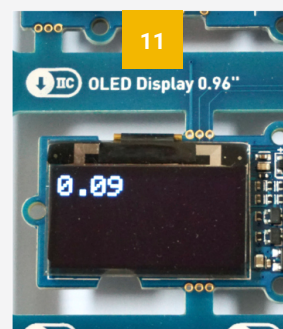
## AÑADE WRITE-TEXT-TO-OLED

Para añadir tu nuevo nodo, simplemente búscalo como de costumbre, o arrastra el patch desde el Project Browser.



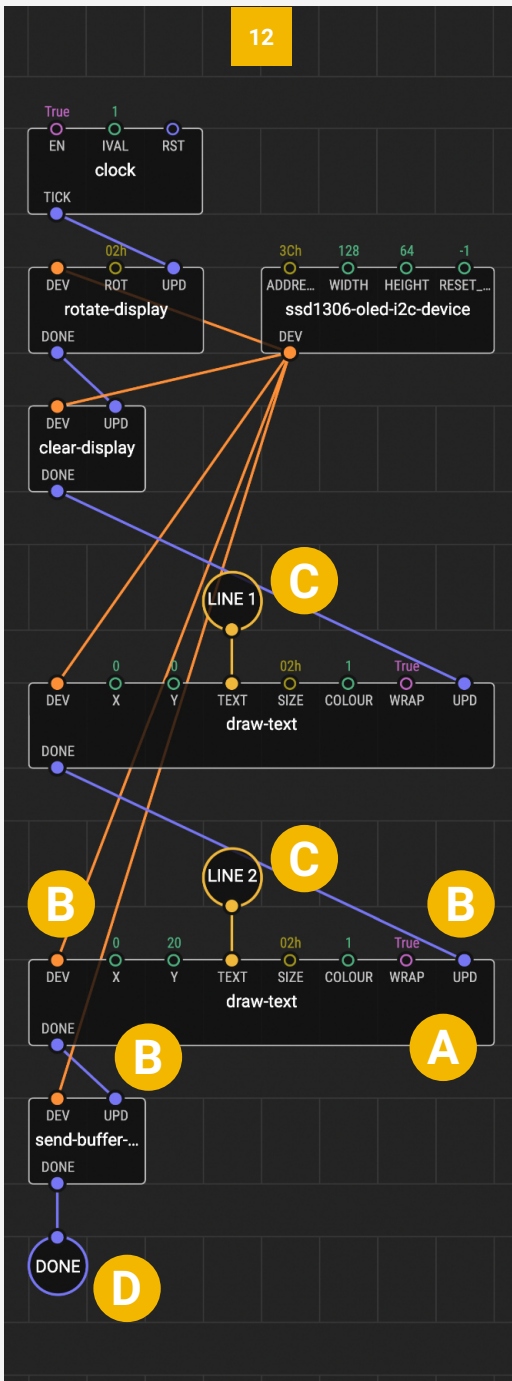
## AÑADE Y CONFIGURA EL NODO ANALOG-SENSOR

Ahora que tenemos un nodo que escribirá texto en la pantalla, vamos a utilizarlo para mostrar la lectura de un sensor. Para muchos sensores analógicos (incluyendo el sensor de sonido) puedes usar el nodo *analog-sensor* (*xod/common-hardware*). Añade este nodo, ajusta PORT a 'A2', y conecta el pin VAL al nodo *write-text-to-oled*.



## PRUEBA EL PATCH

Carga el patch, y deberías ver las lecturas del sensor de sonido mostradas en tu pantalla.



## MODIFICA TU NODO

Escribir una línea de texto directamente en la pantalla OLED está muy bien, pero ¿Qué pasa si necesitamos un nodo más complicado? Por ejemplo, uno que tome múltiples líneas de texto, o uno que envíe una señal una vez que el texto haya sido cargado?

Volvamos a nuestro nodo *write-text-to-display*. Puedes hacer esto abriendo el patch en el Project Browser, o haciendo doble clic en el nodo en tu patch.

Haz los cambios que se indican a continuación para ampliar las capacidades de tu nodo.

**A**

### SEGUNDO DRAW-TEXT NODE

Añade un segundo nodo *draw-text* debajo del primero. Lo utilizaremos para dibujar una segunda línea de texto. Establece el pin Y de este nodo a 20. Esto moverá el texto 20 píxeles hacia abajo, creando una nueva línea.

**B**

### RECONECTA

Elimina el enlace entre el primer nodo *draw-text* y *send-buffer-to-display*. Enlaza el pin DONE del primer nodo *draw-text* con el pin UPD del segundo nodo *draw-text*. Enlaza el pin DONE del segundo nodo *draw-text* con el pin UPD de *send-buffer-to-display*. Conecta el pin DEV de *ssd1306-oled-i2c-device* al pin DEV del segundo nodo *draw-text*.

**C**

### ETIQUETA LOS NODOS INPUT-STRING

Añade un segundo nodo *input-string* y conéctalo al pin TEXT del segundo nodo *draw-text*. Ahora tendremos más de una entrada en nuestro nodo, así que necesitamos darles etiquetas, para evitar confusiones. Haz clic en cada nodo *input-string* uno por uno. Utiliza el panel del Inspector para nombrar tus nodos escribiendo en la caja de texto 'Label' (Etiqueta). Nombra tus nodos 'LINE 1' (línea 1) y 'LINE 2' (línea 2) respectivamente.

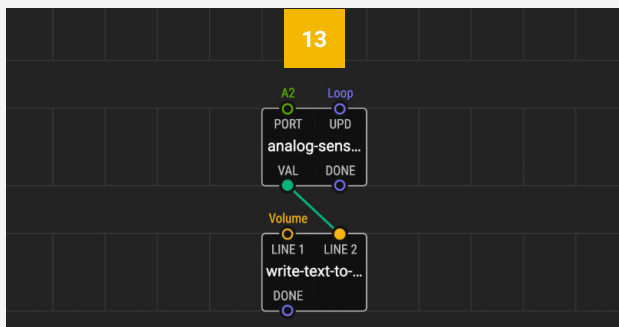
**D**

### AÑADE EL NODO OUTPUT-PULSE

Añade un nodo *output-pulse* (salida pulso) al patch, y conéctalo al pin DONE de *send-buffer-to-display*. Utiliza el Inspector para etiquetar este nodo 'DONE'.

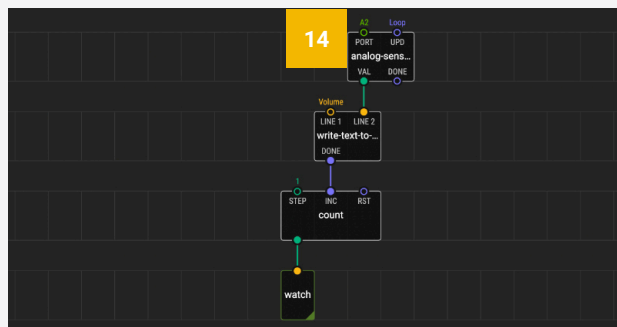


# Crear Nuevos Nodos



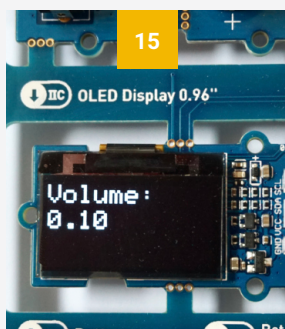
## MODIFICANDO TU PATCH

Vuelve a tu patch *sound-sensor* utilizando la pestaña de la parte superior, o el Project Browser. Notarás que el nodo *write-text-to-oled* ha cambiado. Ahora tiene dos entradas y una salida. Elimina el enlace entre VAL y LINE 1 y enlaza VAL con LINE 2. Utiliza el Inspector para establecer LINE 1 como 'Volume:' (Volumen).



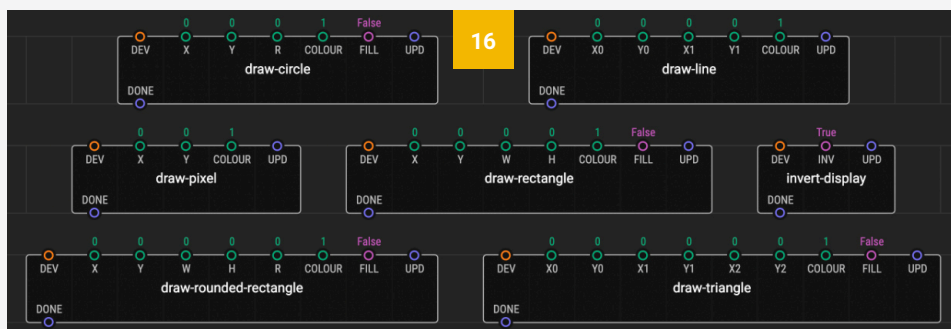
## ¡AÑADE NODOS COUNT Y WATCH

Añade un nodo *count* (*xod/core*) y un nodo *watch* (*xod/debug*). Enlaza el pin DONE de *write-text-to-oled* con el pin INC de *count*, y el nodo *watch* con el pin de salida de *count*. Un nodo *watch* no puede ser vinculado directamente a un pin 'pulse', por lo que este es un truco útil si quieres ver la salida de un pin 'pulse'.



## PRUEBA EL PATCH

Carga y depura. Deberías ver dos líneas de texto en tu pantalla, y el nodo *watch* contará cuando la pantalla se actualice.



## ¡EXPERIMENTA!

La pantalla OLED es un dispositivo realmente útil, y puede utilizarse de múltiples maneras. Prueba jugando con tu nuevo nodo: añadir otra línea, o cambiar la posición y el tamaño del texto. O puedes utilizar el OLED para mostrar los datos de un sensor diferente. También puedes experimentar con algunos de los otros nodos de la librería *wayland/ssd1306-oled-i2c*. Esta librería contiene muchos nodos útiles para dibujar diferentes objetos en la pantalla, así como varios patches de ejemplo para mostrarte cómo funcionan.

## Compartir Nodos y Publicar Librerías

Una de las grandes ventajas de XOD es la creciente comunidad de colaboradores, quienes generan una gama cada vez más amplia de nodos y librerías para que otros los utilicen. Cuando creas nuevos nodos que crees que pueden ser útiles para otros, puedes compartirlos fácilmente con la comunidad XOD publicándolos como una librería.

No hay reglas estrictas sobre lo que constituye una librería, así que un nodo solo puede seguir siendo una librería. Publicar permite que otros usen tus nodos, y también es útil para reutilizar tu propio trabajo, ya que puedes descargar tus propias librerías para usarlas en tus proyectos.

Crear una librería es esencialmente hacer un proyecto con un patch para cada nodo. También puedes incluir patches con ejemplos de cómo usar los nodos, como hay en la librería *wayland-ssd1306-oled-i2c*. Una vez que hayas creado tu proyecto, necesitas establecer los metadatos. Para ello, dirígete a 'Edit > Project Preferences' (Edición > Preferencias del Proyecto) en el menú. Aquí debes introducir un nombre y una descripción para tu librería, así como un tipo de licencia (por ejemplo, GNU, CC-BY, etc. más información en [www.opensource.org/licenses](http://www.opensource.org/licenses)).

Para publicar tu librería, ve a 'File > Publish Library' (Archivo > Publicar Librería) en el menú, haz clic en 'Publish' (Publicar) y ¡ya está! Puedes encontrar más información sobre la publicación de librerías en [www.xod.io/docs/guide/creating-libraries](http://www.xod.io/docs/guide/creating-libraries).

## Documentación de Nodos

Cuando publiques tu trabajo es una buena práctica asegurarte de que tus nodos están bien documentados. Esto ayuda a recordar lo que has hecho, y permite a otros hacerse una idea de cómo se puede utilizar el nodo.

Antes de publicar tu librería, asegúrate de haber descrito el nodo y cada uno de sus pines. Para escribir una descripción del nodo, haga clic en un espacio en blanco del patch y aparecerá un cuadro de 'Description' (Descripción) en el panel del Inspector. Escribe una breve descripción sobre lo que hace el nodo y para qué se utiliza, por ejemplo: 'Este nodo escribe dos líneas de texto en la pantalla OLED del *ssd1306*'.

Para escribir una descripción de los pines, haz clic en un nodo de entrada o de salida y verás la casilla 'Description' en la parte inferior del panel Inspector. Escribe una breve descripción del pin, por ejemplo: 'Cadena para mostrar en la primera línea de texto. El texto aparecerá en las coordenadas 0:0'.

También puedes proporcionar más información sobre el funcionamiento de tu nodo añadiendo comentarios al patch. Para ello, ve a 'Edit > Insert Comment' (Editar > Insertar Comentario) en la barra de menú. Puedes encontrar más información sobre la documentación de nodos en [www.xod.io/docs/guide/documenting-nodes](http://www.xod.io/docs/guide/documenting-nodes).



# Utilizar Buses

## Tarea 7: Requisitos

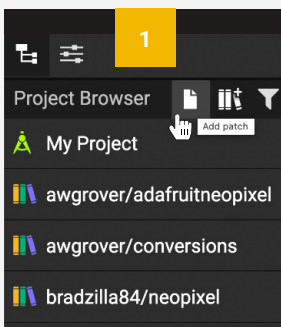
- Computadora con MacOS, Windows o Linux (software XOD y driver USB instalados)
- Grove Beginner Kit for Arduino (módulos de aceleración de 3-ejes y pantalla OLED)
- Cable USB-A a micro USB

En esta tarea veremos otra forma de simplificar nuestros patches: usar buses. Los buses son una forma de enlazar pines de forma 'invisible' para no tener demasiadas intersecciones que hagan el flujo de datos confuso.

Los buses son como los nodos de entrada y salida. Vienen en dos tipos, *to-bus* (hacia bus) y *from-bus* (desde bus), y toman el tipo de datos del pin al que están conectados.

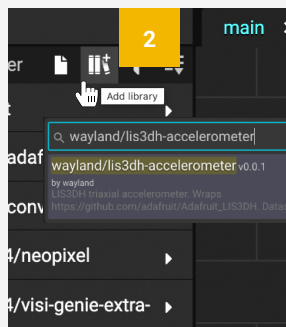
El nodo *to-bus* se utiliza como un nodo de salida y envía información de una salida a un bus. El nodo *from-bus* actúa como un nodo de entrada y recupera información del bus del mismo nombre.

Practicaremos el uso de buses mostrando la salida de nuestro sensor de aceleración de 3 ejes (también conocido como acelerómetro o sensor de inclinación) en la pantalla.



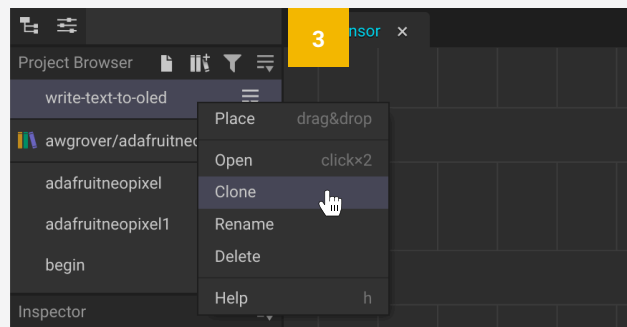
### HAZ UN NUEVO PATCH

Añade un nuevo patch al proyecto, y nómbralo 'tilt-sensor' (sensor de inclinación) (o pasa al tuto701).



### AÑADE LIBRERÍA

Para trabajar con el acelerómetro necesitarás instalar la librería *wayland/lis3dh-accelerometer*.

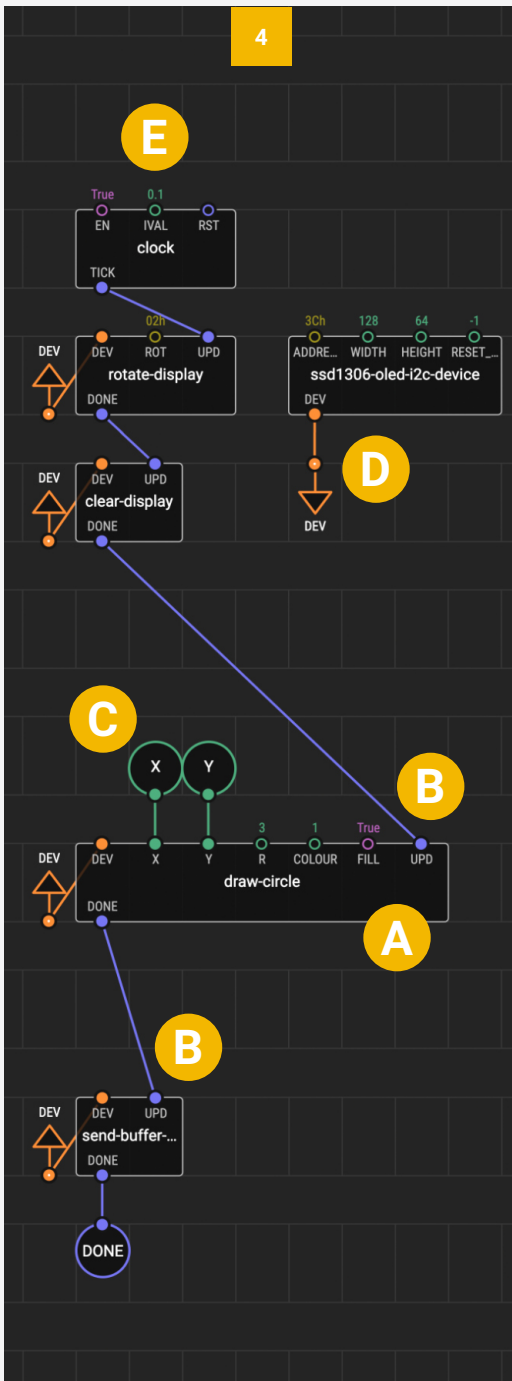


### CLONA EL PATCH WRITE-TEXT-TO-OLED

Volveremos a utilizar la pantalla OLED, pero de una manera diferente. Para no tener que empezar de nuevo, podemos clonar el patch *write-text-to-oled* haciendo clic con el botón derecho del ratón sobre el patch en el Project Browser y seleccionando 'Clone' (Clonar). Cambia el nombre del nuevo patch a 'write-dot-to-oled'.



4



## MODIFICA TU NODO

Esta vez queremos dibujar un pequeño círculo en la pantalla en lugar de texto. El círculo se moverá por la pantalla a medida que inclines el tablero. Sigue las siguientes instrucciones para modificar el patch 'write-dot-to-oled' (escribir punto en oled) para este nuevo propósito.

A

### AÑADE Y CONFIGURA DRAW-CIRCLE

Elimina ambos nodos *draw-text* y los nodos *input-string* asociados. Añade un nodo *draw-circle* (dibujar círculo). Ajusta R (radio) a '3' para hacer un círculo de 3 píxeles de ancho. Deja COLOR como '1'. Ajusta FILL (relleno) en 'True' para obtener un círculo sólido.

B

### RECONECTA

Enlaza el pin DONE de *clear-display* con el pin UPD de *draw-circle*. Enlaza el pin DONE de *draw-circle* con el pin UDP de *send-buffer-to-display*.

C

### AÑADE NODOS INPUT-NUMBER

Añade dos nodos *input-number* (entrada número) y conéctalos a los pines X y Y. Utiliza el Inspector para etiquetarlos 'X' y 'Y'.

D

### SUSTITUYE LOS ENLACES POR BUSES

Aunque el patch OLED funcionaba, era muy desordenado. Los enlaces se entrecruzaban, y sería fácil perder una conexión. Vamos a mejorar esto sustituyendo los enlaces con un bus. Elimina los enlaces naranja entre *ssd1306-oled-i2c-device* y los demás nodos. Añade un nodo *to-bus* (hacia bus) (*xod/patch-nodes*) y enlázalo a la pin de salida de *ssd1306-oled-i2c-device*. Usa el Inspector para etiquetar este nodo 'DEV'. Añade un nodo *from-bus* (desde bus) (*xod/patch-nodes*). Etiquetalo 'DEV', ya que los buses sólo pueden comunicarse si tienen el mismo nombre. Repite este proceso, o copia y pega el nodo *from-bus* hasta que tengas cuatro en total. Conecta estos a cada uno de los pines de entrada de los nodos *rotate-display*, *clear-display*, *draw-circle* y *send-buffer-to-display*.

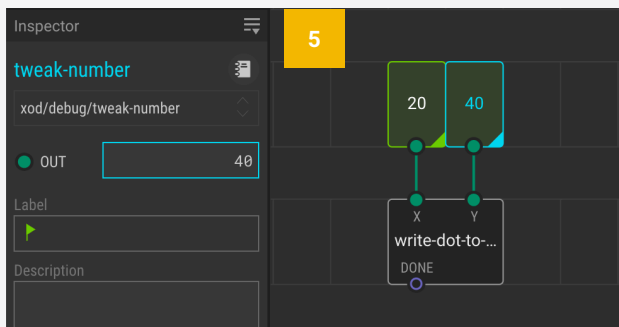
E

### CAMBIA EL TIEMPO DEL RELOJ

Ajusta el pin IVAL de *clock* a 0.1, para que se actualice con mayor frecuencia.

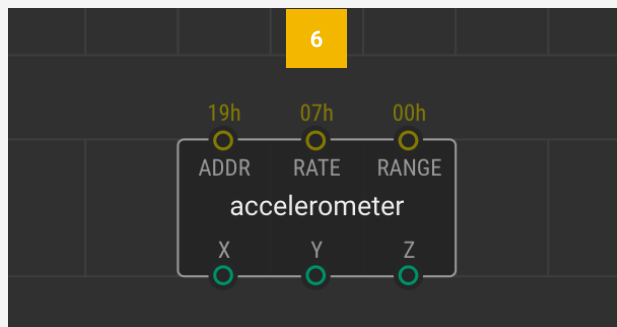


# Utilizar Buses



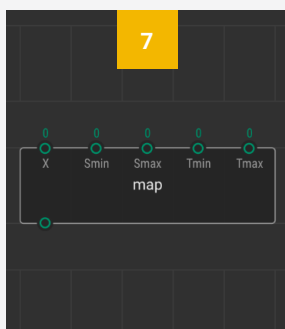
## PRUEBA EL NODO

Prueba el nodo volviendo a tu patch *tilt-sensor*, añadiendo un nodo *'write-dot-to-oled'*, y conectando dos nodos *tweak-number* a las entradas X y Y. Carga y depura. Haz clic en los nodos *tweak-number* y utiliza el Inspector para cambiar sus valores. Observa cómo se desplaza el punto por la pantalla.



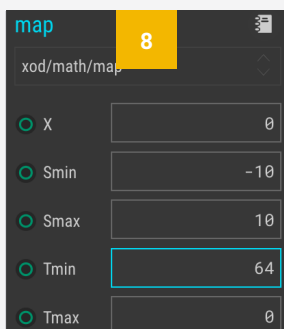
## AÑADE UN NODO ACELEROMETER

Añade un nodo *accelerometer* (acelerómetro) (*wayland/lis3dh-accelerometer*). Utilizaremos las salidas de este nodo para ubicar del punto en la pantalla. Podrías conectar los pines X y Y de *accelerometer* directamente a los pines X y Y de *write-dot-to-oled*, pero no funcionaría ya que los rangos de los nodos no coinciden.



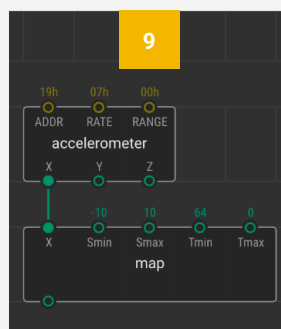
## AÑADE UN NODO MAP

Añade un nodo *map* (*xod/math*). Este nodo mapeará el rango de salida de *accelerometer* al rango de entrada de *write-dot-to-oled*.



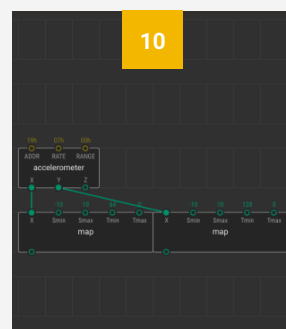
## AJUSTA EL NODO MAP

Ajusta Smin a '-10' y Smax a '10' (el rango del acelerómetro). Ajusta Tmin a '64' (la altura de la pantalla) y Tmax a '0'.



## CONECTA EL NODO MAP

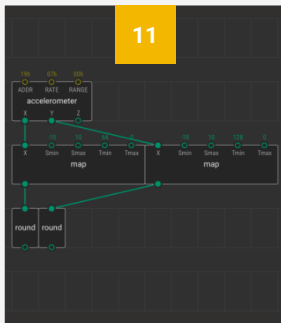
Enlaza los pines X de *accelerometer* y *map*. El nodo convierte el rango X del acelerómetro en valores dentro de la altura de la pantalla.



## REPITE EL NODO MAP

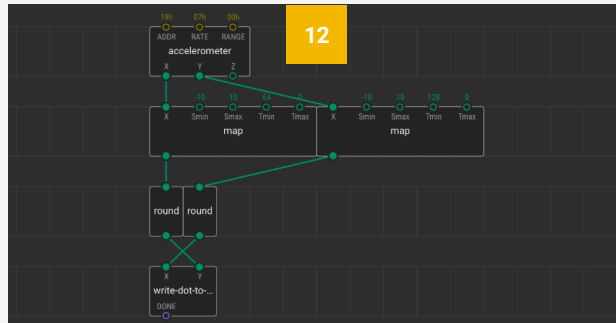
Añade otro nodo *map* y enlázalo a la salida Y. Ajusta Smin a '-10', Smax a '10', Tmin a '128' (el ancho de la pantalla) y Tmax a '0'.





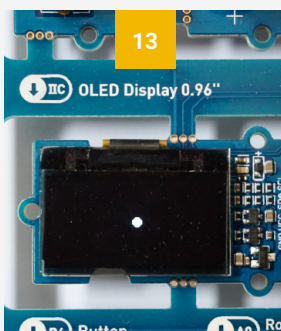
### AÑADE NODOS ROUND

Añade dos nodos *round* (redondos) (*xod/math*). Conéctalos a las salidas de los nodos *map*. Ésto redondeará las salidas.



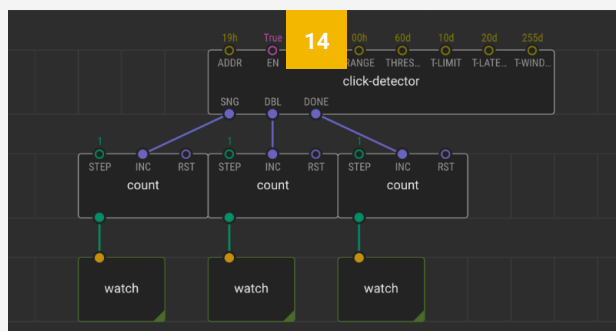
### CONECTA ACCELEROMETER AL OLED

Borra los nodos *tweak-number*. Conecta la pin de salida del nodo *round* vinculada a la pin X de *accelerometer* al pin de entrada Y, y la pin de salida del nodo *round* vinculada a la pin Y de *accelerometer* al pin de entrada X. Esto parece contraintuitivo, pero se debe a la configuración de los diferentes nodos.



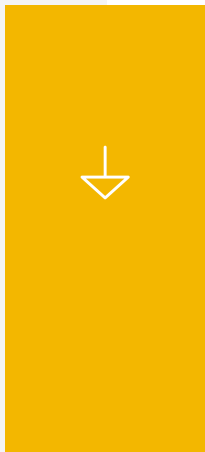
### PRUEBA EL PATCH

Carga el patch. Inclina tu placa a la izquierda y a la derecha, hacia delante y hacia atrás, ¡y observa cómo se mueve el puntito!



### ¡EXPERIMENTA!

Experimenta con los nodos de este patch. ¿Puedes lograr que aparezca algún texto cuando el punto se posa en el centro? También vale la pena explorar la librería *wayland/lis3dh-accelerometer*, ya que tiene nodos útiles y un montón de demostraciones. Por ejemplo, el nodo *click-detector*, que detecta los toques del sensor.



# Programas Lógicos

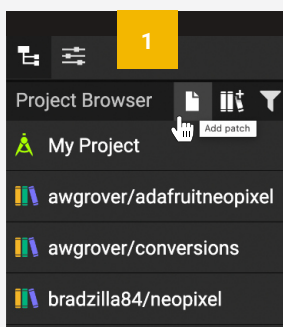
## Tarea 8: Requisitos

- Computadora con MacOS, Windows o Linux (software XOD y driver USB instalados)
- Grove Beginner Kit for Arduino (módulos de sensor de luz y pantalla OLED)
- Cable USB-A a micro USB

En esta tarea tomaremos algunas de las habilidades que hemos aprendido hasta ahora en esta lección y las utilizaremos para crear un programa más complejo que utilice la lógica para indicar a la placa sobre lo que debe hacer.

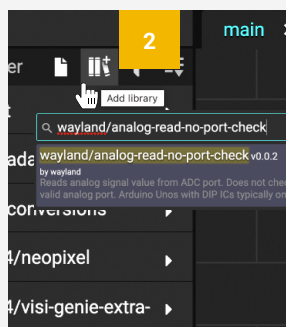
Utilizaremos los módulos del sensor de luz y de la pantalla OLED para crear un sencillo dispositivo de detección de luz.

Aunque se trata de un dispositivo bastante sencillo, contiene muchas de las funciones básicas que puedes utilizar para tus propios instrumentos: una entrada en forma de sensor; un programa lógico que ordena a la placa lo que debe hacer en función del valor de esta entrada; y una salida que cambia algo, en este caso el texto que aparece en una pantalla.



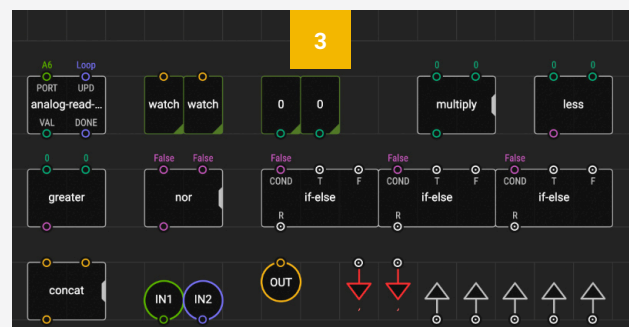
### HAZ UN NUEVO PATCH

Añade un nuevo patch al proyecto, y nómbralo 'light-sensor' (sensor de luz) (o pasa al tuto801).



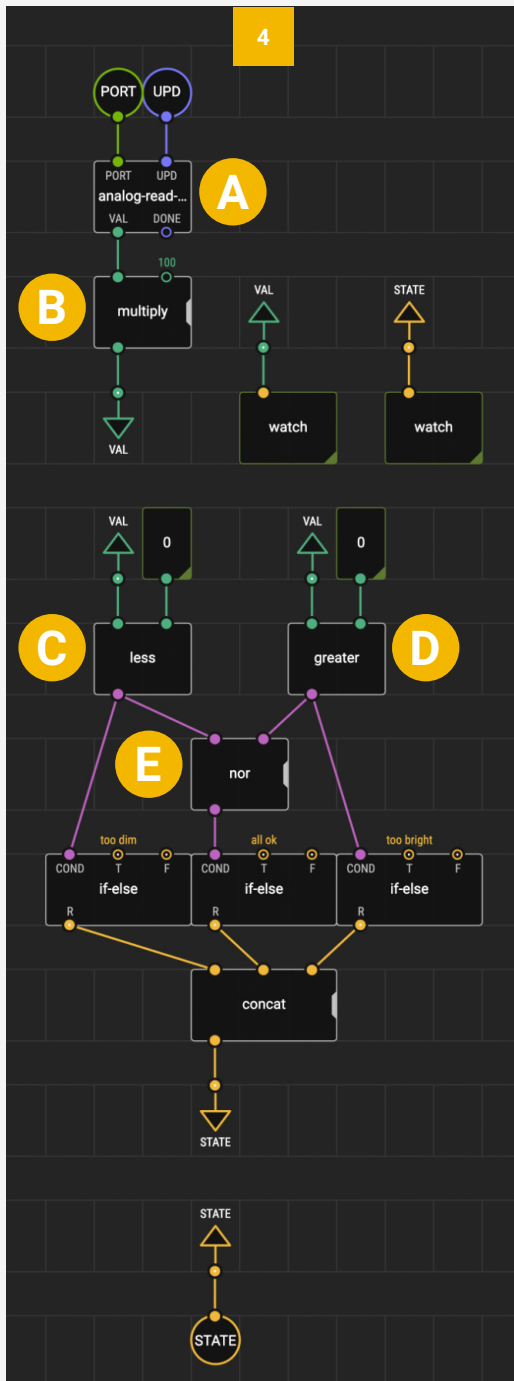
### AÑADE LIBRERÍA

Para trabajar con el sensor de luz necesitarás instalar la librería *wayland/analog-read-no-port-check*.



### AÑADE NODOS

Añade los siguientes nodos al patch *light-sensor*: *analog-read-no-port-check* (*wayland/analog-read-no-port-check*), *watch* x2, *tweak-number* x2 (*xod/debug*), *multiply*, *less*, *greater*, *nor*, *if-else* x3, *concat* (*xod/core*), *input-port*, *input-pulse*, *output-string*, *to-bus* x2, *from-bus* x5 (*xod/patch-nodes*).



## CONFIGURA EL NODO LIGHT-SENSOR (PARTE 1)

Queremos que nuestro nodo devuelva una de tres lecturas en función de la intensidad de la luz: 'too dim' (demasiado tenue), 'all ok' (todo bien) o 'too bright' (demasiado brillante). Utilizaremos una combinación de funciones lógicas para programar esta capacidad. Sigue las instrucciones a continuación para configurar tu patch.

**A ANALOG-READ-NO-PORT-CHECK**  
Este nodo (leer análogo sin verificar puerto) representa el sensor de luz. Conecta VAL a la primera entrada del nodo *multiply*.

**B MULTIPLY**  
Usaremos el nodo *multiply* (multiplicar) para aumentar la salida del nodo sensor. El nodo multiplicará los valores de entrada, así que pongamos la segunda entrada a '100'. Conecta uno de los nodos *to-bus* a la salida de *multiply* y etiquétalo como 'VAL'.

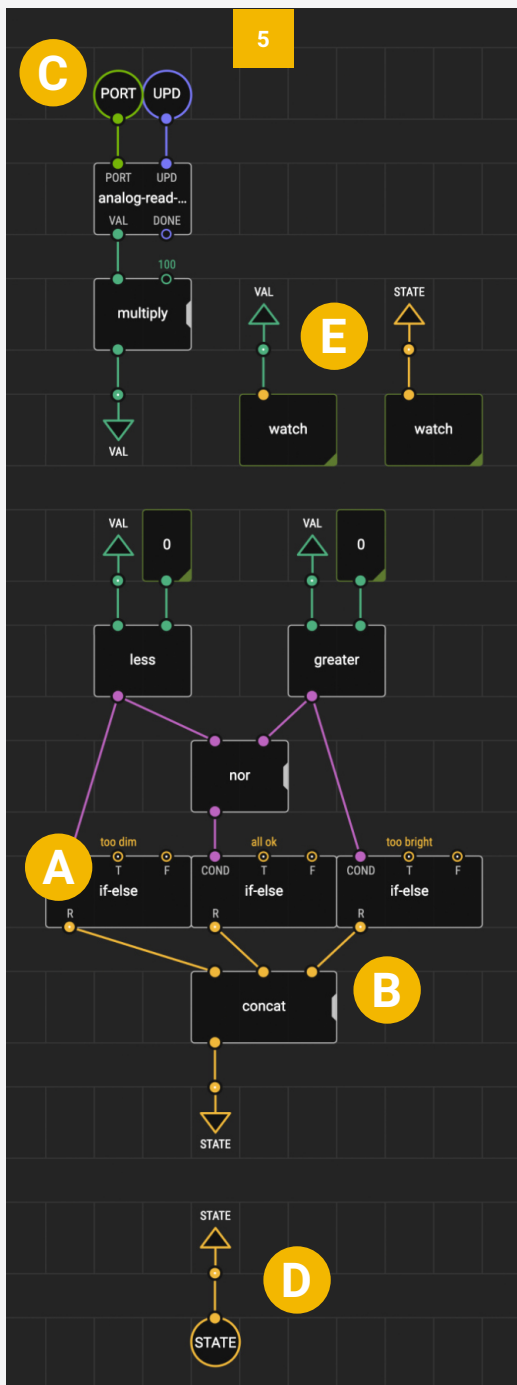
**C LESS**  
Este nodo (menor) toma el primer valor de entrada y lo compara con el segundo. Devolverá 'True' si el primer valor es menor que el segundo, y 'False' si no lo es. Conecta uno de los nodos *from-bus* a la primera entrada y etiquétalo como 'VAL' para que reciba el valor del bus. Conecta uno de los nodos *tweak-number* a la segunda entrada para que puedas ajustar el límite inferior.

**D GREATER**  
El nodo *greater* (mayor) es muy similar al nodo *less*, pero devolverá 'True' sólo cuando la primera entrada sea mayor que la segunda. Conecte un nodo *from-bus* a la primera entrada y etiquétalo como 'VAL'. Luego conecta un nodo *tweak-number* a la segunda entrada para permitirte ajustar el límite superior.

**E NOR**  
El nodo *nor* (ni) sólo devolverá 'True' si ambas entradas leen 'False'. Queremos un tercer estado que se active si la intensidad de la luz no es menor que el límite inferior, ni mayor que el límite superior. Utilizaremos el nodo *nor* para conseguirlo. Conecta la salida del nodo *menor* a la primera entrada, y la salida del nodo *mayor* a la segunda entrada.



# Programas Lógicos



## CONFIGURA EL NODO DEL SENSOR DE LUZ (PARTE 2)

Ahora debería haber completado la primera mitad del patch. Continúa configurando la segunda mitad siguiendo las instrucciones siguientes.

### A IF-ELSE

El nodo *if-else* (si-entonces) emitirá un valor (T) si la condición (COND) que recibe es verdadera, y otro (F) si es falsa. Queremos configurar los tres nodos *if-else* para que cada una de las condiciones anteriores (menor que el límite inferior, mayor que el límite superior, o ninguna de las dos) devuelva una línea de texto diferente.

Ajusta el pin T del primer nodo *if-else* a 'too dim' (demasiado tenue) y conecta el pin COND a la salida de *less*. Ajusta el pin T del segundo nodo *if-else* a 'all ok' (todo bien) y conecta el pin COND a la salida de *nor*. Ajusta el pin T del tercer nodo *if-else* a 'too bright' (demasiado brillante) y conecta el pin COND a la salida de *greater*. Dejaremos los pines F en blanco, para que no se devuelva nada cuando las condiciones sean falsas.

### B CONCAT

Utiliza la pestaña *variadic* del nodo *concat* para ampliarlo a tres entradas. Conecta las tres salidas de los nodos *if-else* a las entradas de *concat*. Esto combinará las tres respuestas en una sola cadena. Debido a las condiciones lógicas, sólo se devolverá una cadena a la vez y cada uno de los otros dos nodos devolverá un valor en blanco. Conecta el segundo nodo *to-bus* a la salida de *concat* y etiquétalo como 'STATE'.

### C NODOS INPUT

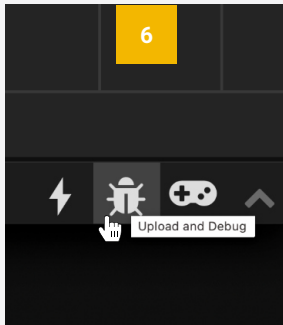
Conecta el nodo *input-port* (entrada puerto) al pin PORT de *analog-read-no-port-check* y etiquétalo 'PORT'. Conecta el nodo *input-pulse* (entrada pulso) al pin UPD de *analog-read-no-port-check* y etiquétalo 'UPD'.

### D NODO OUTPUT

Conecta un nodo *from-bus* al nodo *output-string* (salida cadena) y etiqueta ambos nodos como 'STATE'. Esto parece redundante, pero nos ayudará con el siguiente paso.

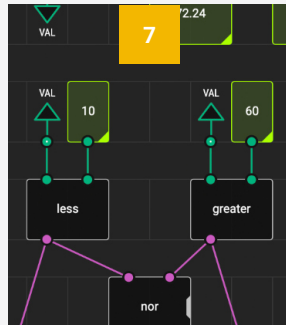
### E NODOS WATCH

Conecta un nodo *from-bus* a cada nodo *watch*. Etiqueta uno 'VAL' y otro 'STATE'. Esto enlazará uno a la salida multiplicada por 100 del nodo sensor para que puedas ver la lectura del sensor, y otro a la salida final del nodo *concat* para que puedas ver el estado actual. Añadiendo los buses aquí podemos poner los dos nodos de vigilancia juntos y ver fácilmente las salidas una al lado de la otra, en lugar de tener que moverse por la pantalla.



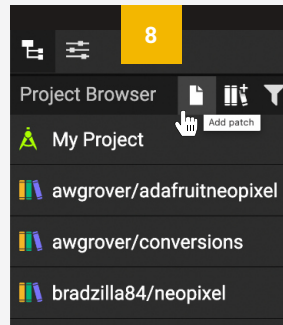
### CARGAR Y DEPURAR

Carga y depura el patch *light-sensor* usando el botón de la mariquita.



### ESTABLECE EL RANGO

Utiliza los nodos *tweak-number* para encontrar un rango. Hemos utilizado 10-60 pero puedes ajustarlo a tu entorno.



### HAZ UN NUEVO PATCH

Añade un nuevo patch al proyecto, y nómbralo 'light-sensor-display' (mostrar sensor de luz) (o pasa al tuto810).



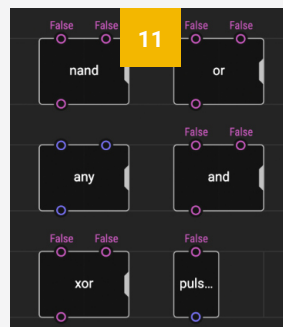
### AÑADE Y CONECTA LOS NODOS

Añade tu nodo *light-sensor* y un nodo *write-text-to-oled* al patch. Ajusta LINE1 a 'Light:' (Luz:) y conecta STATE a LINE2.



### PRUEBA EL PATCH

Carga el patch. Cambia la intensidad de la luz moviendo o cubriendo la placa. Observa cómo afecta el texto en la pantalla.



### ¡EXPERIMENTA!!

Explora los otros nodos lógicos de la librería *xod/core*. O prueba a utilizar *pulse-on-true* con *if-else* para controlar un programa.



# Secuencias y Bucles

## Tarea 9: Requisitos

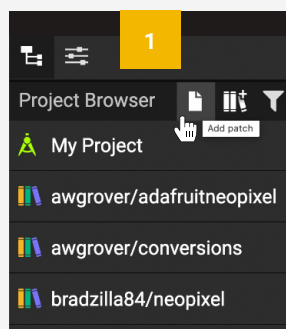
- Computadora con MacOS, Windows o Linux (software XOD y driver USB instalados)
- Grove Beginner Kit for Arduino (módulo buzzer)
- Cable USB-A a micro USB

En esta última tarea exploraremos una de las habilidades más útiles: la creación de secuencias y bucles ( o 'loops').

Programando una secuencia de eventos, utilizando la lógica e introduciendo bucles podemos fabricar dispositivos útiles para tareas como la automatización, el monitoreo y la respuesta a las condiciones ambientales.

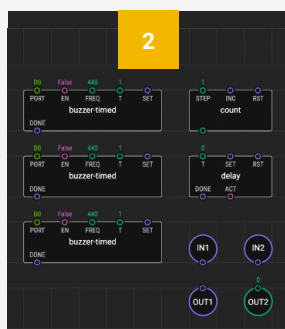
Esta tarea introducirá estas habilidades utilizando el buzzer para tocar una melodía sencilla 'Hot Cross Buns' (pan de cruz caliente - una canción infantil inglesa).

Crearemos dos secuencias separadas y usaremos nodos lógicos para instruir al programa sobre cuándo reproducirlas. Utilizaremos dos métodos diferentes para hacer las secuencias, y crearemos un nodo distinto para cada una.



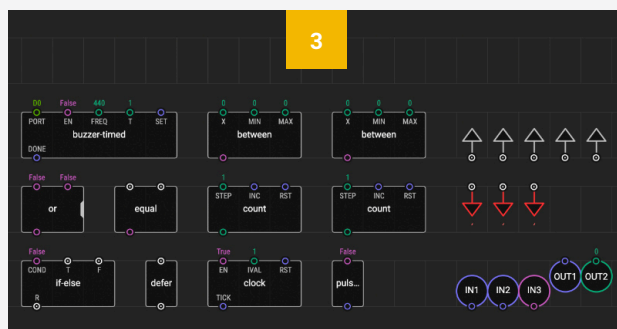
### HAZ TRES NUEVOS PATCHES

Añade tres patches y nómbralos 'bar124' (compás 124), 'bar3' (compás 3) y 'play-tune' (tocar melodía) (o pasa al tuto901)



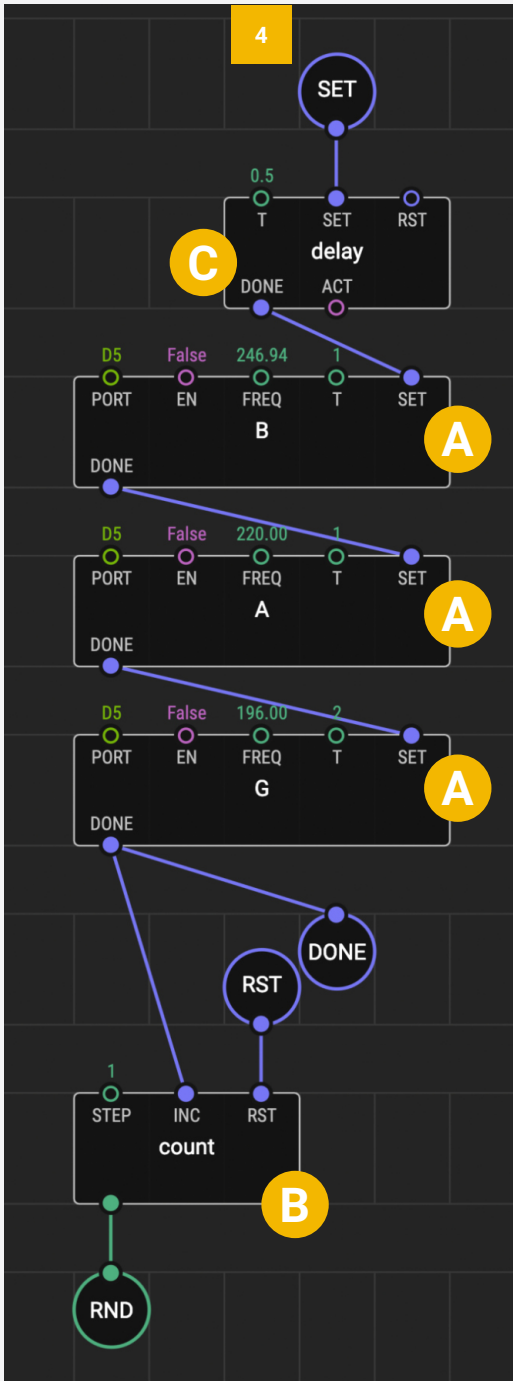
### AÑADE NODOS A BAR124

Añade: *buzzer-timed* x3 (*marcoaita/malibrary*), *delay*, *count* (*xod/core*), *input-pulse* x2, *output-number* (*xod/patch-nodes*).



### AÑADE NODOS A BAR3

Añade los siguientes nodos al patch *bar3*: *buzzer-timed* (*marcoaita/malibrary*), *clock*, *count* x2, *or*, *defer*, *if-else*, *equal*, *pulse-on-true* (*xod/core*), *between* x2 (*e/comparison* - necesitarás instalar esta librería) *input-pulse* x2, *input-boolean*, *output-pulse*, *output-number*, *to-bus* x3, *from-bus* x5 (*xod/patch-nodes*).



## CONFIGURA TU NODO BAR124

En este patch crearemos la secuencia de notas (B,A,G) que componen los compases 1, 2 y 4 de la melodía.

### A NODOS BUZZER-TIMED

Ajusta cada uno de los nodos *buzzer-timed* (zumbador cronometrado) para que toquen una nota diferente cambiando sus pines de frecuencia. Ajusta uno a '246.94' Hz y nómbralo 'B' usando el Inspector. Ajusta uno a '220' Hz y nómbralo 'A'. Ajusta uno a '196' Hz y nómbralo 'G'. Ajusta PORT a 'D5' y EN a 'False' en los tres. Ajusta T (tiempo) a '1' en los nodos A y B y a '2' en el nodo G. Esto producirá una nota final más larga. Conecta el pin DONE de B al pin SET de A y el pin DONE de A al pin SET de G. Añade el nodo *output-pulse* al pin DONE de G y nómbralo 'DONE', para que podamos ver cuando la secuencia está completa.

### B COUNT

Conecta el pin DONE de G al pin INC de *count* para que la cuenta aumente cada vez que se complete la secuencia. Añade el nodo *output-number* (salida número) al pin de salida y nómbralo 'RND' (ronda), para que podamos ver el número de veces que se ha reproducido la secuencia. Añade un nodo *input-pulse* al pin RST (reiniciar) y nómbralo 'RST'. Lo utilizaremos para reiniciar la cuenta cuando la melodía termine.

### C DELAY

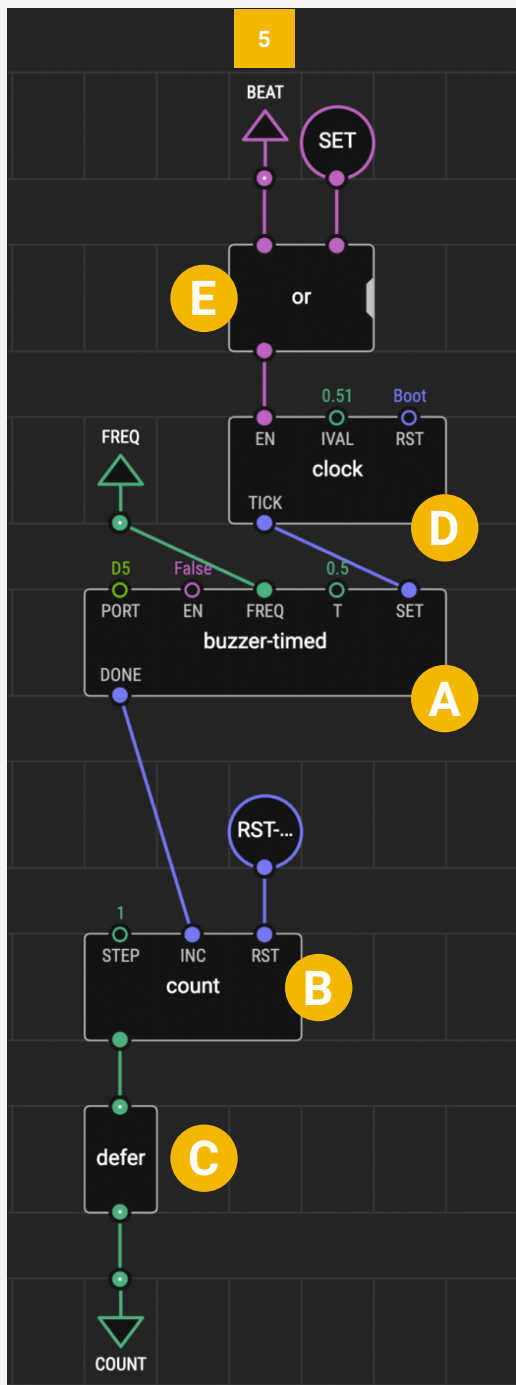
Utilizaremos el nodo *delay* (retraso) para introducir un espacio de medio segundo entre cada compás. Ajusta T (tiempo) a 0.5, y conecta el pin DONE de *delay* al pin SET de B. Añade un nodo *input-pulse* al pin SET y nómbralo 'SET'. Utilizaremos esta entrada para iniciar la secuencia.

Para probar este nodo, puedes añadir un nodo *tweak-pulse* al pin SET de *delay*. Carga y depura, luego usa el nodo *tweak-pulse* para iniciar la secuencia. Este patch utiliza una serie de nodos con pines SET y DONE conectados para crear una secuencia, de forma similar al nodo *write-text-to-oled* en la **Tarea 6** (p49). El patch *bar3* creará una secuencia de una manera diferente.





# Secuencias y Bucles

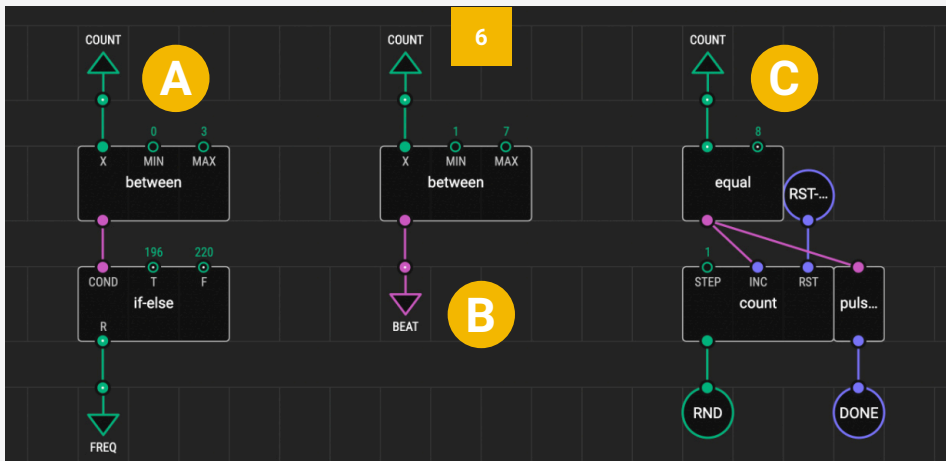


## CONFIGURA EL NODO BAR3 (PARTE 1)

En este patch crearemos la secuencia de notas (G x 4, A x 4) que compone el compás 3 de la melodía. Lo haremos de una forma diferente a la del nodo bar134. Primero configuraremos una secuencia para cronometrar las notas, luego usaremos nodos lógicos para controlar la frecuencia y el número. Esto creará un bucle que alimenta la información que sale del programa de vuelta a la secuencia. Utilizaremos buses para conectar las dos mitades de este nodo.

- A BUZZER-TIMED**  
Ajusta PORT a 'D5' y EN a 'Falso'. Ajusta T (tiempo) a 0.5. Añade un nodo *from-bus* al pin FREQ y nómbralo 'FREQ'. Esto permitirá a nuestros nodos lógicos establecer la frecuencia de la nota dependiendo del número de compás.
- B COUNT**  
Conecta el pin DONE de *buzzer-timed* al pin INC de *count*, para que la cuenta aumente cada vez que suene el buzzer. Añade un nodo *input-pulse* al pin RST y nómbralo 'RST-BEAT'. Lo utilizaremos para reiniciar la cuenta al final.
- C DEFER**  
El nodo *defer* (diferir) es la clave para crear bucles en XOD. En este caso, estamos creando un bucle que lee el número de compás, y cambia la frecuencia de la nota y decide si se repite en base a esto. Conecta el pin de salida de *count* al pin de entrada de *defer* para informar a XOD de este bucle. Añade un nodo *to-bus* al pin de salida y nómbralo 'COUNT'. Esto alimentará nuestros nodos lógicos.
- D CLOCK**  
Queremos que el buzzer suene una nota corta repetidamente. Conecta el nodo *clock* al pin SET de *buzzer-timed*. Ajusta RST a 'On Boot' y ajusta el pin IVAL (intervalo) a 0.51, que es un poco más largo que el tiempo del buzzer.
- E OR**  
Con *clock* hemos hecho que el buzzer suene regularmente, pero no queremos que suene todo el tiempo. Tenemos que añadir condiciones que especifiquen cuándo se activa el reloj. El buzzer debe sonar O BIEN al comienzo del tercer compás, O BIEN cuando la secuencia ha comenzado pero aún no ha terminado, es decir, después de los compases 1-7, pero no después del compás 8. Conecta el nodo *or* (o) al pin EN de *clock*. Añade un nodo *input-pulse* a uno de los pines de entrada y llámalo 'SET'. Lo utilizaremos para iniciar la secuencia al principio del compás 3. Añade un nodo *from-bus* al otro pin de entrada y llámalo 'BEAT'. Esto continuará la secuencia después de los compases 1-7.





## CONFIGURA EL NODO BAR3 (PARTE 2)

Sigue las instrucciones siguientes para completar la segunda mitad del bucle.

**A**

### FRECUENCIA

Esta parte cambia la nota entre G y A en función del número de cuenta. Utiliza uno de los nodos *between* (entre). Añade un nodo *from-bus* al pin X y nómbralo 'COUNT' para que reciba el número de conteo. Ajusta MIN a '0' y MAX a '3'. Conecta la salida al pin COND del nodo *if-else*. Ajusta T a '196' para que el buzzer toque un G durante las cuatro primeras notas (cuenta entre 0-3). Ajusta F a '220' para que el zumbador toque un A en caso contrario. Añade un nodo *to-bus* al pin de salida de *if-else* y llámalo 'FREQ'. Esto se retroalimentará en el bucle para ajustar el pin FREQ del nodo *buzzer-timed*.

**B**

### RITMO

Esta parte iniciará el buzzer después de las cuentas 1-7. Utiliza el otro nodo *between*. Añade un nodo *from-bus* a X y nómbralo 'COUNT'. Establece MIN a '1' y MAX a '7'. Añade un nodo *to-bus* a la salida y nómbralo 'BEAT'. Esto se retroalimentará en el bucle como una de las condiciones que iniciarán el reloj.

**C**

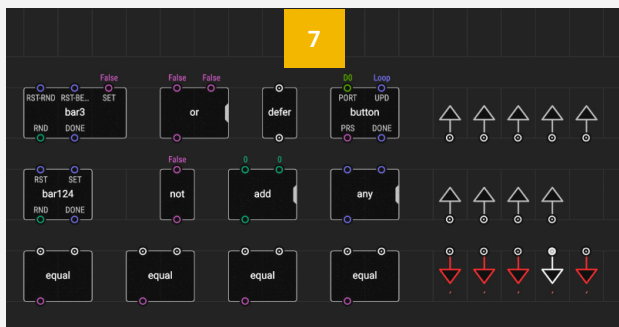
### RONDA

Esta parte registra cuando la secuencia ha terminado. Conecta un nodo *from-bus* a un pin de entrada de *equal* (igual) y nómbralo 'COUNT'. Pon el otro pin de entrada a '8'. Conecta el pin de salida al pin INC de *count* para que el conteo aumente cuando la secuencia haya terminado. Conecta el nodo *input-pulse* al pin RST de *count* y llámalo 'RST-RND'. Lo utilizaremos para reiniciar el conteo. Añade el nodo *output-number* a la salida de *count* y nómbralo 'RND' para que podamos ver la ronda de la secuencia. Conecta el nodo *pulse-on-true* (pulsar en verdadero) a *equal* y luego añade el nodo *output-pulse* y nómbralo 'DONE'. Esto nos permitirá ver cuando la secuencia está completa.

Tu nodo bar3 está ahora completo. Probar este nodo añadiendo los nodos *flip-flop* y *tweak-pulse* a una de las entradas de *or*. Pulsa esto dos veces para iniciar la secuencia.

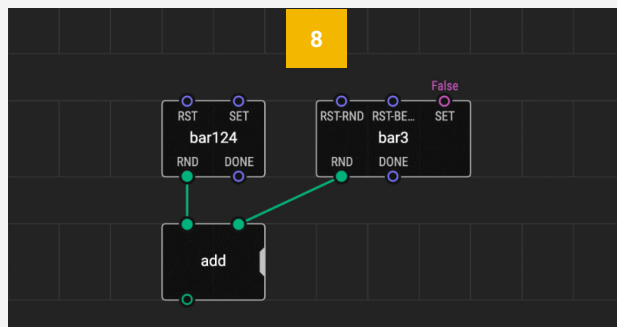


# Secuencias y Bucles



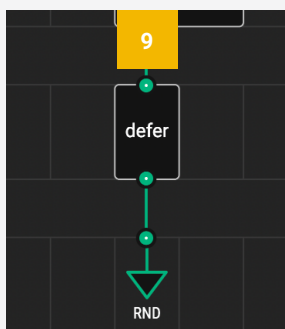
## AÑADIR NODOS TO PLAY-TUNE

Ahora configuraremos una serie de condiciones lógicas para decidir cuándo tocar cada compás. Añade los siguientes nodos al patch *play-tune*: *bar124*, *bar3*, *add*, *defer*, *equal* x4, *or*, *any*, *not* (*xod/core*), *button* (*xod/common-hardware*), *to-bus* x5, *from-bus* x9 (*xod/patch-nodes*).



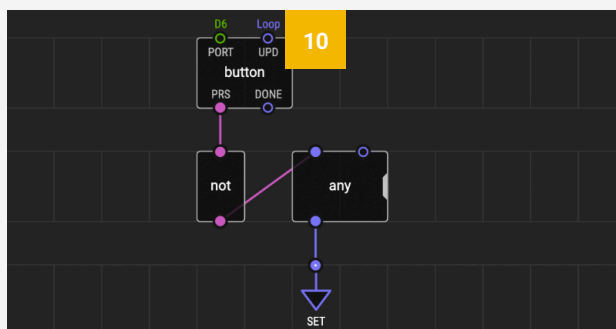
## CONTAR EL NÚMERO DE LA RONDA

Para empezar necesitamos saber el número de la ronda actual. Añadimos un nodo *output-number* (*RND*) a cada uno de nuestros nodos (*bar124* y *bar3*) para contar cuántas veces se ha tocado. Conecta ambos pines *RND* al nodo *add* (añadir) para sumar estos dos valores y encontrar el número de la ronda.



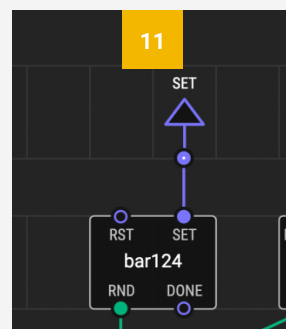
## DEFER Y BUS DE RND

Crearemos un bucle de retroalimentación. Añade un nodo *defer* a la salida de *add*. Luego agrega un nodo *to-bus* y nómbralo 'RND'.



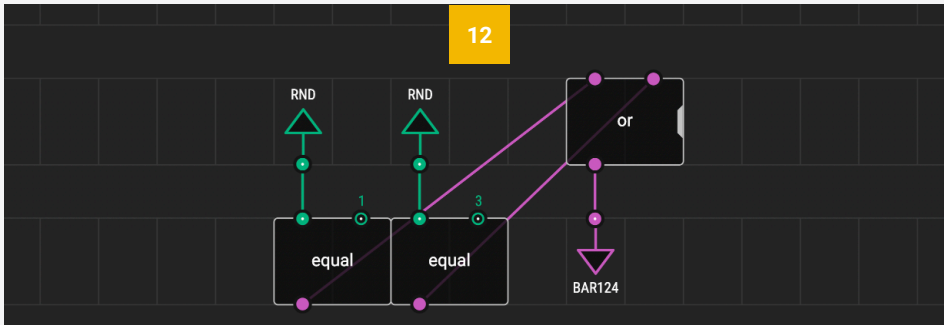
## TOCAR EL COMPÁS 1

Utilizaremos el botón de la placa para iniciar la ronda 1. Pon el pin *PORT* de *button* a 'D6' y el pin *PRS* a 'True'. Luego conecta el pin *PRS* al nodo *not*. Conecta la salida de *not* a una de las entradas de *any*. Deja la otra entrada sin conectar por ahora. Añade un nodo *to-bus* al pin de salida de *not* y nómbralo 'SET'.



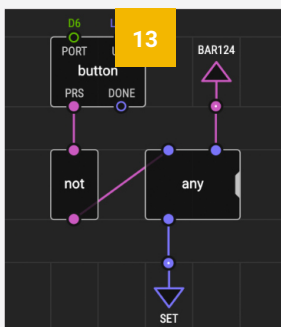
## BUS DE SET

Añade un nodo *from-bus* al pin *SET* de *bar124* y nómbralo 'SET'. Este bus iniciará la secuencia de *bar124*.



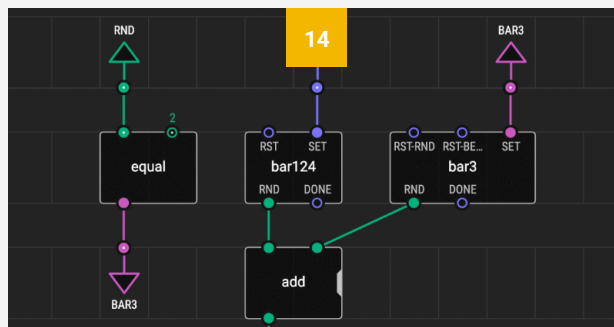
### TOCANDO LOS COMPASES 2 Y 4

También queremos iniciar *bar124* al final de los compases 1 y 3. Utiliza dos nodos *equal*. Conecta un nodo *from-bus* a la primera entrada de cada uno y llámalos a ambos 'RND' para que reciban el número de la ronda. Pon la segunda entrada de uno en '1' y la del otro en '3'. Conecta las salidas de ambos nodos a los pines de entrada del nodo *or*. Añade un nodo *to-bus* a la salida de *or* y nómbralo 'BAR124'.



### BUS DE BAR124

Añade un nodo *from-bus* a la otra entrada de *any* (ver **Paso 10**). Nómbralo 'BAR124'. *bar124* tocará después de los compases 1 y 3.

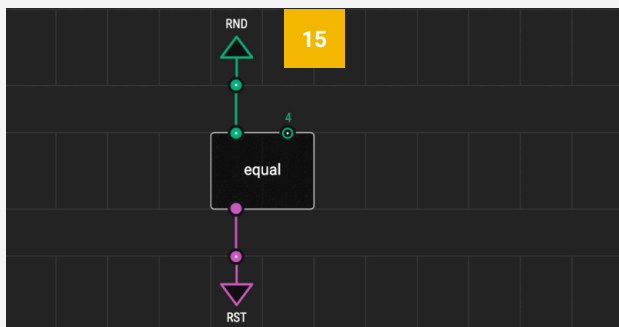


### TOCANDO EL COMPÁS 3 Y EL BUS DE BAR3

Queremos que *bar3* suene al final del compás 2. Añade un nodo *from-bus* a la entrada del tercer nodo *equal* y llámalo 'RND'. Pon la segunda entrada de *equal* a 2. Añade un nodo *to-bus* a la salida y llámalo 'BAR3'. Añade un nodo *from-bus* al pin SET de *bar3* y nómbralo 'BAR3'. Este bus iniciará *bar3* después del compás 2.

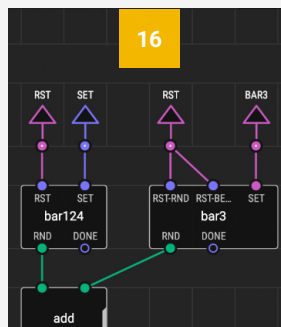


# Secuencias y Bucles



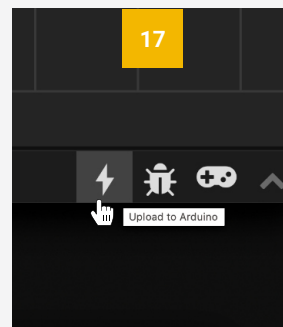
## REINICIA LA SECUENCIA

Después de la ronda 4 queremos reiniciar la secuencia. Para ello añadimos pines RST a nuestros nodos bar124 y bar3. Añade un nodo *from-bus* a la primera entrada del nodo igual y nómbralo 'RND'. Ajusta la otra entrada a '4' para que el reinicio ocurra después de la compás 4. Añade un nodo *to-bus* al pin de salida y nómbralo 'RST'.



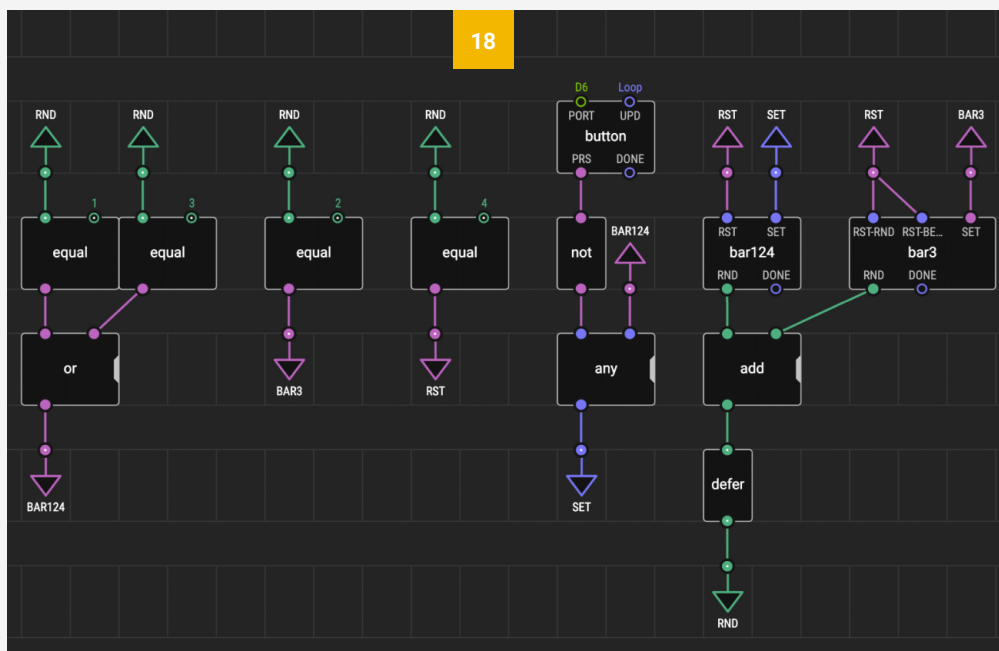
## BUS DE RST

Nombra ambos nodos *from-bus* 'RST'. Añade uno al pin RST de *bar124* y otro a los pines RST (RST-RND, RST-BEAT) de *bar3*.



## CARGA Y PRUEBA

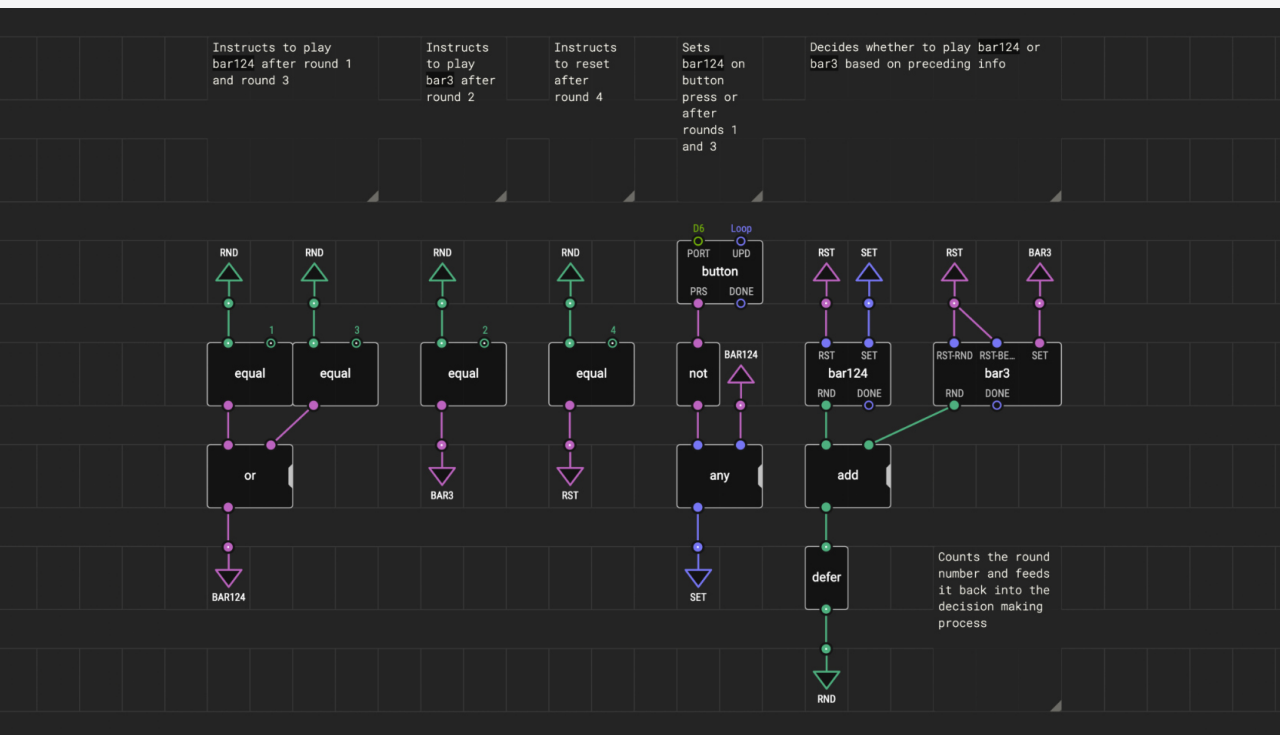
Por último, carga el programa y Pruébalo pulsando el botón de tu placa.



## ¡EXPERIMENTA!

Felicidades, ¡has completado la tarea final! Tu patch final debería ser algo parecido a esto.

Juega con este patch y con los nodos que has utilizado. Intenta crear una melodía diferente. O intenta hacer que el LED parpadee al ritmo de las notas. Las posibilidades son infinitas.



## Cajas de Comentarios

Cuando se crea un programa más complejo como este, a menudo es útil incluir cajas de comentarios, tanto para ayudar a mantener un registro de lo que estás haciendo, como para hacer más fácil para otros seguir tu flujo de trabajo. Esto es el equivalente en XOD a 'comentar' las notas cuando se escribe código.

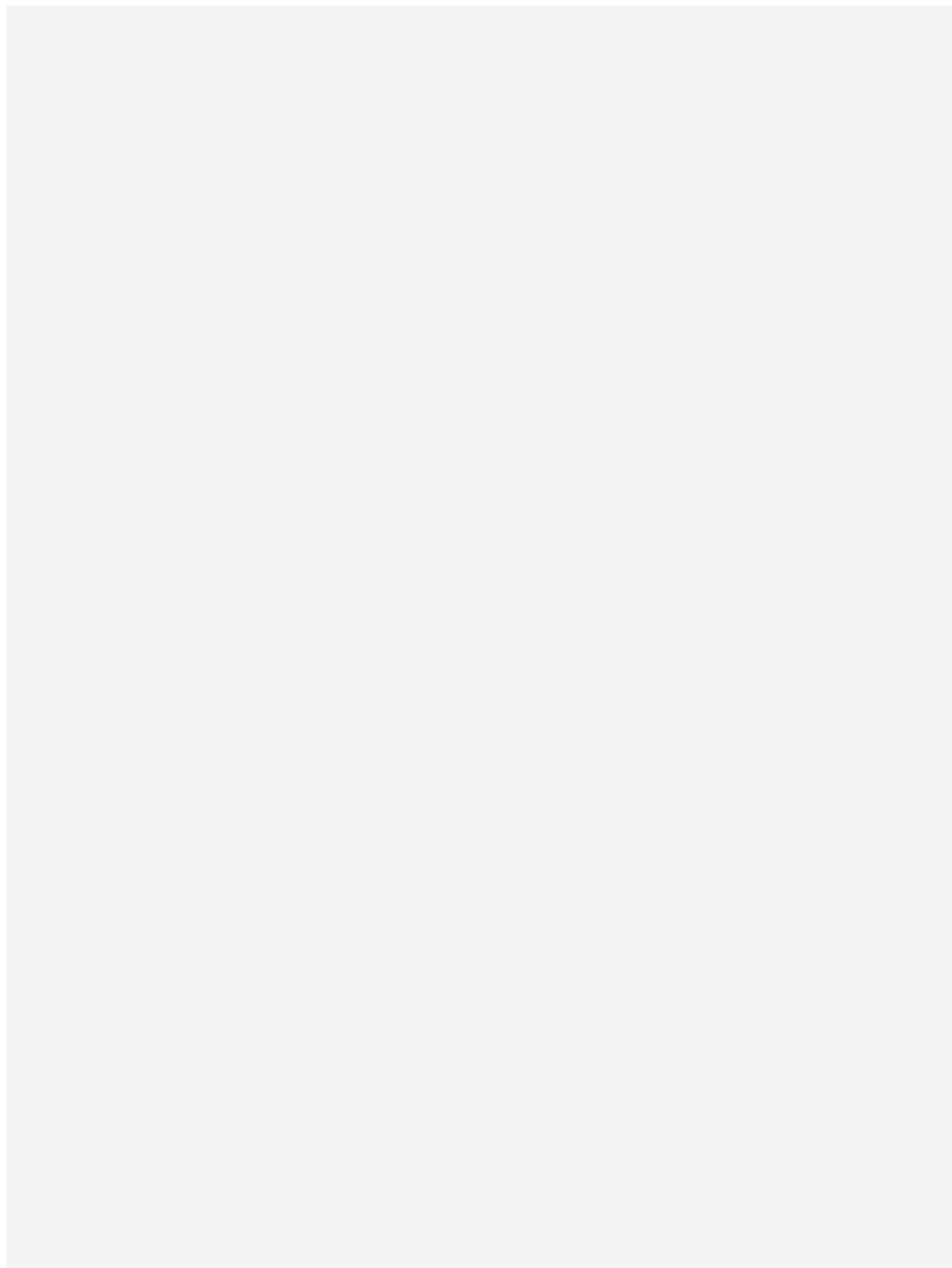
En el ejemplo anterior puedes ver que se han añadido comentarios en cada sección del programa para describir lo que hace esa parte del patch. Intenta añadir tus comentarios para anotar tu patch *play-tune*. Puedes añadir una caja de comentarios navegando hasta 'Edit > Insert Comment' (Editar > Insertar comentario) en la barra de menú.

También puedes añadir formato a tus cajas de comentarios XOD, por ejemplo:

- **\*Rodea el texto con estrellas para añadir texto blanco en negrita\***
- **\*\*Rodea el texto con dos estrellas para añadir un texto rojo en negrita\*\***
- - Utiliza un guión antes del texto para añadir una lista de viñetas
- 1. Utiliza un número y un punto antes del texto para añadir una lista de números

Puedes leer más sobre cómo añadir comentarios para documentar tus nodos y el 'markdown' (formato) de XOD en la página web de XOD [www.xod.io/docs/guide/documenting-nodes](http://www.xod.io/docs/guide/documenting-nodes).





# Lección 5: Siguientes Pasos



**Ampliando Tu Capacidad**



**Estudios de Caso**



**Información Adicional**

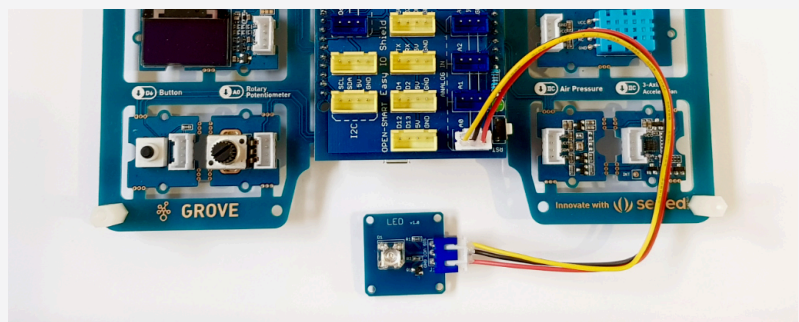


# Siguientes Pasos

Felicidades, ¡has completado el curso de Programación Sin Código para Biología! Empezando por la comprensión de tu placa y cómo programarla, hasta la construcción de tus propios nodos y secuencias complejas, esta guía te ha llevado a través de cómo utilizar cada uno de los dispositivos de la placa, así como a realizar una serie de funciones útiles en XOD.

Ahora deberías sentirte cómodo con tu placa y el software XOD, y deberías entender mejor cómo se pueden utilizar estas herramientas para crear programas y dispositivos que respondan a su entorno e influyan en él. Una vez que entiendas los principios básicos de la programación de microcontroladores como éste, las posibilidades de aplicación son infinitas.

En este capítulo proporcionaremos información sobre cómo puedes aprovechar estas habilidades para empezar a desarrollar tus propios dispositivos personalizados, así como dónde encontrar componentes, información y ayuda para guiarte en tus próximos pasos. También ofreceremos algunos ejemplos de cómo los anteriores participantes de Biomaker han hecho uso de estos conocimientos, aplicándolos al mundo real para contribuir a la investigación biológica en el laboratorio y en el campo. Por último, proporcionamos información adicional útil, incluyendo una visión general de las placas de desarrollo alternativas, una lista de sitios web útiles, una ficha de apoyo para la placa Grove, una lista de nodos XOD utilizados en esta guía y un glosario de términos.



*'Open Smart easy-plug LED breakout board' conectada a la placa Grove*

## OBJETIVOS

Al final de este capítulo deberás ser capaz de:

- Recordar dónde encontrar componentes adicionales compatibles con tu placa Grove.
- Reconocer las diferentes formas de conectar nuevos componentes a tu placa Grove.
- Localizar nodos XOD compatibles con nuevos dispositivos.
- Describir algunos ejemplos de cómo se pueden aplicar estos conocimientos a la investigación en el campo de la biología.
- Recordar dónde encontrar información adicional y ayuda para construir tus propios dispositivos.



# Ampliando Tu Capacidad

## Componentes Adicionales

La gran variedad de componentes de bajo costo disponibles para trabajar con Arduino puede ser desalentadora, y entender cómo usar y conectar estos componentes a tu placa puede parecer complejo. A continuación describimos algunos de los sistemas sencillos disponibles para conectar nuevo hardware a tu placa, y en la siguiente página hablaremos de cómo conectar o cablear estos componentes.

**COMPONENTES GROVE** La placa Grove está fabricada por la empresa de hardware libre Seeed Studio, la cual proporciona toda una serie de componentes Grove que son compatibles con la placa Grove a través de simples conectores 'plug-and-play'. Estos componentes pueden conectarse directamente a la placa utilizando los enchufes blancos situados en el centro de la placa y los cables incluidos en el kit. Puedes consultar los componentes compatibles con Grove en el sitio web de Seeed Studio ([www.seeedstudio.com](http://www.seeedstudio.com) > Shop > Grove).

**M5STACK** M5Stack es otra empresa de hardware que vende sus propios componentes compatibles con Grove, ellos los denominan 'units' (unidades). Puedes consultar los componentes compatibles con M5Stack en el sitio web de M5Stack ([www.m5stack.com](http://www.m5stack.com) > Store > Unit).

**COMPONENTES OPEN SMART** Open Smart proporciona un sistema 'easy-plug' similar al sistema Grove, así como una mayor variedad de componentes que requieren un cableado sencillo y sin soldaduras. Los conectores de Grove no son compatibles con los conectores Open Smart, por lo que es necesario un 'shield' de expansión para conectar estos componentes a la placa (ver p77). Puedes consultar los componentes Open Smart en la tienda Open Smart Ali Express ([www.open-smart.aliexpress.com](http://www.open-smart.aliexpress.com)).

**OTROS COMPONENTES** Grove y Open Smart proporcionan sistemas fáciles de usar para conectar componentes a tu Grove (o cualquier otra placa Arduino), sin embargo, también hay una variedad de otros proveedores y componentes disponibles si introduces una pequeña cantidad de cableado y soldadura. Compañías como Adafruit, SparkFun y Seeed Studio proporcionan útiles módulos electrónicos que pueden ser fácilmente conectados a tu placa mediante soldadura, o utilizando un 'shield' de prototipo (ver p76).

Adafruit y SparkFun tienen su sede en EE.UU., y Seeed Studio (Grove) tiene su sede en China. Puedes comprar componentes directamente de sus páginas web, o a través de distribuidores locales. Hay listas de sus distribuidores en sus páginas web: [www.adafruit.com/distributors](http://www.adafruit.com/distributors), [www.sparkfun.com/distributors](http://www.sparkfun.com/distributors), [www.seeedstudio.com/distributors.html](http://www.seeedstudio.com/distributors.html).

M5Stack y Open Smart tienen su sede en China y puedes comprar componentes en sus cuentas de Ali Express ([www.open-smart.aliexpress.com](http://www.open-smart.aliexpress.com) and [www.m5stack.aliexpress.com](http://www.m5stack.aliexpress.com)).



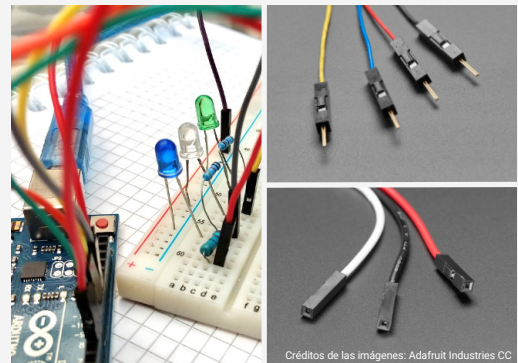
# Ampliando Tu Capacidad

## Conexión de Componentes

Hay varias formas de conectar componentes adicionales a tu placa, la mayoría de las cuales utilizan los 'headers' de la placa. Se trata de los conectores de plástico amarillos situados en los bordes izquierdo y derecho del módulo central de la placa. Los componentes electrónicos pueden conectarse a estas tomas (directamente o a través de una placa de pruebas), añadirse como parte de un 'shield' o conectarse a través de un módulo 'breakout'. A veces se utiliza una combinación de estos métodos. A continuación explicamos cada una de estas opciones.

### CABLEADO ELECTRÓNICO

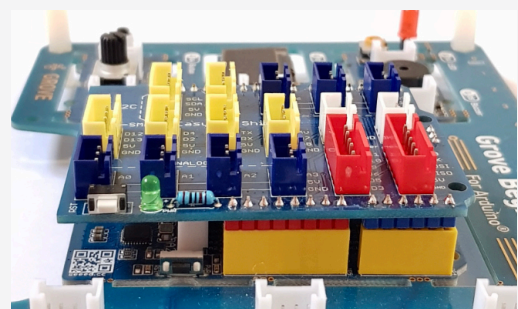
Cada uno de los headers de la placa se conecta a uno de los pines de la placa (ver p11). Los componentes pueden conectarse directamente a estos, pero lo más habitual es utilizar una placa de pruebas o un shield. Las placas de pruebas (o 'protoboards') son una forma de crear y probar fácilmente los circuitos electrónicos. Puedes obtener más información sobre ellas en el sitio web de Tu Electronica ([www.tuelectronica.es/que-es-la-protoboard](http://www.tuelectronica.es/que-es-la-protoboard)). Los cables de conexión utilizados para conectar los componentes suelen tener dos tipos de extremos: macho y hembra. Los extremos hembra son como enchufes individuales, y los extremos macho son pines metálicos, que encajan en los headers, los conectores hembras y las placas de pruebas.



Izquierda: Cableado a un Arduino utilizando una placa de pruebas  
Cables macho (arriba) y hembra (abajo)

### SHIELDS

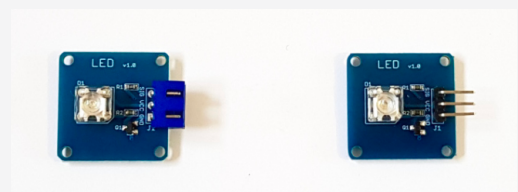
Los shields son placas modulares que se unen a tu Arduino para dotarlo de una funcionalidad extra. Utilizan una serie de pines que encajan directamente en los headers. Los shields pueden añadir una variedad de funciones, por ejemplo, permitir que la placa se comunique vía WiFi, añadir capacidad de almacenamiento adicional o acceder al GPS. Los shields de expansión y prototipado proporcionan headers o pines adicionales que permiten ampliar la capacidad de la placa y conectar fácilmente cualquier número de componentes personalizados.



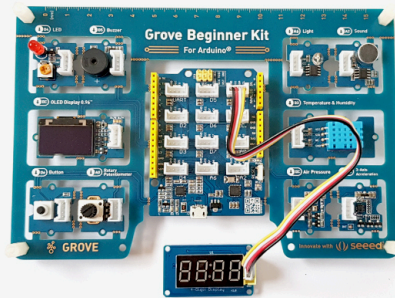
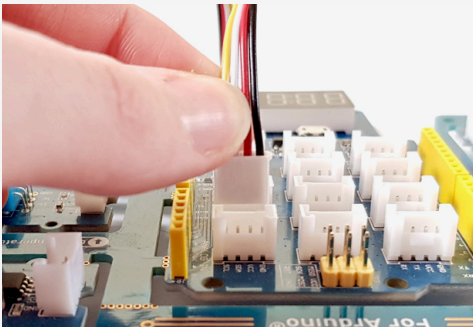
Open Smart shield conectado a la placa Grove

### MÓDULOS BREAKOUT

Un módulo breakout es similar a un shield, pero suele incluir un solo componente electrónico o un pequeño número de ellos. Se utilizan para facilitar el cableado de los componentes. Cada uno de los módulos (LED, buzzer, etc.) de tu placa Grove es esencialmente un breakout incorporada. Es muy fácil comprar componentes como parte de un módulo breakout y luego utilizar el cableado electrónico o un shield para conectarlo al Arduino.

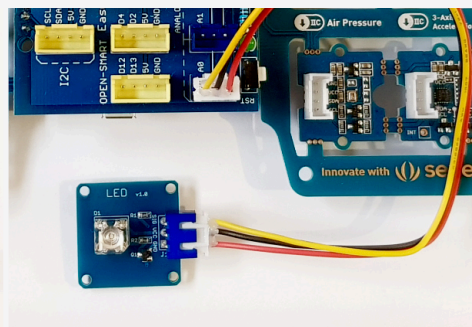
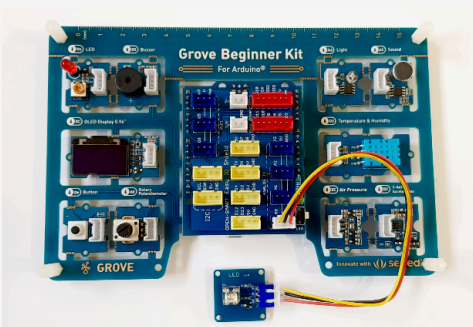


Módulo breakout LED de Open Smart easy-plug (izquierda) y estándar (derecha)



### CONEXIÓN DE LOS COMPONENTES GROVE

Los componentes Grove vienen en módulos breakout con conectores Grove blancos incluidos. Puedes utilizar los cables Grove (se incluyen seis en el Kit Grove para principiantes) para conectarlos a la placa. Sólo tienes que conectar un extremo al breakout y otro al conector blanco de la placa. Fíjate en el nombre del puerto que estás utilizando (A0, D5). Está escrito debajo del conector. Utiliza A0, A2 y A6 para dispositivos analógicos, D2-7 para dispositivos digitales e I2C para dispositivos I2C.

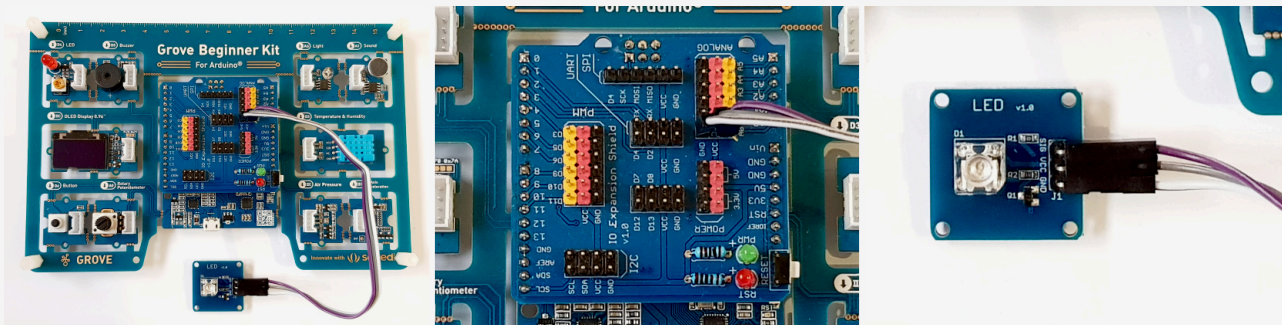


### CONEXIÓN DE COMPONENTES OPEN SMART EASY-PLUG

Los componentes Open Smart 'easy-plug' son similares a los componentes Grove, pero los conectores no son compatibles. Para ello, conecta el shield de expansión easy-plug en los headers, y luego utiliza los cables easy-plug para conectar los componentes. Ten en cuenta el nombre del puerto. Utiliza A0-A3 para dispositivos analógicos, D3, D5 y D6 para dispositivos digitales e I2C para dispositivos I2C. Más información en la página web: [www.biomaker.org/s/Biomaker-Easy-Plug-Expansion-Kit-Information-Sheet.pdf](http://www.biomaker.org/s/Biomaker-Easy-Plug-Expansion-Kit-Information-Sheet.pdf).



# Ampliando Tu Capacidad



## CONEXIÓN DE COMPONENTES OPEN SMART ESTÁNDAR

Los módulos breakout estándar de Open Smart vienen con pines macho conectados. Puedes utilizar el shield de expansión Open Smart y los cables hembra-hembra para conectar fácilmente estos componentes. Conecta el shield de expansión en los headers, luego utiliza cables hembra-hembra para conectar los pines del breakout a los pines del shield. Conecta cada pin a su correspondiente pin en el shield. Por ejemplo, conecta VCC a VCC, GND a GND y SIG a un pin relevante (A0-5 para dispositivos analógicos y D7,8,12 y 13 para dispositivos digitales). Para I2C, conecta SDA a SDA y SCL a SCL. Más información en la página web: [www.biomaker.org/s/No-Code-Programming-for-Biology-Handbook.pdf](http://www.biomaker.org/s/No-Code-Programming-for-Biology-Handbook.pdf).

## Conexión de Otros Tipos de Componentes

Los módulos breakout de otras empresas suelen venir sin conectores. Puedes añadirles conectores soldando los pines a ellas. Así podrás utilizarlas como componentes Open Smart. Para un excelente tutorial que demuestra esto, visita [www.rimstar.org/science\\_electronics\\_projects/pin\\_headers\\_soldering\\_cutting\\_male\\_female.htm](http://www.rimstar.org/science_electronics_projects/pin_headers_soldering_cutting_male_female.htm).

## Choque de Componentes

Como tu placa Grove ya tiene componentes conectados a muchos de los pines, puedes encontrarte con choques si intentas conectar un segundo dispositivo al mismo pin. Esto puede o no interrumpir tu programa, dependiendo de los dispositivos involucrados. Para evitarlo, prioriza el uso de pines sin dispositivos ya conectados, como el D2. Si los choques se convierten en un problema, puedes utilizar un Arduino diferente o podrías separar el módulo central de la placa Grove para utilizarlo por separado. Ten en cuenta que los choques no son un problema con los dispositivos I2C, ya que pueden conectarse a los mismos pines e identificarse a través de sus direcciones.

## Encontrar Nodos XOD

Una vez que hayas encontrado un componente que te gustaría utilizar, tendrás que encontrar un nodo XOD que represente ese hardware. La librería `xod/common-hardware` proporciona nodos para un número de componentes comúnmente utilizados y muchos más nodos han sido creados por la comunidad XOD.

Como estándar, los componentes electrónicos tienen asignados un 'designador de referencia'. Esta breve combinación de letras y números identifica componentes específicos; por ejemplo, la placa Grove utiliza el barómetro 'BMP280' y la pantalla OLED 'SSD1306'. Este sistema es útil porque facilita la comparación de componentes, la comprensión del hardware que utilizan otros y la búsqueda de nodos complementarios y/o código para programar el hardware.

En XOD, muchos colaboradores han creado librerías específicas para tratar ciertas piezas de hardware, y puedes buscar estas librerías en el sitio web de XOD en [www.xod.io/libs](http://www.xod.io/libs). Normalmente es más fácil buscar usando el 'designador de referencia' del hardware. Otra forma útil de encontrar librerías es buscar en el foro XOD en [www.forum.xod.io](http://www.forum.xod.io). Esto puede ayudarte a encontrar nodos y librerías relevantes, así como a identificar cualquier problema común que otros hayan tenido al usar piezas específicas de hardware.

## Uso de Arduino IDE

XOD, y otras opciones de programación 'sin código', son una forma muy útil de empezar a trabajar con microcontroladores. Requieren menos aprendizaje inicial y ofrecen una forma alternativa e intuitiva de pensar en programación. Sin embargo, también puedes estar interesado en aprender a programar, ya sea como una extensión de tus habilidades de programación con XOD, o como una alternativa. Arduino proporciona su propio software gratuito para programación: el Arduino IDE (descarga en [www.arduino.cc/en/software](http://www.arduino.cc/en/software)). Este software utiliza el lenguaje C++ para programar. Grove proporciona una guía para programar tu placa usando Arduino IDE en [www.files.seeedstudio.com/wiki/Grove-Beginner-Kit-For-Arduino/res/Grove-Beginner-Kit-For-ArduinoPDF.pdf](http://www.files.seeedstudio.com/wiki/Grove-Beginner-Kit-For-Arduino/res/Grove-Beginner-Kit-For-ArduinoPDF.pdf).

Una de las ventajas de utilizar el Arduino IDE es la gran cantidad de recursos disponibles para trabajar con casi cualquier pieza de hardware. Aunque la comunidad XOD está creciendo rápidamente y se añaden nuevas librerías todo el tiempo, hay componentes que ya no tienen nodos XOD compatibles. En este caso, casi siempre hay una librería de Arduino IDE que se puede utilizar. Otra opción es convertir las librerías existentes de Arduino IDE en librerías XOD. Matt Wayland ha escrito una excelente guía que detalla cómo convertir las librerías de Arduino para su uso en XOD, disponible en [www.biomaker.org/s/converting-arduino-to-xod\\_wayland.pdf](http://www.biomaker.org/s/converting-arduino-to-xod_wayland.pdf). Mayor información sobre cómo crear librerías de XOD en C++ está disponible en las páginas web de XOD [www.xod.io/docs/guide/nodes-for-xod-in-cpp](http://www.xod.io/docs/guide/nodes-for-xod-in-cpp) y [www.xod.io/docs/guide/analog-sensor-node](http://www.xod.io/docs/guide/analog-sensor-node).

También puedes utilizar XOD en combinación con Arduino IDE. Por ejemplo, puedes escribir un programa en XOD, y luego ir a 'Deploy > Show Code for Arduino' en la barra de menú para exportar este programa en código. A continuación, podrías añadir código adicional en Arduino. Por ejemplo, código para controlar un dispositivo no compatible en XOD.





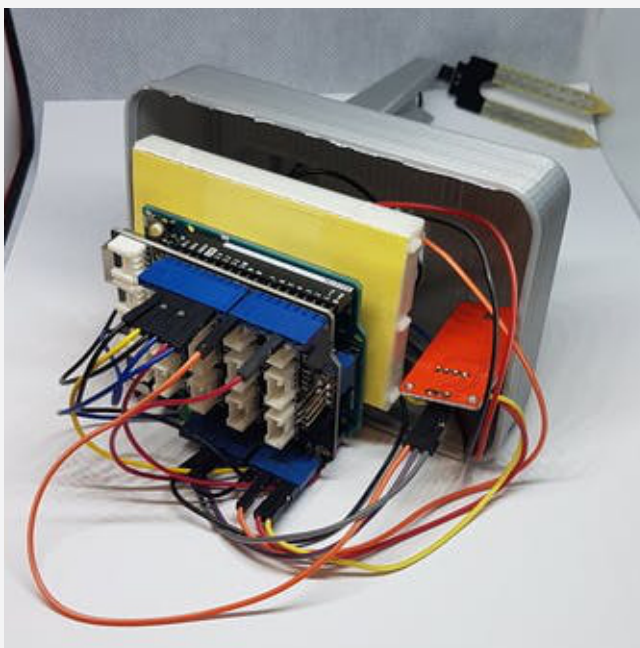
# Estudios de Caso

## eCO-SENSE: Sensores de Suelo Alimentados por la Fotosíntesis de las Plantas

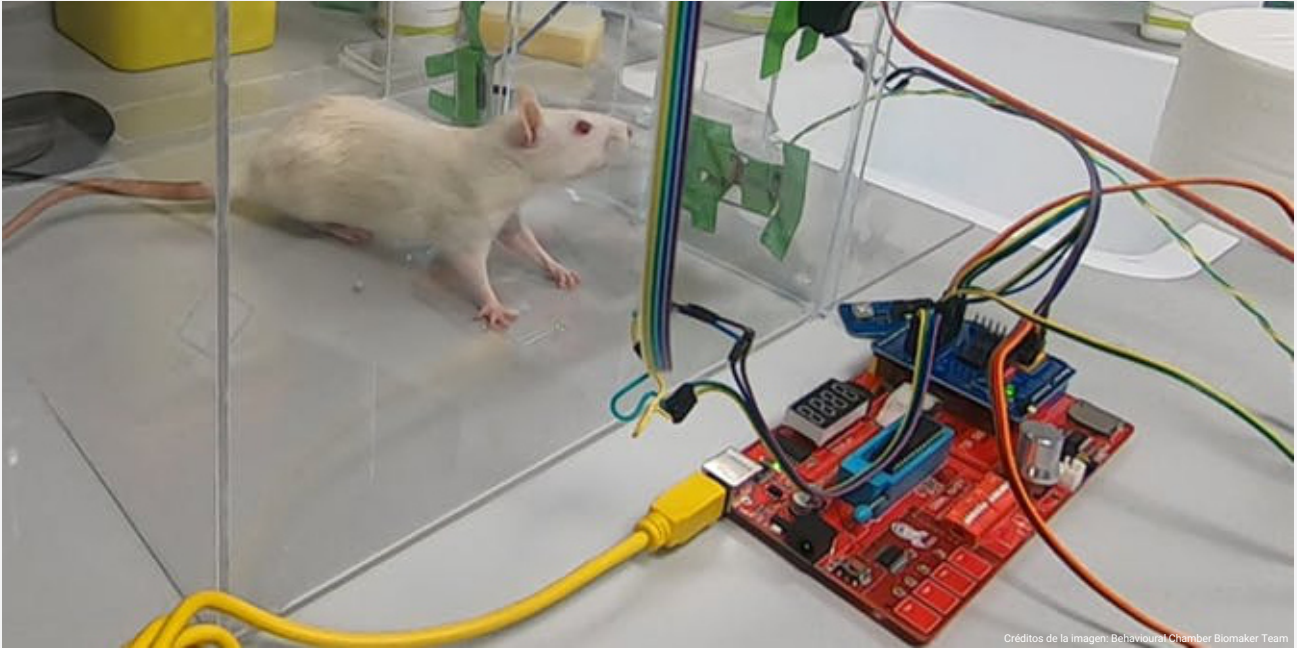
El objetivo de este proyecto es crear un prototipo de sensor de suelo de bajo costo alimentado por energía biofotovoltaica. El dispositivo utiliza un Arduino Uno, un sensor de temperatura, un sensor de humedad y un sensor de gas para medir las condiciones del suelo, y añade un módulo bluetooth para enviar los datos de forma inalámbrica a un teléfono o computadora. En colaboración con el Dr. Paolo Bombelli, de la Universidad de Cambridge, el equipo pretende utilizar celdas biofotovoltaicas para alimentar su dispositivo, lo que permitiría utilizarlo in situ y en entornos con pocos recursos.

El proyecto utilizó un sensor de temperatura y humedad DHT22 (similar al sensor DHT11 de la placa Grove), un sensor de humedad FC-28, un sensor de gas SGP30 y un módulo breakout de bluetooth nRF8001. Las alternativas compatibles con Grove incluyen los sensores Grove DHT11 o DHT22 (usa el nodo xod/dev/dht2x-higrómetro y el conector D2), el sensor de Humedad en Tierra de Grove (usa el nodo xod/common-hardware/analog-sensor y los conectores A0/A2/A6 - cuidado con los choques), el sensor Grove VOC y eCO2 (convierte las librerías SparkFun o Adafruit Arduino y usa el conector I2C, dirección 58h) y el módulo Grove Blueseed (usa el conector UART). La comunicación inalámbrica aún no está totalmente soportada en XOD, pero para un tutorial útil que explora cómo enviar los datos del sensor a tu teléfono Android a través de bluetooth utilizando el Arduino IDE, consulta [www.instructables.com/How-to-Receive-Arduino-Sensor-Data-on-Your-Android](http://www.instructables.com/How-to-Receive-Arduino-Sensor-Data-on-Your-Android).

Puedes leer más sobre el proyecto eCO-SENSE en su página de Hackster: [www.hackster.io/glen-chua/eco-sense-soil-sensors-powered-by-plant-photosynthesis-be80a2](http://www.hackster.io/glen-chua/eco-sense-soil-sensors-powered-by-plant-photosynthesis-be80a2)



Créditos de las imágenes: eCO-SENSE Biomaker Team



## Cámara de Comportamiento para Evaluar el Agarre de las Extremidades Delanteras de los Roedores

Este proyecto utiliza un emisor de luz y un sensor de luz para monitorear cuándo se mueve un roedor más allá de un determinado umbral, y provoca la liberación de una bolita de azúcar (pellets) cuando lo hace.

El proyecto utiliza un puntero láser rojo y un sensor de luz GL5528 para crear un sensor de movimiento que notifica al programa cuando un roedor ha cruzado un límite. A continuación, un controlador de motor ULN2003 se encarga de poner en marcha el dispensador de pellets construido. El recuento de las veces que un roedor ha completado esta tarea se muestra en una pantalla LCD. Las alternativas compatibles con Grove incluyen el Sensor de Luz de Grove (incluido en la placa, usa el nodo xod/common-hardware/analog-sensor y los conectores A0/A2/A6 - cuidado con los choques), el Controlador de Motor I2C de Grove (usa el nodo gweimer/h-bridge/h-bridge-2dir y el conector I2C, dirección variable) y el LCD 16 x 2 de Grove (usa el nodo xod-dev/text-lcd/text-lcd-i2c-16x2 y el conector I2C, dirección 3Eh).

Puedes leer más sobre el proyecto de la cámara de comportamiento en su página de Hackster:

[www.hackster.io/alejandrocarn/a-behavioural-chamber-to-evaluate-rodent-forelimb-grasping-bedb1a](http://www.hackster.io/alejandrocarn/a-behavioural-chamber-to-evaluate-rodent-forelimb-grasping-bedb1a)



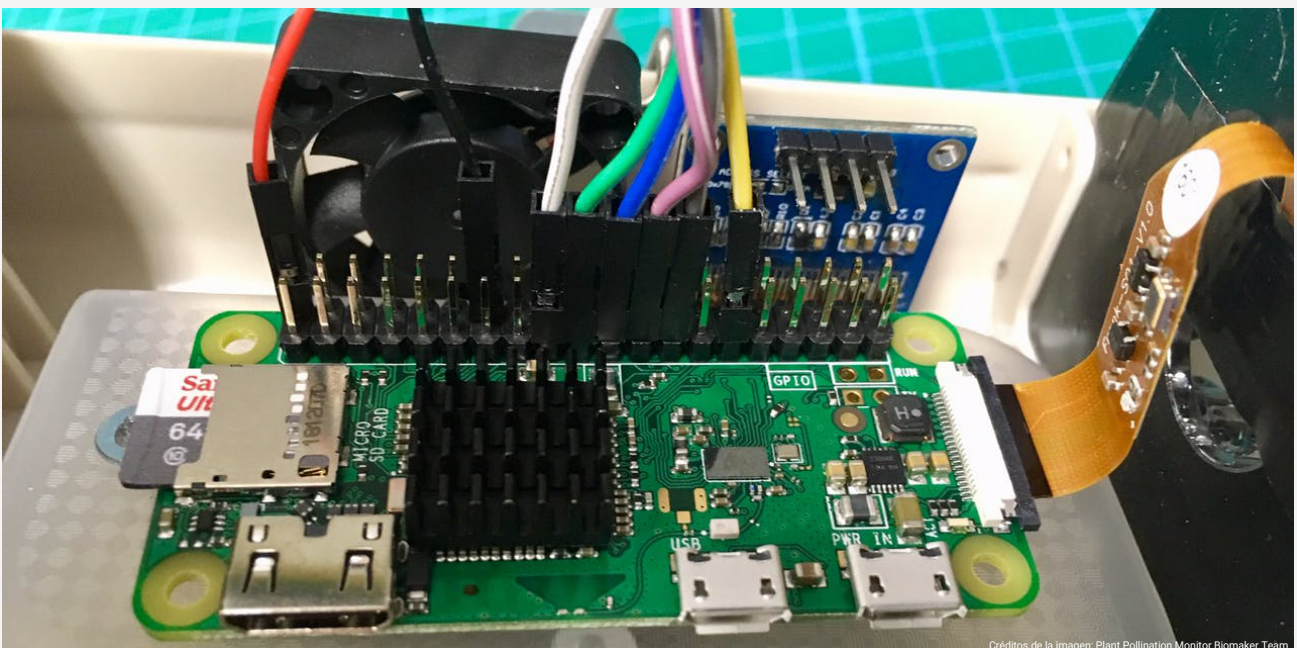
# Estudios de Caso

## Cámara para Monitorear la Polinización de las Plantas

Este proyecto desarrolló un monitor de video y 'time-lapse' para grabar a los polinizadores interactuando con las plantas. El montaje incluía un sensor ambiental para medir la temperatura, la humedad y la presión barométrica y registraba estos datos en los nombres de los archivos de imagen en una tarjeta microSD para su posterior análisis.

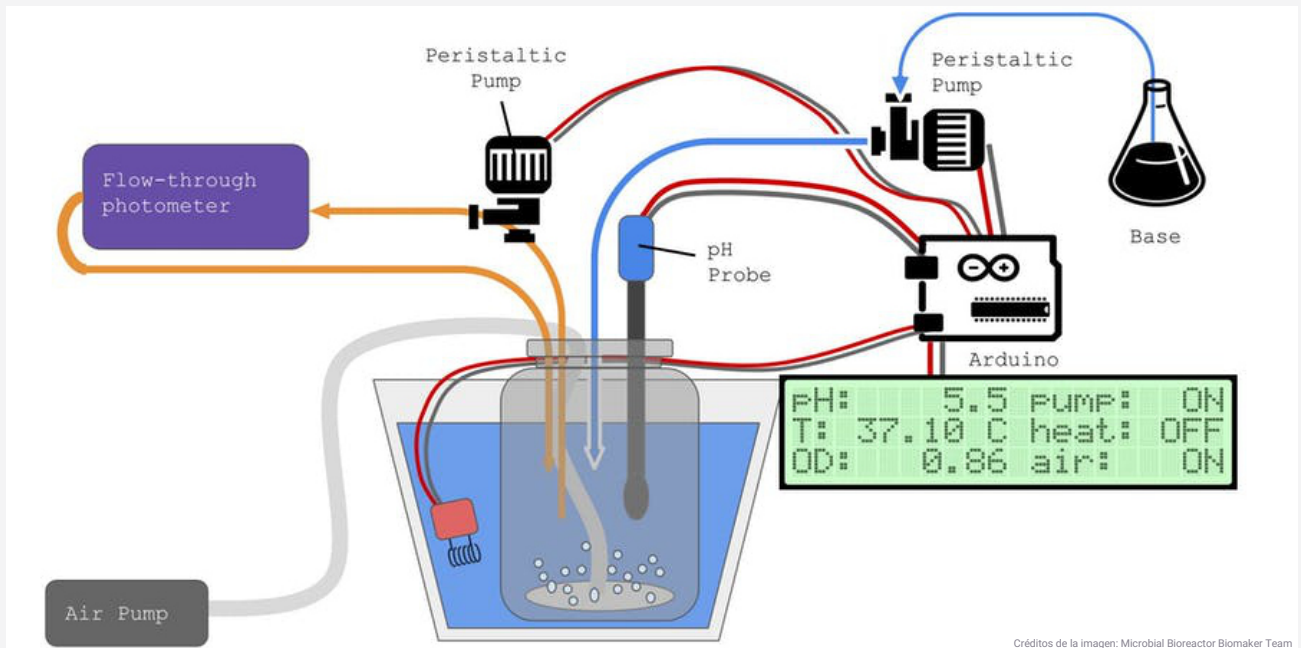
El proyecto utilizó una Raspberry Pi (ver p84), una cámara de enfoque variable de 160°, un sensor de temperatura, presión y humedad BME280 y una pantalla OLED de 128x64. Las alternativas compatibles con Grove incluyen el sensor de barómetro BME280 de Grove (usa el nodo emiliosancheza/bme280- sensor/sensor-bme280 y el conector I2C, dirección 76h), la pantalla OLED de Grove de 0,96 pulgadas (incluida en la placa, utiliza la librería wayland/ssd1306-oled-i2c y el conector I2C, dirección 3Ch) y el kit de Serial Camera de Grove con un shield de tarjeta SD de Grove. Los módulos de cámara aún no están soportados en XOD, pero la información de cómo usar la Grove Serial Camera está disponible en [www.wiki.seeedstudio.com/Grove-Serial\\_Camera\\_Kit](http://www.wiki.seeedstudio.com/Grove-Serial_Camera_Kit). Ten en cuenta que para una cámara de alta resolución con autoenfoco, como la utilizada en este proyecto, Raspberry Pi es una mejor opción que Arduino, ya que estas tareas requieren una gran potencia de procesamiento.

Puedes leer más sobre el proyecto de monitoreo de la polinización de las plantas en su página de Hackster: [www.hackster.io/team-ppi/variable-time-camera-for-monitoring-plant-pollination-events-ad21e7](http://www.hackster.io/team-ppi/variable-time-camera-for-monitoring-plant-pollination-events-ad21e7)



Créditos de la imagen: Plant Pollination Monitor Biomaker Team





## Biorreactor Microbiano de Fuente Abierta

Este proyecto pretende desarrollar un biorreactor con diseño accesible a todos (fuente abierta) para optimizar el rendimiento de las enzimas que producen proteínas recombinantes para la biología molecular. El biorreactor mide la densidad óptica del cultivo y monitorea y regula el pH, la temperatura y la aireación.

El proyecto utiliza un LED y un fotodiodo para medir la densidad óptica, una sonda de pH y una bomba peristáltica para mantener el pH y una pantalla LCD para mostrar las condiciones del reactor. El equipo también pretende añadir un sensor de temperatura y una almohadilla térmica para mantener la temperatura, y un sensor de oxígeno y un dispositivo de agitación para mantener la aireación. Las alternativas compatibles con Grove incluyen el LED de Grove rojo (incluido en la placa, usa el nodo `xod/common-hardware/led` y el conector D4), el Sensor de Luz de Grove (incluido en la placa, usa el nodo `xod/common-hardware/analog-sensor` y los conectores A0/A2/A6 - cuidado con los choques), el sensor de pH de Grove (usa el nodo `xod/common-hardware/analog-sensor` y los conectores A0/A2/A6 - cuidado con los choques), el Controlador de Motor IC2 de Grove para manejar una bomba peristáltica (usa el nodo `gweimer/h-bridge/h-bridge-2dir` y el conector I2C, dirección variable) y el Grove 16 x 2 LCD (usa el nodo `xod-dev/text-lcd/text-lcd-i2c-16x2x2` y el conector I2C, dirección 3Eh).

Puedes leer más sobre el proyecto del biorreactor microbiano en su página de Hackster:  
[www.hackster.io/open-bioeconomy-lab/microbial-bioreactor-d7f61b](http://www.hackster.io/open-bioeconomy-lab/microbial-bioreactor-d7f61b)



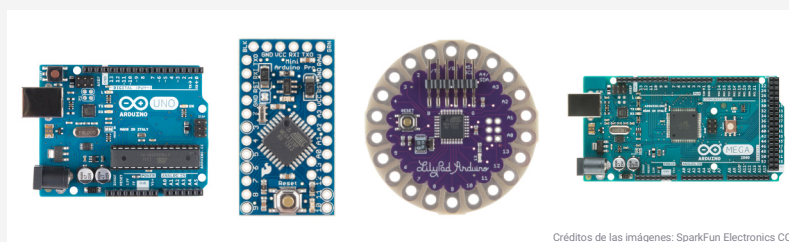
# Información Adicional

## Placas de Desarrollo Alternativas

La placa Grove es una buena forma de comenzar a construir tus propios dispositivos, ya que es sencilla, de bajo costo, fácilmente accesible, y viene con un rango de componentes integrados útiles. Sin embargo, existe una amplia variedad de otras placas disponibles para comenzar con proyectos de este estilo.

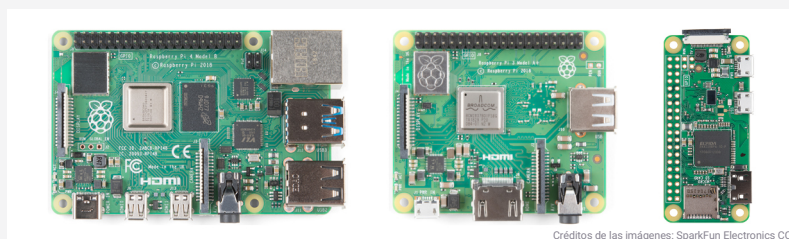
Dos de las placas de desarrollo más usadas son Arduino y Raspberry Pi, y cada una de estas compañías provee un rango de placas para distintos usos. Mientras que las placas Arduino son microcontroladores que pueden llevar a cabo un programa a la vez, las placas Raspberry Pi son computadoras completamente operativas que pueden llevar a cabo múltiples tareas a la vez. Una Raspberry Pi es mejor para proyectos más complejos, pero las placas Arduino son más fáciles de usar y aptas para la mayoría de proyectos sencillos. Ten en cuenta que XOD todavía no es compatible con la programación de placas Raspberry Pi.

Puedes encontrar una comparación entre modelos Arduino en la página web de SparkFun ([www.learn.sparkfun.com/tutorials/arduino-comparison-guide](http://www.learn.sparkfun.com/tutorials/arduino-comparison-guide)) y una comparación de modelos Raspberry Pi en la página web de PiHut ([www.thepihut.com/blogs/raspberry-pi-roundup/raspberry-pi-comparison-table](http://www.thepihut.com/blogs/raspberry-pi-roundup/raspberry-pi-comparison-table)).



Créditos de las imágenes: SparkFun Electronics CC

Placas Arduino, de izquierda a derecha: Arduino Uno, Arduino Pro Mini, Lilypad Arduino, Arduino Mega 2560



Créditos de las imágenes: SparkFun Electronics CC

Placas Raspberry Pi, de izquierda a derecho: Raspberry Pi 4 Model B, Raspberry Pi 3 A+, Raspberry Pi Zero W.

## Enlaces Útiles

|                              |  |
|------------------------------|--|
| <b>BIOMAKER</b>              | <a href="http://www.biomaker.org">www.biomaker.org</a><br>Colección de información técnica, indicaciones hacia tutoriales y recursos de software, información sobre el 'Biomaker Challenge' (Reto Biomaker).   |
| <b>NO-CODE PROGRAMMING</b>   | <a href="http://www.biomaker.org/nocode-programming-for-biology-handbook">www.biomaker.org/nocode-programming-for-biology-handbook</a><br>Información sobre el programa de Programación sin Código para Biología, descargas del manual, tutoriales y videos. |
| <b>HACKSTER</b>              | <a href="http://www.hackster.io/biomaker">www.hackster.io/biomaker</a><br>Plataforma comunitaria Biomaker utilizada para la documentación de libre acceso de proyectos y tutoriales Biomaker.  |
| <b>XOD</b>                   | <a href="http://www.xod.io">www.xod.io</a><br>Descarga del software XOD, librerías, documentos y asesoría por foros.   |
| <b>ARDUINO</b>               | <a href="http://www.arduino.cc">www.arduino.cc</a><br>Repositorio oficial de información sobre Arduino.  |
| <b>ARDUINO CREATE</b>        | <a href="http://www.create.arduino.cc">www.create.arduino.cc</a><br>Recurso integrado para compartir códigos y proyectos.  |
| <b>SEEED STUDIO</b>          | <a href="http://www.seeedstudio.com">www.seeedstudio.com</a><br>Distribuidor de hardware para el Kit de Inicio y componentes Grove.  |
| <b>OPEN SMART</b>            | <a href="http://www.open-smart.aliexpress.com/">www.open-smart.aliexpress.com/</a><br>Fuente de hardware para el 'Biomaker Expansion Kit' (kit de expansión Biomaker).   |
| <b>SPARKFUN</b>              | <a href="http://www.sparkfun.com">www.sparkfun.com</a><br>Buena fuente de información práctica sobre microcontroladores y dispositivos.  |
| <b>ADAFRUIT</b>              | <a href="http://www.adafruit.com">www.adafruit.com</a><br>Buena fuente de información práctica sobre microcontroladores y dispositivos.  |
| <b>INSTRUCTABLES</b>         | <a href="http://www.instructables.com/classes/">www.instructables.com/classes/</a><br>Clases para muchas habilidades de creación, incluyendo electrónica e impresión 3D.   |
| <b>FRITZING</b>              | <a href="http://www.fritzing.org">www.fritzing.org</a><br>Diseño e ilustración de circuitos. Fuente abierta.   |
| <b>PROCESSING</b>            | <a href="http://www.processing.org">www.processing.org</a><br>Cuadernillo de bocetos de software para gráficos dinámicos y artes visuales.   |
| <b>SYNTHETIC BIOLOGY IRC</b> | <a href="http://www.synbio.cam.ac.uk">www.synbio.cam.ac.uk</a><br>Información, noticias y eventos del Synthetic Biology Interdisciplinary Research Centre en University of Cambridge   |
| <b>OPENPLANT</b>             | <a href="http://www.openplant.org">www.openplant.org</a><br>Información, noticias y evento del BBSRC-EPSC Synthetic Biology Research Centre  |

## Contacts

[coordinator@synbio.cam.ac.uk](mailto:coordinator@synbio.cam.ac.uk)

Dr. Steph Norwood: [san43@cam.ac.uk](mailto:san43@cam.ac.uk)

Prof. Jim Haseloff: [jh295@cam.ac.uk](mailto:jh295@cam.ac.uk)



# Información Adicional

## 1

### LED

Nodo: *xod/common-hardware/led*

Ajustes: PORT = D4

LUM = luminosidad (brillo) entre 0-1

ACT = True

Utilizado en: **Tarea 1** (p20-25), **Tarea 4** (p36-41)

## 2

### BUZZER

Nodo: *marcoaita/malibrary/buzzer*

Ajustes: PORT = D5

EN = True

FREQ = 440

Utilizado en: **Tarea 2** (p26-29), **Tarea 9** (p64-71)

## 3

### PANTALLA OLED (SSD1306)

Nodos: *wayland/ssd1306-oled-i2c/ssd1306-oled-i2c-device*

*wayland/ssd1306-oled-i2c/clear-display*

*wayland/ssd1306-oled-i2c/send-buffer-to-display*

Ajustes: ADDRESS = 3Ch

WEIGHT = 128

HEIGHT = 64

RESET = -1

Notas: Añade cualquier otro nodo de *wayland/ssd1306-oled-i2c* entre *clear-display* y *send-buffer-to-display*. Conecta DEV de *ssd1306-oled-i2c-device* a todos los pins DEV. Conecta un nodo *clock* a UPD de *clear-display*, después conecta cada pin UPD uno detrás otro.

Utilizado en: **Tarea 6** (p49-55), **Tarea 7** (p56-59), **Tarea 8** (p60-63)

## 4

### BOTÓN

Nodo: *xod/common-hardware/button*

Ajustes: PORT = D6

UPD = Continuously (Loop)

Notas: Este nodo se enciende automáticamente y apaga al presionarlo. Usa un nodo *not* para invertir esto.

Utilizado en: **Tarea 2** (p26-29)

## 5

### POTENCIÓMETRO ROTATORIO

Nodo: *xod/common-hardware/pot*

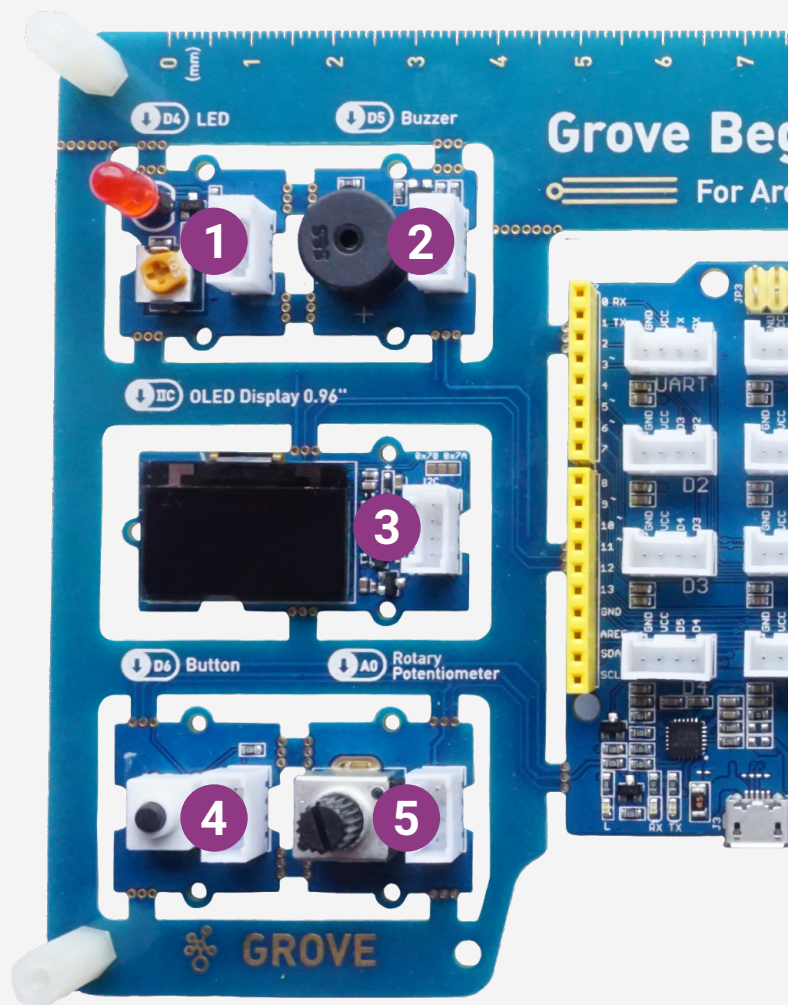
Ajustes: PORT = A0

UPD = Continuously (Loop)

Utilizado en: **Tarea 2** (p26-29)

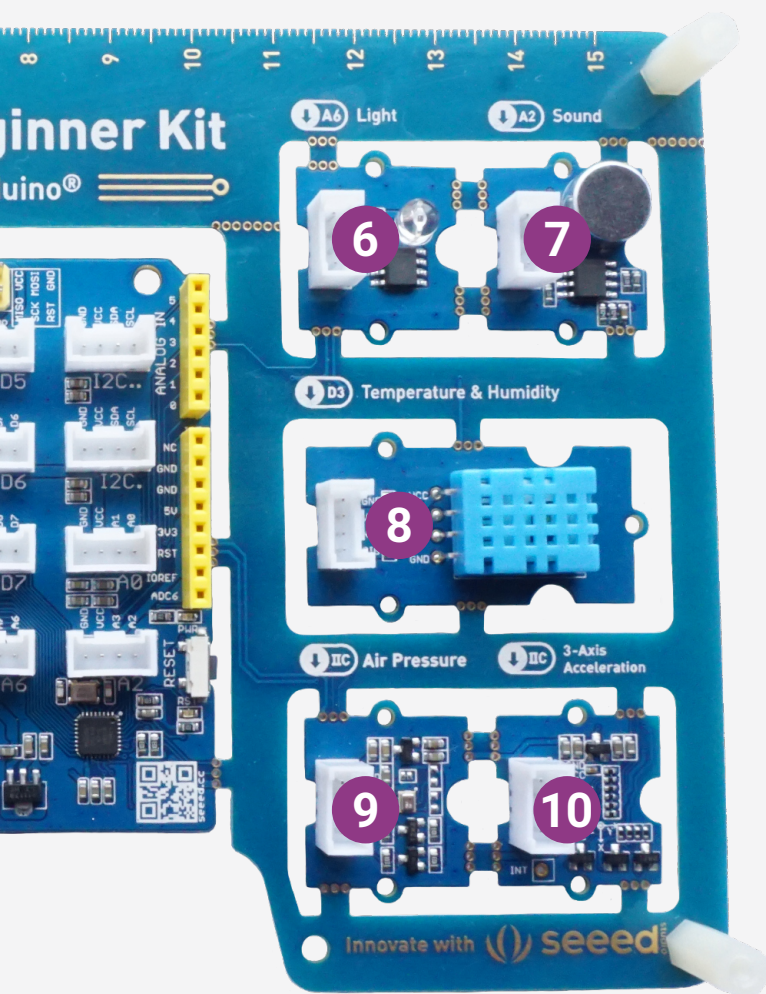
## Ficha de Ayuda para la Placa Grove

Esta ficha de ayuda proporciona una guía rápida sobre cuál nodo XOD utilizar para cada uno de los componentes integrados en el Grove All-In-One Beginner Kit for Arduino.





La guía sugiere un nodo para cada componente, así como algunos ajustes estándar. Otros nodos y ajustes pueden ser utilizados, y alentamos encarecidamente el jugar con los ajustes de los nodos y buscar nodos nuevos.



## 6

### SENSOR DE LUZ

Nodo: `wayland/analog-read-no-port-check/analog-read-no-port-check`  
 Ajustes: PORT = A6  
 UPD = Continuously (Loop)  
 Utilizado en: **Tarea 8** (p60-63)

## 7

### SENSOR DE SONIDO

Nodo: `xod/common-hardware/analog-sensor`  
 Ajustes: PORT = A2  
 UPD = Continuously (Loop)  
 Utilizado en: **Tarea 6** (p49-55)

## 8

### SENSOR DE TEMPERATURA Y HUMEDAD (DHT11)

Nodo: `xod-dev/dht/dht11-hygrometer`  
 Ajustes: PORT = D3  
 UPD = Conecta un nodo `clock`  
 Notas: Ajustar UPD a 'Loop' puede causar errores.  
 Utilizado en: **Tarea 3** (p33-35)

## 9

### SENSOR DE PRESIÓN DEL AIRE (BMP280)

Nodo: `wayland/bmp280-barometer/barometer-thermometer`  
 Ajustes: MODE = 03h  
 OST = 02h  
 OSP = 05h  
 FILT = 04h  
 STDBY = 04h  
 UPD = Continuously (Loop)  
 Utilizado en: **Tarea 5** (p42-45)

## 10

### SENSOR DE ACELERACIÓN DE 3 EJES (LIS3DH)

Nodo: `wayland/lis3dh-accelerometer/accelerometer`  
 Ajustes: ADDR = 19h  
 RATE = 07h  
 RANGE = 00h  
 Utilizado en: **Tarea 7** (p56-59)

# Información Adicional

## Lista de Nodos Utilizados

|                           |   |   |
|---------------------------|---|---|
| accelerometer             | wayland/lis3dh-accelerometer/accelerometer                      | <b>Tarea 7</b> p56-59   |
| add                       | xod/core/add  | <b>Tarea 9</b> p64-71   |
| analog-read-no-port-check | wayland/analog-read-no-port-check/<br>analog-read-no-port-check | <b>Tarea 8</b> p60-63   |
| analog-sensor             | xod/common-hardware/analog-sensor                               | <b>Tarea 6</b> p49-55<br><b>Estudios de Caso</b> p80-83   |
| and                       | xod/core/and  | <b>Tarea 8</b> p60-63   |
| any                       | xod/core/any  | <b>Tarea 8</b> p60-63<br><b>Tarea 9</b> p64-71  |
| bar124                    | custom node created in...                                       | <b>Tarea 9</b> p64-71   |
| bar3                      | custom node created in ..                                       | <b>Tarea 9</b> p64-71   |
| barometer-thermometer     | wayland/bmp280-barometer/barometer-thermometer                  | <b>Tarea 5</b> p42-45   |
| between                   | e/comparison/between  | <b>Tarea 9</b> p64-71   |
| button                    | xod/common-hardware/button                                      | <b>Tarea 1</b> p20-25<br><b>Tarea 2</b> p26-29<br><b>Tarea 9</b> p64-71   |
| buzzer                    | marcoaita/malibrary/buzzer                                      | <b>Tarea 2</b> p26-29   |
| buzzer-timed              | marcoaita/malibrary/buzzer-timed                                | <b>Tarea 9</b> p64-71   |
| clear-display             | wayland/ssd1306-oled-i2c/clear-display                          | <b>Tarea 6</b> p49-55<br><b>Tarea 7</b> p56-59  |
| click-detector            | wayland/lis3dh-accelerometer/click-detector                     | <b>Tarea 7</b> p56-59   |
| clock                     | xod/core/clock  | <b>Tarea 1</b> p20-25<br><b>Tarea 4</b> p36-41<br><b>Tarea 6</b> p49-55<br><b>Tarea 7</b> p56-59<br><b>Tarea 9</b> p64-71 |
| concat                    | xod/core/concat   | <b>Tarea 5</b> p42-45<br><b>Tarea 8</b> p60-63  |
| count                     | xod/core/count  | <b>Tarea 4</b> p36-41<br><b>Tarea 6</b> p49-55<br><b>Tarea 9</b> p64-71   |
| dec-to-2digits            | cesars/utills/dec-to-2digits                                    | <b>Tarea 5</b> p42-45   |
| dec-to-4digits            | cesars/utills/dec-to-4digits                                    | <b>Tarea 5</b> p42-45   |
| defer                     | xod/core/defer  | <b>Tarea 9</b> p64-71   |
| delay                     | xod/core/delay  | <b>Tarea 9</b> p64-71   |
| dht11-hygrometer          | xod-dev/dht/dht11-hygrometer                                    | <b>Tarea 3</b> p33-35   |
| dht2x-hygrometer          | xod/dev/dht2x-hygrometer  | <b>Estudios de Caso</b> p80-83  |
| draw-circle               | wayland/ssd1306-oled-i2c/draw-circle                            | <b>Tarea 6</b> p49-55<br><b>Tarea 7</b> p56-59  |
| draw-line                 | wayland/ssd1306-oled-i2c/draw-line                              | <b>Tarea 6</b> p49-55   |
| draw-pixel                | wayland/ssd1306-oled-i2c/draw-pixel                             | <b>Tarea 6</b> p49-55   |
| draw-rectangle            | wayland/ssd1306-oled-i2c/draw-rectangle                         | <b>Tarea 6</b> p49-55   |
| draw-rounded-rectangle    | wayland/ssd1306-oled-i2c/draw-rounded-rectangle                 | <b>Tarea 6</b> p49-55   |
| draw-text                 | wayland/ssd1306-oled-i2c/draw-text                              | <b>Tarea 6</b> p49-55   |
| draw-triangle             | wayland/ssd1306-oled-i2c/draw-triangle                          | <b>Tarea 6</b> p49-55   |
| equal                     | xod/core/equal  | <b>Tarea 9</b> p64-71   |
| flip-flop                 | xod/core/flip-flop  | <b>Tarea 4</b> p36-41   |
| flip-n-times              | xod/core/flip-n-times   | <b>Tarea 4</b> p36-41   |
| format-number             | xod/core/format-number  | <b>Tarea 5</b> p42-45   |
| from-bus                  | xod/patch-nodes/from-bus  | <b>Tarea 7</b> p56-59<br><b>Tarea 8</b> p60-63<br><b>Tarea 9</b> p64-71   |
| greater                   | xod/core/greater  | <b>Tarea 8</b> p60-63   |
| h-bridge-2dir             | gweimer/h-bridge/h-bridge-2dir                                  | <b>Estudios de Caso</b> p80-83  |
| if-else                   | xod/core/if-else  | <b>Tarea 8</b> p60-63<br><b>Tarea 9</b> p64-71  |

|                           |  |                |                         |
|---------------------------|--|----------------|-------------------------|
| input-boolean             | xod/patch-nodes/input-boolean                    | Tarea 6 p49-55 | Tarea 9 p64-71          |
| input-byte                | xod/patch-nodes/input-byte                       |                | Tarea 6 p49-55          |
| input-number              | xod/patch-nodes/input-number                     | Tarea 6 p49-55 | Tarea 7 p56-59          |
|                           |  |                | Tarea 8 p60-63          |
| input-port                | xod/patch-nodes/input-port                       | Tarea 6 p49-55 | Tarea 8 p60-63          |
| input-pulse               | xod/patch-nodes/input-pulse                      | Tarea 6 p49-55 | Tarea 8 p60-63          |
|                           |  |                | Tarea 9 p64-71          |
| input-string              | xod/patch-nodes/input-string                     |                | Tarea 6 p49-55          |
| input-t1                  | xod/patch-nodes/input-t1                         |                | Tarea 6 p49-55          |
| invert-display            | wayland/ssd1306-oled-i2c/invert-display          |                | Tarea 6 p49-55          |
| join                      | xod/core/join                                    |                | Tarea 5 p42-45          |
| led                       | xod/common-hardware/led                          | Tarea 1 p20-25 | Tarea 4 p36-41          |
|                           |  |                | Estudios de Caso p80-83 |
| less                      | xod/core/less                                    |                | Tarea 8 p60-63          |
| light-sensor              | custom node created in...                        |                | Tarea 8 p60-63          |
| map                       | xod/math/map                                     | Tarea 2 p26-29 | Tarea 7 p56-59          |
| multiply                  | xod/core/multiply                                |                | Tarea 8 p60-63          |
| nand                      | xod/core/nand                                    |                | Tarea 8 p60-63          |
| nor                       | xod/core/nor                                     |                | Tarea 8 p60-63          |
| not                       | xod/core/not                                     | Tarea 1 p20-25 | Tarea 2 p26-29          |
|                           |  |                | Tarea 9 p64-71          |
|                           |  |                | Tarea 5 p42-45          |
| number-split-to-digit     | gst/number-split-to-digit/number-split-to-digit  |                | Tarea 9 p64-71          |
| or                        | xod/core/or                                      | Tarea 8 p60-63 | Tarea 9 p64-71          |
| output-boolean            | xod/patch-nodes/output-boolean                   |                | Tarea 6 p49-55          |
| output-byte               | xod/patch-nodes/output-byte                      |                | Tarea 6 p49-55          |
| output-number             | xod/patch-nodes/output-number                    | Tarea 6 p49-55 | Tarea 9 p64-71          |
| output-port               | xod/patch-nodes/output-port                      |                | Tarea 6 p49-55          |
| output-pulse              | xod/patch-nodes/output-pulse                     | Tarea 6 p49-55 | Tarea 9 p64-71          |
| output-string             | xod/patch-nodes/output-string                    | Tarea 6 p49-55 | Tarea 8 p60-63          |
| output-t1                 | xod/patch-nodes/output-t1                        |                | Tarea 6 p49-55          |
| pot                       | xod/common-hardware/pot                          |                | Tarea 2 p26-29          |
| pulse-on-true             | xod/core/pulse-on-true                           | Tarea 8 p60-63 | Tarea 9 p64-71          |
| rotate-display            | wayland/ssd1306-oled-i2c/rotate-display          | Tarea 6 p49-55 | Tarea 7 p56-59          |
| round                     | xod/math/round                                   |                | Tarea 7 p56-59          |
| send-buffer-to-display    | wayland/ssd1306-oled-i2c/ send-buffer-to-display | Tarea 6 p49-55 | Tarea 7 p56-59          |
| sensor-bme280             | emiliosancheza/bme280-sensor/sensor-bme280       |                | Estudios de Caso p80-83 |
| square-wave               | xod/waves/square-wave                            |                | Tarea 4 p36-41          |
| ssd1306-oled-i2c-device   | wayland/ssd1306-oled-i2c/ssd1306-oled-i2c-device | Tarea 6 p49-55 | Tarea 7 p56-59          |
| text-lcd-i2c-16x2         | xod-dev/text-lcd/text-lcd-i2c-16x2               |                | Estudios de Caso p80-83 |
| to-bus                    | xod/patch-nodes/to-bus                           | Tarea 7 p56-59 | Tarea 8 p60-63          |
|                           |  |                | Tarea 9 p64-71          |
| tweak-boolean             | xod/debug/tweak-boolean                          |                | Tarea 3 p33-35          |
| tweak-byte                | xod/debug/tweak-byte                             |                | Tarea 3 p33-35          |
| tweak-color               | xod/debug/tweak-color                            |                | Tarea 3 p33-35          |
| tweak-number              | xod/debug/tweak-number                           | Tarea 3 p33-35 | Tarea 7 p56-59          |
|                           |  |                | Tarea 8 p60-63          |
| tweak-pulse               | xod/debug/tweak-pulse                            | Tarea 3 p33-35 | Tarea 4 p36-41          |
| tweak-string-16/32/64/128 | xod/debug/tweak-string-16/32/64/128              |                | Tarea 3 p33-35          |
| watch                     | xod/debug/watch                                  | Tarea 3 p33-35 | Tarea 4 p36-41          |
|                           |  | Tarea 5 p42-45 | Tarea 6 p49-55          |
|                           |  |                | Tarea 8 p60-63          |
| write-dot-to-oled         | custom node created in...                        |                | Tarea 7 p56-59          |
| write-text-to-oled        | custom node created in...                        | Tarea 6 p49-55 | Tarea 8 p60-63          |
| xor                       | xod/core/xor                                     |                | Tarea 8 p60-63          |

Nota: para los elementos en gris se da un ejemplo de su uso, pero no se presenta una demostración completa en esta guía.



# Glosario

|                                       |   |
|---------------------------------------|---|
| <b>ACELERÓMETRO</b>                   | Los acelerómetros miden la aceleración de un objeto, es decir, cualquier cambio de velocidad (rapidez y dirección). Un acelerómetro de 3 ejes, como el incluido en la placa Grove, puede detectar cuando la placa es movida mueve o inclinada en cualquier dirección (ejes X, Y y Z).   |
| <b>ACTUADOR</b>                       | Los actuadores son dispositivos de salida que convierten las señales electrónicas en movimiento mecánico. Por ejemplo, motores o bombas.  |
| <b>ADAFRUIT</b>                       | Adafruit Industries es una empresa de hardware de fuente abierta que ofrece componentes electrónicos, herramientas, accesorios y recursos de aprendizaje. Sus componentes son compatibles con el hardware de Arduino y Raspberry Pi.  |
| <b>ANALÓGICO</b>                      | Las señales analógicas, a diferencia de las señales digitales, son continuas y pueden adoptar un número infinito de valores. Los dispositivos analógicos miden variables continuas, como la intensidad de sonido o luz. Muchos sensores ambientales son analógicos. Las computadoras usan señales digitales, por lo que las señales analógicas deben ser convertidas primero en señales digitales por el microcontrolador.  |
| <b>ARDUINO</b>                        | Arduino es una empresa de electrónica de fuente abierta. Elaboran software de programación libremente disponible y hardware de bajo costo para que cualquiera pueda hacer sus propios proyectos electrónicos.   |
| <b>ARDUINO IDE</b>                    | El 'Integrated Development Environment' (Entorno de Desarrollo Integrado, IDE) de Arduino es el software gratuito de Arduino para programar placas Arduino utilizando el lenguaje de programación C++. Es una alternativa al XOD IDE, y puede utilizarse junto con XOD (ver p79).   |
| <b>ARDUINO UNO</b>                    | El Arduino UNO fue la primera placa Arduino basada en USB, compuesta por un chip microcontrolador, una placa de circuito impreso (PCB) y una serie de pines de entrada-salida digitales y analógicos para conectar shields y hardware externo. El Arduino UNO R3 es la tercera revisión de esta placa, y es en la que se basan Seeduino y la placa Grove.   |
| <b>ATMEGA328P</b>                     | El Atmega328P es el chip microcontrolador utilizado en las últimas versiones de la placa Arduino, incluida la placa Grove.  |
| <b>BARÓMETRO</b>                      | Un barómetro es un dispositivo que mide la presión del aire. Puede ser utilizado para monitorear o pronosticar el clima, o para medir la altitud.   |
| <b>BIOMAKER</b>                       | Biomaker es una iniciativa dirigida por el Synthetic Biology IRC (Centro de Investigación Interdisciplinaria en Biología Sintética) de la Universidad de Cambridge y OpenPlant que se centra en la formación y la provisión de fondos y recursos para los investigadores interesados en la intersección de la biología, la ingeniería y la informática. Entre las actividades de Biomaker se encuentran el Biomaker Challenge anual y la formación en Programación Sin Código para la Biología. |
| <b>BUS (XOD)</b>                      | En informática un bus es un sistema de comunicación que transfiere información entre computadoras, o entre diferentes partes de una misma computadora. En XOD, podemos utilizar los buses para transferir información entre una parte de nuestro patch y otra, sin tener que conectarlos mediante enlaces. Esto se hace usando los nodos <i>to-bus</i> y <i>from-bus</i> para enviar información a, y recibir información de un bus.  |
| <b>BUZZER/ZUMBADOR PIEZOELÉCTRICO</b> | Los buzzers o zumbadores piezoeléctricos son dispositivos sencillos que pueden generar pitidos y tonos básicos. Funcionan con un cristal piezoeléctrico, un material especial que cambia de forma cuando se le aplica voltaje. Si el cristal empuja contra un diafragma, puede generar una onda de presión que el oído humano capta como sonido.  |



|                           |  |
|---------------------------|--|
| <b>CABLES DE CONEXIÓN</b> | Los cables de conexión o cables puente se utilizan para conectar componentes a la placa Arduino y existen varios tipos. Los cables hembra-hembra tienen conectores en cada extremo que se enchufan a los pines metálicos de los componentes, de la placa Arduino y de los shields. Los cables macho-macho tienen pines metálicos en cada extremo que se conectan a los enchufes hembra, a los headers o a las placas de pruebas. Cables macho-hembra tienen un conector hembra en un extremo y un conector macho en el otro. Los cables de conexión también pueden venir preconfeccionados con enchufes que encajen en conectores compatibles. Por ejemplo, Grove o Open Smart (JST-XH). |
| <b>DEBUGGER (XOD)</b>     | El Debugger (depurador) es el simulador de XOD. Puede usarse para simular tu patch, o para editar y 'debug' (depurar) tu patch después de cargarlo a la placa. Puedes iniciar el Debugger usando el botón 'Simulate' (Simular), o usando el botón 'Upload and Debug' (Cargar y Depurar) para cargar el patch al mismo tiempo. En el Debugger, puedes utilizar los nodos <i>tweak</i> y <i>watch</i> para editar y monitorear tu patch en tiempo real (ver <b>Tarea 3</b> p33-35).  |
| <b>DIGITAL</b>            | Las señales digitales, a diferencia de las analógicas, sólo pueden tomar valores finitos y discretos. Por ejemplo, un LED puede estar 'encendido' o 'apagado'. Las señales digitales pueden comportarse de forma similar a las señales analógicas, por ejemplo, se puede cambiar el brillo de un LED, pero en última instancia hay un número finito y discreto de valores que puede tomar el brillo de un LED. Las computadoras utilizan señales digitales, y la mayoría de los componentes electrónicos son digitales. Por ejemplo, las pantallas, los botones y algunos tipos de sensores.   |
| <b>DRIVER USB</b>         | Un driver o controlador USB es una pieza de software que permite al sistema operativo de tu computadora comunicarse con el hardware externo, como un disco duro o una placa de desarrollo de microcontroladores como la placa Grove. La placa Grove utiliza el driver CP210 de Silicon Labs, y es posible que tengas que descargarlo para poder utilizar tu placa.   |
| <b>FOTORRESISTENCIA</b>   | Una fotorresistencia o fotorresistor (inglés: light dependent resistor, LDR) es un componente sensible a la luz. Cuando la luz incide sobre él, la resistencia cambia, y este cambio se utiliza como señal electrónica.  |
| <b>GROVE</b>              | Grove es un rango de componentes electrónicos que son fáciles de usar y compatibles con Arduino. Utiliza un sistema 'plug-and-play' de módulos que encajan fácilmente para construir dispositivos. Está desarrollado por la empresa Seeed Studio.  |
| <b>HACKSTER</b>           | Hackster es una plataforma online para registrar y compartir proyectos de electrónica. Ofrece una forma sencilla de documentar y explorar proyectos, y cuenta con una gran comunidad de colaboradores, entre los que se encuentran empresas como Arduino y Seeed Studio.   |
| <b>HEADERS</b>            | Los headers son conectores cableados a la placa PCB y proporcionan enchufes 'hembras'. Nos dan una forma de conectar fácilmente componentes externos a la placa, ya sea a través de cables de conexión, o a través de un shield.   |
| <b>HIGRÓMETRO</b>         | Los higrómetros (inglés: hygrometers) son dispositivos utilizados para medir la humedad, es decir, la cantidad de vapor de agua presente en el aire o en el suelo.   |
| <b>I2C</b>                | Los circuitos inter-integrados (I2C) son un protocolo de comunicación digital utilizado para comunicarse con varios dispositivos a la vez. Con la comunicación I2C se pueden conectar varios dispositivos al mismo pin del microcontrolador, y a cada dispositivo se le da un 'nombre' digital (conocido como dirección). Las direcciones se escriben como XXh, siendo XX un código de dos dígitos de números y letras. Por ejemplo, 19h o 3Ch.  |
| <b>INSPECTOR (XOD)</b>    | En XOD, el panel de 'Inspector' es el lugar donde se puede editar un nodo. Por ejemplo, puedes cambiar los parámetros de los pines de un nodo, cambiar el nombre de un nodo o añadir una descripción. Debes hacer clic en un nodo del patch para que aparezcan estas opciones. El panel de Inspector aparece en la parte izquierda de la pantalla, debajo del panel de 'Project Browser', y puede activarse y desactivarse mediante el botón de la barra deslizante situado en la parte superior izquierda, o bien accediendo a 'View > Toggle Inspector' (Ver > Alternar Inspector) en la barra de menús.   |
| <b>LED</b>                | Un diodo emisor de luz (inglés: light emitting diode, LED) es una fuente de luz brillante de bajo consumo que genera luz haciendo pasar una corriente a través de un diodo semiconductor.  |
| <b>LIBRARY (XOD)</b>      | En XOD (y en otros softwares de codificación como Arduino), las librerías son colecciones de nodos (o código) listos para usar. A menudo están diseñadas para ayudarte a utilizar una pieza específica de hardware (por ejemplo, <i>wayland/bmp280-barometer</i> ) o como una colección de nodos con funciones similares (por ejemplo, <i>xod/math</i> ). El XOD IDE tiene varias librerías preinstaladas, pero puedes añadir más librerías usando el botón 'Add Library' (Añadir Librería) (libros con un símbolo +, en Project Browser) o navegando a 'File > Add Library...' (Archivo > Añadir Librería...) en la barra de menú.  |
| <b>M5STACK</b>            | M5Stack es una empresa de hardware que proporciona su propio sistema de desarrollo con Wi-Fi y Bluetooth, así como una serie de componentes compatibles con Grove llamados 'units' (unidades).   |



# Glosario

|                            |   |
|----------------------------|---|
| <b>MICROCONTROLADOR</b>    | Un microcontrolador es una pequeña computadora de bajo consumo integrada en un dispositivo. A diferencia de una computadora de uso general, como una portátil o un PC, los microcontroladores suelen estar diseñados para realizar una tarea y ejecutar un programa específico. La placa Grove Arduino contiene un microcontrolador reprogramable para que puedas cargar tu programa en la placa y crear tus propios dispositivos.  |
| <b>MÓDULOS BREAKOUT</b>    | También conocido como 'módulos', 'breakouts', 'placas breakout' o 'breakout boards'. Los módulos breakout se utilizan para facilitar el cableado de los componentes electrónicos. Suelen consistir en una pequeña placa PCB con un solo componente electrónico, o con un número reducido de ellos. Por ejemplo, un módulo breakout de pantalla LED u OLED. Se pueden conectar fácilmente a una placa de desarrollo mediante cables, pins y headers, o enchufes.   |
| <b>NODO (XOD)</b>          | En XOD los nodos se utilizan como 'una representación visual de un dispositivo o función física'. Pueden representar un componente electrónico, una función matemática o cualquier otra función que una computadora pueda realizar. Aparecen en un patch como un cuadro gris oscuro delineado en blanco, con el nombre del nodo impreso en el centro. Pueden tener pequeños círculos de colores (pines) en la parte superior e inferior que representan entradas y salidas.   |
| <b>OPEN SMART</b>          | OpenSmart es un grupo de empresas tecnológicas Chinas interesadas en la producción y el desarrollo de hardware abierto.   |
| <b>PANTALLA OLED</b>       | Las pantallas de diodos orgánicos emisores de luz (OLED) son una alternativa a las pantallas LCD que se utilizan principalmente para los televisores. En lugar de tener una luz de fondo para iluminar los píxeles, cada píxel puede producir su propia luz. Esto puede mejorar el contraste.   |
| <b>PATCH (XOD)</b>         | En XOD un patch es el área de trabajo en la que se construye un programa. Es similar a un documento en otros sistemas, pero en lugar de código de texto el patch se construye con nodos.  |
| <b>PCB</b>                 | Una placa de circuito impreso (PCB) está compuesta por una fina placa de plástico con pistas conductoras de metal grabadas en la superficie o entre las capas. Se utilizan para conectar componentes eléctricos, que suelen estar soldados a la placa.  |
| <b>PIN (ARDUINO)</b>       | En electrónica, 'pin' se utiliza para referirse a los contactos eléctricos de un componente, es decir, las partes de un componente que se utilizan para conectarse a otros componentes. En la placa Grove, el chip microcontrolador tiene una serie de pines que se conectan tanto a los componentes de la placa, como a los enchufes Grove y a los headers del módulo central, lo que les permite comunicarse con componentes adicionales. En el XOD los pines de Arduino se denominan 'Ports' (Puertos) para evitar la confusión con los pines del XOD. |
| <b>PIN (XOD)</b>           | En XOD se utiliza 'pin' para referirse a las entradas y salidas asociadas a un nodo específico. Aparecen como pequeños círculos redondos en la parte superior (pines de entrada) e inferior (pines de salida) de un nodo. Los pines están coloreados según su tipo de datos.  |
| <b>PLACA DE DESARROLLO</b> | Una placa de desarrollo de microcontroladores, como la placa Grove Arduino, aloja un chip microcontrolador en una pequeña placa PCB junto con algunas piezas y conexiones adicionales que facilitan que cualquiera pueda programar y conectar componentes a un microcontrolador en casa. Las placas de desarrollo están pensadas para ser baratas y de fácil acceso, y suelen utilizarse para desarrollar prototipos e instrumentos personalizados.   |

|                                  |   |
|----------------------------------|---|
| <b>PLACA DE PRUEBAS</b>          | También conocido como 'breadboard', 'proboards' or 'placa de prototipos'. Las placas de pruebas son dispositivos sencillos para desarrollar y crear prototipos de circuitos eléctricos, sin necesidad de soldar. Constan de filas de enchufes (agujeros) que se conectan mediante cableado eléctrico. Los componentes pueden conectarse directamente a estas tomas mediante cables o pines metálicos. Todos los dispositivos conectados a la misma fila de agujeros estarán conectados entre sí.  |
| <b>PLATAFORMA DE DESARROLLO</b>  | Una plataforma de desarrollo (inglés: development host) es el dispositivo que utilizarás para escribir y desarrollar el programa que quieres instalar en la placa de desarrollo. Para programar la placa Grove utilizarás una computadora portátil o un PC como plataforma de desarrollo.   |
| <b>POTENCIÓMETRO</b>             | Los potenciómetros (a menudo abreviados como 'pot') son resistencias variables que permiten modificar la resistencia, y por tanto la corriente que circula por un circuito, sin necesidad de reprogramar el dispositivo. A menudo se encuentran en forma de perilla, deslizador o tornillo.   |
| <b>PROGRAMACIÓN SIN CÓDIGO</b>   | La programación sin código es un mecanismo cada vez más popular que permite a las personas programar hardware y software mediante una interfaz gráfica de usuario, en lugar de exigirles que aprendan y escriban código a través de texto. La iniciativa No-Code Programming for Biology (Programación Sin Código para Biología) de Biomaker ha adoptado la programación sin código y de bajo código como medio para formar a los biólogos, y a otras personas con poca o nula experiencia en programación/codificación, para que construyan sus propios dispositivos personalizados.   |
| <b>PROJECT BROWSER</b>           | El Project Browser (Navegador de Proyectos) de XOD es donde encontrarás el proyecto actual en el que estás trabajando y las librerías que tienes instaladas. Bajo el desplegable [nombre de tu proyecto] encontrarás todos los parches de tu proyecto. Debajo hay una lista de librerías. Al hacer clic en el botón desplegable de una librería, podrás explorar los nodos de esa biblioteca. Arrastrando un nodo desde Project Browser al patch añadirá ese nodo a tu patch. En la parte superior están los botones 'New Patch' (Nuevo Patch) y 'Add Library' (Añadir Librería). El panel de Project Browser aparece en la parte superior izquierda de la pantalla, y puede activarse y desactivarse utilizando el botón en la parte superior izquierda, o navegando a 'View > Toggle Project Browser' (Ver > Alternar Navegador de Proyectos) en el menú. |
| <b>PROTOCOLO DE COMUNICACIÓN</b> | Un protocolo de comunicación es el método por el que dos o más componentes electrónicos intercambian datos. Ethernet, wi-fi y bluetooth son ejemplos de protocolos de comunicación. Diferentes componentes utilizan diferentes protocolos de comunicación (por ejemplo, analógico, digital, I2C) y por lo tanto tendrán que ser conectados a la placa Arduino de diferentes maneras.  |
| <b>QUICK HELP (XOD)</b>          | El panel de Quick Help (Ayuda Rápida) en XOD es donde puedes encontrar información sobre un nodo y sus pines. Cuando haces clic en un nodo, la información sobre ese nodo y sus pines aparecerá aquí. El panel de Quick Help aparece en la parte superior derecha de la pantalla, y puede activarse y desactivarse usando el botón de signo de interrogación en la parte superior derecha, o navegando a 'View > Toggle Quick Help' (Ver > Alternar Ayuda Rápida) en ella barra de menú.  |
| <b>SEED STUDIO</b>               | Seed Studio es una empresa de hardware de código abierto. Han desarrollado la placa de desarrollo Seeeduno Lotus (basada en la placa de desarrollo Arduino Uno R3) y el sistema de componentes Grove que utilizan enchufes para conectar fácilmente los módulos.  |
| <b>SHIELD</b>                    | Los shields son placas de circuito modulares que se unen a tu Arduino para dotarlo de una funcionalidad extra. Los shields pueden tener funciones específicas, como un shield wifi que permitirá a tu placa transferir información a través de wifi, o puede tener funciones más generales, como un shield de expansión o de creación de prototipos. Estos te permiten conectar fácilmente cualquier número de componentes personalizados. Algunos shields pueden apilarse unos sobre otros para crear combinaciones de módulos y funciones.  |
| <b>SPARKFUN</b>                  | SparkFun es una empresa de hardware de código abierto que ofrece una gama de placas y componentes de desarrollo, así como tutoriales y recursos de aprendizaje sobre programación, electrónica y trabajo con hardware.  |
| <b>TERMINALES (XOD)</b>          | Los nodos terminales XOD (nodos de entrada y salida) se utilizan para permitir que un patch se comunice con 'el mundo exterior'. Añadir nodos terminales a un patch hará que ese patch sea utilizado como un nodo en otros patches, con los nombres y tipos de los terminales correspondientes a los nombres y tipos de los pines en tu nuevo nodo.   |
| <b>XOD</b>                       | XOD es una empresa de software de código abierto que proporciona el Entorno de Desarrollo Integrado (IDE) XOD. El XOD IDE es un software gratuito que permite programar placas de desarrollo basadas en Arduino utilizando la programación visual en lugar de la codificación basada en texto. El software XOD utiliza nodos gráficos para representar funciones, y los nodos se conectan entre sí para visualizar el flujo de datos y programar el hardware.   |



# Índice

## A

|                            |                               |
|----------------------------|-------------------------------|
| 3-Axis Acceleration Sensor | (ver Acelerómetro)            |
| Acelerómetro               | 7, 11, 56-59, 87              |
| Adafruit                   | 75, 85                        |
| Air Pressure Sensor        | (ver Sensor: Presión de Aire) |
| Analógico                  | (ver Comunicación Analógica)  |
| Arduino                    |                               |
| IDE                        | 79                            |
| Página Web                 | 85                            |
| Placas                     | 5-9, 84                       |

## B

|                                |                               |
|--------------------------------|-------------------------------|
| Barómetro                      | (ver Sensor: Presión de Aire) |
| Biomaker                       | i, 2, 5, 85                   |
| Biorreactor                    | 83                            |
| Bluetooth                      | 80                            |
| Boolean (Pin)                  | (ver Booleano (Pin))          |
| Booleano (Pin)                 | 15                            |
| Borrar el Programa de la Placa | 25                            |
| Botón                          | 6, 11, 20-25, 26-29, 86       |
| Breakout                       | (ver Módulos Breakout)        |
| Buses                          | (ver XOD: Buses)              |
| Buzzer                         | 6, 11, 26-29, 64-71, 86       |
| Byte (Pin)                     | 15                            |

## C

|                          |                                 |
|--------------------------|---------------------------------|
| C++                      | 79                              |
| Cableado Electrónico     | 76                              |
| Cadena (Pin)             | 15                              |
| Cajas de Comentarios     | (ver XOD: Cajas de Comentarios) |
| Cámara                   | 81, 82                          |
| Cámara de Comportamiento | 81                              |
| Clock (Nodo)             | 36-41                           |
| CO2 Sensor               | (ver Sensor: CO2)               |
| Componentes              |                                 |
| Adicionales              | 75                              |
| Choques                  | 78                              |
| Integrados               | 6-7, 86-87                      |
| Comunicación             |                                 |
| Analógica                | 10                              |
| Digital                  | 10                              |
| I2C                      | 10                              |
| Concat (Nodo)            | 42-45                           |
| Conexiones de Pines      | 11                              |
| Contactos                | 85                              |
| Controlador de Motor     | 81, 83                          |
| Count (Nodo)             | 36-41                           |
| Crear Nuevos Nodos       | 49-55                           |
| Cultivo Bacteriano       | 83                              |

## D

|                                       |                            |
|---------------------------------------|----------------------------|
| Debugger                              | (ver XOD: Debugger)        |
| Depurador                             | (ver XOD: Debugger)        |
| Diagnóstico y Resolución de Problemas | 25                         |
| Digital                               | (ver Comunicación Digital) |
| Documentación de Nodos y Librerías    | 55                         |
| Driver USB                            | 19                         |

## E

|                     |                    |
|---------------------|--------------------|
| Encontrar Nodos XOD | 79                 |
| Enlaces             | (ver XOD: Enlaces) |

## F

|                      |       |
|----------------------|-------|
| Flip (Nodos)         | 36-41 |
| Format-Number (Nodo) | 42-45 |
| Fritzing             | 85    |

## G

|                          |                    |
|--------------------------|--------------------|
| Gas Sensor               | (ver Sensor: Gas)  |
| Grove                    |                    |
| Beginner Kit for Arduino | (ver Grove: Placa) |
| Componentes              | 75, 77             |
| Página Web               | 85                 |
| Placa                    | 5, 6               |
| Seeduino                 | 6                  |

## H

|                 |                                     |
|-----------------|-------------------------------------|
| Hackster        | 5, 8, 85                            |
| Higrómetro      | (ver Sensor: Temperatura y Humedad) |
| Humidity Sensor | (ver Sensor: Humedad)               |

## I

|               |                               |
|---------------|-------------------------------|
| I2C           | (ver Comunicación: I2C)       |
| IDE           | (ver Arduino: IDE o XOD: IDE) |
| Instructables | 85                            |

## J

|             |       |
|-------------|-------|
| Join (Nodo) | 42-45 |
|-------------|-------|

## L

|              |                             |
|--------------|-----------------------------|
| LCD          | (ver Pantalla LCD)          |
| LED          | 6, 11, 20-25, 36-41, 83, 86 |
| Librería     | (ver XOD: Librería)         |
| Light Sensor | (ver Sensor: Luz)           |
| Lógica       | (ver Nodos Lógicos)         |

## M

|                    |                            |
|--------------------|----------------------------|
| M5Stack            | 75                         |
| Microcontrolador   | 6-9                        |
| Módulos Breakout   | 76                         |
| Monitor de Plantas | 82                         |
| Motor              | (ver Controlador de Motor) |

## N

|                                 |   |
|---------------------------------|---|
| No-Code Programming for Biology | (ver Programación Sin Código para Biología) |
| Nodos                           | (ver XOD: Nodos)                            |
| Nodos Lógicos                   | 60-63, 64-71                                |
| Nuevos Nodos XOD                | (ver Crear Nuevos Nodos)                    |
| Number (Pin)                    | (ver Numero (Pin))                          |
| Número (Pin)                    | 15  |

## O

|            |                     |
|------------|---------------------|
| OLED       | (ver Pantalla OLED) |
| Open Smart | 75, 77-78, 85       |
| OpenPlant  | i, 85               |

## P

|                                       |  |
|---------------------------------------|--|
| Pantalla                              |  |
| LCD                                   | 81, 83                                 |
| OLED                                  | 6, 11, 50-55, 56-59, 60-63, 82, 86     |
| pH Sensor                             | (ver Sensor: pH)                       |
| Pines                                 | (ver Conexiones de Pines o XOD: Pines) |
| Placa de Prototipos                   | (ver Placa de Pruebas)                 |
| Placa de Pruebas                      | 76                                     |
| Placas Breakout                       | (ver Módulos Breakout)                 |
| Plantas                               | (ver Monitor de Plantas)               |
| Port (Pin)                            | (ver Puerto (Pin))                     |
| Potenciómetro Rotativo                | 6, 11, 26-29, 86                       |
| Procesamiento                         | 85                                     |
| Programación Sin Código para Biología |  |
| Guía para Principiantes               | 2, 4                                   |
| Kit para Principiantes                | 6                                      |
| Página Web                            | 5, 85                                  |
| Recursos                              | 4                                      |
| Tutoriales                            | 19                                     |
| Protoborad                            | (ver Placa de Pruebas)                 |
| Publicar Librerías                    | 55                                     |
| Puerto (Pin)                          | 15                                     |
| Pulse (Pin)                           | (ver Pulso (Pin))                      |
| Pulso (Pin)                           | 15, 41                                 |

## R

|                      |                                      |
|----------------------|--------------------------------------|
| Raspberry            | Pi 82, 84                            |
| Resetear             | (ver Borrar el Programa de la Placa) |
| Rotary Potentiometer | (ver Potenciómetro Rotativo)         |

## S

|                       |                          |
|-----------------------|--------------------------|
| Seed Studio           | (ver Grove)              |
| Seeeduino             | (ver Grove: Placa)       |
| Sensor                |                          |
| CO2                   | 80                       |
| Sonido                | 7, 11, 50-55, 87         |
| Gas                   | 80                       |
| Humedad               | 80                       |
| Luz                   | 7, 11, 60-63, 81, 83, 87 |
| pH                    | 83                       |
| Presión de Aire       | 7, 11, 42-45, 82, 87     |
| Temperatura y Humedad | 7, 11, 34-35, 80, 82, 87 |
| Tierra                | 80                       |
| Shield                | 76-78                    |
| Soil Sensor           | (ver Sensor: Tierra)     |
| Sound Sensor          | (ver Sensor: de Sonido)  |
| SparkFun              |                          |
| Componentes           | 75                       |
| Página Web            | 85                       |
| Tutoriales            | 10, 76, 84               |
| String (Pin)          | (ver Cadena (Pin))       |
| Synthetic Biology IRC | i, 85                    |

## T

|                                       |                                     |
|---------------------------------------|-------------------------------------|
| Tarjeta SD                            | 82                                  |
| Temperature and Humidity Sensor       | (ver Sensor: Temperatura y Humedad) |
| Terminales                            | (ver XOD: Terminales)               |
| Tutoriales                            |                                     |
| Programación Sin Código para Biología | 19                                  |
| XOD                                   | 45                                  |
| Tweak (Nodos)                         | 33-35                               |

## U

|            |                  |
|------------|------------------|
| USB Driver | (ver Driver USB) |
|------------|------------------|

## W

|              |           |
|--------------|-----------|
| Watch (Nodo) | 33-35, 41 |
|--------------|-----------|

## X

|                           |  |
|---------------------------|--|
| XOD                       |  |
| Buses                     | 56-59                                    |
| Cajas de Comentarios      | 71                                       |
| Debugger                  | 33-35                                    |
| Enlaces                   | 14                                       |
| Guía                      | 45                                       |
| IDE                       | 5, 12-13, 19                             |
| Librería                  | 12                                       |
| Documentar Librerías      | (ver Documentación de Nodos y Librerías) |
| Publicar Librerías        | (ver Publicar Librerías)                 |
| Nodos                     | 14, 21, 41, 45                           |
| Documentar Nodos          | (ver Documentación de Nodos y Librerías) |
| Encontrar Nodos           | (ver Encontrar Nodos XOD)                |
| Lista de Nodos Utilizados | 88                                       |
| Nuevos Nodos              | (ver Crear Nuevos Nodos)                 |
| Página Web                | 5, 85                                    |
| Pines                     | 14                                       |
| Software                  | (ver XOD IDE)                            |
| Terminales                | 49                                       |
| Terminología              | 14                                       |
| Tipos de Datos            | 15                                       |
| Tutoriales                | 45                                       |

## Z

|          |              |
|----------|--------------|
| Zumbador | (ver Buzzer) |
|----------|--------------|

# Reconocimientos

## **AUTORES**

Stephanie Norwood  
Jim Haseloff

## **TRADUCTORES**

Liliana Artega Dorantes  
Mario Artega Vazquez  
Fernando Guzman Chavez

## **IMÁGENES**

Stephanie Norwood  
SparkFun Electronics, Inc.  
Adafruit Industries, LLC  
Biomaker Challenge Participants (Participantes del Reto Biomaker)

## **DISEÑO**

Stephanie Norwood

## **HARDWARE**

Seeed Technology Co.,Ltd.  
Arduino S.r.l.  
Adafruit Industries, LLC  
M5stack-store  
OpenSmart Tech.  
Spark Fun Electronics, Inc.

## **SOFTWARE**

XOD, Inc.  
Arduino S.r.l.  
Hackster, Inc.

## **CREADORES DE LIBRERÍA XOD**

Matt Wayland  
Marco Aita  
Cesar Sosa  
XOD user: gst  
XOD user: e  
XOD user: gweimer

## **FINANCIADORES Y PATROCINADORES**

Biotechnology and Biological Sciences Research Council (BBSRC) (BB/L014130/1)  
Engineering and Physical Sciences Research Council (EPSRC) (BB/L014130/1)  
Natural Environment Research Council (NERC) (NE/T012293/1)  
National Science Foundation (NSF) (NE/T012293/1)  
CONACYT (A1-S-38383)  
OpenPlant  
Synthetic Biology Interdisciplinary Research Centre (SynBio IRC), University of Cambridge

## **CONTACTOS**

Stephanie Norwood  
coordinator@synbio.cam.ac.uk  
Jim Haseloff  
jh295@cam.ac.uk



Esta obra está bajo una [Licencia Creative Commons Atribución-  
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).





**Diseñada para todos aquellos con poca o nula experiencia trabajando con programación o hardware, esta guía hace uso del software gratis de código abierto y hardware de bajo costo para presentarte los principios detrás de la fabricación de tus propios instrumentos.**

Aprende cómo:

- Entender y controlar una placa Arduino
- Programar sin usar código
- Utilizar dispositivos electrónicos sencillos tales como pantallas y sensores
- Construir tus propios dispositivos para la investigación biológica



Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

