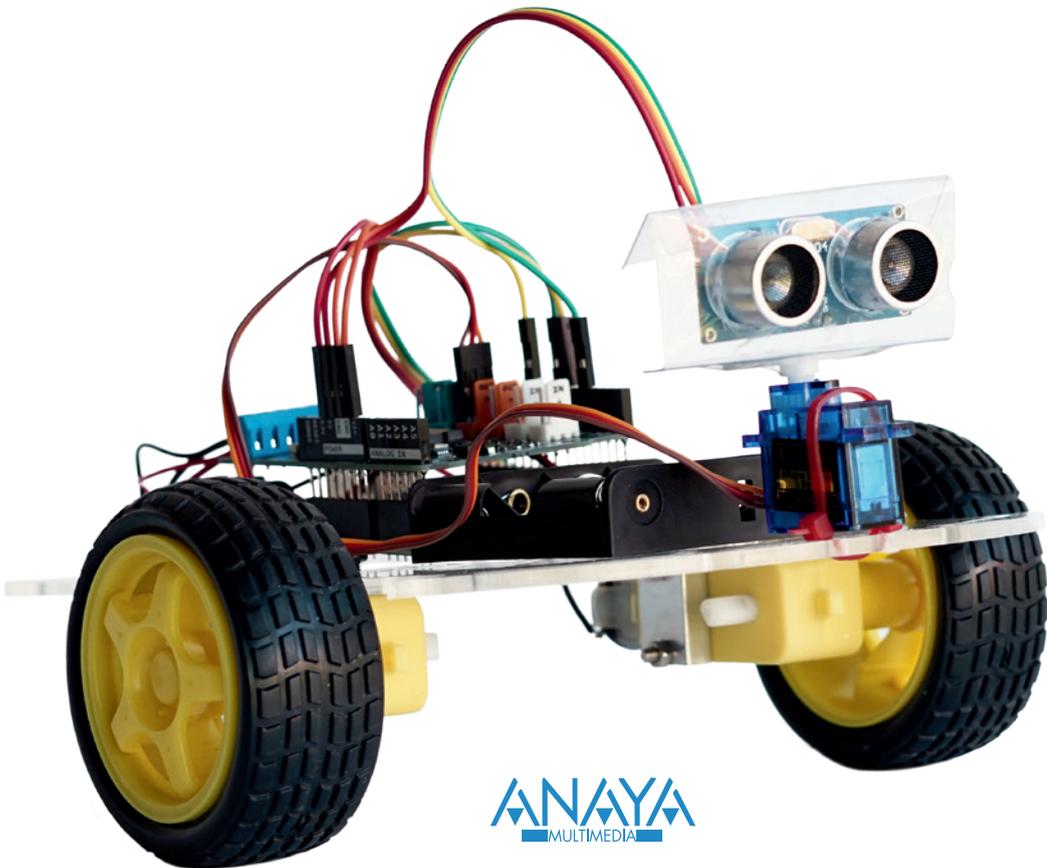




FORMA ROBOTI-K
creatividad y robótica educativa

ARDUINO, PROGRAMACIÓN Y ROBÓTICA

CREA PROYECTOS PASO A PASO



ANAYA
MULTIMEDIA

ÍNDICE DE CONTENIDOS

Antes de empezar...	12
Conocimientos	12
Ejemplos del libro	12
Sobre Arduino	
Sobre el lenguaje C	12
Sobre MIT App Inventor	12
Sobre Raspberri Pi	13
Parte I. Arduino paso a paso	14
1. Electrónica	17
Arduino	18
A. ¿Qué es Arduino?	18
B. ¿De dónde viene?	18
C. ¿A dónde va?	19
D. Entorno de programación Arduino (IDE)	19
E. Configuración	20
Componentes	22
A. Componentes básicos	22
Arduino Uno	22
Cable de conexión USB A/B M/M	22
Cables de conexión Dupont	23
<i>Protoboard</i> (placa de prototipos)	23
B. Componentes electrónicos	24
Pasivos	24
Resistencias	24
Fotorresistencia LDR	26
Sensor de temperatura LM35	26

Activos	26	I. Luminosidad led según luz (1)	79
Pulsador	27	J. Luminosidad led según luz (2)	81
Led	27	K. Luminosidad led según luz (3)	83
Zumbador piezoeléctrico (buzzer activo)	28	L. Luminosidad con pulsador	85
Microservomotor	28	M. Termómetro de ledes	87
Señales analógicas y digitales	29	N. Termostato	94
A. Señales analógicas	29	Parte II. Invernadero sensorizado	96
B. Señales digitales	29	4. Electrónica	99
2. Programación	31	Componentes	100
Programando Arduino en C	32	A. Arduino Nano	100
A. Estructura	32	B. Reloj RTC	100
B. Signos de puntuación	33	C. Sensor de luz	101
C. Operadores	34	D. Sensor de humedad en tierra	101
D. Variables	35	E. Pantalla LCD	101
E. Tipos de datos	36	F. Potenciómetro	102
F. Constantes	40	G. Relé de 2 canales	102
G. Funciones	41	H. Enchufe, cable y casquillo	102
Funciones digitales	42	I. Lámpara LED	103
Funciones analógicas	44	J. Cable de conexión USB 2.0 - Mini USB 5P	103
Funciones de tiempo	46	K. Transformador USB	103
Funciones matemáticas	47	L. Más componentes	103
Funciones para generación de aleatorios	47	5. Programación	105
Funciones de comunicación serie	48	Librerías	106
H. Estructuras de control de flujo	53	A. Liquid Crystal	107
I. Comunicación serie	56	B. Wire	107
J. Palabras reservadas del IDE de Arduino	56	C. ABlocks_DS3231rtc	107
3. Prácticas	59	D. Math	107
A. Led parpadeante	60	Prácticas	108
B. Secuencia de ledes	61	A. Led	108
C. SOS con un led	64	B. Conexión serie	112
D. Cruce de semáforos	67	C. Sensor de luz analógico	115
E. Coche fantástico	69	D. Sensor de temperatura analógico	119
F. Secuencia de ledes con pulsador	72	E. Pantalla LCD	122
G. Ruleta de la fortuna	75		
H. Intensidad luminosa del led	77		

6. Desarrollo del proyecto	127	8. Desarrollo del proyecto	179
Montaje	128	Montaje	180
A. Herramientas, materiales y componentes	128	A. Herramientas, materiales y componentes	180
B. Instrucciones de montaje	129	B. Instrucciones de montaje	181
Programación	138	Programación	194
Proyecto: mi primera planta	140	Configuración	200
Parte III. Robot explorador	142	Parte IV. CÁMARA DE VIGILANCIA	202
7. Electrónica y programación	145	9. Electrónica y programación	205
Componentes electrónicos	146	Componentes electrónicos	206
A. Controladora de motores	146	A. Raspberry Pi	206
Alimentación	146	B. Cámara web (<i>webcam</i>)	206
Entradas y salidas	147	C. Batería externa (<i>power bank</i>)	206
Conexión de motores: motor de CC sin escobilla	148	D. Tarjeta de memoria (MicroSD)	207
Características	148		
Prácticas	149	App Inventor	207
B. Sensor de ultrasonidos	155	A. Abriendo App Inventor	207
Operación	155	B. Descripción del entorno de desarrollo	209
Prácticas	156	Menús	209
C. Servomotor	160	MENÚ PRINCIPAL	209
Conexiones	161	MENÚ DE LA APLICACIÓN	210
Valores de velocidad y de torque	161	Modo diseñador (<i>Designer</i>)	210
Biblioteca servo	160	PALETA (<i>Pallette</i>)	211
Funciones	162	VISOR (<i>Viewer</i>)	211
D. Comunicación <i>Bluetooth</i>	166	COMPONENTES (<i>Components</i>)	211
Módulo <i>Bluetooth</i> HC-06	167	PROPIEDADES (<i>Properties</i>)	211
Complementos de programación	168	Modo bloques (<i>Blocks</i>)	212
A. Biblioteca SoftwareSerial	168	BLOQUES (<i>BLOCKS</i>)	212
Limitaciones	168	VISOR (<i>VIEWER</i>)	212
Funciones	168	AVISOS (<i>SHOW WARNINGS</i>)	213
		MOCHILA	213
		BOTONES DE AJUSTE (CÓDIGO Y PAPELERA)	213

C. Programación	214
Bloques mutadores	214
 Bloques de control (Control)	214
 Bloques lógicos (Logic)	217
 Bloques de matemáticas (Math)	219
 Bloques de texto (Text)	220
 Bloques de lista (Lists)	223
 Bloques de color (Colors)	227
 Bloques de variables (Variables)	228
 Bloques de procedimientos (Procedures)	229
Prácticas	231
A. Hola Mundo	231
B. Cambio de color	232
C. Cambio de ventana	234
D. Variables	236
E. Brújula	238
F. Tamaño	241
G. Laberinto	243
10. Desarrollo del proyecto	251
Montaje	252
A. Herramientas, materiales y componentes	252
B. Instrucciones de montaje	253
Programación	258
Configuración	265
A. Raspberry Pi	265
B. Wifi	266
C. Aplicación	267
Amplía información...	268

Antes de empezar...

CONOCIMIENTOS

- Hemos diseñado esta obra para guiar al lector paso a paso en la programación de Arduino y en la construcción de tus propios proyectos, sin necesidad de que tengas conocimientos previos. Los contenidos de este libro son útiles para desarrollar las materias STEAM: Ciencia, Tecnología, Ingeniería, Arte y Matemáticas, muy demandadas en los centros educativos actualmente.

EJEMPLOS DEL LIBRO

- Los ejemplos y los recursos de este libro se encuentran disponibles en la página web de Anaya Multimedia (<http://www.anayamultimedia.es>) en la opción «Complementos» de la ficha correspondiente a este libro.

Sobre Arduino

Arduino es una marca que identifica una plataforma electrónica de código abierto, basada en hardware y software fácil de usar.

- Web oficial de Arduino: «www.arduino.cc/»
- Web oficial de Arduino en español: «playground.arduino.cc/Es/Es»

ARDUINO® es una marca registrada de Arduino AG.

Sobre el lenguaje C

C es un lenguaje de programación con el cual se desarrollan tanto aplicaciones como sistemas operativos a la vez que forma la base de otros lenguajes más actuales como Java, C++ o C#. Es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear softwares de sistemas y aplicaciones.

Sobre MIT App Inventor

MIT APP Inventor es una marca que identifica un entorno de programación de fácil uso.

- Web oficial de App Inventor: «<http://appinventor.mit.edu/explore/>»

MIT App Inventor es una marca registrada de Massachusetts Institute of Technology.

Sobre Raspberry Pi

Raspberry Pi es un ordenador de bajo coste y tamaño reducido desarrollado en el Reino Unido por la Raspberry Pi Foundation, que buscaba la promoción de la enseñanza de informática en las escuelas y que ahora está a disposición de todo el mundo para desarrollar la informática y la creación digital.

- Web oficial de Raspberry Pi: «www.raspberrypi.org/»

Raspberry Pi es una marca registrada de la Raspberry Pi Foundation.

Sobre este libro

Este libro permitirá al lector iniciarse en la robótica educativa asentando las bases de la programación de una manera totalmente práctica.

El libro consta de 4 partes, en cada una de ellas se combina teoría y práctica. La parte teórica, además, incluye pequeños ejercicios para que el lector pueda comprender con mayor profundidad la teoría de la programación. La parte práctica está basada en la realización, paso a paso, de un proyecto completo. Incluye, además, el detalle de todo el material necesario para la realización de dichos proyectos, así como ilustraciones para que el lector pueda seguir todas las fases de realización sin perderse.

En la primera parte, «Arduino paso a paso», el lector se iniciará en la programación con Arduino con el lenguaje de programación C, revisando de una manera práctica todos los componentes y elementos de la famosa placa.

En la segunda y tercera parte, seguiremos profundizando en la programación de la placa Arduino para que el lector pueda llevar a cabo un «invernadero sensorizado», que mide los niveles de temperatura, luz y humedad ambiente, así como los niveles de humedad en tierra, y un «robot explorador».

En la cuarta y última parte, nos adentraremos en la programación de la CPU Raspberry, para llevar a cabo una «cámara de vigilancia», que programaremos con App Inventor. Esta cámara podremos adaptarla al robot explorador.

Es un libro muy completo, que incluye todo lo que el lector necesita para aprender a programar de manera gradual, disfrutando mientras lleva a cabo divertidos proyectos.

PASO 6

Una vez configurado el ordenador y la placa, para ejecutar un programa se debe pulsar la flecha (→) que aparece en la cabecera.



Verificar



Subir



Nuevo



Abrir



Salvar

Componentes

Los componentes son los diferentes elementos físicos que componen un proyecto. En el caso específico de este libro se realizarán exclusivamente prototipos.

En caso de haber adquirido este libro junto con el kit de material, estos son los componentes más relevantes para el desarrollo de las prácticas incluidos.

A Componentes básicos

Son los componentes mínimos necesarios para construir y programar un prototipo.



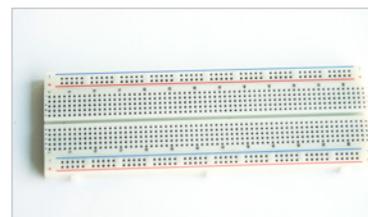
ARDUINO UNO

Arduino UNO es una placa controladora que porta un ATmega328p. Se alimenta tanto por la conexión USB como por la conexión VIN con un voltaje entre 6 y 20 V. Esta placa tiene el tamaño aproximado de una tarjeta de crédito, aunque en realidad posee una gran capacidad y versatilidad para controlar muchos componentes.



CABLE DE CONEXIÓN USB A/B M/M

Cable de conexión de datos para USB tipo «A macho - B macho». Se emplea para conectar la placa Arduino al ordenador.



PROTOBOARD (PLACA DE PROTOTIPOS)

Es una placa de forma rectangular con 810 perforaciones conectadas de forma interna que nos permitirán comunicar los diferentes elementos de nuestro prototipo a una placa Arduino mediante los cables de conexión.

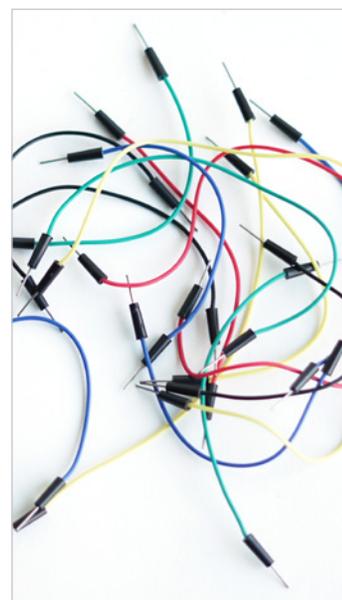
En los laterales de la placa se pueden ver dos franjas, una roja y una azul, con sendas líneas de pines junto a ellas conectados entre sí, por lo que habitualmente son usados para distribuir el positivo (línea roja) y el negativo (línea azul) a lo largo de la placa de prototipos, facilitando el acceso a la corriente de los diferentes elementos y pudiendo mantener voltajes de trabajo diferentes a cada lado de la *protoboard*.

El resto de orificios están separados por una línea central que delimita el paso de corriente de una zona a otra. Los pines de esta sección se comunican desde el pin A al pin E.

Ejemplo

Si introducimos nuestro cable en el pin A de la fila 1, estará comunicado con los pines B1, C1, D1, E1, no así con los pines F, G, H, I y J de la fila 1.

De la misma forma, si conectamos el cable al pin A1, este no está conectado al pin A2, A3...



CABLES DE CONEXIÓN DUPONT

Son los encargados de comunicar los diferentes componentes, la *protoboard* y la controladora Arduino. Existen de varios tamaños, colores y tipos de terminal de conexión, pudiendo ser machos o hembras. En función del tipo de terminación se clasifican en:

- Macho – Macho
- Macho – Hembra
- Hembra – Hembra

El tipo de cable de conexión que usaremos en cada momento dependerá de las necesidades de cada componente y situación. Esto no supone un cambio en el funcionamiento de los dispositivos.

Solución 2

```
int tcorto = 100;
int tlargo = 300;
int pausa = 300;
int espera = 1000;
int n = 0;

void setup() { // Comienza la configuración
  pinMode(13, OUTPUT);
}

void s() {
  // Comienza el bucle para la letra «S»
  for (n = 0; n < 3; n++) {
    digitalWrite(13, HIGH);
    delay(tcorto);
    digitalWrite(13, LOW);
    delay(tcorto);
  }
}

void o() { // Comienza el bucle para la letra «O»
  for (n = 0; n < 3; n++) {
    digitalWrite(13, HIGH);
    delay(tlargo);
    digitalWrite(13, LOW);
    delay(tlargo);
  }
}

void loop() {
  // Se ejecuta el bucle principal en el orden siguiente
  s();
  delay(pausa);
  o();
  delay(pausa);
  s();
  delay(espera);
}
```

Objetivos

- Familiarizarse con el entorno de programación.
- Aprender a declarar variables tipo lista de valores.

D Cruce de semáforos

En esta práctica simularemos un cruce de semáforos.

Semáforo 1:

- Pin «11» (rojo)
- Pin «12» (ámbar)
- Pin «13» (verde)

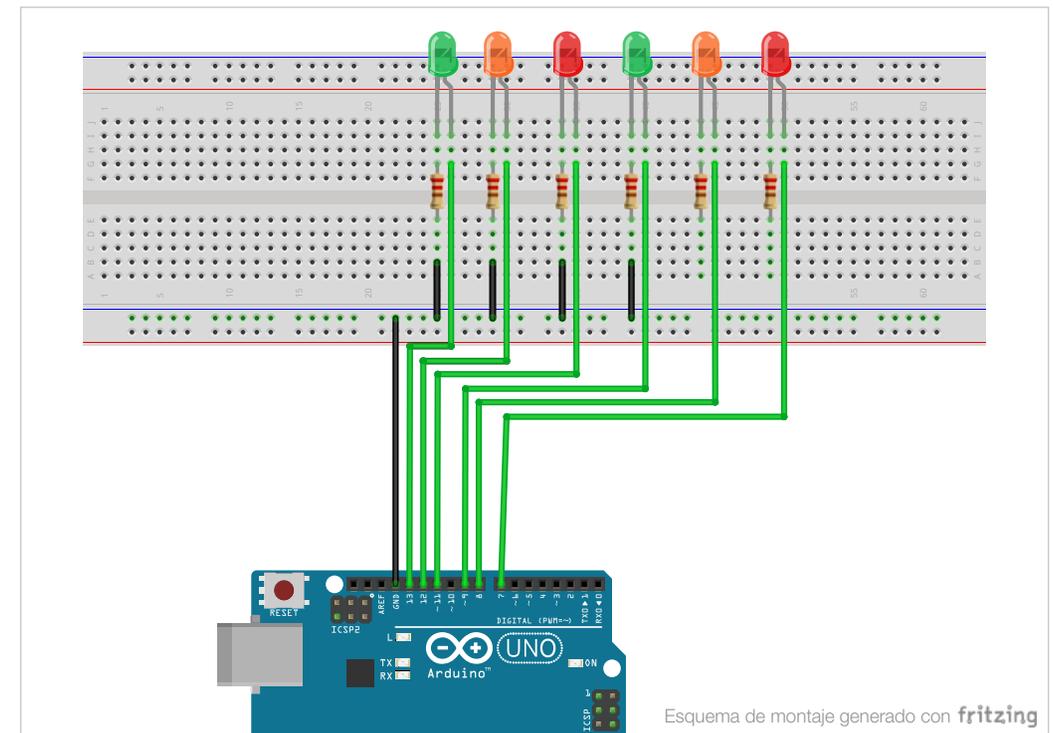
Semáforo 2:

- Pin «7» (rojo)
- Pin «8» (ámbar)
- Pin «9» (verde)

El funcionamiento de los semáforos será el siguiente:

1. Rojo 1 y Verde 2 (durante 3000 ms)
2. Parada (durante 2000 ms)
3. Rojo 1 y Ámbar 2 (durante 500 ms)
4. Parada (durante 2000 ms)
5. Verde 1 y Rojo 2 (durante 3000 ms)
6. Parada (durante 2000 ms)
7. Ámbar 1 y Rojo 2 (durante 500 ms)

El número al lado de cada color indica la posición del led en la placa de prototipos.



Crea un programa que muestre un párrafo de forma que pueda leerse correctamente en la pantalla.

Solución

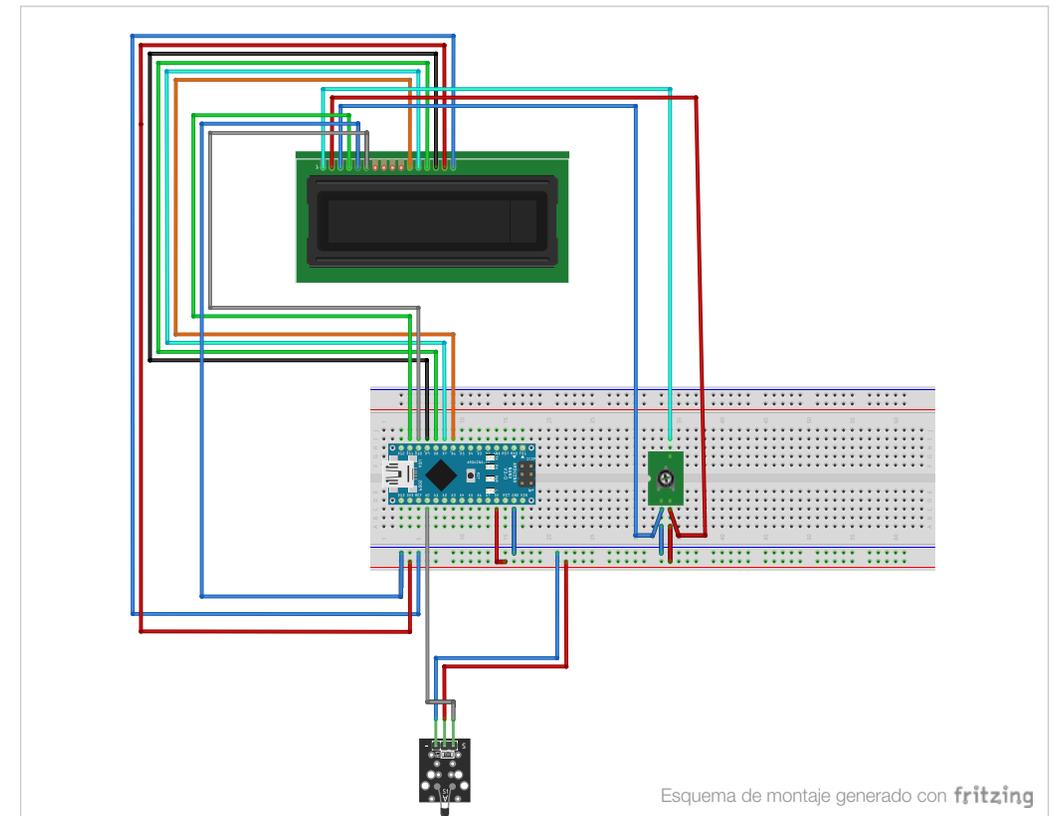
```
void loop()
{
  lcd.setCursor(0, 0);
  // Coloca el cursor en la columna «0» y fila «0» de la pantalla
  lcd.print("Programar es muy fácil");
  // Muestra el mensaje «Programar es muy fácil» en la pantalla
  delay(500);
}
```

Incorpora un sensor de luz al proyecto y muestra su valor en la pantalla LCD.

Solución

```
void setup() {
  lcd.begin(16, 2);
  pinMode(A0, INPUT);
}

void loop()
{
  lcd.setCursor(0, 0);
  // Coloca el cursor en la columna 0, fila 0 de la pantalla
  lcd.print(analogRead(A0))
}
```



Modifica el código anterior para mostrar el valor del sensor de luz en forma de porcentaje en la pantalla LCD.

Solución

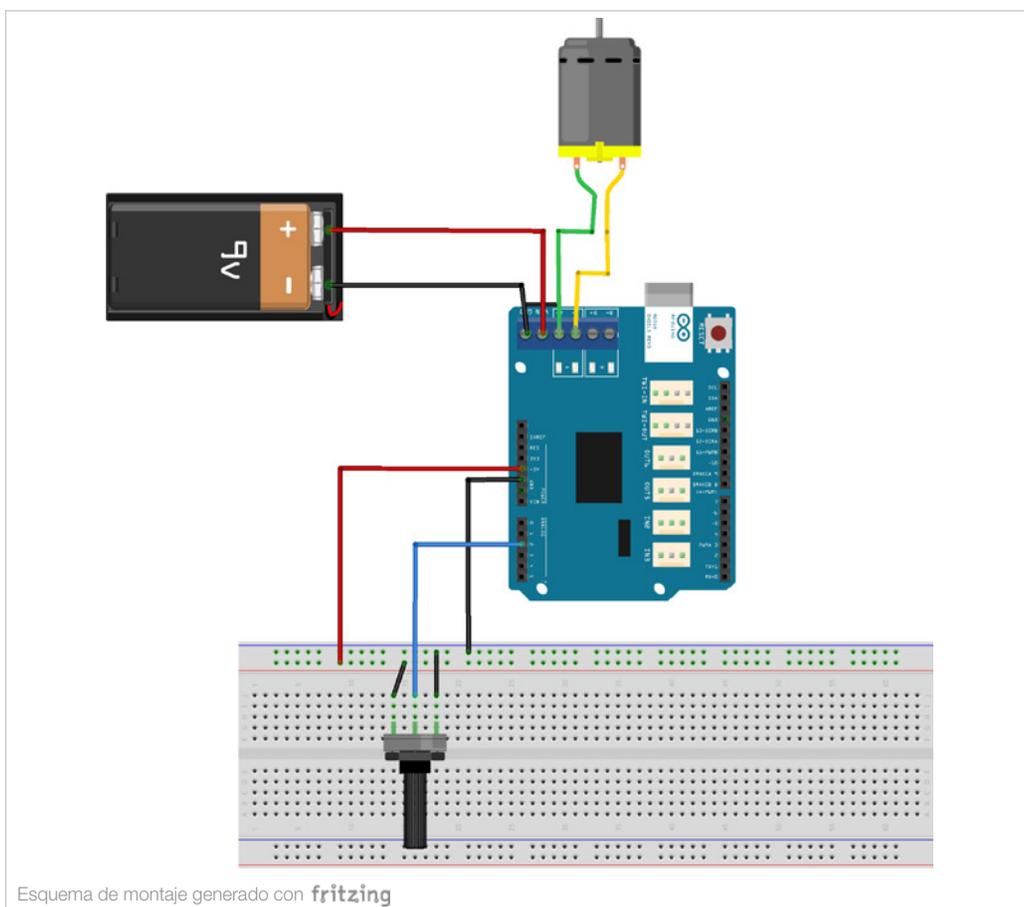
```
void loop()
{
  lcd.setCursor(0, 0);
  // Coloca el cursor en la columna «0» y fila «0» de la pantalla
  int porcentaje = map(analogRead(A0), 0, 1024, 0, 100);
  lcd.print(porcentaje + " % ");
}
```

```

int valorPot = 0;
void setup() {
  pinMode(BRAKE_A, OUTPUT); // Pin de freno en el «canal A»
  pinMode(DIR_A, OUTPUT); // Pin de dirección en el «canal A»
  pinMode(POT, INPUT); // Pin de entrada del potenciómetro
}

void loop() {
  valorPot = map(analogRead(POT), 0, 1023, 0, 256);
  // Transforma el valor del potenciómetro a una escala entendible
  // por el motor
  analogWrite(PWM_A, valorPot); // Envía el valor del sensor mapeado
  // al motor
}

```



Esquema de montaje generado con fritzing

Añade un segundo motor y modifica el código generado en la práctica anterior para controlar los dos motores al unísono.

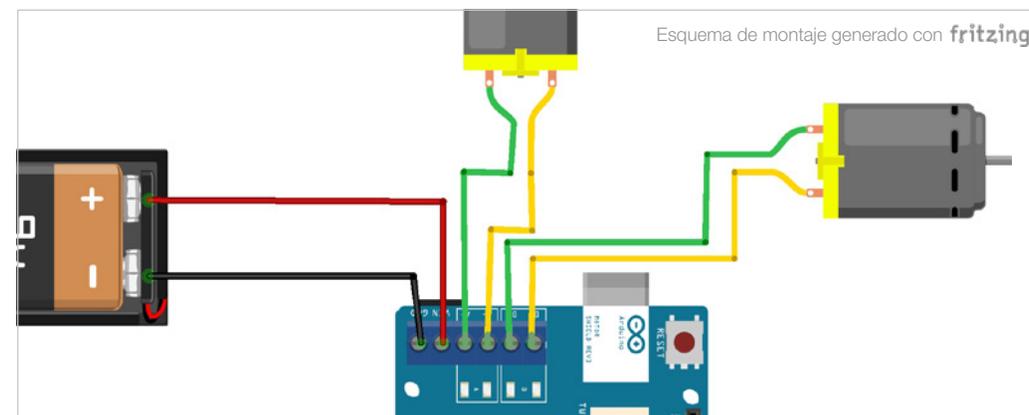
Solución

```

const int
PWM_A = 3,
DIR_A = 12,
PWM_B = 11,
DIR_B = 13,
POT = A2;
int valorPot = 0;
void setup() {
  pinMode(BRAKE_A, OUTPUT); // Pin de freno en el «canal A»
  pinMode(DIR_A, OUTPUT); // Pin Dirección en el «canal A»
  pinMode(BRAKE_B, OUTPUT); // Pin de freno en el «canal B»
  pinMode(DIR_B, OUTPUT); // Pin Dirección en el «canal B»
  pinMode(POT, INPUT); // Pin entrada del potenciómetro
}

void loop() {
  valorPot = map(analogRead(POT), 0, 1023, 0, 256);
  // Transforma el valor del potenciómetro a una escala entendible
  // por el motor
  analogWrite(PWM_A, valorPot);
  // Envía el valor del sensor mapeado al motor
  analogWrite(PWM_B, valorPot);
  // Envía el valor del sensor mapeado al motor
}

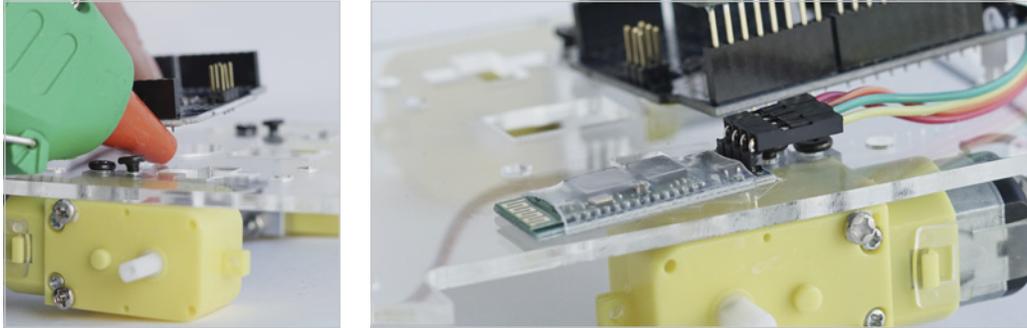
```



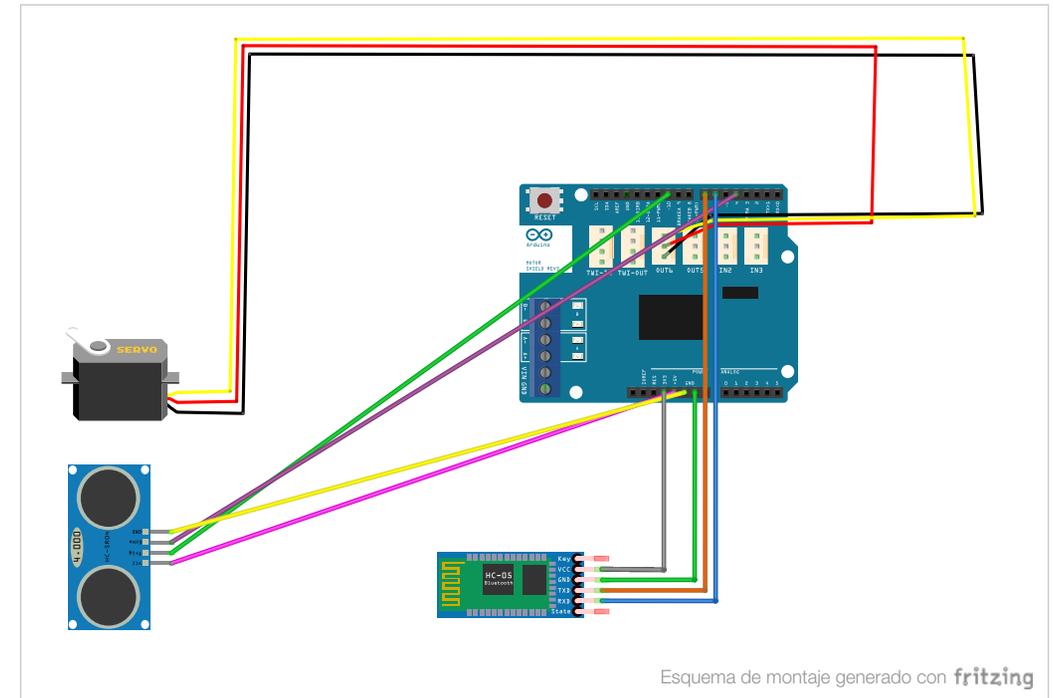
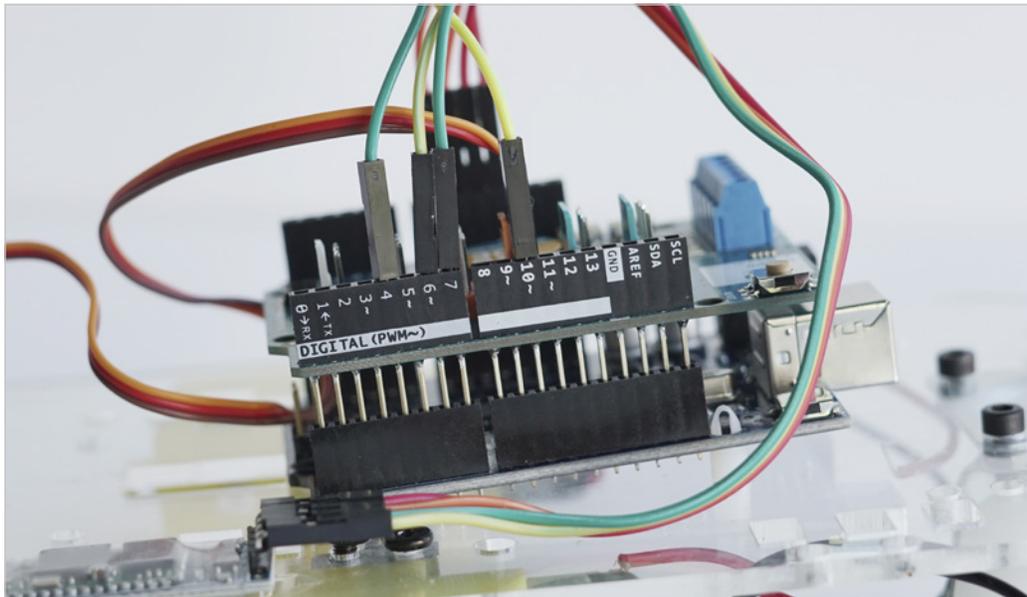
Esquema de montaje generado con fritzing

PASO 12

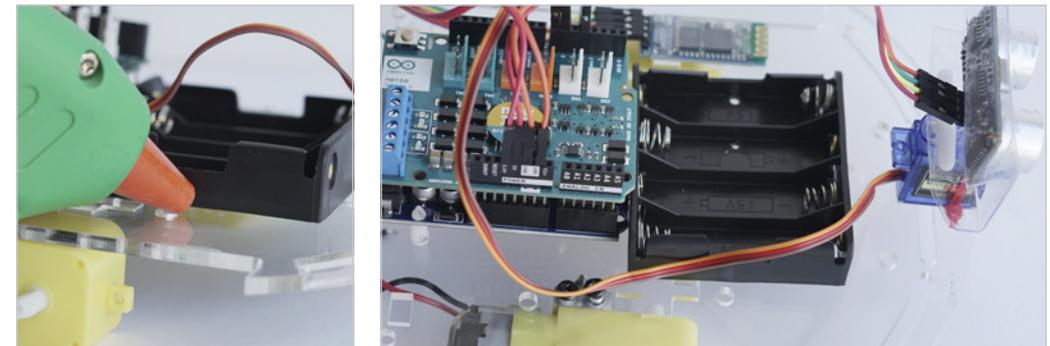
Fija con la termoselladora el dispositivo *Bluetooth* en el lateral del chasis y conéctale la faja de cuatro cables.

**PASO 13**

Conecta el pin «RX» del *Bluetooth* al pin «6» y el «TX» al pin «7».

**PASO 14**

Fija con la termoselladora el portapilas en la parte delantera del chasis y asegúrate de que los cables apunten hacia la controladora.

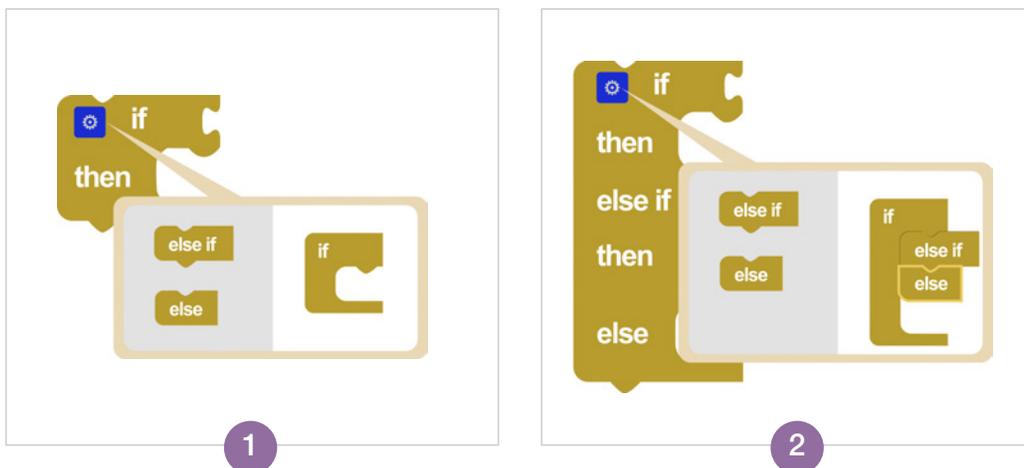


C Programación

En este apartado se presentan las instrucciones y funciones comunes a todas las aplicaciones desarrolladas en App Inventor, que se encuentran agrupadas en función de la acción que ejecutan. Cada una de las instrucciones que se usarán para programar aplicaciones se han diseñado en forma de bloques interconectables, lo que agiliza el trabajo en gran medida.

Bloques mutadores

Son bloques cuya característica especial es poder modificarse. Se pueden distinguir por el icono azul con un engranaje que aparece siempre en la esquina superior izquierda. Al hacer clic en dicho icono, se despliega un menú mediante el cual podremos arrastrar nuevas secciones al bloque mutador.



Hacemos clic en el icono azul para desplegar las propiedades del bloque mutador

Arrastramos el bloque a la parte derecha para formar el bloque final

Bloques de control (Control)

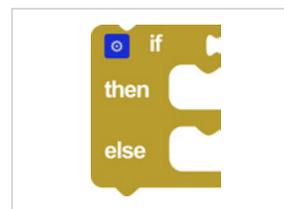
Con los bloques de control se puede definir el flujo del código.



IF & IF ELSE

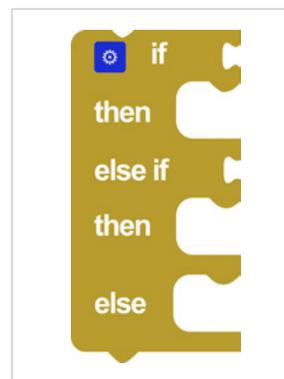
Comprueba la condición asignada a la instrucción.

- En caso de ser verdadera, ejecutará las instrucciones contenidas en el apartado «then».



Comprueba la condición asignada a la instrucción.

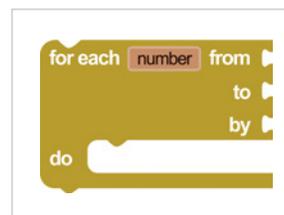
- En el caso de ser verdadera, ejecutará las instrucciones contenidas en el apartado «then».
- En el caso de que la condición devuelva una respuesta falsa, se ejecutará el código contenido en la sección «else».



Comprueba la condición asignada a la instrucción.

- En el caso de ser verdadera, ejecutará las instrucciones contenidas en el apartado «then».
- En el caso de que la condición devuelva una respuesta falsa, se comprobará el resultado de la segunda condición.
- En el caso de que todas las condiciones generen un resultado con valor «false», se ejecutará el código contenido en la sección «else».

FOR EACH FROM TO



Las instrucciones contenidas en la sección «do» se ejecutarán una vez por cada valor desde «from» hasta «to». En cada ejecución el valor se modificará en función del dato que se aporte a la sección «by».

FOR EACH IN LIST



Recorre una lista dada en la sección «in list». El código contenido en la sección «do» se ejecutará una vez por cada registro que contenga la lista.

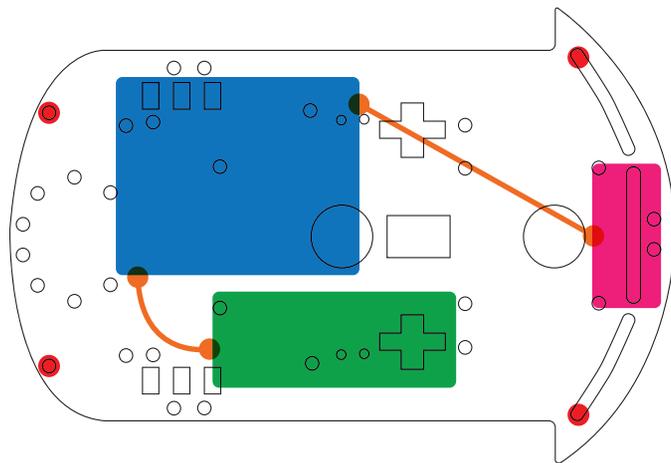
WHILE



El funcionamiento es similar a la instrucción «if». Comprueba una condición y, mientras esta devuelva un resultado de verdadero, se ejecutarán las instrucciones de la sección «do». Esto sucederá de forma repetitiva hasta que la condición devuelva un resultado «false».

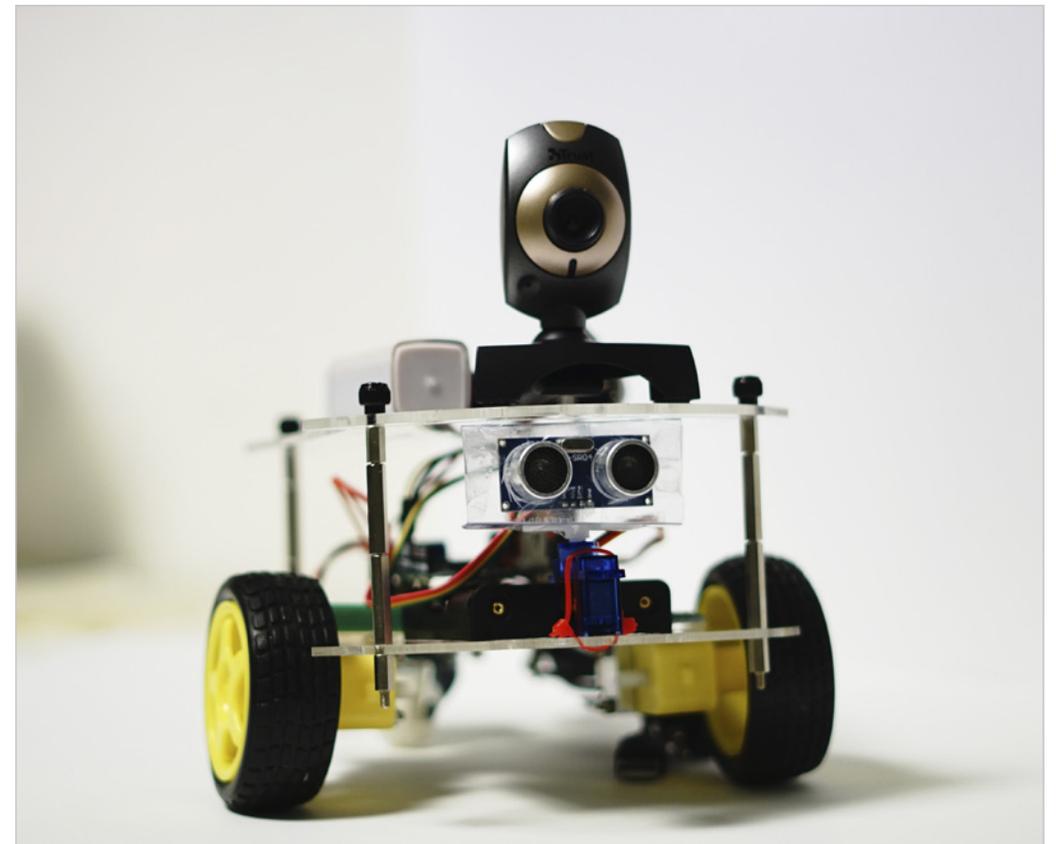
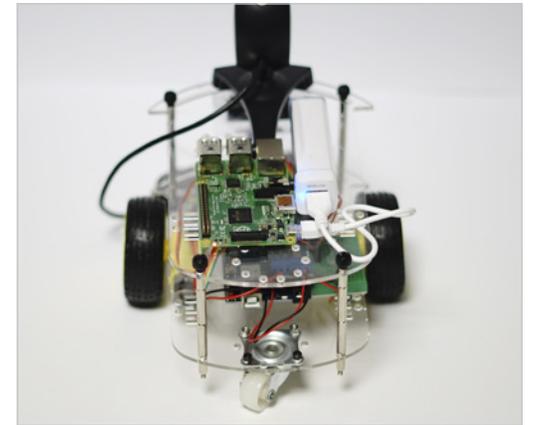
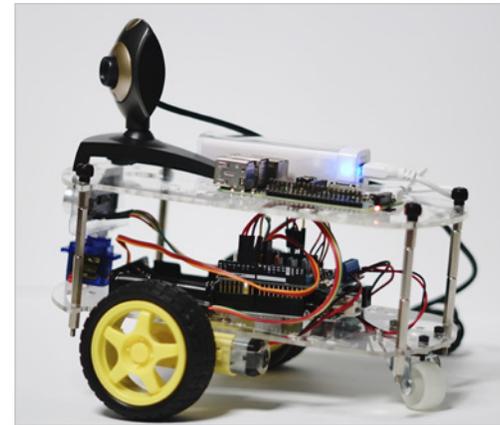
PASO 6

Conecta el cable USB-MicroUSB desde la batería hasta la Raspberry Pi. Una vez conectado el cable, el equipo quedará encendido. Para apagar el equipo basta con desconectar este cable.


 **Consejo**

Si este es tu primer montaje, desconecta el equipo hasta completar la instalación de la aplicación para ahorrar batería.

RESULTADO FINAL



Si eres un amante de la tecnología y quieres iniciarte en el mundo de la robótica educativa, ¡este libro es para ti!

A lo largo de varios proyectos explicados paso a paso, podrás aprender a programar de forma gradual sin necesidad de tener conocimientos previos.

Aprenderás a manejar los componentes, los elementos y las características de las placas Arduino para programarlas con el lenguaje C y descubrirás el potencial de bibliotecas como Software Serial.

También trabajarás con la aplicación App Inventor 2 para programar acciones de envío y respuesta desde una Raspberry Pi hasta un dispositivo Android.

Para poner en práctica todo lo aprendido, desarrollarás tres proyectos diferentes: un «invernadero sensorizado», un «robot explorador» y una «cámara de vigilancia» que podrás incorporar al robot explorador.

En definitiva, un libro muy completo con todo lo que necesitas conocer para llevar a cabo proyectos increíbles de una forma didáctica y divertida.

