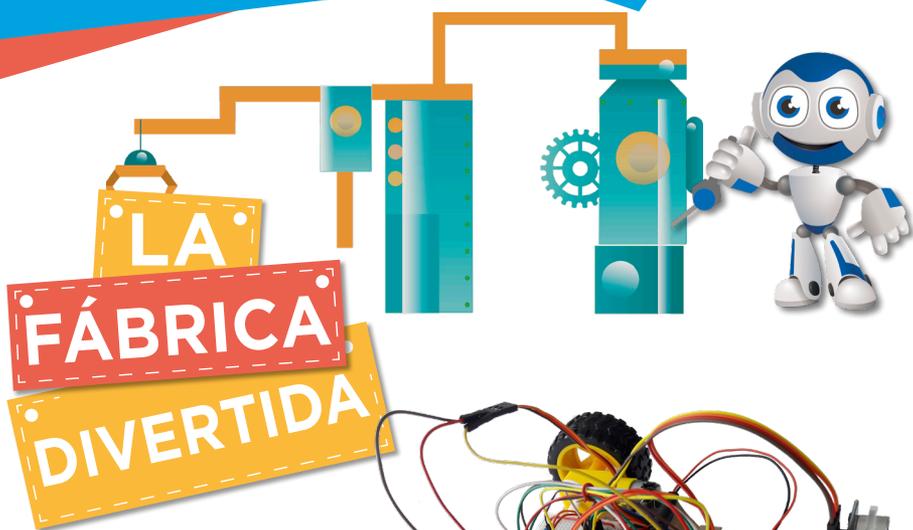
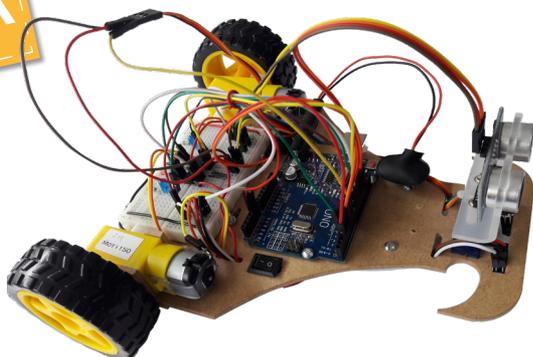


Móvil bluetooth



1 HC-06 Módulo bluetooth maestro – esclavo



1 Resuelve laberintos



1 Resistencia de carbón
1 k Ohms ($k\Omega$)



4 Cables puente macho-hembra de 20 cm



1 Cable puente macho – macho de 20 cm



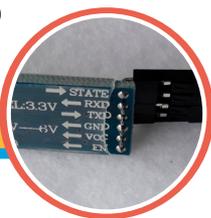
1 Resistencia de carbón
2.2 k Ohms ($k\Omega$)

También necesitarás:

- 1** Desarmador plano
- 1** Desarmador de cruz
- 1** Pinza para pelar cables
- 1** Pila de 9 V
- 4** Pilas AA
- Pegamento: silicón, Kola-loka, etc.

Nota: No desconectes ni desatornilles nada del Resuelve laberintos.

1 Conecta el extremo hembra de cuatro cables puente macho – hembra en las terminales RXD, TXD, GND y VCC del módulo bluetooth.



2 Conecta el extremo macho del cable puente de TXD en RX-0 de la Arduino.



Nota: El extremo hembra tiene un orificio al centro.

3 Conecta el extremo macho del cable puente de RXD en [G, 4] de la protoboard.

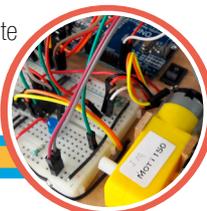


4 Conecta el extremo macho del cable puente de GND en el tercer orificio negativo del primer bloque que se encuentra del lado posterior de la protoboard.



Nota: El extremo macho del cable puente es el que tiene un pin.

5 Conecta el extremo macho del cable puente de VCC en el tercer orificio positivo del primer bloque que se encuentra del lado posterior de la protoboard.

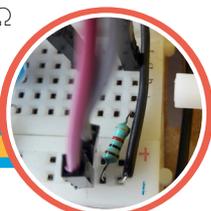


6 Conecta el extremo de un cable puente macho – macho en [G, 4] de la protoboard y el extremo libre conéctalo en TX-1 de la Arduino.



7 Conecta el extremo de una resistencia de 1 k Ω en [J, 1], y el extremo libre de esta resistencia conéctala en el segundo orificio negativo del primer bloque posterior de la protoboard.

Nota: La resistencia de 1 k Ω es la que tiene una línea café, una negra, una roja y una dorada.

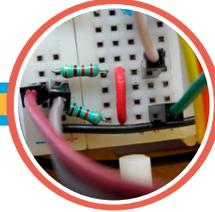


Nota: La resistencia de 2.2 k Ω es la que tiene tres líneas rojas y una dorada.

8 Conecta el extremo de la resistencia de 2.2 k Ω en [J, 4] de la protoboard, y el extremo libre de esta resistencia conéctala en el cuarto orificio positivo del primer bloque posterior de la protoboard.

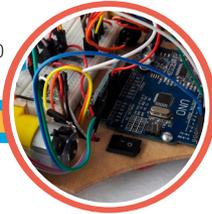


9 Conecta [I, 1] con [I, 4] usando un tramo de 2 cm de alambre.



10

Pega el bluetooth a lado de la protoboard.



Nota: Puedes ver los pasos para configurar tu interface del programa en el video de la siguiente liga https://youtu.be/IE_x-85F-28

11 En tu navegador de internet abre la página web <http://appinventor.mit.edu/explore/> y da clic en el botón crea aplicaciones o **create apps** ubicado en la esquina superior derecha de tu pantalla.



12 Asigna un correo y acepta los términos y condiciones del programa para poder usarlo. Cambia el idioma a español de la interface dando clic en la opción **English** y selecciona el lenguaje **Español**.



13 Da clic en el botón **Comenzar un proyecto nuevo** y asigna un nombre a tu proyecto, da clic en aceptar y comienza a crear.



20 Agrega un **Selector de Lista**, que esté dentro del menú **Interfaz de usuario** del panel **Paleta**.



19

Agregar otro componente **Disposición Horizontal** del menú **Disposición** y agrega dentro dos componentes **Botón** más del menú **Interfaz de usuario**.



18 Arrastra otro **Botón** del menú **Interfaz de usuario** de la **Paleta** de componentes hacia el centro de la pantalla de teléfono.



17 Arrastra el componente **Disposición Horizontal** que está en el panel **Paleta** dentro del Menú **Disposición**. Agrega 3 componentes **Botón** del menú **Interfaz de usuario** dentro del cuadro que aparece.



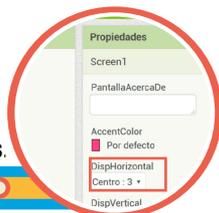
16

Arrastra hacia el centro de la pantalla del teléfono el componente **Botón** que se encuentra dentro del menú **Interfaz de usuario** en el panel **Paleta**.



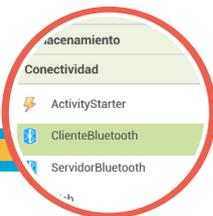
15

Da clic en la imagen de pantalla de teléfono que está dentro del panel **Visor**, después modifica la **Disposición Horizontal** a **Centro** en el panel **Propiedades**.



14 Accede a la interfaz **Diseño** de tu aplicación, la cual posee diferentes paneles como: **Paleta**, que contiene los componentes para formar tu app; el **Visor**, donde previsualizas tu aplicación; **Propiedades**, donde modificas cada componente que vayas agregando; **Componentes**, que muestra la lista de los elementos que has agregado; **Medios**, que contiene los archivos que vas subiendo a tu aplicación.

21 Arrastra la opción **ClienteBluetooth**, que se encuentra en el menú **Conectividad** dentro del Panel **Paleta**.



22 Selecciona el primer botón que agregaste (**Botón1**), da clic en el panel **Propiedades** y busca la opción **Texto** y elimina el contenido que tiene dentro.

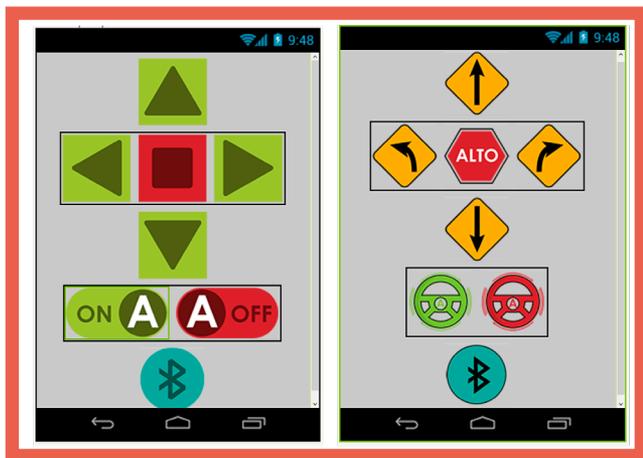


23 Da clic en la siguiente liga <http://bit.ly/2AyKCnc> y descarga la carpeta **Móvil bluetooth**. Dentro hay dos carpetas de imágenes selecciona el grupo de botones que utilizarás para hacer tu aplicación: **IMAGENES APK OP1** o **IMAGENES APK OP2**.

24 Da clic en la opción **Imagen** de ese mismo panel y da clic en el botón **Subir archivo**. En la ventana que aparece da clic en el botón **Examinar**, selecciona la imagen con la flecha hacia arriba y da clic en **Aceptar**.



25 Repite el paso anterior con cada uno de los botones para que al final la pantalla de tu programa quede como la que te mostramos.



```

cuando SelectorDeLista1 .AntesDeSelección
ejecutar poner SelectorDeLista1 .Elementos como ClienteBluetooth1 .DireccionesYNombres

cuando SelectorDeLista1 .DespuésDeSelección
ejecutar poner SelectorDeLista1 .Selección como SelectorDeLista1 .Selección

cuando Botón1 .Clic
ejecutar llamar ClienteBluetooth1 .EnviarTexto
      texto "a"

cuando Botón2 .Clic
ejecutar llamar ClienteBluetooth1 .EnviarTexto
      texto "b"

cuando Botón3 .Clic
ejecutar llamar ClienteBluetooth1 .EnviarTexto
      texto "c"

cuando Botón4 .Clic
ejecutar llamar ClienteBluetooth1 .EnviarTexto
      texto "d"

cuando Botón5 .Clic
ejecutar llamar ClienteBluetooth1 .EnviarTexto
      texto "e"

cuando Botón6 .Clic
ejecutar llamar ClienteBluetooth1 .EnviarTexto
      texto "f"

cuando Botón7 .Clic
ejecutar llamar ClienteBluetooth1 .EnviarTexto
      texto "g"

```

Nota: Los botones deben coincidir de la siguiente manera

- "a" con Botón1 (flecha hacia arriba)
- "b" con Botón2 (flecha hacia la izquierda)
- "c" con Botón3 (alto)
- "d" con Botón4 (flecha hacia la derecha)
- "e" con Botón5 (flecha hacia atrás)
- "f" con Botón6 (encendido del sensor automático)
- "g" con Botón7 (apagado del sensor automático)

27

Arrastra las opciones del panel **Bloques** hasta que la programación quede como la que se muestra en la imagen, localiza los bloques dando clic en el componente que está marcado en la programación. Para los bloques integrados te puedes guiar por los colores.

26

Programa los botones dando clic en el botón **Bloques** de la esquina superior derecha, aparecerá una interfaz de manera semejante a arduBlock y mBlock que ya has utilizado antes.

Nota. También puedes programar tu aplicación siguiendo las indicaciones del video en: <https://youtu.be/L3hiQ50j0nA>

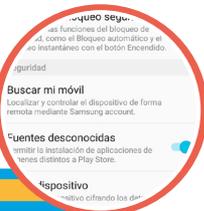


28 Da clic en el botón **Generar** del menú principal y selecciona la opción **Guardar archivo apk en mi ordenador**.



29

Una vez que tengas el archivo en tu computadora cópialo en tu teléfono o tableta usando el cable USB. Modifica los permisos de seguridad desde el botón **Ajustes** en tu dispositivo para permitir instalar la aplicación de origen desconocido (**Fuentes desconocidas**).



30 Abre el IDE (programa) de Arduino, conecta tu robot a tu computadora con la ayuda del cable USB y programa tu móvil bluetooth, para ello sigue las instrucciones en el video <https://youtu.be/FetJalPgaVI> o arma la programación tecleando el código para que quede igual al que te damos.

Nota: Recuerda respetar los espacios, signos de puntuación, mayúsculas y minúsculas de la programación, de lo contrario no funcionará. Los colores se asignan en automático al teclear correctamente el comando.

```
sketch_dec20a Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda
sketch_dec20a $
#include <Servo.h>
int Buzzer = 2;
int LEDDer = 3;
int LEDIzq = 4;
int ptrig = 5;
int pecho = 6;
Servo MyServo;
```

Aquí es donde se declaran las variables y se indica en cual pin del Arduino está conectado cada componente.

Nota: Verifica que los pines de programación correspondan con la conexión de tu robot.

```
int MotorDerA = 9;
int MotorDerB = 10;
int MotorIzqA = 11;
int MotorIzqB = 12;
int estado = 'g';

int duracion, distancia;
```

Aquí es donde se declaran las variables y se indica en qué pin del Arduino está conectado cada componente.

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin (9600);
  pinMode (MotorDerA, OUTPUT);
  pinMode (MotorDerB, OUTPUT);
  pinMode (MotorIzqA, OUTPUT);
  pinMode (MotorIzqB, OUTPUT);
  pinMode (pecho, INPUT);
  pinMode (ptrig, OUTPUT);
  pinMode (13, OUTPUT);
  MyServo.attach(7);
}
```

Aquí es donde se declara el modo como funcionará cada Variable, si será entrada o salida.

```
void loop() {
  // put your main code here, to run repeatedly:
```

```
if(Serial.available() > 0) {
  estado = Serial.read();
}
```

Condicional que indica la espera de lectura de un estado: 'a', 'b', 'c', 'd', 'e', 'f' y 'g'.

```
MyServo.write (90);
```

Indica que el servo comienza a 90°, viéndolo de frente.

```
if(estado=='a'){
  digitalWrite (MotorDerA, HIGH);
  digitalWrite (MotorDerB, LOW);
  digitalWrite (MotorIzqA, HIGH);
  digitalWrite (MotorIzqB, LOW);
  digitalWrite (LEDDer, HIGH);
  digitalWrite (LEDIzq, HIGH);
}
```

Programación correspondiente al Botón1, hacia adelante, por tanto las dos luces LED deben estar encendidas.

Nota: Los estados 'a' al 'e' son para manejar manualmente tu robot con la aplicación que creaste.

```

if(estado=='b') {
    digitalWrite (MotorDerA, LOW);
    digitalWrite (MotorDerB, LOW);
    digitalWrite (MotorIzqA, HIGH);
    digitalWrite (MotorIzqB, LOW);
    digitalWrite (LEDDer, LOW);
    digitalWrite (LEDIzq, HIGH);
}

```

Programación correspondiente al Botón2, hacia la izquierda, por tanto la LED izquierda debe estar encendida.

```

if(estado=='c') {
    digitalWrite (MotorDerA, LOW);
    digitalWrite (MotorDerB, LOW);
    digitalWrite (MotorIzqA, LOW);
    digitalWrite (MotorIzqB, LOW);
    digitalWrite (LEDDer, LOW);
    digitalWrite (LEDIzq, LOW);
}

```

Programación correspondiente al Botón3, stop, por tanto todo va apagado.

```

if(estado=='d') {
    digitalWrite (MotorDerA, HIGH);
    digitalWrite (MotorDerB, LOW);
    digitalWrite (MotorIzqA, LOW);
    digitalWrite (MotorIzqB, LOW);
    digitalWrite (LEDDer, LOW);
    digitalWrite (LEDIzq, HIGH);
}

```

Programación correspondiente al Botón4, hacia la derecha, por tanto la LED derecha debe estar encendida.

```

if(estado=='e') {
    digitalWrite (MotorDerA, LOW);
    digitalWrite (MotorDerB, HIGH);
    digitalWrite (MotorIzqA, LOW);
    digitalWrite (MotorIzqB, HIGH);
    digitalWrite (LEDDer, LOW);
    digitalWrite (LEDIzq, LOW);
}

```

Programación correspondiente al Botón5, hacia la atrás, por deben encender ambas LED.

```

if(estado=='f') {
    digitalWrite (ptrig, LOW);
    delayMicroseconds (2);
    digitalWrite (ptrig, HIGH);
    delayMicroseconds (10);
    digitalWrite (ptrig, LOW);
    duracion = pulseIn(pecho, HIGH);
    distancia = (duracion/2) / 29;
    Serial.println(distancia);
}

```

Programación correspondiente al Botón6 encendido del Móvil bluetooth en modo automático.

```

if(distancia <=15 && distancia >=2) (
    digitalWrite (MotorDerA, LOW);
    digitalWrite (MotorDerB, LOW);
    digitalWrite (MotorIzqA, LOW);
    digitalWrite (MotorIzqB, LOW);
    digitalWrite (LEDDer, LOW);
    digitalWrite (LEDIzq, LOW);
    tone(2, 440, 100);
    delay (200);

    digitalWrite (MotorDerA, LOW);
    digitalWrite (MotorDerB, HIGH);
    digitalWrite (MotorIzqA, LOW);
    digitalWrite (MotorIzqB, HIGH);
    digitalWrite (LEDDer, LOW);
    digitalWrite (LEDIzq, LOW);
        delay (500);
    digitalWrite (MotorDerA, LOW);
    digitalWrite (MotorDerB, LOW);
    digitalWrite (MotorIzqA, HIGH);
    digitalWrite (MotorIzqB, LOW);
    digitalWrite (LEDDer, LOW);
    digitalWrite (LEDIzq, HIGH);
    tone (2, 440, 100);
        delay (1100);
        digitalWrite(13, LOW);
)
else(
    digitalWrite (MotorDerA, HIGH);
    digitalWrite (MotorDerB, LOW);
    digitalWrite (MotorIzqA, HIGH);
    digitalWrite (MotorIzqB, LOW);
    digitalWrite (LEDDer, HIGH);
    digitalWrite (LEDIzq, HIGH);
)
}
if(estado='g'){
    digitalWrite (MotorDerA, LOW);
    digitalWrite (MotorDerB, LOW);
    digitalWrite (MotorIzqA, LOW);
    digitalWrite (MotorIzqB, LOW);
    digitalWrite (LEDDer, LOW);
    digitalWrite (LEDIzq, LOW);
}
}

```

Programación correspondiente al Botón6, encendido del Móvil bluetooth en modo automático.

Programación correspondiente al Botón7, apagado del Móvil bluetooth.

31 Carga el *sketch* en tu Móvil bluetooth usando el botón  o da clic en el menú **Programa** y en **Subir**.

32 Energiza tu robot. Coloca el broche de barril en la Arduino y conéctale una pila de 9 V y coloca cuatro pilas AA en el portapilas inferior.

33

Enciende la conexión bluetooth de tu dispositivo y busca el dispositivo HC05 o HC06 y vincula tu robot al dispositivo.

Bluetooth

Su dispositivo (Galaxy J7) es actualmente visible para los dispositivos cercanos.

Dispositivos vinculados

 HC-06

KAISER

34

Abre la aplicación que creaste y da clic en el botón bluetooth, selecciona el HC05 o HC06. Ya estás listo para poner a prueba tu robot.

Robot Move Bluetooth



35 Pon en funcionamiento tu robot manejándolo con tu dispositivo móvil o en modo automático.

Robot

terminado



Alfaomega Grupo Editor

Te acerca al conocimiento

www.alfaomega.com.mx

Soporte escolar

e-mail: soporteescolar@alfaomega.com.mx

Si te resulta complicado cargar la programación, contáctanos en www.robotik-age.com