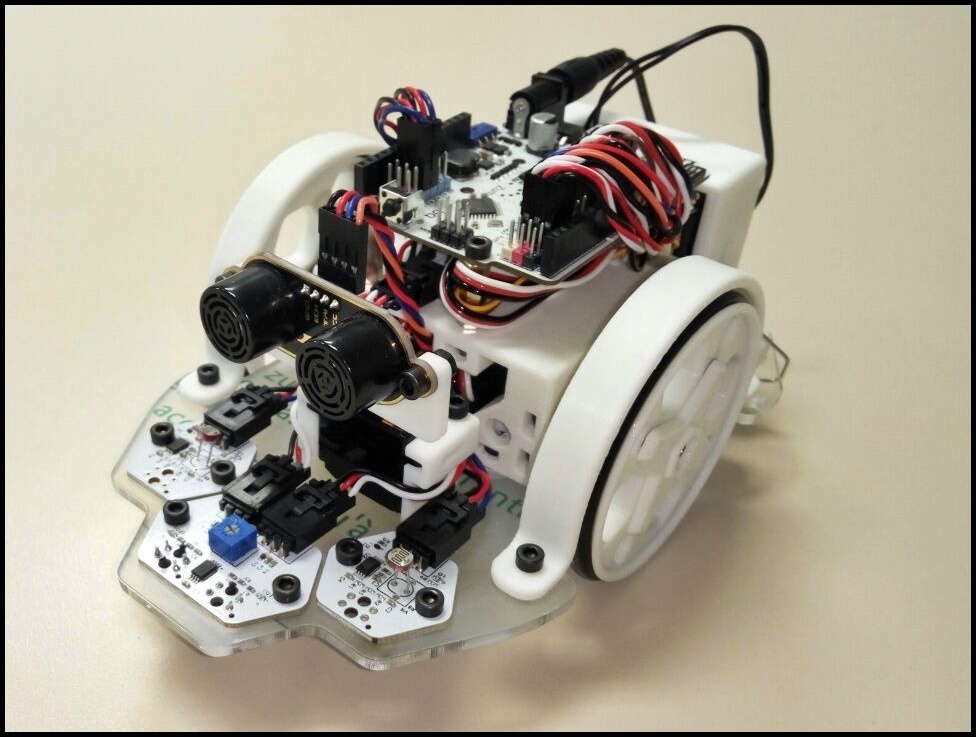
**PROYECTO CONTROL REMOTO**



**DESARROLLO APLICACIONES MULTIPLATAFORMA**

**ARMANDO CALDERÓN FREIRE & DAVID ROMÁN REY**

Tutor proyecto: José Luis Álvarez

**Curso Académico 2016-2018**

# Índice General

[Índice General 1](#_Toc516045347)

[Índice de imágenes 1](#_Toc516045348)

[Introducción 2](#_Toc516045349)

[1. Elementos del robot 3](#_Toc516045350)

[2. Metodología Scrum 4](#_Toc516045351)

[a. Primer Sprint (02/03/2018) 4](#_Toc516045352)

[b. Segundo Sprint (11/05/2018) 6](#_Toc516045353)

[c. Tercer Sprint (11/06/2018) 8](#_Toc516045354)

[d. Cuarto Sprint (Futuro) 12](#_Toc516045355)

[3. Manual de usuario 13](#_Toc516045356)

[a. Iniciar Sesión 13](#_Toc516045357)

[b. Crear cuenta 13](#_Toc516045358)

[c. Control Remoto 14](#_Toc516045359)

[4. Tecnologías usadas 16](#_Toc516045360)

[5. Conclusiones 16](#_Toc516045361)

[6. Código 17](#_Toc516045362)

[7. Bibliografía 19](#_Toc516045363)

# Índice de imágenes

[1 Piezas del Robot 3](#_Toc516045364)

[2 Placa bq zum bt-328 11](#_Toc516045365)

[3 Control Remoto 14](#_Toc516045366)

[4 Cambiar nombre bluetooth. 17](#_Toc516045367)

[5 Función Automático 17](#_Toc516045368)

[6 Avanzar 18](#_Toc516045369)

[7 Triángulo 18](#_Toc516045370)

# Introducción

**PRESENTACIÓN**

El proyecto es un **robot** de conducción manual y autónoma a través de servomotores. Este robot es alimentado por pilas y dirigido por una **aplicación móvil** para Android.

La aplicación se sincroniza mediante Bluetooth con diferentes opciones de control en el que antes tendremos que registrarnos para poder controlarlo.

**OBJETIVOS**

Nuestro objetivo era tener dos robots y que lucharan entre ellos, pero finalmente hemos pensado que mejor ver y controlar uno por nosotros mismos, y una vez terminado analizar y programar una gran variedad de opciones.

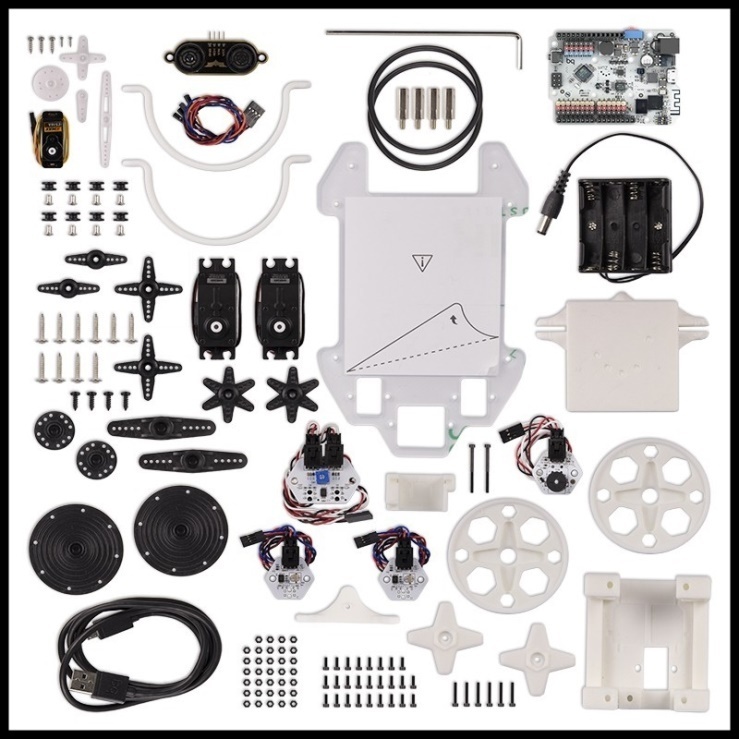
* **Dibujar figuras.**
  + Círculo.
  + Triangulo.
  + Cuadrado.
* **Utilización de sensor:**
  + Luz.
* **Movilidad de la cabeza y ruedas:**
* Motor servo.

**¿POR QUÉ HEMOS DECIDIDO HACER ESTE PROYECTO?**

Llevamos estudiando juntos el mismo ciclo desde hace 5 años y no podíamos perder esta oportunidad de poder hacer el proyecto de DAM juntos. Habíamos pensado en hacer un proyecto por separado muy parecido, pero en grado medio hicimos una página web y no queríamos volver a repetirlo y sabíamos que podíamos dar un paso adelante con algo nuevo.

También recordamos que en el grado medio nos trajeron un robot, el cual lo programo el profesor y nos dio bastante curiosidad, hasta que, con los conocimientos adquiridos en estos años, nos hemos atrevido hacer este proyecto.

# Elementos del robot



1 Piezas del Robot

El **pack del robot** está compuesto por la base donde apoyaremos la **placa Arduino**, tres **servos motores**, varios **sensores** (ultrasonido, infrarrojos, luz) **zumbador, conectores** y pines, dos ruedas, un portapilas, y un conjunto de piezas para testear el robot y poder hacer más formas con él.

Es un pack bastante completo en el que nos vienen todos los componentes para poder hacer un robot muy útil. Se le puede añadir más componentes desde una pinza hasta un mini bote de pintura, todo ello dependiendo de los conocimientos que tenga el usuario en programación y lo manitas que sea a la hora de modificar partes criticas como la placa base.

# Metodología Scrum

1. **¿Qué es la metodología Scrum y que es Trello?**

**Scrum** es un **proceso** en el que se aplica un conjunto de prácticas organizadas para **trabajar** en **equipo** y obtener el mejor **resultado** posible de un proyecto.

Para ello hemos utilizado **Trello:** es una herramienta de organización de proyectos online simple, en la que nos hemos repartido el trabajo según las fechas de entrega previstas. Dentro de esta herramienta puedes clasificar el trabajo en “**TO DO**”: tareas a realizar, “**DOING**”: tareas que se están haciendo, “**DONE**”: tareas ya hechas.

Estas tareas se dividen por tramos fijados (fechas), en las que en cada sprint hay que tener unos objetivos acordados y organizados al principio.

## Primer Sprint (02/03/2018)

* + **Buscar información sobre la placa y sus componentes.**
    - El robot tiene una placa llamada **ZUM BT 328** que lleva Bluetooth integrado, de esta forma nos facilitara poder sincronizarnos con cualquier tipo de control remoto. En el robot vamos a crear varias funciones para poder utilizarlo con nuestro mando, para ello hemos añadido una **librería** llamada **servo.h** para poder controlar nuestros servos (ruedas) de la propia página oficial de **Arduino**. El robot contiene un buffer para almacenar datos recibidos por el Bluetooth y el cual enviará los datos a un menú donde realizará las funciones que haya recibido anteriormente.
  + **Crear PowerPoint e integrar datos.**
    - Creamos el PowerPoint para este primer sprint en el que enseñamos el montaje del robot, las funciones, partes de la placa del robot y finalmente los próximos objetivos de cara al siguiente sprint.
  + **Fotos – Montaje del robot**

## Segundo Sprint (11/05/2018)

* + **Diseño del boceto de la aplicación (Control remoto).**
    - Creamos en papel y con bolígrafo el dibujo de lo queremos ilustrar en la aplicación. Dibujamos cada uno un boceto y después sacamos ideas hasta finalmente tener una principal.
  + **Crear aplicación con usuario/contraseña y registro.**
    - La aplicación finalmente la hemos hecho con Scratch, en ai2.appinventor.mit ya que comenzamos haciéndola en Xamarin y había muy poca información y documentación. Finalmente queríamos centrarnos más en la programación del robot y nos pasamos a Scratch que para nosotros era más rápido y útil.
    - En un principio queríamos crear un inicio de sesión simple que guardara el usuario y contraseña en el propio programa. El usuario podrá interactuar con una ventana de registro que no iba a tener ninguna funcionabilidad, la cual solo nos iba a servir para mostrarla en este sprint. Una vez dado con el boceto que queríamos, empezamos a trabajar con el inicio de sesión, primero con los campos de usuario, contraseña y el botón de iniciar sesión. Cuando comprobamos que realmente hacia lo correcto implementamos el control remoto como segunda ventana.
  + **Empezar documentación del proyecto y PowerPoint.**
    - Hacemos una estructura en papel de lo que queremos poner en el Word. Tenemos unas ideas las cuales contrastamos con el tutor para saber si nos va a faltar algo o se nos va a quedar corto.
    - Unimos las anteriores diapositivas con las creadas para este Sprint. Mejoramos el diseño de todo el PowerPoint (imágenes, fondos, letra), le hacemos un lavado de cara para que quede más serio ya que se nos sugiere cambiarlo.
  + **Crear aplicación con los controles para manejar el robot.**
    - Hemos creado un control remoto con 4 botones con la funcionabilidad de poder mover el robot hacia todos los lados.
  + **La conexión Bluetooh:**

Al principio no sabíamos que señal utilizaba el Bluetooth para la comunicación. Consultamos el manual de **Arduino** y vimos que el Bluetooth utilizaba la señal en **baudios**(que es la unidad de medida que se utiliza para la transmisión de datos en bits).

Primero hicimos un **Serial.Begin**(establece la velocidad de datos en bits por segundo) con la señal **9600 baudios** y un **Serial.print** para comprobar que esa señal era la correcta. Después abrimos el monitor serial, le indicamos con que señal estábamos trabajando para comprobar si recibía algún dato.

void setup() {

Serial.begin(9600);}

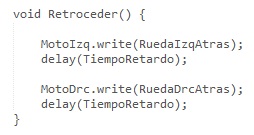
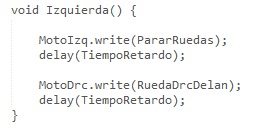
void loop() {

Serial.print("Recibiendo señal");}

Tras varias pruebas realizadas conseguimos dar con la señal correcta: **19200 baudios.**

Una vez conseguido la comunicación, vimos el nombre del Bluetooth y la contraseña con los comandos que tenía el módulo Bluetooth, por defecto trae el nombre del Bluetooth **zumbt-368** y la contraseña (**1234**).

* + **Crear funciones de movilidad para el robot.**

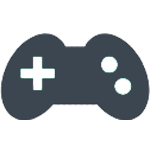
Creamos 4 funciones dentro de **Arduino** las cuales son avanzar, retroceder, izquierda y derecha.

Dentro de cada función tenemos los motores (derecha e izquierda) los cuales reciben el movimiento que les hayamos puesto, estos datos se envían como un byte o una serie de bytes. Como por ejemplo en **Retroceder**; envía al motor de la izquierda un byte con el valor de **RuedaIzqAtras**, el cual ha sido definido con valor 180.

Después mandamos un delay, que es el tiempo que queremos que dure ese proceso, en nuestro caso **TiempoRetardo** definido como 1000 (milisegundos).

## Tercer Sprint (11/06/2018)

* + **Conexión con autenticación y aplicación.**
    - En primer lugar, hay que crear un proyecto en **Firebase.** Dentro tiene un apartado llamado autentificación y tiene varias opciones para autentificarnos, nosotros hemos elegido que los usuarios se registren mediante un correo y una contraseña que el cliente elija.
    - Para poderlo aplicarlo en nuestra aplicación lo que tenemos que hacer primero es añadir un fichero Firebase en la aplicación y después coger nuestro token (ID o código) de Firebase y asignarlo a nuestra aplicación. Finalmente añadir nuestro token para que tengamos conexión.
    - Nuestro token sirve para poder verificar que añadimos a cada usuario, con esto a la hora de registrarnos se le asigna una id con ese token que luego se comprueba cuando iniciamos sesión.
  + **Crear icono para la aplicación.**
    - El icono de la aplicación se hizo con 3 imágenes y con Photoscape. Buscamos información de las dimensiones (120X120) y el formato que debía tener (.ico o .png) ya que no aceptaba cualquier formato.
    - Primero se hizo una base de color azul, después añadimos el mando al fondo y finalmente colocamos el icono de Bluetooh encima del mando.

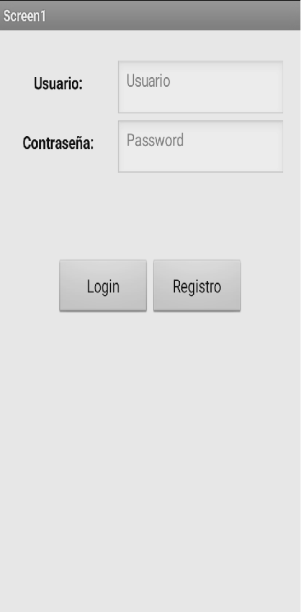
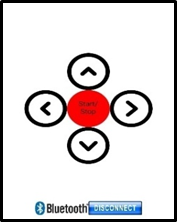
****

* + **Finalizar la aplicación.**

Lo primero que hicimos fue mejorar el entorno visual de la aplicación, que fuera más sencillo y practico hacia el usuario.

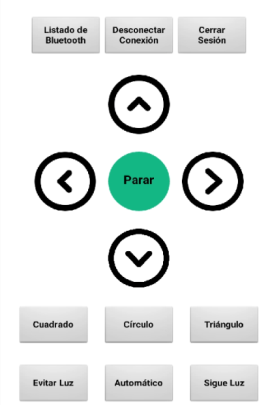
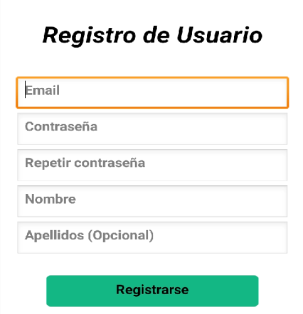
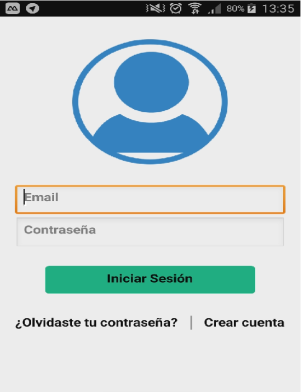
* + - **Aplicación antigua:**

En esta primera aplicación **priorizamos** tener los **objetivos**, que eran un inicio de sesión y unos botones simples pero los cuales funcionaran y conectaran con el robot.

** **

* + - **Aplicación nueva:**

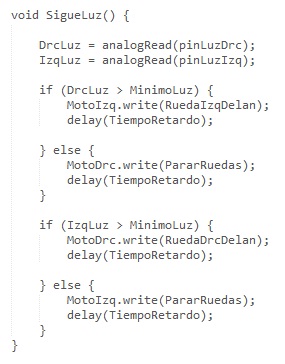
Para esta segunda aplicación queríamos darle un lavado de **imagen** y ya que teníamos una **aplicación funcional**, decidimos hacerla más seria y robusta con colores más llamativos y no tan pesados.



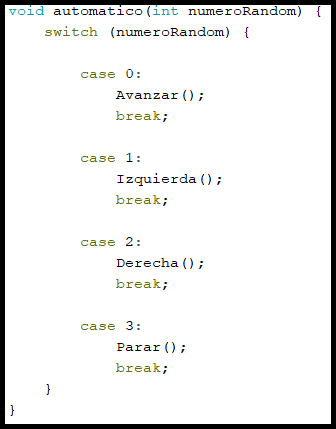
* + **Crear funciones para darle utilidad a los sensores de luz.**
    - Utilizamos la función **analogRead(pin):**

Asignamos a cada variable (DrcLuz o IzqLuz) lo que lee en los pines **pinLuzDrc** o **pinLuzIzq**.

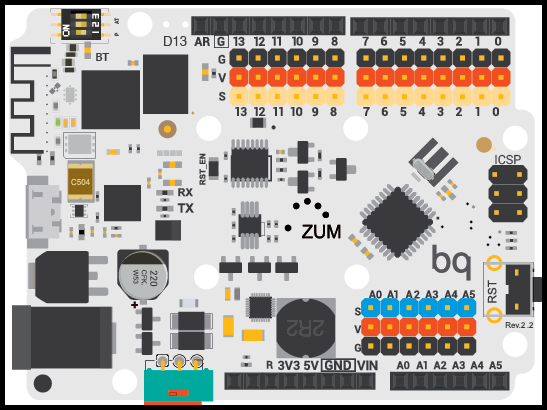
* + - Después creamos una variable con el mínimo de luz que tiene que recibir.

****

* + **Crear función robot automático.**
    - Creamos una función dentro de **Arduino** para que sin nosotros le demos órdenes, se vaya moviendo solo por el espacio.



* + **Cambiar nombre y contraseña a la placa.**



2 Placa bq zum bt-328

* + - **Pasos para modificar la placa.**

Para realizar el cambio lo que hay que hacer, es **levantar** los dos **pines** (2 - 3) y ponernos en **ON**.

Después conectamos la placa y abrimos **Arduino** y en **herramientas** seleccionamos monitor serie.

A continuación, dentro del **terminal** hay que cambiar los baudios a 19200 para sincronizarlo con el Bluetooth y ponerlo en modo **retorno de carro**.

Para hacer posible la modificación hay que utilizar los comandos siguientes:

* **AT + NAME**: para cambiar el nombre.
* **AT + PIN**: para cambiar la contraseña.
* **AT + HELP**: saca una guía de comandos disponibles.

Una vez finalizados estos pasos hay que volver a poner los pines en su estado inicial.

* + **Testeo del robot y de la aplicación.**
    - Realizamos pruebas de las funciones implementadas desde las formas (que las haga correctamente) hasta la función sigue luz, si esta coge luz de todos los ángulos o solo desde enfrente.
* **Finalizar documentación del proyecto.**
  + Finalizamos todo el Word, detallando algunos apartados y mejorando partes específicas que nos aconseja el tutor.

## Cuarto Sprint (Futuro)

* + **Funcionalidad con el sensor ultrasonido.**

Con este sensor nos gustaría utilizarlo para detectar objetos cuando se presente a su alrededor y lo esquive de forma autónoma.

* + **Funcionalidad con los sensores infrarrojos**.

Utilizar este sensor para detectar rallas por el suelo. Este robot tiene los infrarrojos hacia el suelo únicamente, asique limita mucho poder hacer cosas distintas a un sigue líneas.

* + **Mejoras en la aplicación.**

Implementar nuevas funcionalidades dentro del inicio de sesión y control remoto.

* + **Nuevas funciones en el robot.**

Preguntar al usuario una palabra y que el robot dibuje la palabra.

* + **Innovar en nuevas tecnologías para la aplicación**.

Hacer una página web y poder controlarlo desde el ordenador.

# Manual de usuario

## Iniciar Sesión

Una de las **funcionalidades** que tiene al iniciar sesión, es si el campo email está vacío, saltará una **notificación** diciéndonos que debemos rellenar el email para poder iniciar sesión, y si ese email no existiera nos avisara de que no existe tal correo.

Por otro lado, si el campo **contraseña** lo introduces mal nos notificará con un error de contraseña. Si nos hemos olvidado la contraseña, tenemos un botón llamado **olvidaste** tu contraseña, si pulsamos sobre él nos enviará un correo electrónico a nuestra cuenta de email para que podamos modificar nuestra contraseña.

En el botón iniciar sesión si todo está correcto, tanto el campo email y el campo contraseña, nos redirigirá a nuestro control remoto.

* La **función** que tiene el campo **olvidar contraseña** es la siguiente:
  + Tenemos un **autentificador** de correos que nos comprueba si ese correo ya se ha registrado en su base de datos y después le envía el correo para modificar la **contraseña**.

## Crear cuenta

Una vez que estamos en la pantalla de iniciar sesión tenemos un botón llamado **Crear cuenta**.

Dentro de nuestro registro tenemos varios campos obligatorios:

* **Email:** debemos introducir un email correcto.
* **Contraseña:** la **contraseña** no tiene unos **requisitos** mínimos puedes poner letras, números, caracteres especiales, pero tiene que ser máximo de 6 dígitos.
* **Repetir contraseña:** la contraseña tiene que ser idéntica a la que has escrito anteriormente.
* **Nombre:** el campo nombre es **necesario** rellenarlo. Si estos campos no se rellenan nos avisará con una notificación diciéndonos que debemos rellenar los campos obligatorios.

El campo apellidos es un campo opcional, en nuestro caso no le damos importancia porque solo es necesario registrarnos con un email y una contraseña.

Una vez rellenado todos los campos obligatorios, pulsamos en el botón **registrarse** y se agregara a nuestra base de datos (**Firebase**). A partir de aquí, ya podemos iniciar sesión con nuestro email y contraseña.

## Control Remoto

Para poder comunicarnos con nuestro robot tenemos que activar nuestro **Bluetooth** primero, una vez buscado nuestro dispositivo llamado (**Proyecto-Robot**) nos pedirá una contraseña que es (**arda**). Una vez hecho esos pasos nos metemos a nuestro control remoto y pulsamos en el botón llamado **listado de Bluetooth**. Nos aparecerá nuestro robot, en el tendremos disponibles varias funcionalidades.

Una vez conectado con el robot, podremos utilizar el controlador de movimientos que son:

* **Avanzar.**
* **Parar.**
* **Retroceder.**
* **Izquierda.**
* **Derecha.**

También podemos enviar funciones al robot para que las realice como:

* **Cuadrado.**
* **Triángulo.**
* **Evitar la luz.**
* **Sigue la luz.**
* **Círculo.**
* **Automático.**

Si queremos desconectarnos del robot hay un botón que se llama **desconectar conexión:** una vez pulsado dejaremos de sincronizarnos con el robot y no emitirá ninguna señal. También podemos cerrar sesión y salirnos del control remoto.



3 Control Remoto

¿Cómo funciona el robot?

El robot tiene dos modos de manejarlo **automático y manual**:

* **Automático:** para poner el modo tenemos que estar en la aplicación y pulsar en el botón de automático y el robot se moverá de **forma aleatoria** con cuatro **movimientos**; adelante, retroceder, izquierda y derecha. En cuanto pulsamos en el botón de parar el robot sale del modo automático y se queda esperando a la siguiente instrucción.
* **Manual:** si el robot nos lo encontramos parado, estará por defecto en manual y lo podemos **controlar** nosotros con los **botones** de control.

También tenemos varias funciones para que el robot haga figuras como (Circulo, Triangulo y Cuadrado).

**Los movimientos de los servos:**

Nuestros servos (Motores de las ruedas) se mueven mediante los **grados** asignados cuando lo estas programando. Aquí ponemos un ejemplo simple:

#include <Servo.h>

ServoservoIzq;

ServoservoDrc;

voidsetup ()

{

servoIzq.attach(8); // Pin rueda izquierda.

servoDrc.attach(9); // Pin rueda derecha.

}

voidloop ()

{

servoIzq.write(45); // Girar la rueda izquierda 45 grados.

delay(1000); //Espera 1 milisegundo.

servoDrc.write(45); //Girar la rueda derecha 45 grados.

delay(1000); // Espera 1 milisegundo.

Los servos normalmente giran 180 grados con el punto medio de 90 grados en la posición central, y se pueden colocar en cualquier punto intermedio.

# Tecnologías usadas

|  |  |  |
| --- | --- | --- |
| **NOMBRE** | **LENGUAJE** | **USO** |
| **ARDUINO** | **C++** | **Plataforma basada en una placa diseñada para facilitar el desarrollo de proyectos electrónicos.** |
| **FIREBASE** | **SQL** | **Plataforma de desarrollo móvil que permite desarrollar aplicaciones multiplataforma con datos en nube.** |
| **MY APP INVENTOR** | **SCRATCH** | **Creación de aplicaciones en el entorno Android.** |
| **TRELLO** | **-** | **Herramienta de gestión de proyectos online, diseñada para metodología ágil.** |

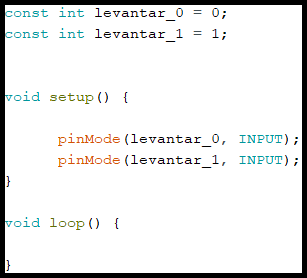
# Conclusiones

Aparentemente habíamos pensado en que lucharan dos robots dentro de un área, y programarlos para después poder controlarlos. Pero la idea finalmente era acabar uno y ver el resultado final y ver que funcionalidades se le puede añadir.

Para nosotros el resultado obtenido es bastante satisfactorio, ya que uno de los objetivos principales del proyecto era empezar un proyecto nuevo y diferente para nosotros, con tecnologías que habíamos dado muy poco y poder enseñar un robot con su aplicación móvil.

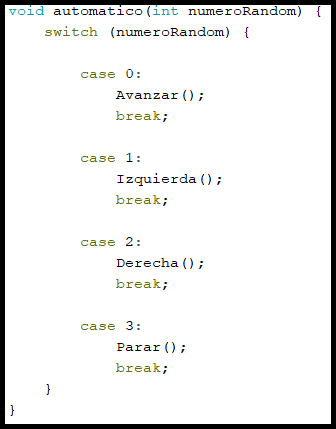
Para un futuro tenemos pensado seguir mejorando el robot y que pueda hacer nuevas funciones. Creemos que con el trabajo realizado hemos aprendido bastante y podemos seguir innovando como una función que hemos pensado que es que el usuario introduzca una palabra y el mismo robot la dibuje sin nosotros dar órdenes.

# Código



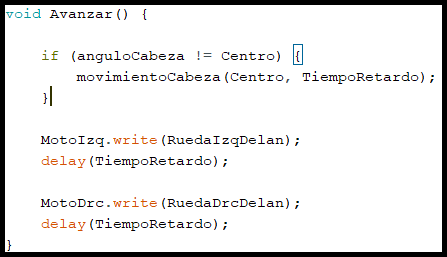
Para poder acceder a la placa debemos crear primero dos variables, indicando los pines que estén vacíos. En nuestro caso, utilizamos el 0 y 1 en los cuales no tenemos nada conectado. Dentro del **setup**, asignamos dos variables con entrada de datos.

4 Cambiar nombre bluetooth.



En la función **automático** tenemos unos casos que dependiendo del número que recibamos realizara una **función** concreta.

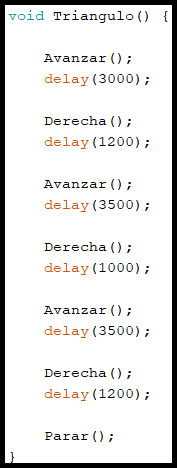
5 Función Automático



Esta es una de nuestras funciones en las que se establece el **giro** de **cabeza** y la **dirección** del robot.

Para calcular el ángulo de la cabeza le indicamos al servo que se coloque en 80 grados.

6 Avanzar



En esta función lo que queremos conseguir es que el robot nos dibuje un **triángulo.**

Para ello, usamos las funciones de dirección y calculamos el tiempo de ejecución para hacer posible el dibujo.

7 Triángulo

# Bibliografía

* **Manual de Arduino:**

<https://arduinobot.pbworks.com/f/Manual+Programacion+Arduino.pdf>

* **Manual de Scratch:**

<http://www.areatecnologia.com/informatica/scratch-2-tutorial-online.html>

* **Librería Servo añadida a nuestro robot:**

<https://github.com/arduino-libraries/Servo/blob/master/src/Servo.h>

* **Manual para la conexión entre la aplicación y la base de datos de Firebase**

<https://rominirani.com/tutorial-mit-app-inventor-firebase-4be95051c325>

* **Manual para reemplazar la contraseña olvidada mediante Firebase:**

<https://firebase.google.com/docs/reference/rest/auth/?hl=es-419#section-change-password>

* **Manual configuración servos.**

<http://www.hobbytronics.co.uk/arduino-tutorial2-servos>

* **Manual Serial.Begin Arduino.**

<https://playground.arduino.cc/ArduinoNotebookTraduccion/Serial>

* **Comandos para la configuración del bluetooth.**

<http://diwo.bq.com/wp-content/uploads/2014/11/BLK-MD-BC04-B_AT-COMMANDS.pdf>