

# IME

Diseño de vehículo/Juguete didáctico  
controlado por voz

Marcos Daniel García Cruz

Universidad Nacional Autónoma de México  
Facultad de Estudios Superiores Cuautitlán

Asesor: David Tinoco Varela

## Abstract

---

Hoy en día, la tecnología y su desarrollo juega un papel muy importante en prácticamente cualquier campo o área, incluyendo nuestra vida cotidiana. Obviamente, un deseo ideal sería el que cada persona o sector poblacional pudiera interactuar con ella sin limitantes, y más aún, que pudiera desarrollarla. Lamentablemente, existen diversos sectores poblacionales que no pueden interactuar en la misma forma con los elementos tecnológicos, como un juguete por ejemplo, debido a que pueden tener capacidades diferentes.

En este proyecto se describe el desarrollo en hardware y software de un vehículo que puede ser considerado un juguete controlado por voz, permitiendo que personas que puedan tener deficiencias motrices, puedan interactuar con él de forma natural y sencilla.

Adicionalmente, se describe todo el proceso, con la intención de alentar a sectores poblacionales jóvenes al desarrollo de proyectos, partiendo de la idea de que si se les elimina el miedo a la realización de algo “difícil”, serán motivados a la generación de diferentes tipos de prototipos.

## Introducción

---

Es bien sabido que los juguetes son una parte significativa en el desarrollo durante la infancia, estos permiten desarrollar diferentes aspectos como lo son la atención, motricidad, memoria, la coordinación, la creatividad, etc. Desafortunadamente la mayoría de los juguetes no son adaptados a los niños que cuenten con algún tipo de discapacidad, por lo que, experimentan mayor restricción y mayores dificultades para poder desarrollarse a través del juego, ya sea porque no puedan sujetarlos, saber como funcionan, no cuentan con formas que reconozcan fácilmente, las instrucciones llegan a ser un poco difíciles de entender, los colores no pueden ser los adecuados, etc.

No es una sorpresa que la tecnología se ha convertido en una parte muy importante en la vida diaria en la actualidad, es difícil pensar que las personas no tengan

habilidad tecnológica, el problema de que la tecnología se encuentre muy presente en la sociedad actualmente y avance muy rápido es que a las personas discapacitadas les cuesta comprender y adaptarse a las diferentes herramientas tecnológicas con las que se cuentan hoy en día.

Afortunadamente, los juguetes y la tecnología se han relacionado dando lugar al desarrollo de juguetes adaptados para aquellos niños con capacidades diferentes, apoyando así su desarrollo en diferentes habilidades cognitivas, por ejemplo:

- Nivel de excitación: El estar alerta y/o interesado a estímulos ambientales.
- Orientación: Identificar personas, lugares, horarios y situaciones.
- Reconocimiento: Identificar rostros, formas y materiales a los cuales hayan estado expuestos anteriormente.
- Concentración: Enfocarse en una tarea o actividad en un lapso.
- Memoria: Codificar, almacenar y recuperar información del pasado.
- Secuenciación: Ordenar información, conceptos y acciones.
- Formación de conceptos: Formular pensamientos e ideas a partir de una experiencia previa.
- Resolución de problemas: Ante una problemática, formar planes que puedan resolver dicha problemática.

En el presente proyecto, se pretende motivar a los jóvenes a desarrollar un vehículo controlado por voz, viéndolo más como un juguete esperando a ser armado. De esta manera, se trata de atacar dos problemas, por un lado, se plantea el uso de controles no convencionales para el manejo de un juguete, buscando que personas con capacidades diferentes puedan interactuar con ellos en diferentes formas; por otro lado, se busca generar interés en las mentes jóvenes en los ámbitos de electrónica y programación. Obviamente, el prototipo aquí presentado, es una base que puede ser modificado a presentaciones particulares.

Para el diseño de este proyecto, se ocuparán herramientas que sean amigables y eficientes al usuario, como lo son *Arduino* y *App Inventor*, así como conexiones básicas de electrónica. Estas herramientas están al alcance de cualquier interesado por lo que esperamos que sea una motivación adicional para motivar al desarrollo.

## Desarrollo del proyecto

---

Antes de empezar a desarrollar el proyecto, hay que conocer las herramientas que se utilizarán. Para que el lector pueda irse familiarizando con ellas.

### ¿Qué es Arduino?

Arduino es una plataforma de desarrollo cuyo elemento principal es una placa electrónica de *hardware libre*, se compone de un microcontrolador reprogramable y una serie de pines hembra, para permitir conexiones entre el microcontrolador y los distintos dispositivos electrónicos que le permiten interactúa con el entorno tal como sensores y actuadores de manera muy sencilla y simplificada. En la figura 1, se muestran algunos de los diferentes tipos de placas Arduino que existen:



Figura 1. Tipos de placas Arduino.

Una de las principales ventajas con las que cuenta Arduino, es su facilidad de manejo, lo que ha generado una comunidad internacional enorme, dispuesta a siempre ayudar con ideas y código que fluye de manera libre por Internet, adicionalmente podemos mencionar que es una placa multiplataforma, su lenguaje de programación no es complicado, los precios son accesibles y es reutilizable. El proyecto Arduino, puede encontrarse en el siguiente link:

<https://www.arduino.cc/>

### ¿Qué es App Inventor?

App Inventor es un entorno que permite el desarrollo de aplicaciones para dispositivos *Android*, para poder utilizarlo y generar aplicaciones, solo necesitamos un navegador web y un dispositivo que contenga o emule el sistema operativo *Android*, cuenta con una gestión de proyectos en la Nube, lo que nos permite guardar los códigos generados e irles dando seguimiento a estos mismos. Una de las grandes ventajas de esta herramienta, es la forma de programación, ya que esta se realiza de forma gráfica, lo que permite que hasta un programador no experimentado pueda explotar sus virtudes. El siguiente link nos lleva hacia el proyecto principal de App Inventor.

<https://appinventor.mit.edu/>

Dentro del entorno, nosotros encontramos:

- *App Inventor Designer*: En esta, se construye la interfaz de usuario, eligiendo y situando los elementos con los que interactúa el usuario y los componentes que forman la aplicación.
- *App Inventor Blocks Editor*: En este apartado se definirá el comportamiento de los diferentes componentes que conforman la aplicación.

Esos elementos pueden ser observados en la figura 2.

### ¿Qué es el reconocimiento de voz?

El reconocimiento de voz es la capacidad que tienen ciertos programas y máquinas que les hacen posible identificar palabras o frases completas habladas y codificarlas de tal manera que la máquina y/o programa sea capaz de entender.

Generalmente, se requiere un proceso de aprendizaje para que el programa sepa interpretar las palabras o frases que se le indican para realizar determinadas acciones, previamente definidas.

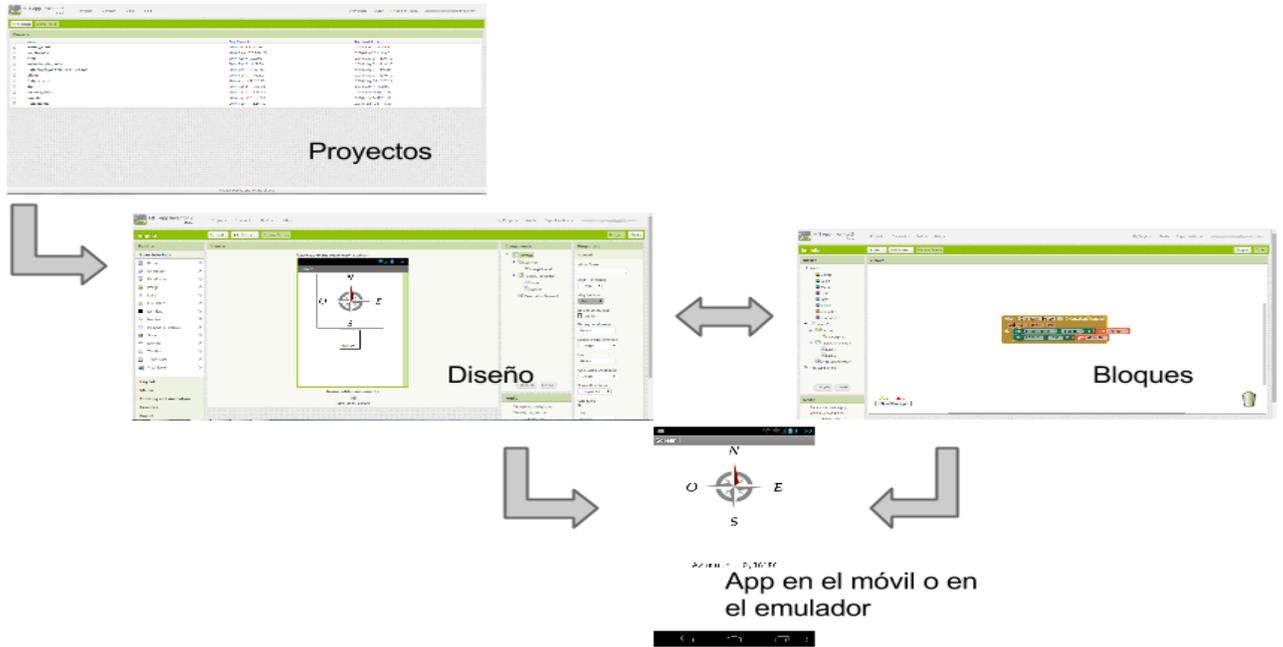


Figura 2. Programa App Inventor y sus interfaces.

El reconocimiento de voz, es un área computacional en constante crecimiento, aunque existen diversos sistemas modernos que pueden identificar varias palabras o frases largas, tal como Alexa, Siri, Cortana, etc.

Algunos usos y aplicaciones que puede tener el reconocimiento de voz son:

- Realizar un texto por medio del dictado.
- Generar una clave de voz para proteger datos.
- Controlar dispositivos a distancia.

### **Materiales y Equipo**

Los materiales que se utilizaron para la elaboración del proyecto son los siguientes, de igual manera se informa del costo aproximado del material (A Septiembre del 2021):

- Equipo:
  - Una computadora que cuente con conexión a internet y el IDE de trabajo de Arduino.
  - Pistola de silicón.
  - Cautín.

- Material:
  - Una placa protoboard (\$50 pesos).
  - Una placa Arduino Uno (\$150-\$300 pesos).
  - Un módulo bluetooth HC-05 (\$100 pesos).
  - Un regulador de voltaje LM7805 (\$40 pesos).
  - 2 motorreductores de doble eje tipo I de 3Vcc (\$49 pesos c/u).
  - Llantas para motorreductores (\$65 pesos).
  - Una rueda loca (\$29 pesos).
  - Un puente H L293d (\$45 pesos).
  - Una pila de 9V (\$60 pesos)
  - Cable conductor.
  - Cables macho-macho
  - Palitos de paleta.
  - Barras de silicón.
  - Liga.

El costo aproximado total del proyecto es \$700 pesos mexicanos.

### Construcción del vehículo

Aunque la idea principal es la realización de un juguete tecnológico e inclusivo con personas de capacidades diferentes, la falta de mayor recurso nos permitió generar solo el prototipo sin un diseño industrial adecuado, sin embargo, esto puede posteriormente ser resuelto por medio de una impresión 3D.

A continuación se describen los pasos realizados para la construcción del prototipo.

1. Recortar 4 palitos de paleta, cortando los laterales de manera tal que sea del mismo tamaño de la *protoboard*, como se ve en la figura 3.
2. Pegar con la ayuda de la pistola de silicón los motorreductores a las bases creadas de los palitos de paleta y soldar los cables con ayuda de un caufín, en la forma que se observa en la figura 4.
3. Acoplar las llantas a los motorreductores. Este es un proceso sencillo, ya que ambas partes están diseñadas para acoplarse automáticamente, como se ve en la figura 5.



Figura 3. Base de las llantas.

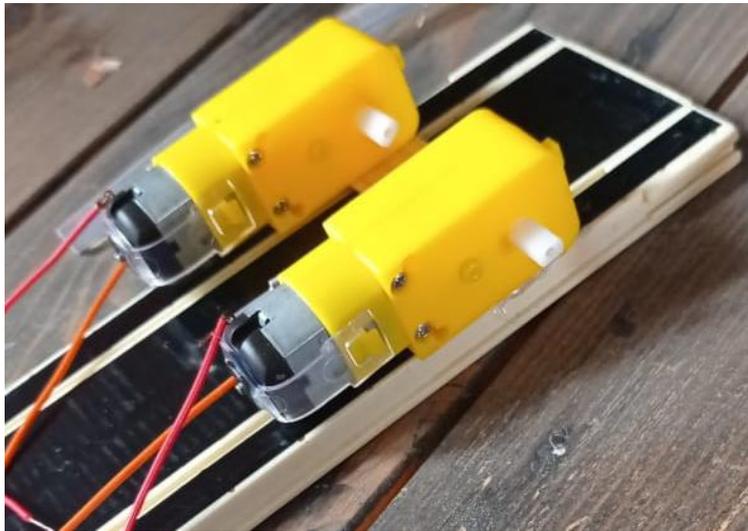


Figura 4. Motorreductores del vehículo.

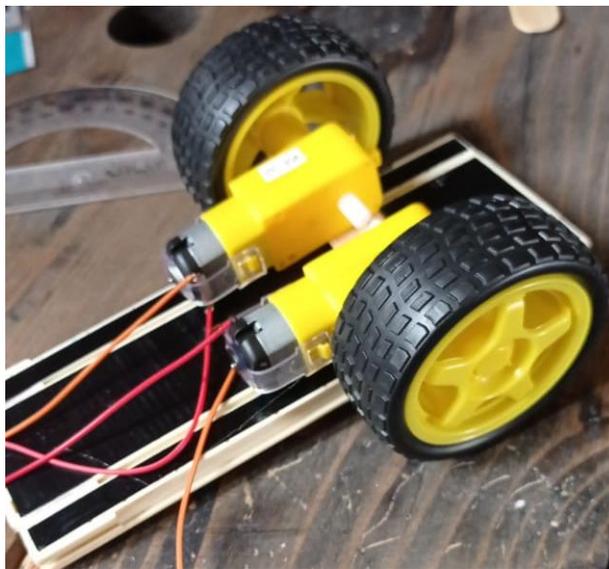


Figura 5. Acoplamiento de las llantas y los motorreductores.

4. Pegar los palitos cortados de base, pegarlos a la protoboard y pegar la rueda loca como se muestra en la figura 6.

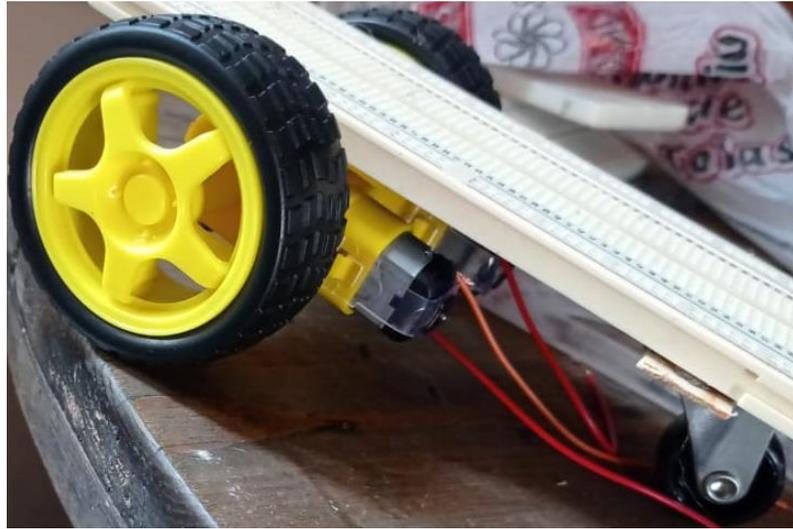


Figura 6. Rueda loca colocada en la parte frontal de la protoboard.

5. Con la ayuda de una liga, acomodar la placa Arduino a la protoboard, como en la figura 7.

**Nota:** Puede usar una placa de cualquier tipo (uno, nano, etc.).

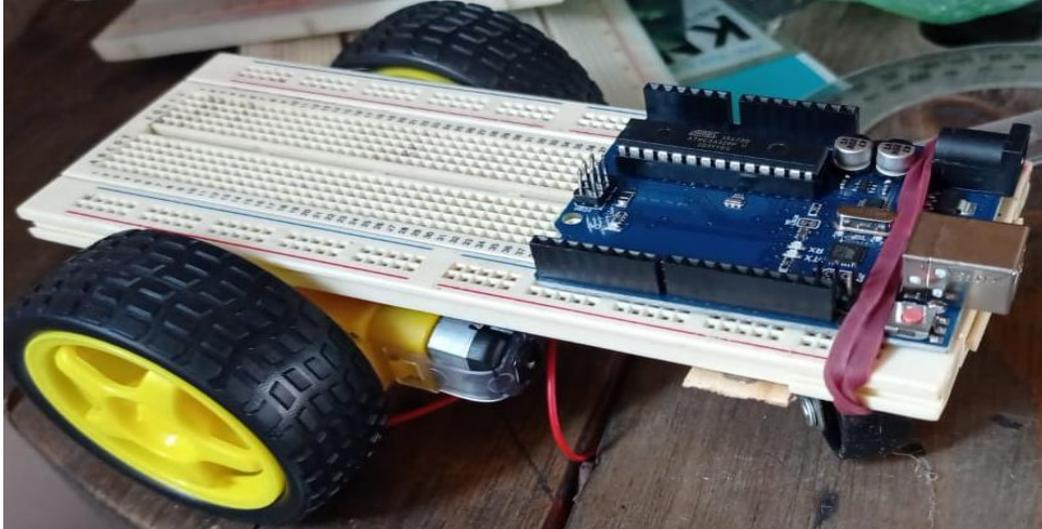


Figura 7. Placa Arduino ajustada a la protoboard por medio de una liga.

6. Colocar el regulador de voltaje y el módulo bluetooth de manera que se puedan realizar las conexiones a la placa Arduino.
7. En la figura 8, se muestra el esquema de conexión del puente H al Arduino. Realizar dicho diagrama en la protoboard, como se ejemplifica en la figura 9. El puente H brindará la potencia suficiente para poder echar a andar los

motores si no se tiene un puente H, puede ser sustituido por alguna otra fuente de corriente.

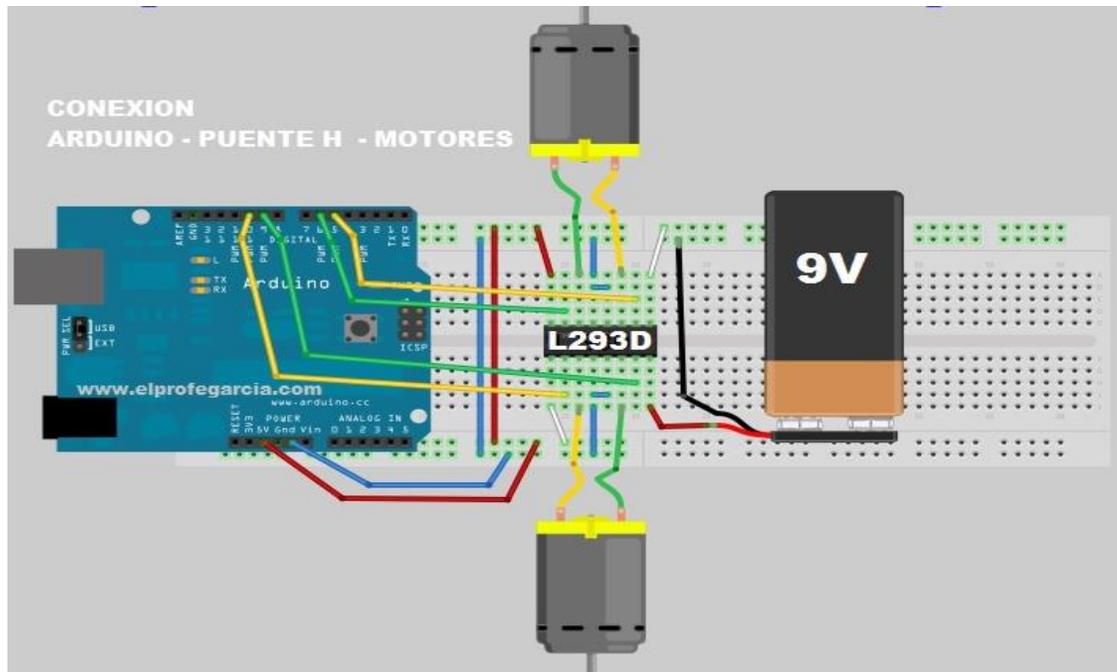


Figura 8. Diagrama de conexión del puente H.

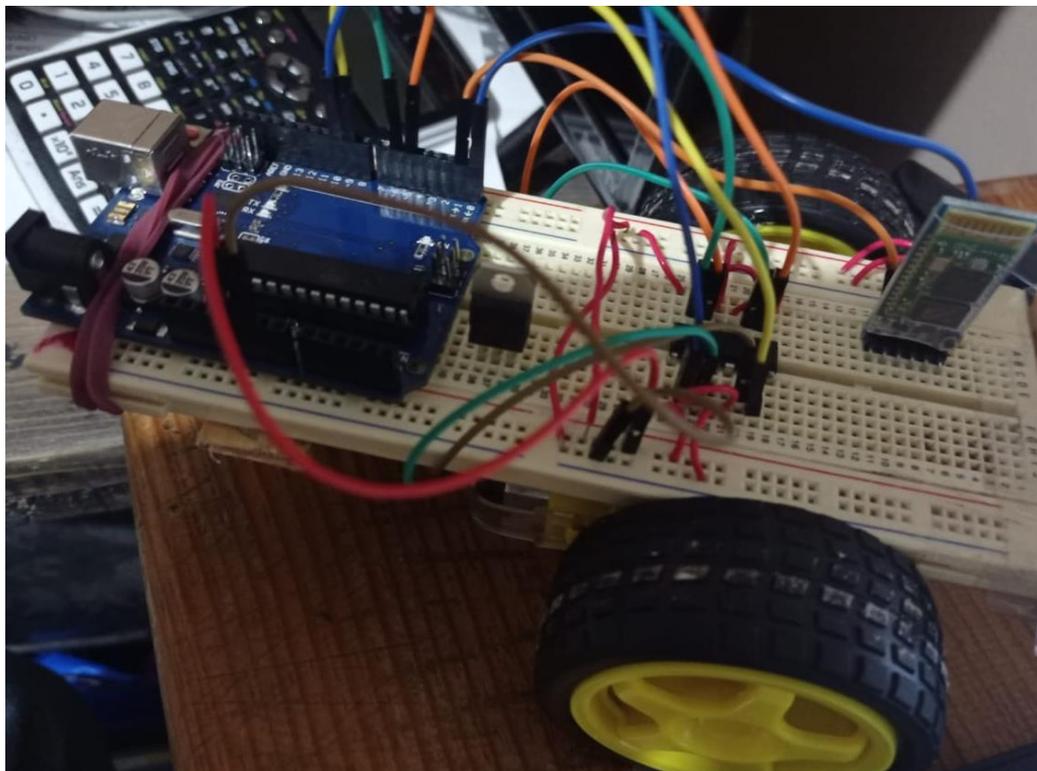
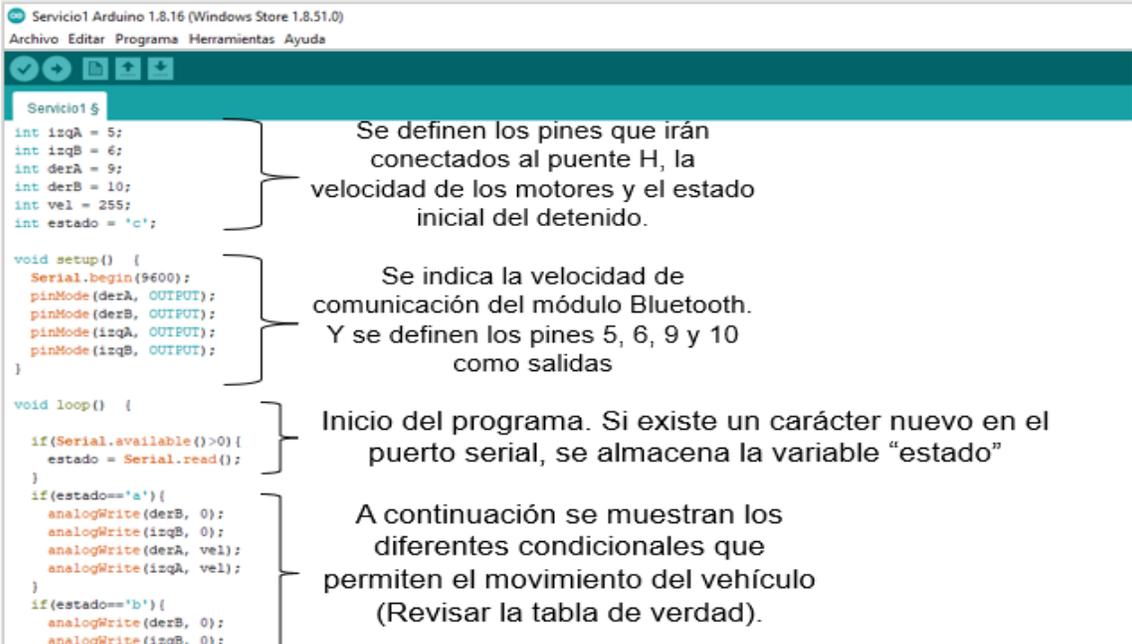


Figura 9. Conexiones del puente H al Arduino.

## Programación del Arduino

Como se mencionó desde el inicio, la programación utilizada para el desarrollo de este proyecto, es una programación sencilla y fácil de seguir, se realiza en dos partes, programación sobre la tarjeta y sobre el dispositivo de control.

A continuación, se muestra el programa utilizado para la programación del vehículo:



```
Servicio1 $
int izqA = 5;
int izqB = 6;
int derA = 9;
int derB = 10;
int vel = 255;
int estado = 'c';

void setup() {
  Serial.begin(9600);
  pinMode(derA, OUTPUT);
  pinMode(derB, OUTPUT);
  pinMode(izqA, OUTPUT);
  pinMode(izqB, OUTPUT);
}

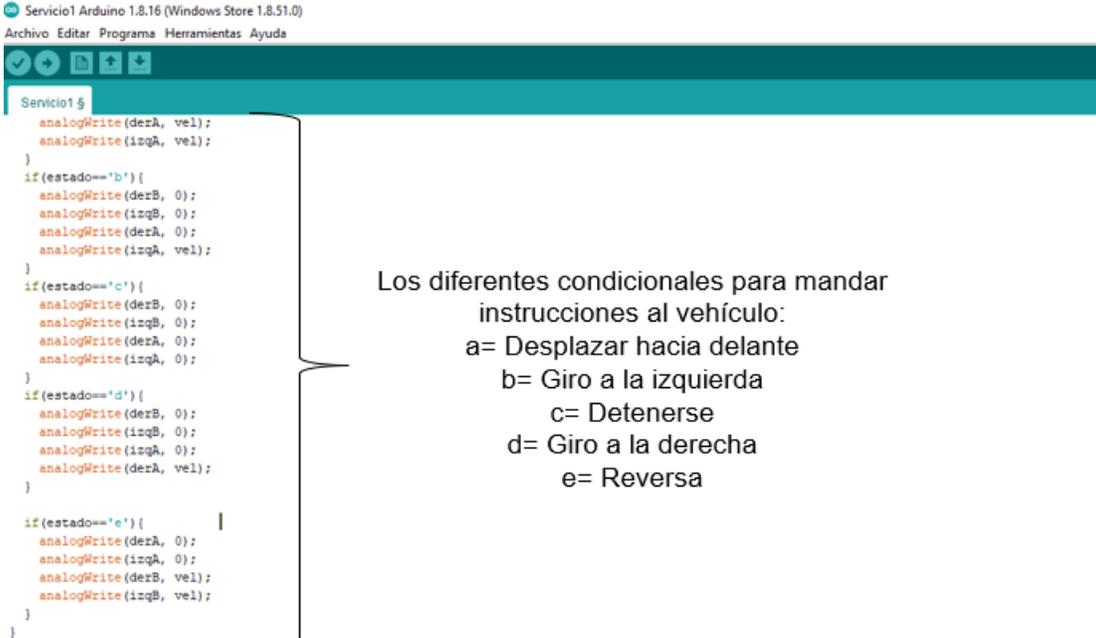
void loop() {
  if (Serial.available() > 0) {
    estado = Serial.read();
  }
  if (estado == 'a') {
    analogWrite(derB, 0);
    analogWrite(izqB, 0);
    analogWrite(derA, vel);
    analogWrite(izqA, vel);
  }
  if (estado == 'b') {
    analogWrite(derB, 0);
    analogWrite(izqB, 0);
  }
```

Se definen los pines que irán conectados al puente H, la velocidad de los motores y el estado inicial del detenido.

Se indica la velocidad de comunicación del módulo Bluetooth. Y se definen los pines 5, 6, 9 y 10 como salidas

Inicio del programa. Si existe un carácter nuevo en el puerto serial, se almacena la variable "estado"

A continuación se muestran los diferentes condicionales que permiten el movimiento del vehículo (Revisar la tabla de verdad).



```
Servicio1 $
    analogWrite(derA, vel);
    analogWrite(izqA, vel);
  }
  if (estado == 'b') {
    analogWrite(derB, 0);
    analogWrite(izqB, 0);
    analogWrite(derA, 0);
    analogWrite(izqA, vel);
  }
  if (estado == 'c') {
    analogWrite(derB, 0);
    analogWrite(izqB, 0);
    analogWrite(derA, 0);
    analogWrite(izqA, 0);
  }
  if (estado == 'd') {
    analogWrite(derB, 0);
    analogWrite(izqB, 0);
    analogWrite(izqA, 0);
    analogWrite(derA, vel);
  }
  if (estado == 'e') {
    analogWrite(derA, 0);
    analogWrite(izqA, 0);
    analogWrite(derB, vel);
    analogWrite(izqB, vel);
  }
}
```

Los diferentes condicionales para mandar instrucciones al vehículo:

- a= Desplazar hacia delante
- b= Giro a la izquierda
- c= Detenerse
- d= Giro a la derecha
- e= Reversa

El puente H será el encargado de controlar los motores, como se ve en el código anterior, se definen diferentes formas de funcionamiento de las llantas para realizar diferentes movimientos y giros, estos movimientos están definidos por la tabla de verdad del puente H, esta se observa en la tabla 1.

Tabla 1. Tabla de verdad del puente H L293D.

derB	izqB	derA	izqA	
0	0	0	0	Parar
0	0	1	1	Al Frente
0	0	0	1	Giro Izquierda
0	0	1	0	Giro Derecha
1	1	0	0	Reversa

**Nota:** Para poder cargar el código al Arduino desde la computadora, necesita estar desconectadas las terminales TX y RX del módulo Bluetooth, posteriormente se conectan en la forma mostrada en la tabla 2:

Tabla 2. Conexiones del módulo Bluetooth al Arduino y protoboard.

Arduino	Bluetooth
0 (RX)	TX
1 (TX)	RX
Protoboard	
Negativo	GND
Positivo	VCC

Para comprobar que el circuito está bien conectado y no cuenta con cortocircuitos, conecte la pila de 9v en las terminales V1 y GND del Arduino, si enciende el LED indicador "ON" del Arduino y el LED indicador del módulo Bluetooth, quiere decir que el circuito funciona correctamente, tal como se ve en la figura 10.

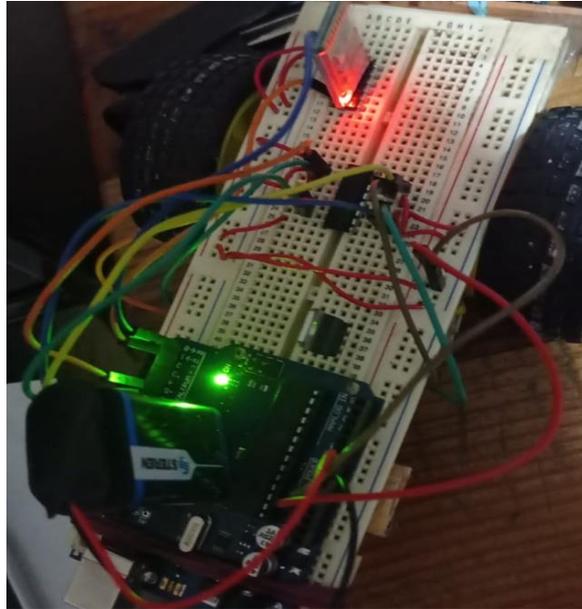


Figura 10. Circuito funcionando correctamente.

### Creación de la aplicación de control por voz

En este punto se va a describir la forma de programación de nuestra aplicación en Android. A continuación se describe cada paso:

1. Para la elaboración de la aplicación de control de voz, abrimos el siguiente enlace en nuestro navegador: <http://ai2.appinventor.mit.edu/> en el iniciamos un nuevo proyecto mediante “*Start new Project*”, como se ve en la figura 11.

**Nota:** Debe de contar con una cuenta de Google para poder trabajar.

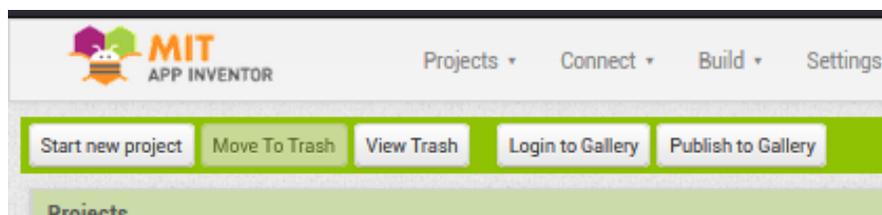


Figura 11. Inicio de nuevo proyecto en App Inventor.

2. Nombrar el proyecto a realizar. Hay que definir el nombre deseado para el proyecto, en la ventana que se muestra en la figura 12.

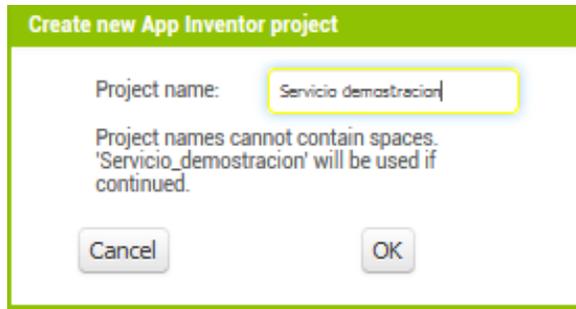


Figura 12. Nombramiento del proyecto.

3. Aparecerá la interfaz de la figura 13, aquí se diseñará la aplicación, puede seleccionar el tamaño de acuerdo con el tamaño del dispositivo (Figura 14) con el que controlará el vehículo/juguete.

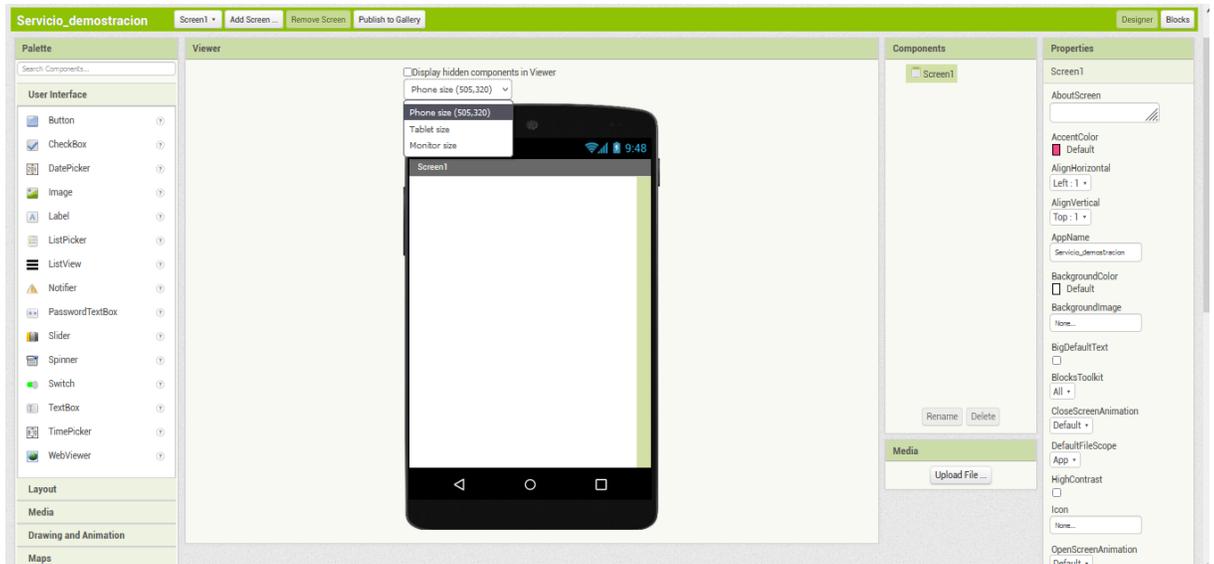


Figura 13. Interfaz de diseño de App Inventor.

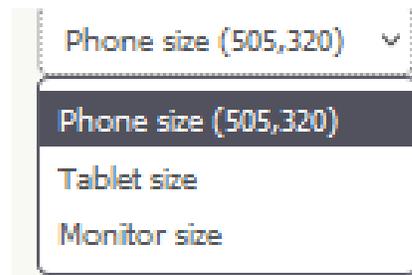


Figura 14. Ventana de selección de tamaño de vista de la app.

4. Añada los elementos de conexión Bluetooth “BluetoothClient” que puede obtener de la pestaña lateral izquierda llamada “Connectivity”,

“SpeechRecognizer” de la pestaña “Media” y “Clock” de la pestaña “Sensors”, arrastrándolos hacia dentro del dispositivo, ver figura 15.

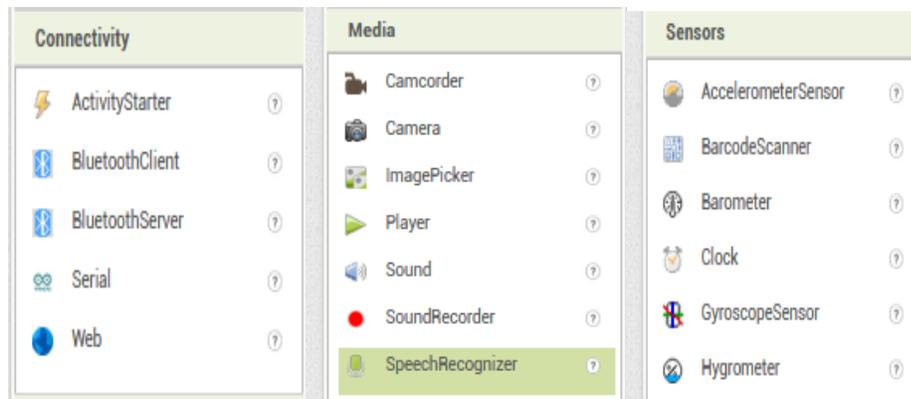


Figura 15. Elementos.

5. Proceda a añadir los botones para abrir el micrófono y apagar la conexión Bluetooth, se encuentran en la pestaña “Use Interface” con el nombre de “Button”, añade un “ListPicker” (Tiene como función ver los dispositivos Bluetooth a los cuales conectarse), añade un “Label” (En la pestaña lateral derecha puede modificar las propiedades de dichos botones, ya sea su tamaño, color, texto, etc.), tal como se ve en la figura 16.

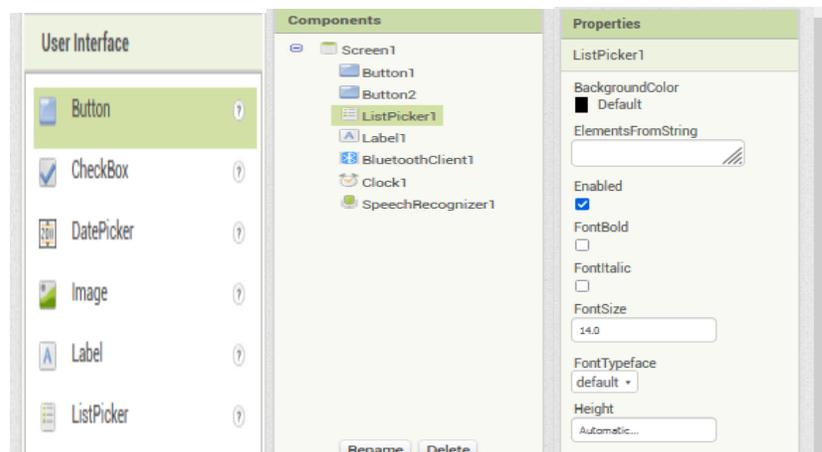


Figura 16. Ventanas de los elementos de la aplicación.

6. La pantalla con los elementos colocados en los pasos anteriores queda como se muestra en la figura 17. En la figura 18 se muestra la aplicación que servirá para explicación de la programación de bloques.

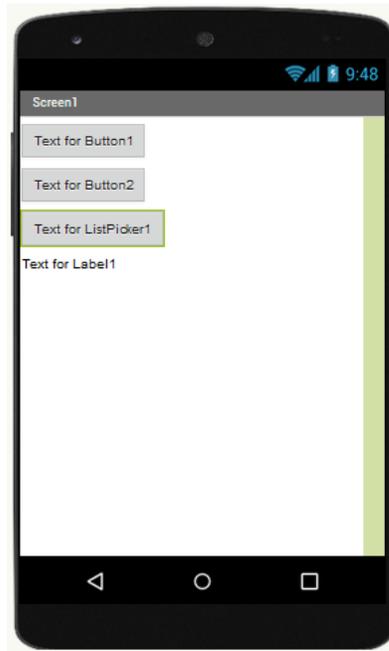


Figura 17. Pantalla del diseño de la aplicación desarrollada.

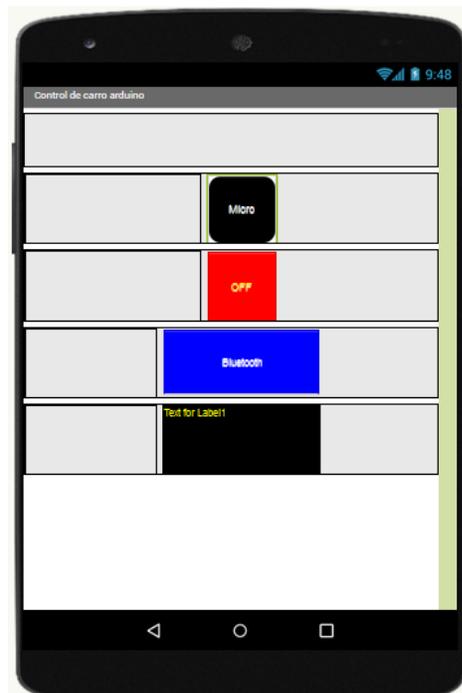


Figura 18. Ejemplo de aplicación de control.

7. A continuación, seleccione el apartado de “Blocks” que se encuentra en la esquina superior derecha, figura 19.



Figura 19. Apartado de Designer y Blocks.

8. Se muestra una imagen de la interfaz de la figura 20, del apartado Blocks.

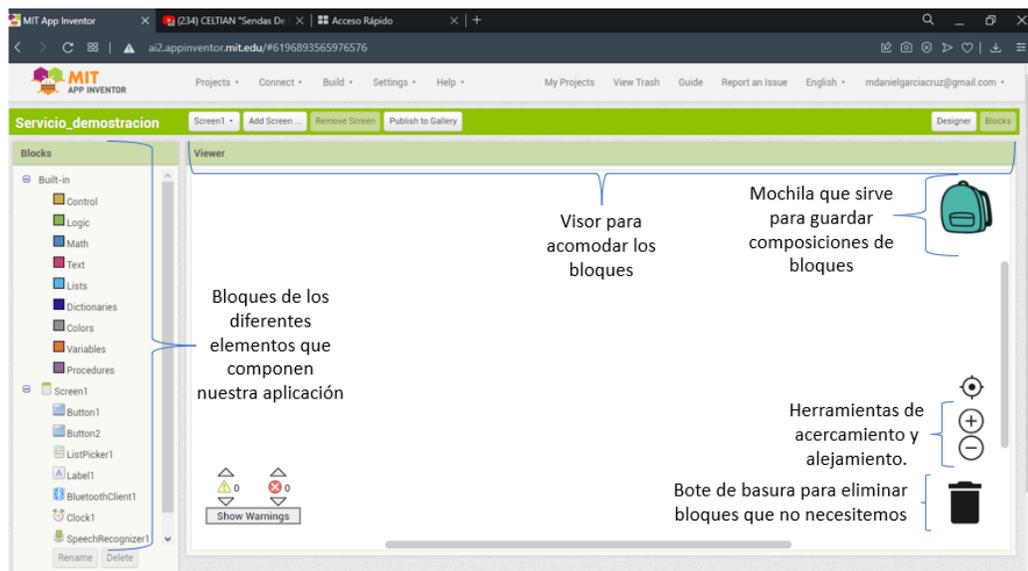


Figura 20. Interfaz del apartado Block de App Inventor.

9. Empezaremos por armar el bloque correspondiente a la función Clock, este nos indicará cuando la aplicación se encuentre conectada con el vehículo, marcando de color verde “Conectado” y de color rojo “Desconectado”. El proceso se muestra de la figura 21 a la figura 27, se hace notar que estas figuras no tiene descripción, ya que la descripción de las mismas, está inmersa en la propia imagen.

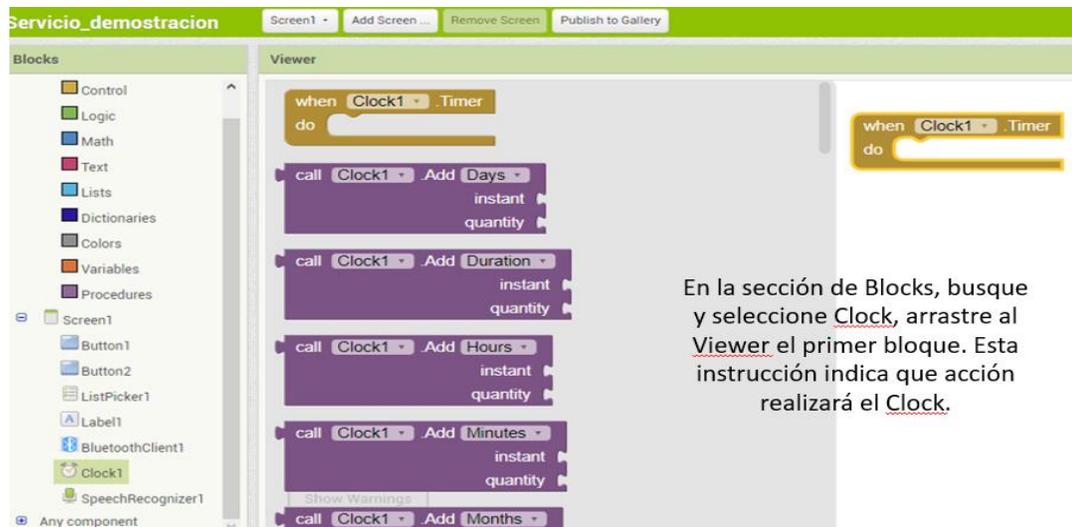


Figura 21.

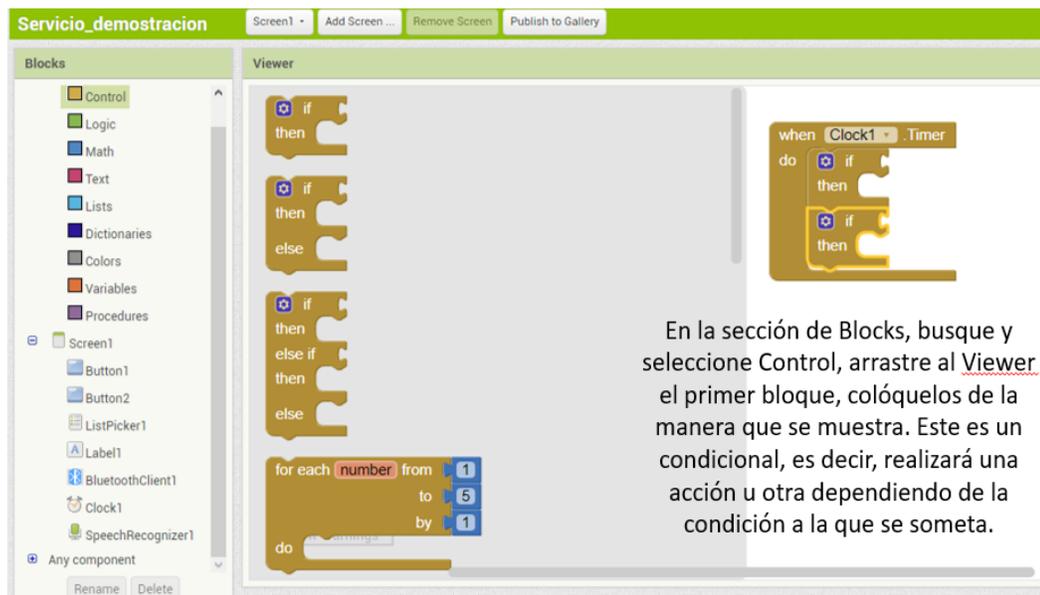


Figura 22.

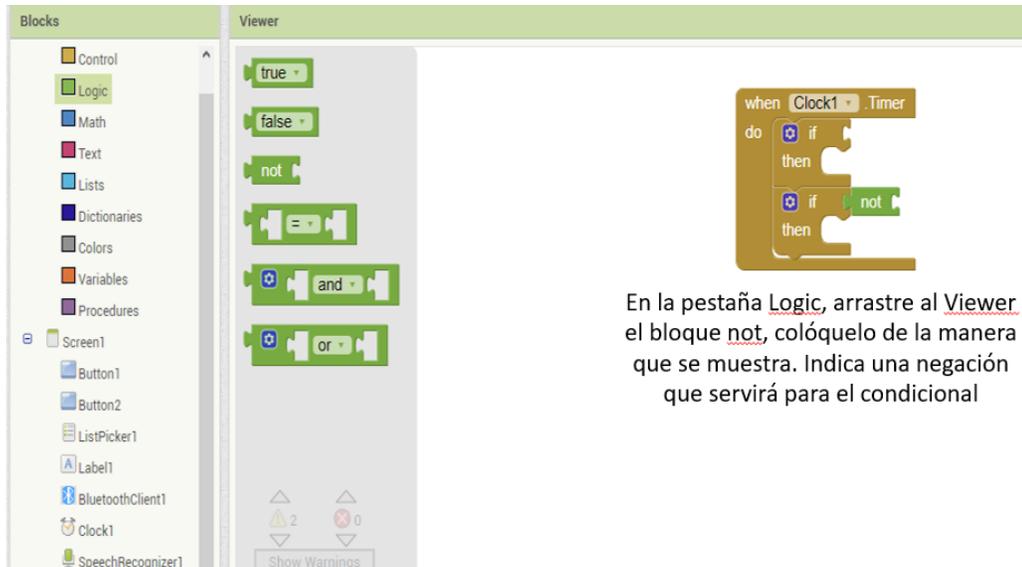


Figura 23.

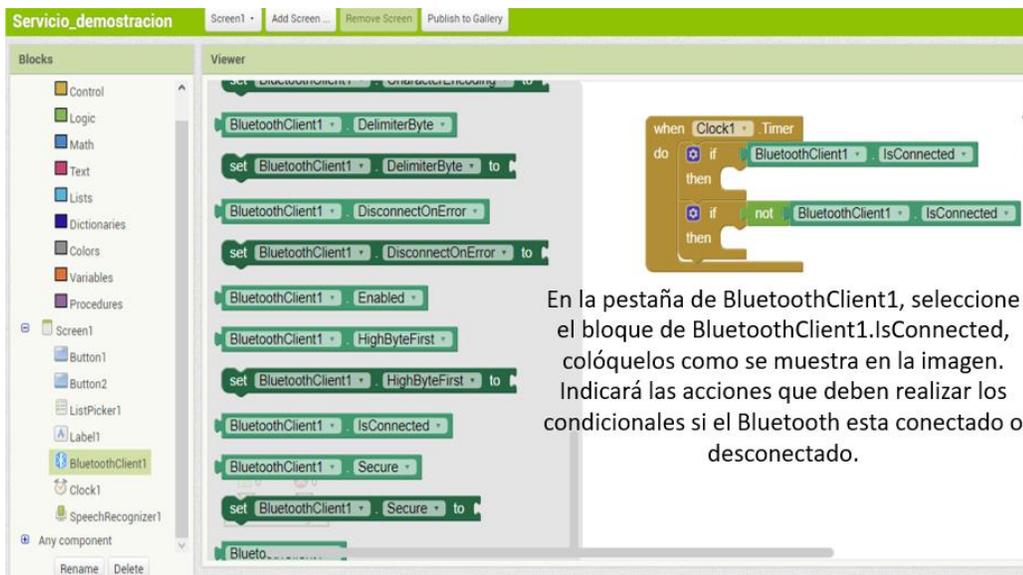


Figura 24.



Figura 25.

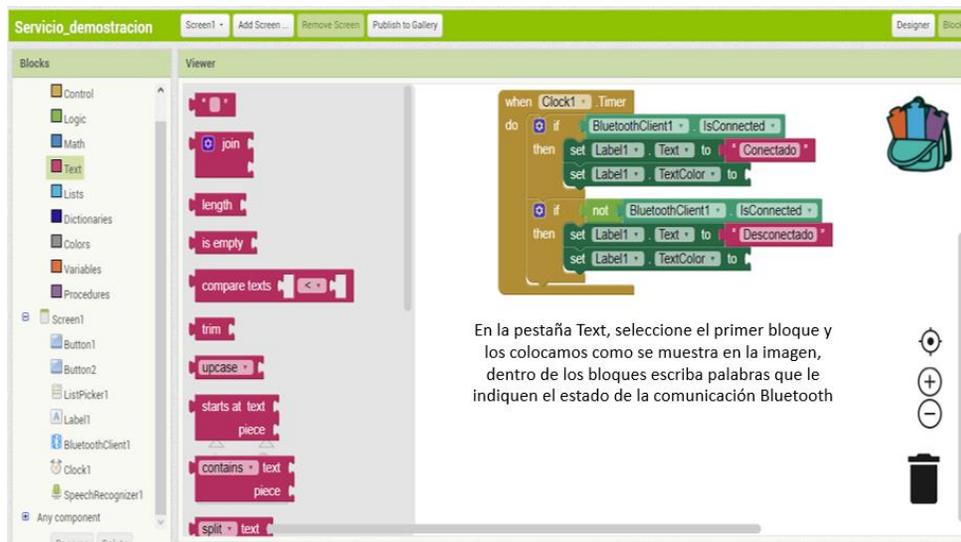


Figura 26.



Figura 27.

10. A continuación, configuraremos la selección de comunicación Bluetooth, es decir, veremos los dispositivos bluetooth que podemos seleccionar para controlar. El proceso se muestra de la figura 28 a la figura 33, se hace notar que estas figuras no tiene descripción, ya que la descripción de las mismas, está inmersa en la propia imagen.



Figura 28.



Figura 29.



De la pestaña ListPicker, seleccionamos set ListPicker1.Elements to y lo colocamos en el condicional when, en la pestaña de BluetoothClient1 seleccionamos BluetoothClient1.AddressesAndNames.

Figura 30.



En la pestaña BluetoothClient1, seleccionamos el tercer bloque, este sirve para implementar la comunicación Bluetooth, lo colocamos como la condición a cumplir.

Figura 31.

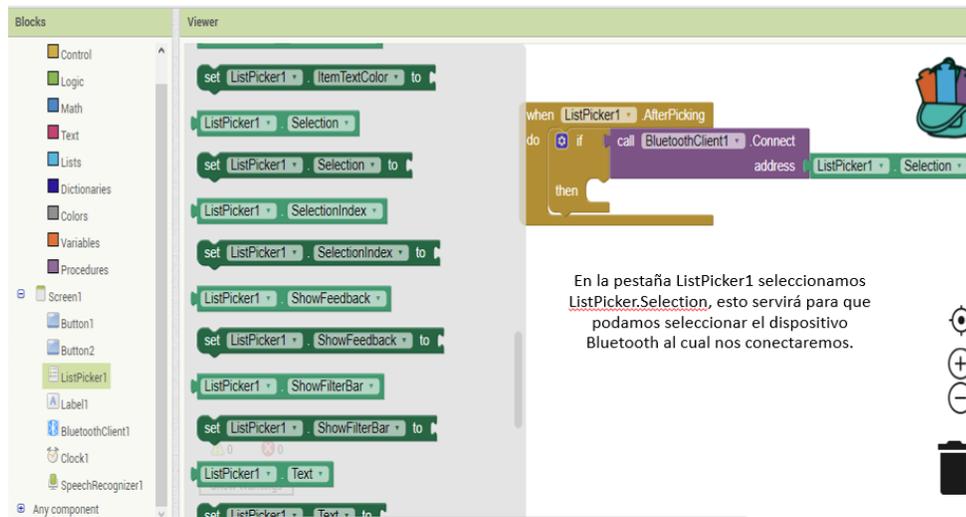
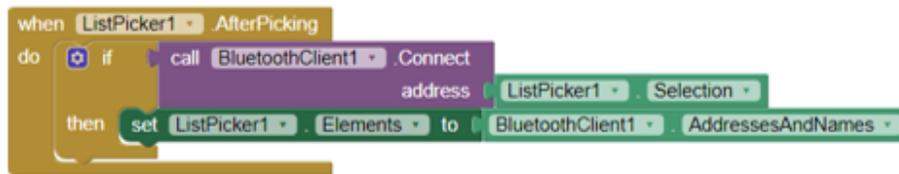


Figura 32.



Seleccionamos los elementos mostrados en las figuras 30 y 31 para colocarlos en la respuesta del condicional if, cuando se cumpla, abrirá la lista de dispositivos a los cual se pueda conectar.

Figura 33.

11. Procedemos a configurar los botones de la aplicación, el primer botón abrirá el micrófono para darle instrucciones al vehículo y el segundo botón desconectará la comunicación Bluetooth. El proceso se muestra de la figura 34 a la figura 37, se hace notar que estas figuras no tiene descripción, ya que la descripción de las mismas, está inmersa en la propia imagen.



Figura 34.



Figura 35.



Figura 36.



Figura 37.

12. Para finalizar la programación de bloques, armaremos las condiciones para mandar instrucciones al micrófono y que de esa forma se mueva el vehículo. El proceso se muestra de la figura 38 a la figura 44. se hace notar que estas figuras no tiene descripción, ya que la descripción de las mismas, está inmersa en la propia imagen.

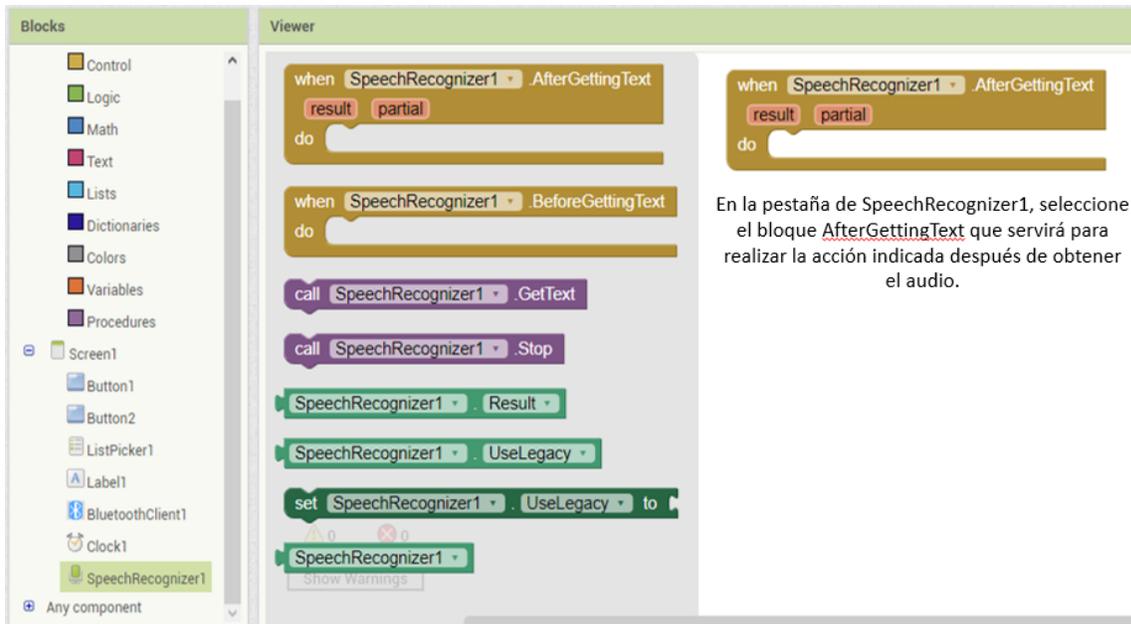
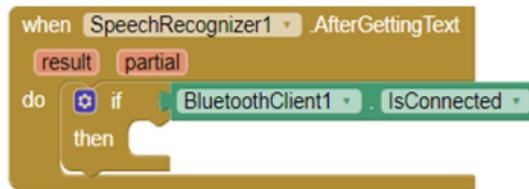
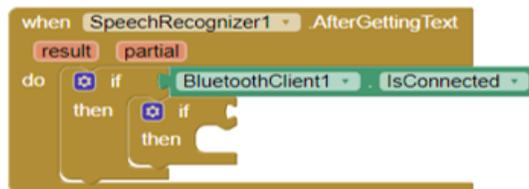


Figura 38.



Añada un condicional if-then, en la parte de if, añade el bloque BluetoothClient1.IsConnected.



Dentro de then añade otro condicional.

Figura 39.

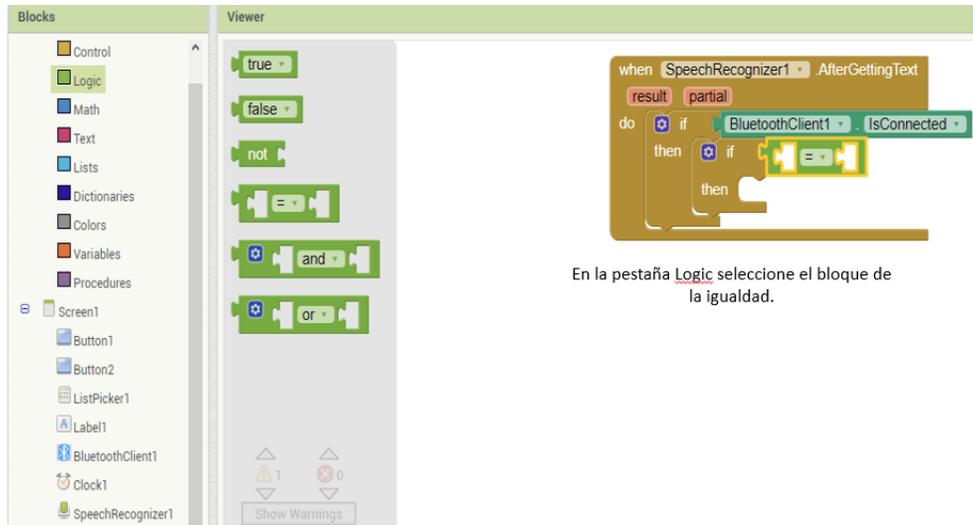


Figura 40.



Figura 41.

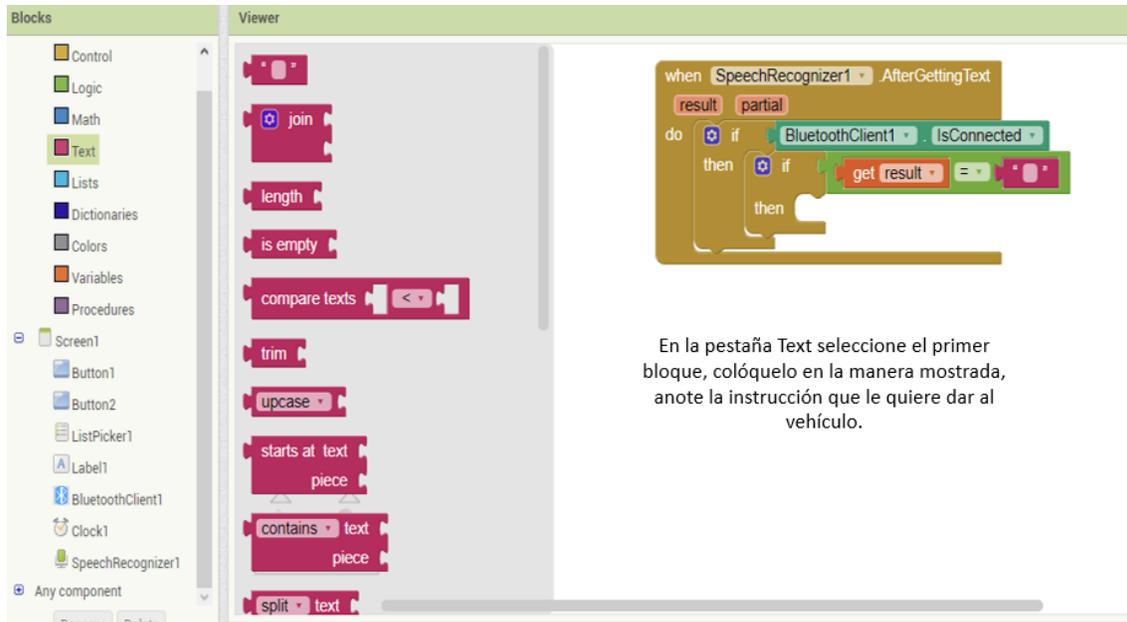


Figura 42.

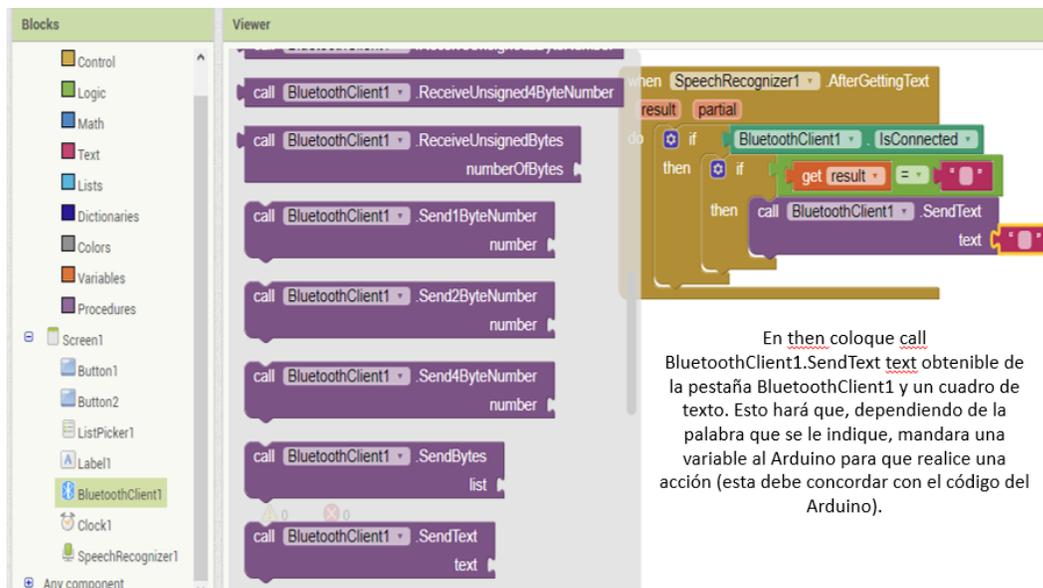
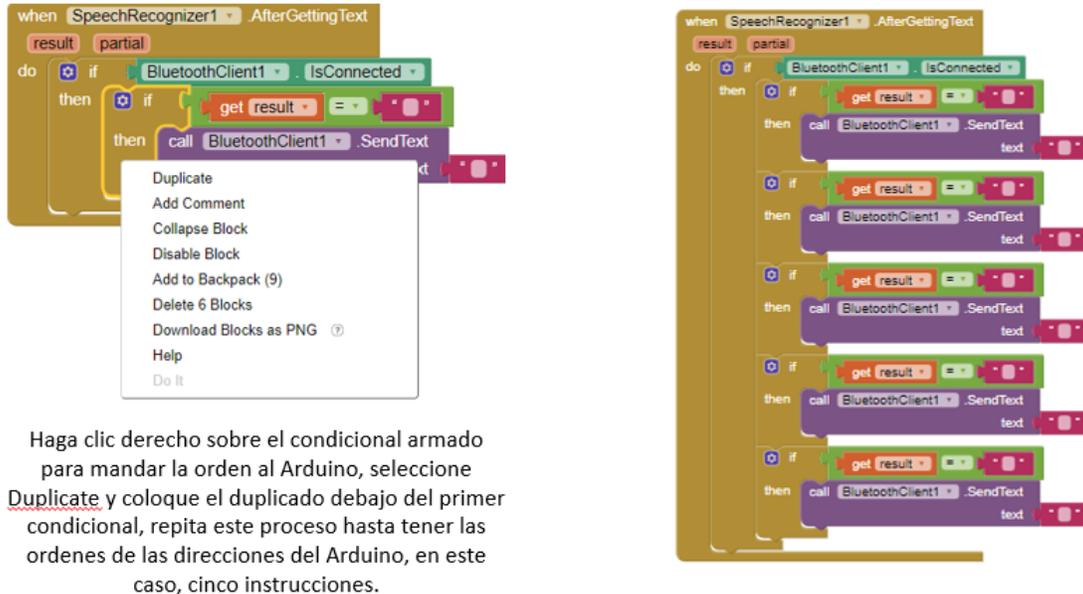


Figura 43.



Haga clic derecho sobre el condicional armado para mandar la orden al Arduino, seleccione Duplicate y coloque el duplicado debajo del primer condicional, repita este proceso hasta tener las ordenes de las direcciones del Arduino, en este caso, cinco instrucciones.

Figura 44.

13. El diagrama de bloques quedará tal y como se muestra en la figura 45.

**NOTA:** Para las instrucciones pruebe con diferentes palabras y escriba como las detecta el micrófono, pues, puede no recibir la orden si esta mal escrita o escrita de manera diferente en el micrófono.

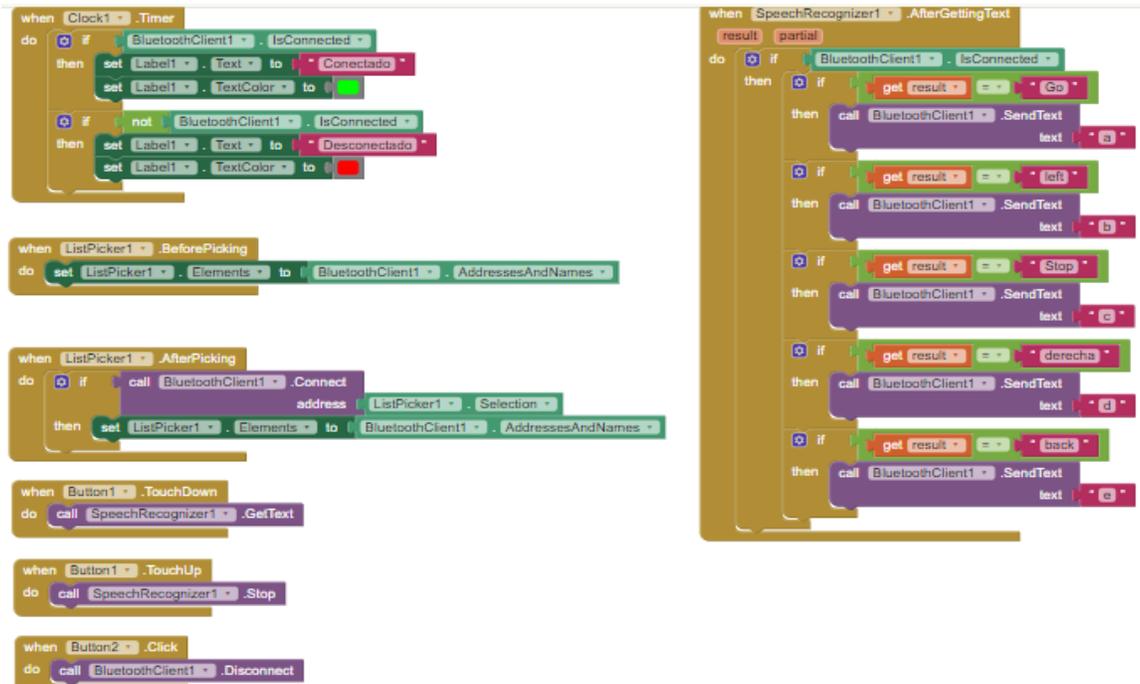


Figura 45. Programación de bloques completa.

14. Para crear la aplicación seleccione la pestaña Build y seleccione el formato de la app, si es en teléfono o computadora. Espere a que termine el proceso termine y descárguela, como se fe en las figuras 46 y 47. Una vez compilada, se podrá descargar la aplicación que podrá ser instalada en un dispositivo Android, como se ve en la figura 48.

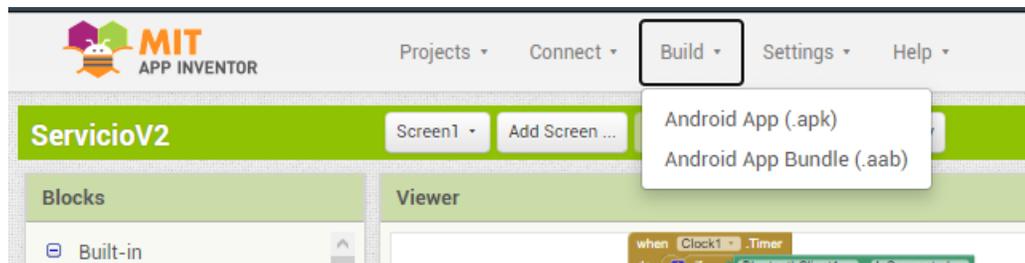


Figura 46. Pestaña Build y sus opciones.

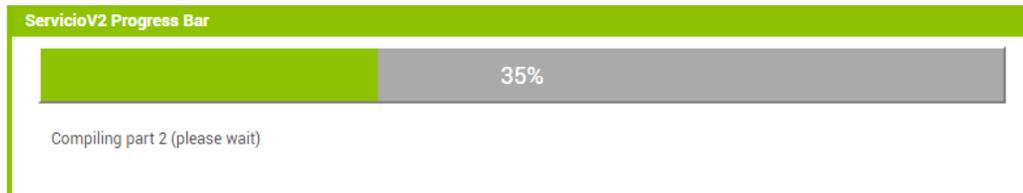


Figura 47. Ventana del proceso de compilación de la aplicación,

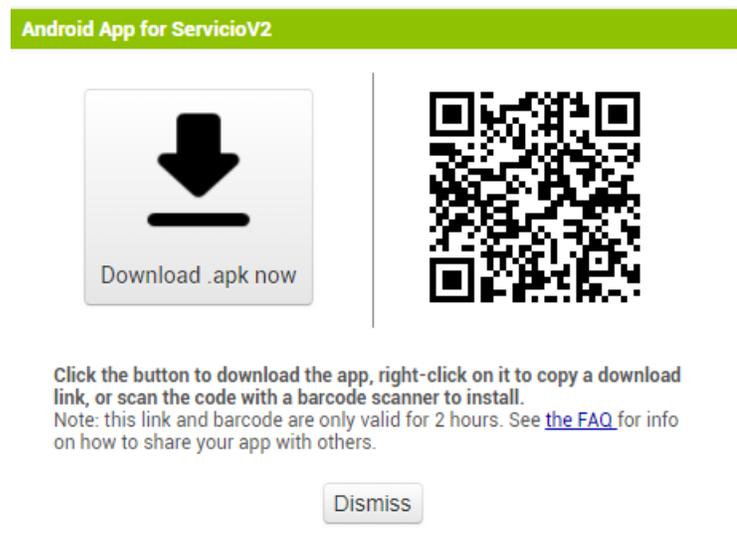


Figura 48. Ventana de descarga de la aplicación.

15. En este caso, se compilo como una apk, descárguela e instálela en su dispositivo Android. A continuación, se observan las diferentes pantallas de la aplicación, incluyendo la espera del mensaje de voz, figuras 49 a 52.

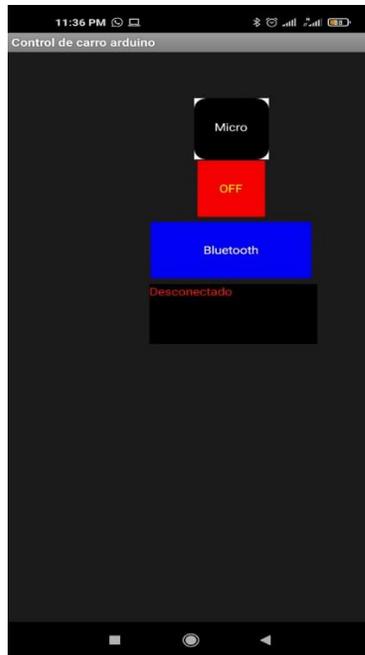


Figura 49. Pantalla principal de la aplicación.

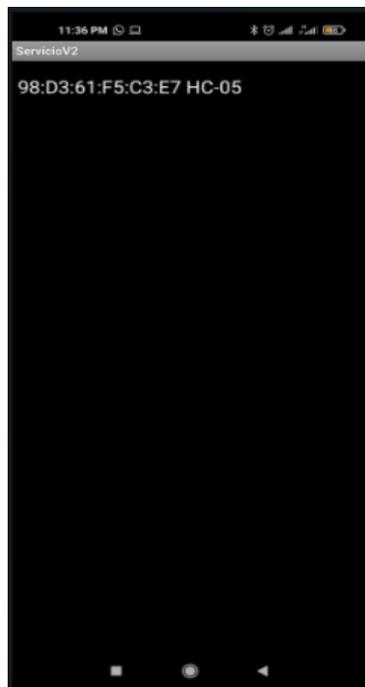


Figura 50. Selección del dispositivo Bluetooth.

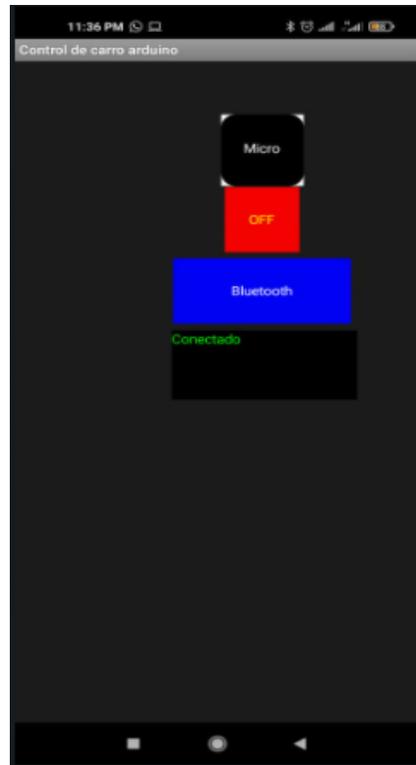


Figura 51. Pantalla principal con el Bluetooth activado.

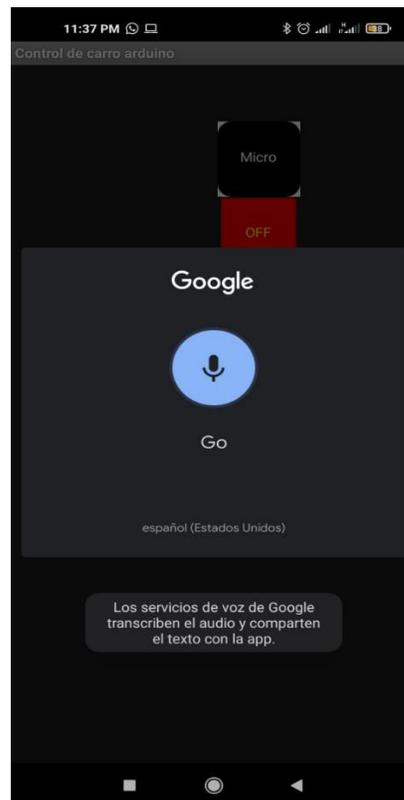


Figura 52. Pantalla del micrófono con una instrucción.

La integración total del proyecto, nos permitirá poder controlar un vehículo/juguete que pueda ser manejado por medio de elementos no convencionales, en este caso la voz, permitiendo que sectores poblacionales que presenten algún problema o falta de movilidad en sus manos, puedan interactúa con él de forma sencilla y natural. Este diseño puede enfocarse también a la generación de interés en sectores poblacionales jóvenes, ya que se presenta toda la secuencia de pasos para generar un diseño que presenta características tecnológicas interesantes. .

## Conclusión

---

En conclusión, el propósito de este proyecto es incitar a los niños, y sectores poblacionales jóvenes, a desarrollar interés en las áreas de electrónica, programación y tecnología en general por medio de un proyecto que puedan construir desde cero, incitando a la creación de su propio juguete a control remoto, teniendo herramientas que sean amigables para su aprendizaje. Los niños con capacidades diferentes se pueden adaptar a la elaboración de este proyecto, ya sea con ayuda de un adulto o por su propia cuenta, en caso de discapacidad intelectual, los colores y las formas de los bloques para programar la aplicación permiten que sean mas entendibles y puedan comprenderlas con mayor facilidad, en el caso de discapacidad motriz, en especial las manos, el mandar las instrucciones por medio de voz permite que puedan usar el vehículo sin mayor problema. Permitiendo que la tecnología sea una aliada muy fuerte en la inclusión de todo tipo de sectores poblacionales en la interacción con elementos tecnológicos.