

Creación de carro funcional con arduino para la enseñanza de la física

Criação de um carrinho funcional com arduino para o ensino da física

DOI:10.34117/bjdv8n3-384

Recebimento dos originais: 27/01/2022

Aceitação para publicação: 25/02/2022

Diego Fernando Galviz Cataño

Phd. Administración Gerencial

Institución: Corporación Universitaria Americana sede Medellín

E-mail: dgalviz@americana.edu.co

Alexander Jiménez Guzmán

Doctorante en Ciencias de la Educación

Institución: Universidad del Rosario, Argentina

E-mail: ajimenezg@americana.edu.co

Jorge Oliverio Suaza Arcila

Phd. Administración Gerencial

Institución: Corporación Universitaria Americana, sede Medellín

E-mail: josuaza@americana.edu.co

Juan Carlos Cardona Acosta

Phd. (C) en Educación

Institución: Universitaria Pascual Bravo - Facultad de Ingenierías

E-mail: juan.cardona@pascualbravo.edu.co

RESUMEN

El presente informe describe los materiales y pasos necesarios a seguir para la construcción de un carro funcional a instrucciones mediante el uso de hardware y software Arduino. Una vez construido el prototipo, se diseñó una práctica de laboratorio para la enseñanza de la física mecánica en un programa de Ingenierías para la Corporación Universitaria Americana, sede Medellín, Colombia. En la práctica se realizó una serie de experimentos a diferentes velocidades para analizar la distancia recorrida, la velocidad y la aceleración del carro. Este tipo de métodos de enseñanza han demostrado mejorar los resultados de los estudiantes facilitando el aprendizaje y aumentando la practicidad de los problemas.

Palabras clave: aprendizaje, enseñanza, arduino, ciencia, física, carro.

RESUMO

Este relatório descreve os materiais e os passos necessários para construir um carro funcional às instruções usando hardware e software Arduino. Uma vez construído o protótipo, uma prática laboratorial foi projetada para o ensino de física mecânica em um programa de engenharia na Corporación Universitaria Americana, Medellín, Colômbia. Na prática, uma série de experimentos foi realizada em diferentes velocidades para analisar a distância percorrida, a velocidade e a aceleração do carro. Estes tipos de métodos de ensino têm demonstrado melhorar o desempenho dos estudantes, facilitando o aprendizado e aumentando a praticidade dos problemas.

Palavras-chave: aprendizagem, ensino, arduino, ciência, física, carro.

1 INTRODUCCIÓN

A lo largo de los últimos años se han realizado diferentes proyectos enfocados al uso de la tecnología en las aulas de clase, con el propósito de mejorar el aprendizaje de los alumnos y la experiencia aplicativa de conceptos teóricos al ámbito práctico. Uno de estos métodos es la integración de las TIC (tecnologías de información y comunicación) las cuales son un conjunto de tecnologías desarrolladas para una información y comunicación eficiente. Las TIC son de suma importancia en el ámbito educativo ya que permite a los docentes la posibilidad de generar contenido educativo didáctico, que resulte de interés para los estudiantes.

En este caso se llevó a cabo un concepto similar para el aprendizaje de la física, por medio de un carro desarrollado a partir de arduino y algunos otros componentes. A este auto se le adaptó un taller de física, con el objetivo de mejorar el análisis teórico - práctico en los estudiantes. En el presente artículo científico se describen los diferentes pasos, materiales y talleres necesarios para la realización e implementación del carro a un ambiente de aprendizaje.

El Arduino presenta grandes ventajas por su disponibilidad, bajo costo y además “el movimiento open-source, conocido en español como libre o de fuente abierta, ha mostrado que se puede crear software útil y robusto a través de un equipo distribuido de voluntarios dispuestos a compartir libremente los resultados de sus desarrollos y proyectos. Dichos proyectos a menudo reúnen grandes y fuertes comunidades de personas que trabajan de manera concurrente en varios niveles” (Gómez, Castillo & Gómez, 2015).

2 DESARROLLO METODOLÓGICO

Se utilizó una metodología mixta inductiva debido a que el proyecto se plantea desde un punto específico, el cual es la realización del prototipo. Para esto se planeó la búsqueda de los diferentes métodos existentes para alcanzar este propósito; así pues y por medio de diferentes pruebas se obtuvo el primer prototipo funcional del carro orientado a la enseñanza de la física. Además de lo anterior, se plantea la necesidad de la implementación de la tecnología en las aulas y sus diferentes beneficios, debido a esto es

menester realizar pruebas cuantitativas al prototipo para verificar su eficacia en las aulas y si realmente representa un beneficio para los estudiantes en Medellín, Colombia.

3 ASPECTOS GENERALES DEL PROYECTO

Como se mencionó anteriormente, la creación de este carro Arduino está pensado para el uso de la enseñanza de la física de un modo didáctico donde se estudiarán temas tales como la velocidad, desplazamiento y distancia, entre otras. Teniendo en cuenta el grado de complejidad de la misma, se presenta un desinterés notorio por parte de los estudiantes, esto debido a los sistemas convencionales de enseñanza, posiblemente porque la enseñanza de las ciencias suele olvidar que su principal finalidad es preparar a los ingenieros y a los ciudadanos para participar en una sociedad que día a día se ve rodeada e influenciada por la tecnología. Debido a esto se optó por un método de enseñanza que resulta de interés para el alumnado, donde la tecnología funciona como mediadora y así mismo facilita el proceso de enseñanza. Por otro lado, “existen muchas dificultades para conseguir material de laboratorio apto para la realización de experiencias dentro del aula. Por un lado, por la falta de equipamiento, por otro, por los elevados costos que requiere.” (Llera, Scagliotti, Jorge, 2017). En el mismo sentido, Christiansen, Miguel y Pereyra (2016) manifiestan que “Al no poder interactuar de forma directa, los estudiantes se dispersan y pierden el interés por el ejercicio. Frente a esta situación surge la necesidad de desarrollar nuevos dispositivos más económicos capaces de reemplazar a los otros y promover una participación activa de los alumnos”. Concordando con lo recién expresado, Celin, Solano y Molina (2016) coinciden en que “La necesidad que se presenta en las instituciones educativas con respecto de equipos e instrumentos en el laboratorio de física, ha llevado como iniciativa de incrementar el número de proyectos de aula encaminados en la construcción y diseño de montajes experimentales a bajo costo, con la aplicación de la electrónica básica”

4 OBJETIVOS

4.1 OBJETIVO GENERAL

Creación de carro funcional con arduino para la enseñanza de la física.

4.2 OBJETIVOS ESPECÍFICOS

- Aplicar los diferentes conocimientos en robótica para la creación del carro funcional con Arduino, incluyendo la programación adecuada para su operación.

- Crear una práctica de laboratorio didáctica para permitir una experiencia tecnológica del proceso enseñanza - aprendizaje que mejore el interés y el nivel de adquisición del conocimiento de la ciencia de la Física, mediante un carro funcional con Arduino.

5 MATERIALES DE TRABAJO

5.1 SOFTWARE

El software Arduino 1.8.9 es definido por su desarrolladora como un software de código abierto hecho para escribir código fácilmente y ser usado con tarjetas de desarrollo Arduino (Arduino, 2019). Este software está disponible para las plataformas de Linux, Windows y Mac OS X; en este caso se usó el sistema operativo Windows 10.

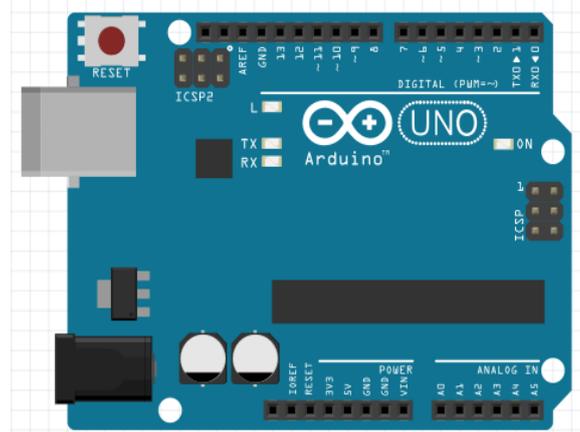
5.2 HARDWARE

Para el desarrollo del hardware se utilizaron diferentes componentes de fácil alcance económico, esto debido a que es un proyecto educativo y es necesario que pueda llegar al mayor número de personas posibles.

5.3 ARDUINO UNO

El primer componente es la placa de desarrollo arduino uno, esta es una placa con microcontrolador distribuido como hardware y software libre compuesto por 14 pines los cuales funcionan como entrada o salida. De estos pines dos de ellos son el RX y TX que se utilizan para recibir y transmitir datos en serie, 2 son entradas GND y una entrada para voltaje; además de esto, seis de ellos pueden funcionar como entradas analógicas y las otras seis como salidas PWM. La placa de desarrollo cuenta con un conector USB tipo A/B el cual permite que el dispositivo sea programado, una fuente de alimentación externa y un botón de reset.

Imagen 1: Tarjeta de desarrollo Arduino uno.

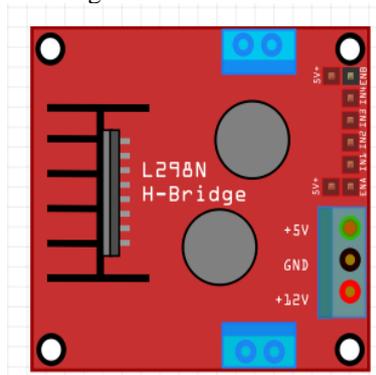


Fuente: Autores.

5.3.1 Puente H L298n

Es una tarjeta para el control de motores de corriente directa. Compuesta principalmente por cuatro salidas para los motores A y B, una entrada de 5V, una entrada de 12V y una entrada GND.

Imagen 2: Puente H L298N.

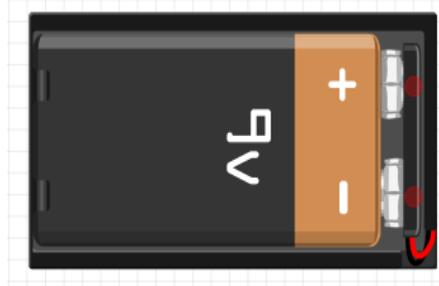


Fuente: Autores.

5.3.2 Pilas 9V

Dos pilas de 9V (Voltios) y dos porta baterías de 9V. Es importante hacer énfasis que una de las porta baterías debe tener un plug DC para poder entregar voltaje directo al arduino. La otra porta batería se encarga de suministrar el voltaje al controlador de motores L298N.

Imagen 3: Porta batería con pila de 9V.

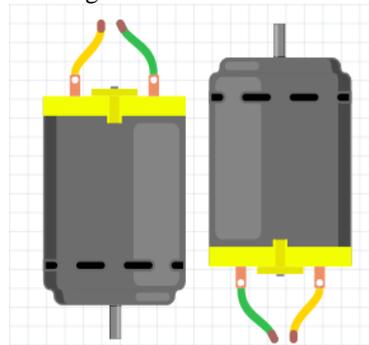


Fuente: Autores.

5.3.3 Motorreductores

Dos motorreductores junto con sus respectivas llantas. Estos se encargan del movimiento del carro.

Imagen 4: Motorreductores.



Fuente: Autores.

5.3.4 Rueda giratoria

Es la encargada de hacer que el carro pueda girar más fácilmente.

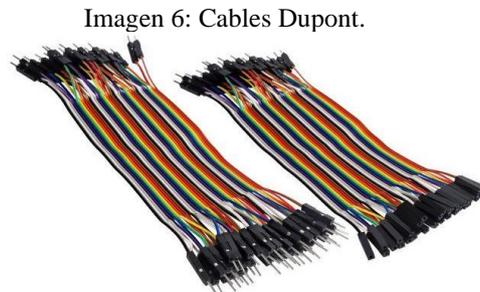
Imagen 5: Rueda giratoria.



Fuente: (SanDonRobotics, 2019).

5.3.5 Cables Dupont

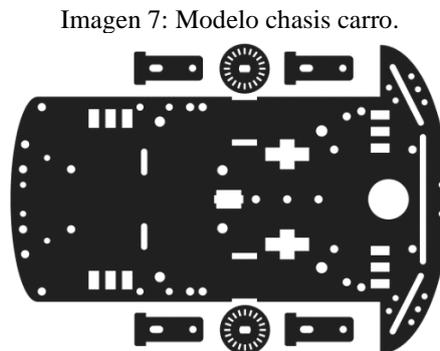
Cables macho-hembra y hembra-hembra (en su defecto puede cortarse un macho-hembra para el reemplazo del hembra-hembra). Estos son los encargados de transmitir la carga eléctrica por el sistema para su adecuada comunicación.



Fuente: (Stereon, 2019).

5.3.6 Chasis

Es el encargado de soportar todo el sistema del carro. Este se puede obtener mediante una impresora 3D o realizar un modelo similar a base de ingenio.

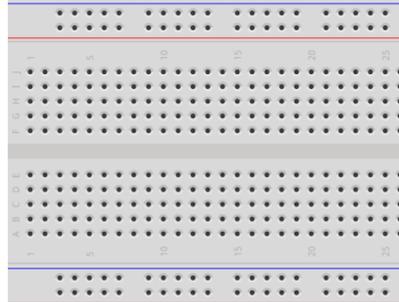


Fuente: (Thingiverse, 2015)

5.3.7 Protoboard

Se utilizó una protoboard de 230 puntos y 2 buses, con el propósito de realizar las conexiones con mayor eficacia y orden.

Imagen 8: Protoboard 230 puntos y 2 buses.

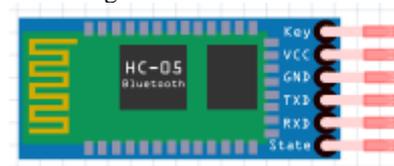


Fuente: Autores.

5.3.8 Arduino HC-05

Se utilizó el modulo bluetooth HC-05, el cual permitió enviar los datos de distancia al monitor de manera inalámbrica.

Imagen 9: Arduino HC-05.

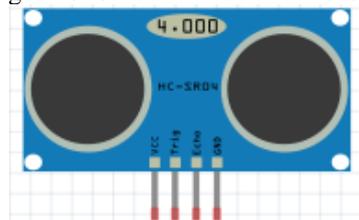


Fuente: Autores.

5.3.9 Sensor ultrasónico HC-SR04 de Arduino

Se utilizó el sensor ultrasónico de Arduino HC-SR04, este funciona por medio de dos sensores, un emisor (Trig) y un receptor (Echo); el emisor envía un pulso de alta frecuencia el cual rebota en superficies sólidas y se refleja hacia el receptor, el cual posee un micrófono que le permite captar estas frecuencias; obteniendo así la distancia del sensor al objeto de choque. Esto permitió calcular la distancia del auto hasta una placa de poliestireno expandido. “Los sensores ultrasónicos son sensibles al voltaje aplicado. Además, pueden tener puntos ciegos, ya sea que el objeto esté ligeramente fuera del rango de lectura, o si el objeto está pegado al sensor ultrasónico.” (Campo, Sastre, Gamas, Samaniego, 2017).

Imagen 10: Sensor ultrasónico HC-SR04.



Fuente: Autores.

5.3.10 Poliestireno expandido

Se utilizó una placa de poliestireno expandido de 1x1. Esta es la encargada de hacer la función objeto de choque para el sensor ultrasónico, permitiendo baja frecuencia de datos atípicos.

Imagen 11: Poliestireno expandido 1x1.



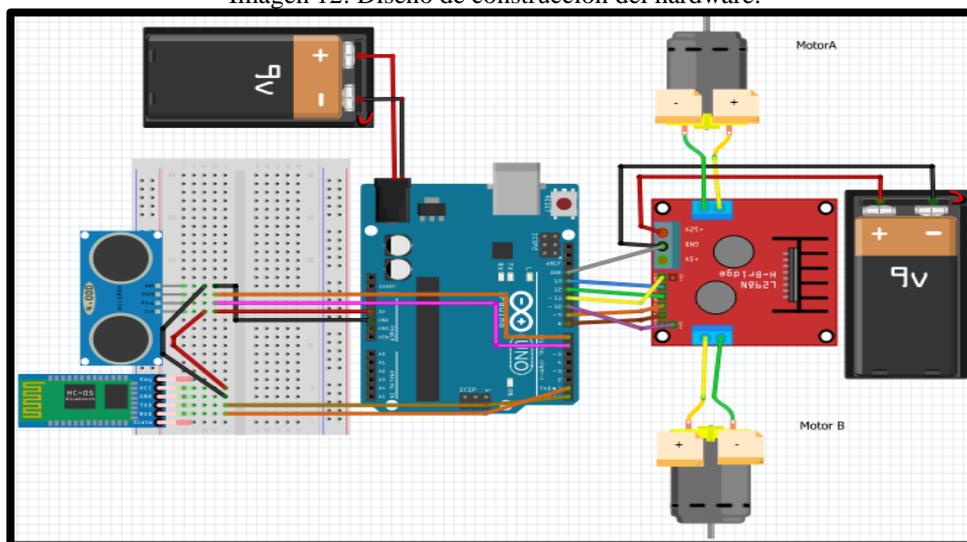
Fuente: (Bauhaus, 2019).

6 DISEÑO FUNCIONAL DEL CARRO

6.1 DISEÑO DEL HARDWARE

Por medio de los dispositivos electrónicos anteriormente mencionados se realizó un diseño del carro enfocado a la enseñanza; este se compone de una tarjeta Arduino Uno, Puente H o L298N, dos motorreductores, dos pilas de 9V, una protoboard, un Arduino HC-05, un sensor ultrasónico HC-SR04, una placa de poliestireno expandido de 1x1 m y los diferentes cables que se utilizaron. Todo el diseño del Hardware se realizó en el software Fritzing, el cual permite simular prototipos reales. En la siguiente imagen se muestra el diseño del carro.

Imagen 12: Diseño de construcción del hardware.

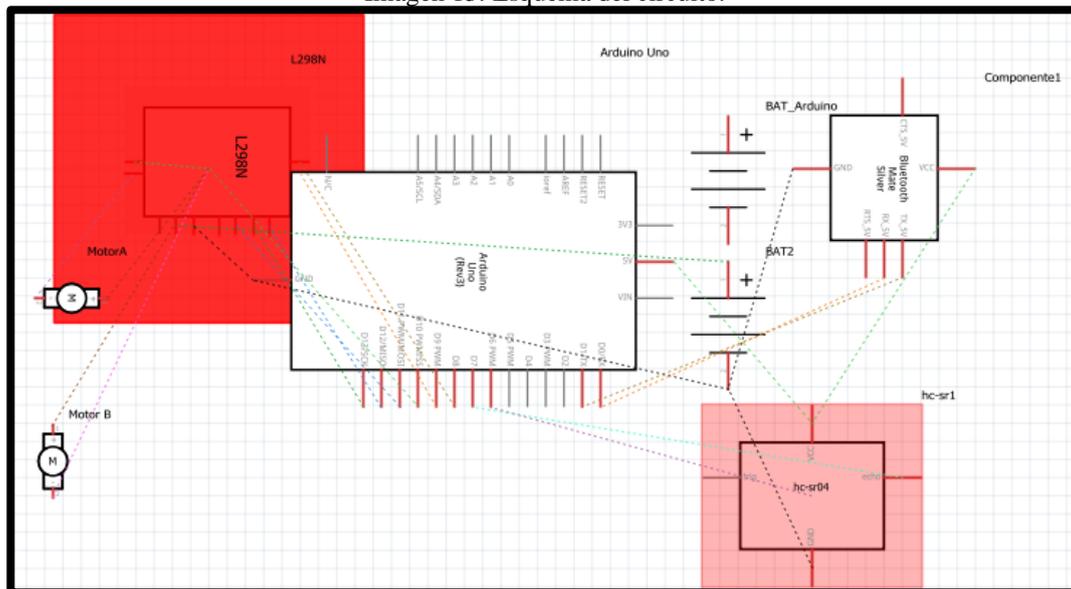


Fuente: Autores.

6.2 ESQUEMA DEL CARRO ARDUINO

De igual forma se realizó un esquema de las conexiones por medio del software Fritzing, con el propósito de representar el funcionamiento del circuito. En este se representan cada una de las interacciones entre los materiales electrónicos previamente enunciados; a continuación se muestra el respectivo diagrama.

Imagen 13: Esquema del circuito.



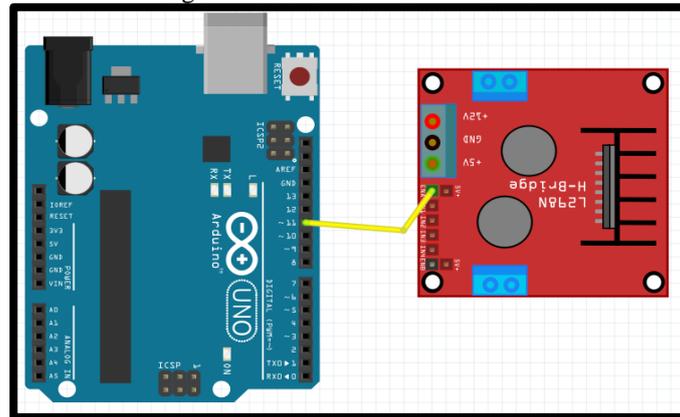
Fuente: Autores.

6.3 CONSTRUCCIÓN DEL HARDWARE

Mediante los materiales previamente enunciados en “Materiales de trabajo” se realizó un diseño efectivo del carro arduino, en el cual se evidencian las conexiones pertinentes para el correcto funcionamiento del hardware del carro. A continuación se describe el paso a paso de la realización del carro con Arduino.

6.4 Para comenzar se conectó el cable amarillo a la salida PWM número 11 de arduino y en ENA del L298N (Puente H) este funciona como un controlador de velocidad del Motor A.

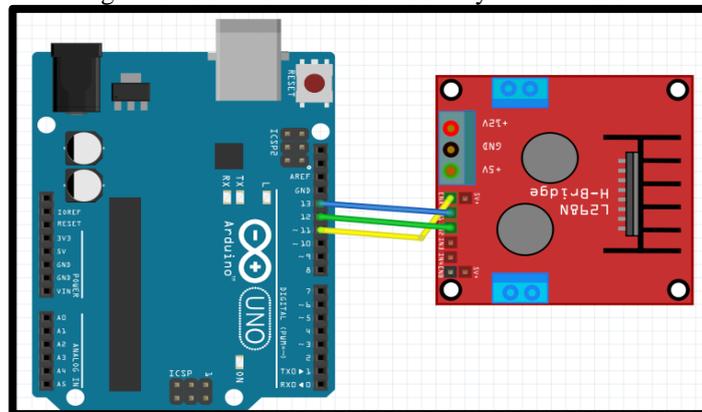
Imagen 14: Conexión PWM 11 a ENA.



Fuente: Autores.

6.5 Luego se procedió a realizar la conexión de los cables azul y verde, los cuales se ubican al lado del cable amarillo de velocidad anteriormente mencionado, estos controlan el sentido de las llantas; así pues se conectó el cable azul a PIN13 y OUT1 y el cable verde a PIN12 y OUT2, todo lo anterior con el fin de controlar la velocidad y el sentido del Motor A.

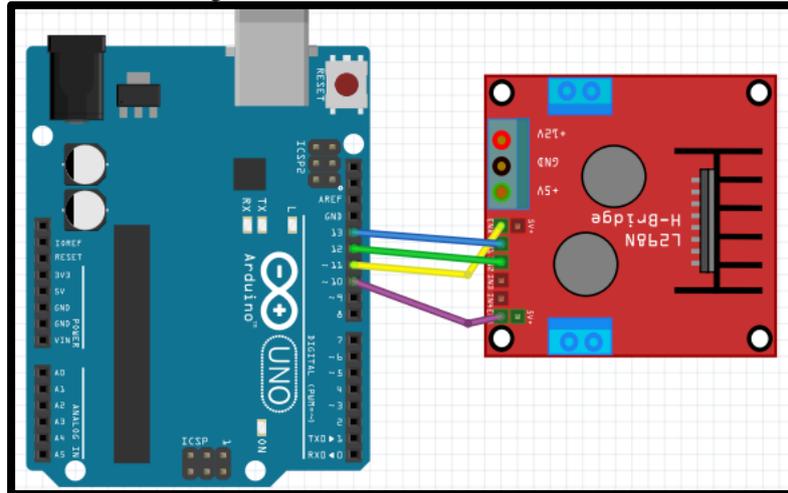
Imagen 15: Conexión PIN 13 a OUT1 y PIN12 a OUT2.



Fuente: Autores.

6.6 De modo similar se compone el funcionamiento del MOTOR B, donde se conectó el cable violeta a la salida PWM número 10 del Arduino Uno y ENB del Puente H este es el que controla la velocidad del Motor B.

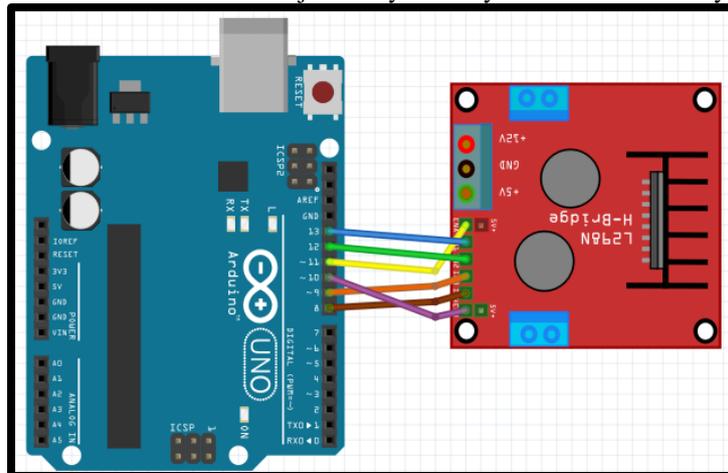
Imagen 16: Conexión PWM 10 a ENB.



Fuente: Autores.

6.7 Del mismo modo se conectaron los cables naranja al PIN9 y OUT3 y marrón al PIN8 y OUT4 del Arduino Uno y Puente H, estos controlan el sentido de las llantas.

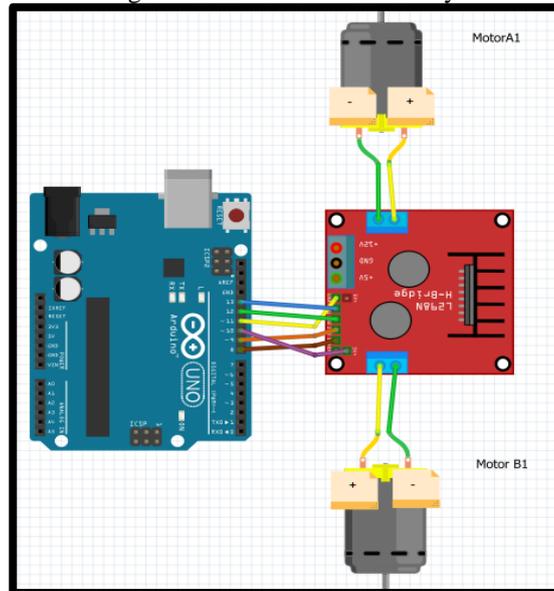
Imagen 17: Conexión cable naranja PIN9 y OUT3 y cable marrón PIN8 y OUT4.



Fuente: Autores.

6.8 Se continuó realizando la respectiva conexión de los motores A y B, los cuales se encargan de hacer girar las ruedas del carro. En este caso, ambos motores cuentan con un cable positivo y uno negativo los cuales deben conectarse a la entrada positiva y negativa del Puente H, tal y como se muestra en la imagen.

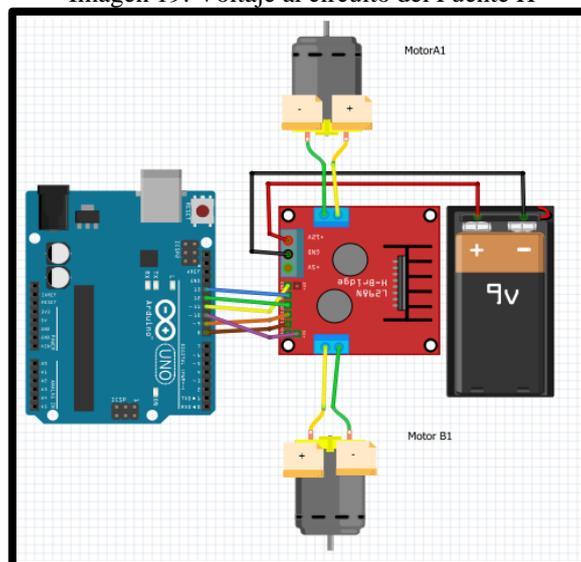
Imagen 18: Conexión motores A y B.



Fuente: Autores.

6.9 Luego se procedió a dar voltaje al Puente H, esta conexión se realizó conectando el cable negativo del porta pila a la entrada GND (Tierra) del Puente H y el cable positivo del porta pila a la entrada 12V del Puente H; tal y como se muestra en la siguiente imagen:

Imagen 19: Voltaje al circuito del Puente H

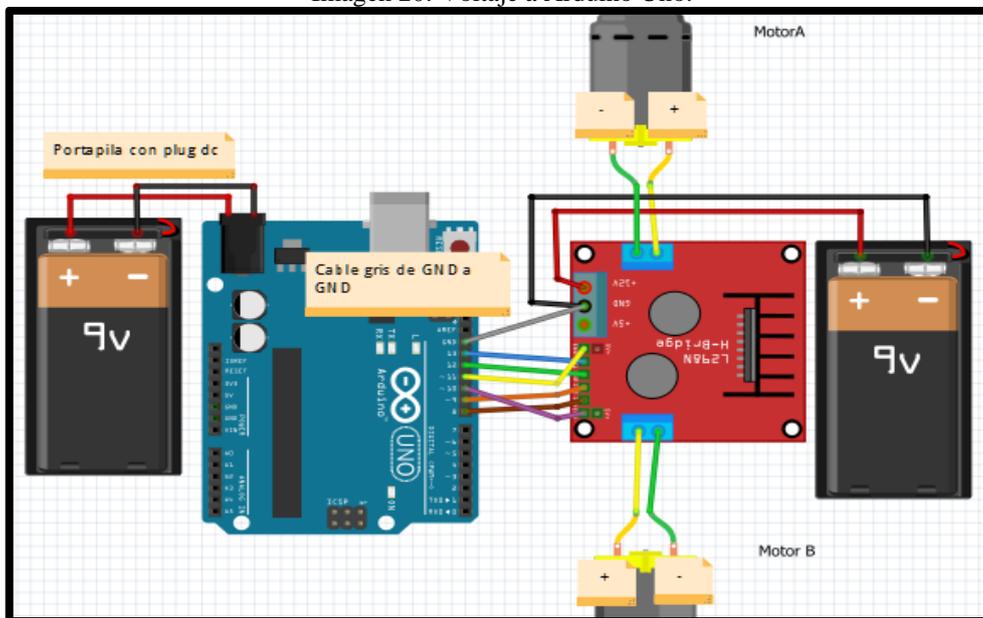


Fuente: Autores.

6.10 Se procedió a dar voltaje al Arduino Uno, por medio de una pila de 9V conectada a una porta pila con plug dc, en el sistema el Arduino uno es el encargado de alimentar la entrada de 5v del sensor ultrasónico y el módulo bluetooth. Además de lo anterior se

conectó un cable conectado de GND del Arduino Uno a GND del Puente H. Tal y como se muestra en la imagen 20.

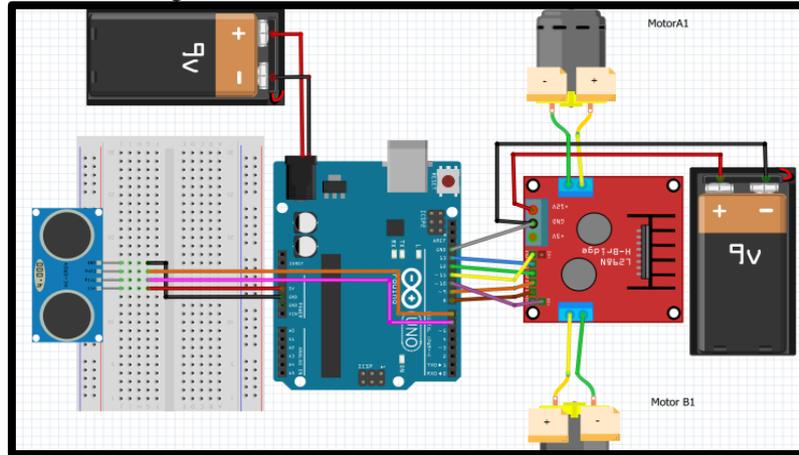
Imagen 20: Voltaje a Arduino Uno.



Fuente: Autores.

6.11 Se procedió a realizar la conexión del sensor ultrasónico para esto se utilizó la protoboard, en la cual se posicionó el sensor ultrasónico, este fue puesto en la fila J(10,11,12,13) y luego se conectó el pin “Trig” del HC-SR04 al pin número 6 del Arduino Uno. Luego de esto se conectó el pin “Echo” al pin número 7 del Arduino Uno. Por último, se posicionó la columna J10 de la protoboard al pin GND del Arduino y la columna J13 a la entrada 5V del Arduino. Tal como se muestra en la imagen (Tener en cuenta que para la correcta visualización de la imagen se optó por posicionar el módulo ultrasónico en posición espejo):

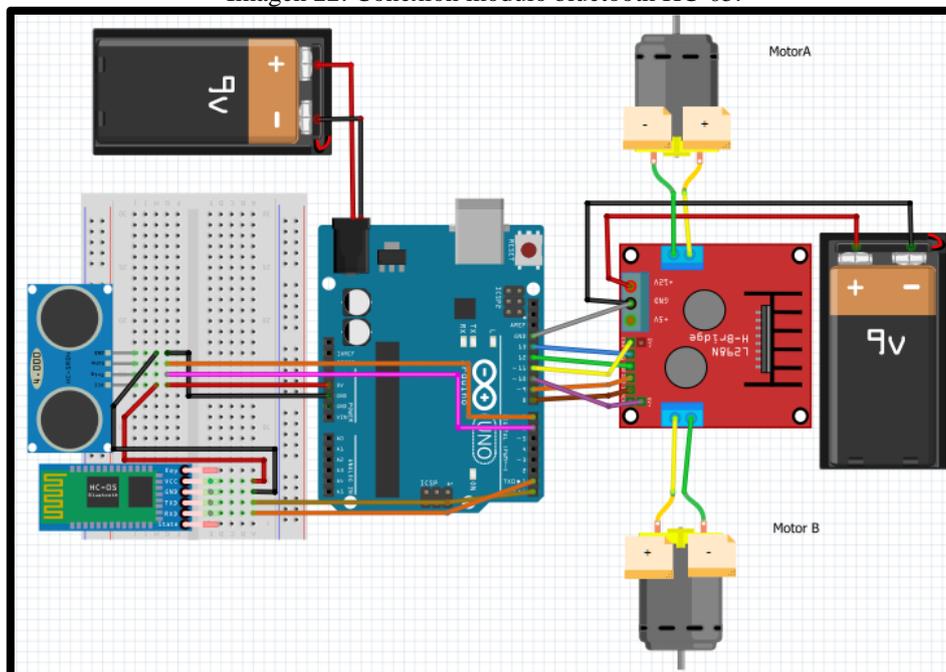
Imagen 21: Conexión módulo ultrasónico HC-SR04.



Fuente: Autores.

6.12 Para finalizar, se adaptó al sistema un módulo bluetooth HC-05, el cual se posicionó en la fila E (19, 20, 21, 22). Luego se conectó GND del HC-05 a la columna F13 de la protoboard, la cual previamente había sido conectada a GND del Arduino Uno; del mismo modo, se conectó 5V del HC-05 a F(10) de la protoboard, por último, se conectó Tx del HC-05 al Rx del Arduino Uno, y Rx del HC-05 al Tx del Arduino Uno. Tal como se muestra en la imagen 22.

Imagen 22: Conexión módulo bluetooth HC-05.

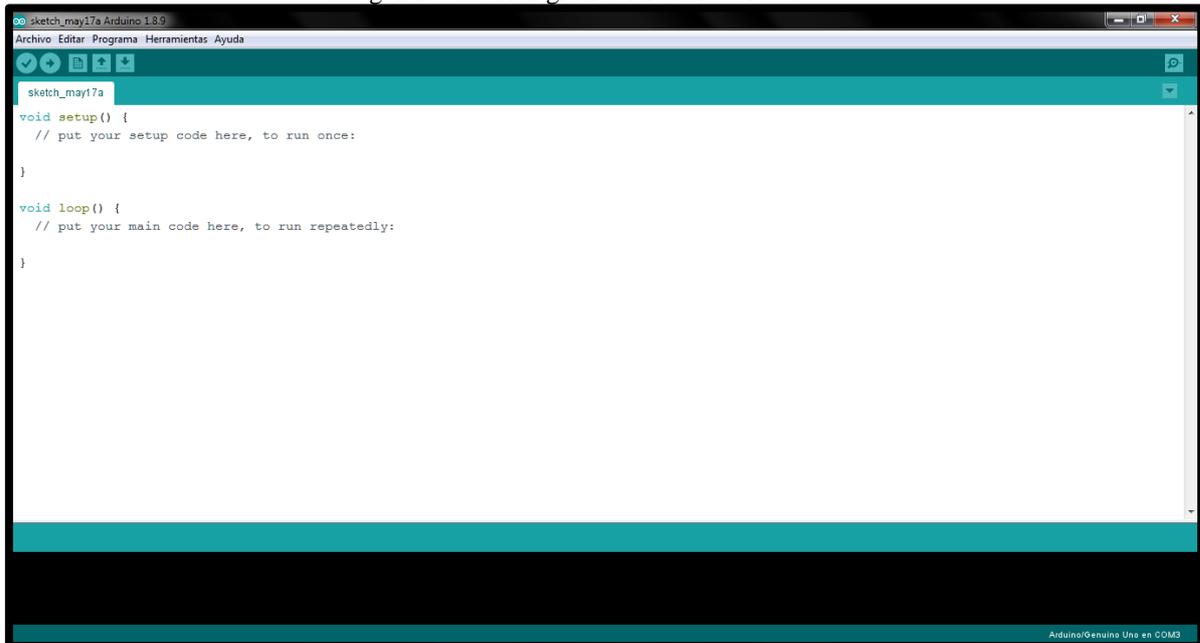


Fuente: Autores.

6 PROGRAMACIÓN EN SOFTWARE ARDUINO

Como se mencionó anteriormente, el código se programó con Arduino Software 1.8.9, este además de estar basado en Java permite visualizar y exportar los diferentes datos de mediciones con los que se trabajó posteriormente. A continuación se muestra una imagen de la interfaz gráfica de Arduino software.

Imagen 23: Interfaz gráfica Arduino software.

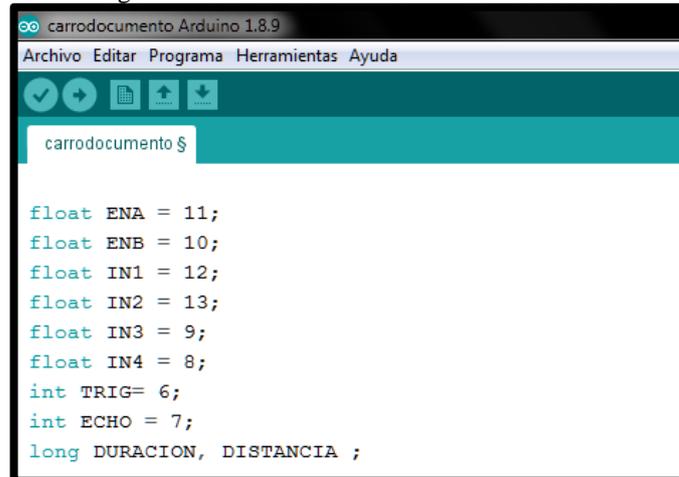


Fuente: Autores.

6.1 CREACIÓN DEL SOFTWARE PARA MOVIMIENTO DE LAS LLANTAS

Para comenzar, se crearon las variables, las cuales se crearon para controlar los pines que se conectaron anteriormente entre el Arduino Uno y el Puente H, por lo que las variables para controlar el Motor A fueron llamadas ENA, IN1, IN2 y las variables para controlar el Motor B fueron llamadas ENB, IN3 e IN4. Además de esto se adicionaron las variables "TRIG", "ECO", "DURACION" y "DISTANCIA".

Imagen 24: Variables creadas en Arduino software.

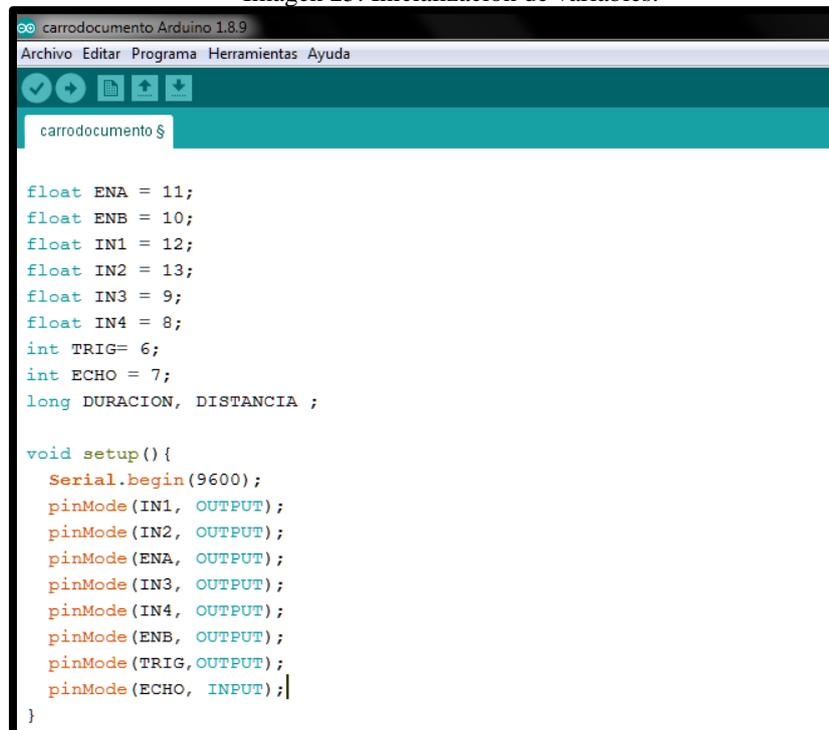


```
carrodocumento Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
carrodocumento $
float ENA = 11;
float ENB = 10;
float IN1 = 12;
float IN2 = 13;
float IN3 = 9;
float IN4 = 8;
int TRIG= 6;
int ECHO = 7;
long DURACION, DISTANCIA ;
```

Fuente: Autores.

6.2 Posteriormente, se inicializaron las variables en el void setup; para esto se asignó OUTPUT a todas las variables que controlan los pines, a excepción del “ECHO” el cual se le asignó la condición de INPUT; además de esto se creó el “Serial.begin” encargado de inicializar la conexión del arduino HC-05 en 9600 baudios.

Imagen 25: Inicialización de variables.



```
carrodocumento Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
carrodocumento $
float ENA = 11;
float ENB = 10;
float IN1 = 12;
float IN2 = 13;
float IN3 = 9;
float IN4 = 8;
int TRIG= 6;
int ECHO = 7;
long DURACION, DISTANCIA ;

void setup() {
  Serial.begin(9600);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
}
```

Fuente: Autores.

6.3 En tercer paso, se procedió a realizar el código para el accionar de los motores. Esto se posicionó el código en la función void loop, la cual se encarga de ejecutarlo. El código está compuesto en tres etapas, la primera se encarga de hacer que el sensor ultrasónico envíe y reciba los pulsos para determinar la distancia; la segunda encargada de convertir estos datos recolectados a una unidad de medida estándar, tal como lo es centímetros, posteriormente se agrega “Serial.println (DISTANCIA);” para imprimir los valores que obtenga el ultrasónico; por último se agregó un tiempo de retraso o delay entre cada dato obtenido por el sensor, a lo cual se le asignó un valor de 500 milisegundos.

Imagen 26: void loop en Arduino software.



```

carrodocumento Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
carrodocumento $
pinMode(ENA, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
pinMode(ENB, OUTPUT);
pinMode(TRIG, OUTPUT);
pinMode(ECHO, INPUT);
}

void loop(){
  //Sensor ultrasónico
  digitalWrite(TRIG,HIGH);
  delay(1);
  digitalWrite(TRIG,LOW);

  //Conversión de datos del sensor a centímetros
  DURACION = pulseIn(ECHO, HIGH);
  DISTANCIA= DURACION/58.2;
  //Imprimir datos sensor ultrasónico
  Serial.println(DISTANCIA);
  //Retraso entre datos obtenidos por el ultrasónico
  delay(500);

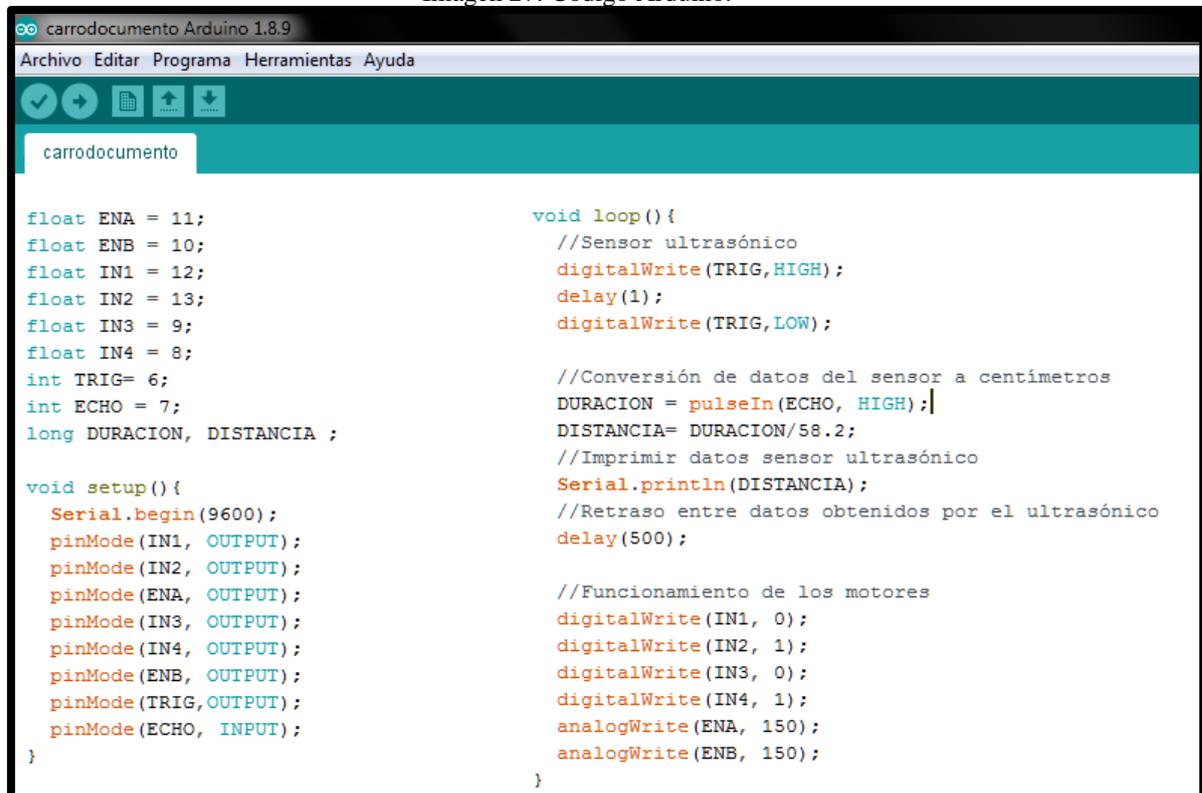
```

Fuente: Autores.

6.4 Por último, se creó el código de funcionamiento de los motores, para este se le asignó un número booleano a cada una de las variables a las que previamente se les fue asignado un pin en el Arduino Uno, este indica si el pin esta encendido (1) o apagado (0),

además de esto se asignó una velocidad inicial de 150, con esto se logró que el carro tomara dirección en línea recta y recolectará los datos de la distancia de él hacia el lugar de colisión más cercano. El código completo se muestra en la siguiente imagen:

Imagen 27: Código Arduino.



```

carrodocumento Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda

carrodocumento

float ENA = 11;
float ENB = 10;
float IN1 = 12;
float IN2 = 13;
float IN3 = 9;
float IN4 = 8;
int TRIG= 6;
int ECHO = 7;
long DURACION, DISTANCIA ;

void setup() {
  Serial.begin(9600);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(TRIG,OUTPUT);
  pinMode(ECHO, INPUT);
}

void loop() {
  //Sensor ultrasónico
  digitalWrite(TRIG,HIGH);
  delay(1);
  digitalWrite(TRIG,LOW);

  //Conversión de datos del sensor a centímetros
  DURACION = pulseIn(ECHO, HIGH);
  DISTANCIA= DURACION/58.2;
  //Imprimir datos sensor ultrasónico
  Serial.println(DISTANCIA);
  //Retraso entre datos obtenidos por el ultrasónico
  delay(500);

  //Funcionamiento de los motores
  digitalWrite(IN1, 0);
  digitalWrite(IN2, 1);
  digitalWrite(IN3, 0);
  digitalWrite(IN4, 1);
  analogWrite(ENA, 150);
  analogWrite(ENB, 150);
}

```

Fuente: Autores.

6.5 Para finalizar, se crearon diferentes derivaciones de “//Funcionamiento de los motores” con la intención de generar códigos para cada evento, estos pueden ser enlazados uno tras de otro, con la finalidad de crear infinitas rutas posibles para la medición de cm del carro con Arduino. A continuación se muestran los diferentes códigos generados junto con el respectivo evento que ejecutan.

Tabla 1: Eventos correspondientes a cada derivación de código.

<p>Avanzar durante 10 segundos</p>	<pre>digitalWrite(IN1, 0); digitalWrite(IN2, 1); digitalWrite(IN3, 0); digitalWrite(IN4, 1); analogWrite(ENA, 150); analogWrite(ENB, 150); delay (10);</pre>
<p>Retroceder durante 10 segundos</p>	<pre>//Retroceder digitalWrite(IN1, 1); digitalWrite(IN2, 0); digitalWrite(IN3, 1); digitalWrite(IN4, 0); analogWrite(ENA, 150); analogWrite(ENB, 150); delay (10);</pre>
<p>Girar 90° a la izquierda</p>	<pre>//Girar izquierda digitalWrite(IN1, 0); digitalWrite(IN2, 1); digitalWrite(IN3, 1); digitalWrite(IN4, 0); analogWrite(ENA, 150); analogWrite(ENB, 150); delay (0,5);</pre>
<p>Girar 90° a la derecha</p>	<pre>//Girar derecha digitalWrite(IN1, 1); digitalWrite(IN2, 0); digitalWrite(IN3, 0); digitalWrite(IN4, 1); analogWrite(ENA, 150); analogWrite(ENB, 150); delay (0,5);</pre>
<p>Parar durante 10 segundos</p>	<pre>//Parar digitalWrite(IN1, 0); digitalWrite(IN2, 0); digitalWrite(IN3, 0); digitalWrite(IN4, 0); analogWrite(ENA, 150); analogWrite(ENB, 150); delay (10);</pre>

Fuente: Autores.

7 RESULTADOS OBTENIDOS

Luego de los anteriores pasos se realizó una prueba de funcionamiento al prototipo. Estas pruebas se realizaron en una habitación con suelo de baldosa en la cual se colocó el carro a circular en línea recta y a diferentes velocidades en contra de una

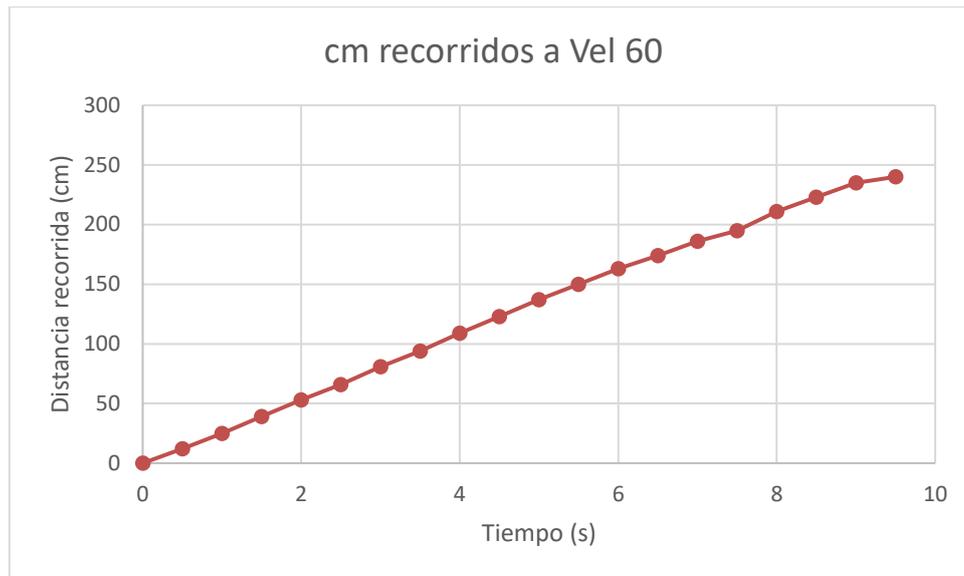
placa de poliestireno expandido con tamaño de 1x1 ubicada aproximadamente a 140 cm de distancia. En la prueba se observó que el sensor puede obtener algunos datos atípicos los cuales se muestran en rojo. Estos datos fueron almacenados y se presentan en la siguiente tabla:

Tabla 2: Datos obtenidos prueba funcional del carro

Tiempo (Seg)	cm recorridos a Vel 60	cm recorridos a Vel 70	cm recorridos a Vel 80	cm recorridos a Vel 90	cm recorridos a Vel 100	cm recorridos a Vel 110	cm recorridos a Vel 120	cm recorridos a Vel 130	cm recorridos a Vel 140	cm recorridos a Vel 150
0	0	0	0	0	0	0	0	0	0	0
0,5	12	8	12	11	37	15	17	17	106	0
1	25	24	189	34	62	42	44	183	189	193
1,5	39	41	50	74	90	69	72	77	78	116
2	53	60	70	193	212	98	100	107	189	76
2,5	66	77	90	102	141	125	131	138	138	111
3	81	95	110	126	191	153	158	165	169	178
3,5	94	114	130	148	190	176	186	195	198	174
4	109	131	150	170	214	204	213	222	227	204
4,5	123	149	169	192	237	227	239	239	239	232
5	137	164	187	213	239	239	240	240	240	238
5,5	150	179	206	236	240	240	240	240	240	239
6	163	198	224	239	240	240	240	240	240	240
6,5	174	213	239	240	240	240	240	240	240	240
7	186	228	239	240	240	240	240	240	240	240
7,5	195	239	240	240	240	240	240	240	240	240
8	211	240	240	240	240	240	240	240	240	240
8,5	223	240	240	240	240	240	240	240	240	240
9	235	240	240	240	240	240	240	240	240	240
9,5	240	240	240	240	240	240	240	240	240	240

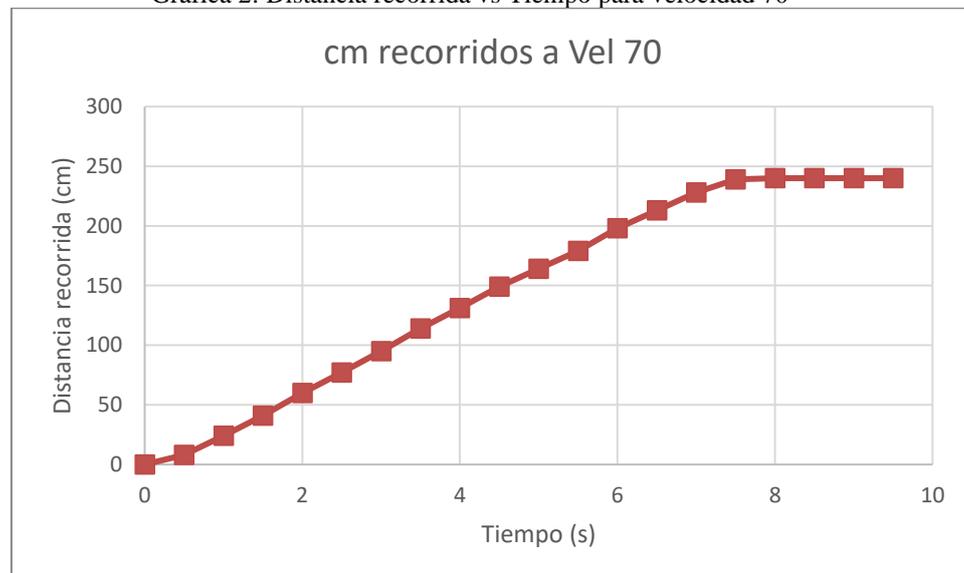
Fuente: Autores.

Gráfica 1: Distancia recorrida vs Tiempo para velocidad 60



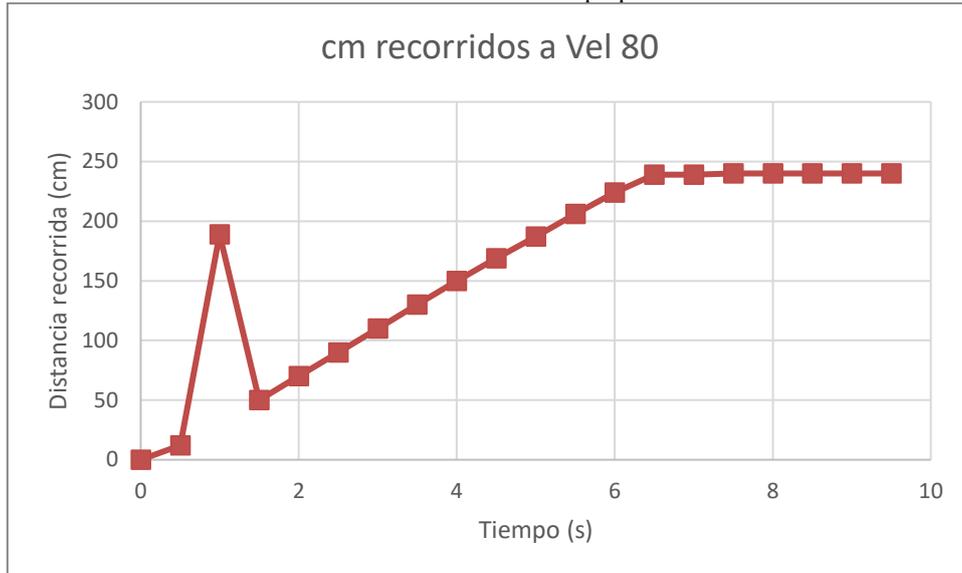
Fuente: Autores.

Gráfica 2: Distancia recorrida vs Tiempo para velocidad 70



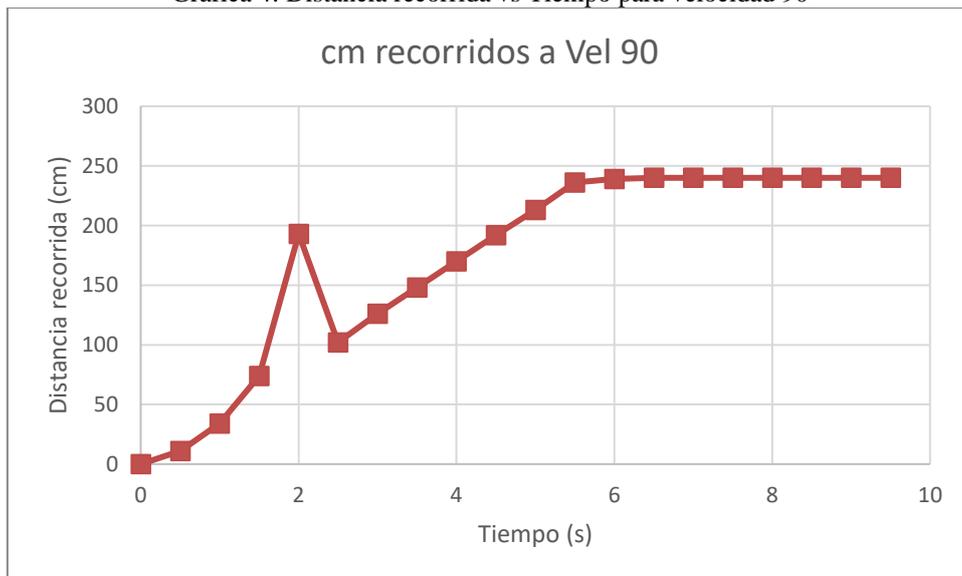
Fuente: Autores.

Gráfica 3: Distancia recorrida vs Tiempo para velocidad 80



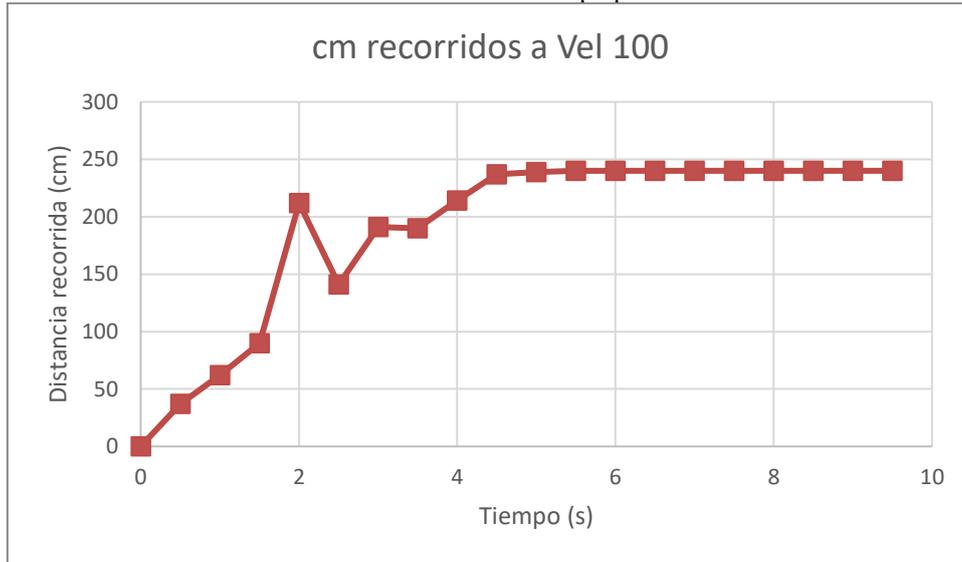
Fuente: Autores.

Gráfica 4: Distancia recorrida vs Tiempo para velocidad 90



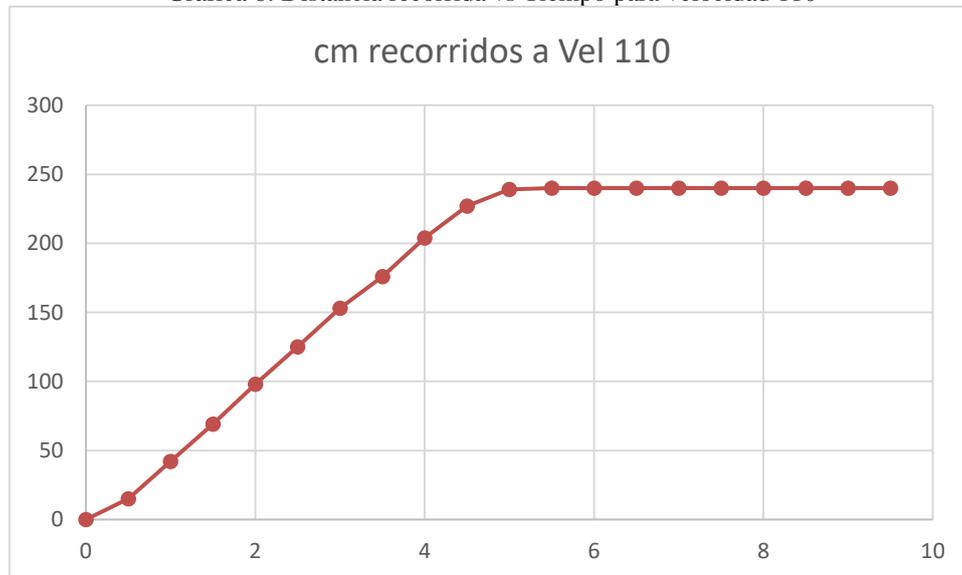
Fuente: Autores.

Gráfica 5: Distancia recorrida vs Tiempo para velocidad 100



Fuente: Autores.

Gráfica 6: Distancia recorrida vs Tiempo para velocidad 110



Fuente: Autores.

Gráfica 7: Distancia recorrida vs Tiempo para velocidad 120



Fuente: Autores.

Gráfica 8: Distancia recorrida vs Tiempo para velocidad 130



Fuente: Autores.

Gráfica 9: Distancia recorrida vs Tiempo para velocidad 140



Fuente: Autores.

Gráfica 10: Distancia recorrida vs Tiempo para velocidad 150



Fuente: Autores.

7.1 ANÁLISIS DE RESULTADOS

De las gráficas de distancia recorrida vs tiempo, se puede observar que en la mayoría de ellas (velocidad: 60, 70, 110, 120) presentan datos lógicos, con una línea prácticamente recta. Otro grupo de gráficas (velocidad: 80, 90, 130) presenta 1 dato desviado con respecto a la tendencia de línea recta. La gráfica de velocidad 100 presenta 2 datos desviados con respecto a la tendencia rectilínea. La gráfica de velocidad 140 presenta 3 datos fuera de la tendencia rectilínea. Y la gráfica de velocidad 150 presenta unos 6 datos desviados de la tendencia rectilínea.

La primera conclusión que se puede inferir de las gráficas es que a una velocidad de 150, se presentan demasiados datos desviados y se podría pensar en no usar esta velocidad en los laboratorios sino solo hasta 140. Si de los demás grupos de datos se eliminan de la gráfica los datos desviados se obtienen líneas rectas, lo cual permite observar la velocidad promedio en cada escenario. En Excel se puede obtener la ecuación de la línea recta de cada gráfica y la pendiente de cada línea recta representa la velocidad respectiva, medida en cm/s. En la siguiente tabla se muestran las pendientes de cada gráfica (velocidad), eliminando los datos desviados de la tendencia rectilínea. Para ello se usó la fórmula de Excel: =ESTIMACION.LINEAL (P44:P59;M44:M59), donde los datos azules corresponden al eje Y (distancia recorrida) y los datos verdes corresponden al eje X (tiempo).

7.2

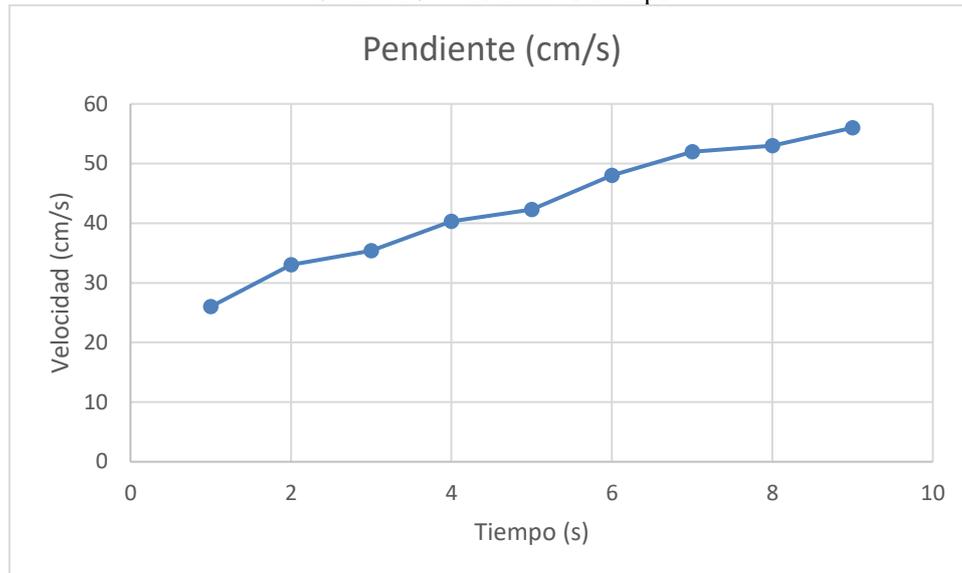
Tabla 3: Pendientes de las rectas obtenidas en cada escenario.

Escenario	Pendiente (cm/s)
Pendiente (vel 60)	26
Pendiente (vel 70)	33
Pendiente (vel 80)	35,4
Pendiente (vel 90)	40,3
Pendiente (vel 100)	42,3
Pendiente (vel 110)	48
Pendiente (vel 120)	52
Pendiente (vel 130)	53
Pendiente (vel 140)	56
Pendiente (vel 150)	NA

Fuente: Autores.

Otro ejercicio que se puede realizar con los datos anteriores es graficar cada una de las velocidades obtenidas en cada escenario, asumiendo que entre cada escenario transcurriera 1 segundo. Así, la gráfica de velocidad (cm/s) vs tiempo (s) mostrará una línea cuya pendiente sería la aceleración o ritmo con que la velocidad ha aumentado de un escenario a otro. En la siguiente gráfica se puede observar este comportamiento.

Gráfica 10: Velocidad vs Tiempo



Fuente: Autores.

En el gráfico 10 se observa que la pendiente varía entre cada punto, por lo tanto la aceleración igualmente es variable.

8 CONCLUSIONES

Se debe aplicar en el aula de clase de Física Mecánica el laboratorio propuesto en este artículo informe de investigación. Se debe evaluar con algún método referenciado el impacto del uso del carro Arduino en el proceso enseñanza - aprendizaje de los estudiantes de la Facultad de Ingenierías de la Corporación Universitaria Americana, sede Medellín y posteriormente realizar el registro y análisis de los resultados obtenidos en un nuevo artículo de investigación.

REFERENCIAS

Barrachina, S., Fabregat, G., Martí, J. (2015). Utilizando Arduino Due en la docencia de la entrada/salida. Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática, 58-65.

Campo, D., Sastre, M., Gamas, I., Samaniego, E. (2017). Diseño y construcción de un prototipo robótico de un automóvil para transporte personal y económico con la tecnología Arduino. Revista de Iniciación Científica, Journal of Undergraduate Research (3), 110 – 123.

Christiansen, R., Miguel, F., Agüero, E., Pereyra, N. (2016). Revista de Enseñanza de la Física. (28), 23-28.

Celin, W., Solano, C., Molina, J. (2017). Plano Inclinado con dos sensores para la enseñanza del movimiento rectilíneo uniformemente acelerado. Revista Espacios, (38), 9-19.

Gómez, C., Castillo, A., Gómez, A. (2015). Arduino como una herramienta para mejorar el proceso de enseñanza – aprendizaje de las ciencias, tecnologías e ingenierías en la Universidad Politécnica De Tapachula. Revista QUID, (24), 13-20.

Larrazza-Mendiluze, E., Garay, N., Martin, J., Muguerza, J., Ruiz-Vazquez, T., Soraluze, I., Lukas, J., Santiago, K. (2013). Game-Console-Based Projects for Learning the Computer Input/Output Subsystem, IEEE Transactions on Education (56), 453–458, 2013.

Llera, M., Scagliotti, A., Jorge, G. (2017). Conectando ciencias: interfaces educativas usando el entorno Arduino. Revista de Enseñanza de la Física (29), 381–389.

Rubio, M., Mañoso, C., Pérez, A. (2013). Using Arduino To Enhance Computer Programming Courses In Science And Engineering. In Proceedings of the EDULEARN13. Conference, Barcelona, 5127–5133.