

Anexos

Anexo A. Consentimientos informados.

PROTOCOLO DE COMPROMISO ÉTICO Y ACEPTACIÓN DE LAS PARTICIPANTES EN LA INVESTIGACIÓN

Robótica educativa una propuesta de enseñanza para la construcción de significados en física.

En los últimos años, se ha visto un gran avance en las tecnologías de la información y la comunicación TIC en nuestro país, en la ciudad; y el colegio Santa clara de Asís no es ajeno a esto, es por lo anterior que se pretende implementar la robótica educativa en el curso de física de las estudiantes del grado décimo; que tiene como objetivo identificar las distintas dificultades y aciertos en el aprendizaje de los conceptos en física, generar diferentes espacios y contextos actuales que vayan de la mano de la ciencia mediante las diferentes tecnologías y la construcción de significados a partir del aprendizaje colaborativo.

Por tanto, las estudiantes del colegio Santa clara de Asís se beneficiarán de participar en la propuesta de intervención y así acercarse al conocimiento de las nuevas tecnologías en la construcción del conocimiento científico y de nuevos significados en ciencias.

la propuesta de investigación se realizará durante el cuarto período del año escolar que corresponde a partir del 24 de septiembre del presente año. durante la investigación el proceso de intervención se pretende llevar a cabo toma de datos o recolección de información propias de la investigación cualitativa, las cuales constan de observación, entrevistas semiestructuradas videos, fotos, actividades propuestas y diarios de campo.

De ser aprobada la propuesta de intervención por el padre de familia, es de vital importancia que el participante se comprometa con el tiempo que se invertirá en la investigación y con los diferentes datos que han de tomarse (fotos, videos, entrevistas) que aporten contenido al proceso de intervención. De igual manera, la estudiante que no desee participar en el proceso de intervención se requiere también realizar las actividades pactadas, ya que hacen parte del proceso de evaluación de desempeños del colegio correspondientes al período en curso.

Durante los procesos de recolección de datos, se hará uso adecuado y discrecional de la información recolectada con el fin de lograr los objetivos propuestos en el trabajo de investigación. los datos recogidos sólo serán evaluados por los docentes investigadores quienes recolectaron la información necesaria para la correcta construcción de éste, evitando la alusión de nombres propios y valorando el aporte de cada uno de los participantes, haciendo saber los resultados finales, los cuales serán publicados en el informe final de la investigación.

Habiendo conocido la información anterior, autoriza la participación del estudiante _____ en el proceso de investigación robótica educativa: una propuesta de enseñanza para la construcción de significados en física.

Firma padre de familia o acudiente.

Firma estudiante participante.

PROTOCOLO DE COMPROMISO ÉTICO Y ACEPTACIÓN DE LOS REPRESENTANTES DE LA INSTITUCIÓN EDUCATIVA SANTA CLARA DE ASÍS

Nombre de la Investigación: Propuesta de enseñanza para la construcción de significados en física con el modelo pedagógico de robótica educativa (MOPRE)

Investigadores: Sergio Andrés Correa Díez
Alejandra Pérez Arango

Grupo investigado: Décimo

Presentamos ante ustedes nuestro compromiso ético. Entendemos como imperativo y debe, hacer uso adecuado y discrecional de la información recolectada en el marco de esta investigación, con el único fin de lograr los objetivos del estudio en cuestión y en la perspectiva de contribuir con aportes para el mejoramiento de la enseñanza de la Física a partir de la implementación de la Robótica Educativa como herramienta facilitadora para la adquisición de nuevos conocimientos a través de una propuesta didáctica en el contexto de la Institución Educativa Santa Clara de Asís, así como contribuir con cuestiones teóricas y metodológicas a la línea de investigación en Tecnologías de la Información y la Comunicación para la Enseñanza de las Ciencias y la Matemática del programa Licenciatura en Matemáticas y Física de la Universidad de Antioquia.

El uso discrecional y adecuado de la información recogida y de su análisis, implica que la misma sólo será utilizada para los propósitos enunciados en el marco de este trabajo investigativo, que se evitará la alusión a nombres propios y se valorará con respeto y responsabilidad los aportes de cada uno de los participantes. Los análisis y resultados serán dados a conocer en primera instancia a los participantes.

Desde esta perspectiva, las personas que firman este documento autorizan a los investigadores para que las fuentes de información como escritos, entrevistas, observaciones, fotos, grabaciones en audio y video, etc.; se constituyan en bases de datos para dicha investigación.

Gracias por su atención y colaboración.

FIRMA COORDINADORA

FIRMA DIRECTORA DE GRUPO

FIRMA DEL RECTOR

Anexo B. Preguntas entrevista semiestructurada tercera fase.

1. ¿De qué manera le aportó el trabajo colaborativo durante el proyecto?
2. ¿Cuál fue la experiencia que tuvo al utilizar el lenguaje de programación para el robot?

3. A partir del lenguaje de programación ¿qué conceptos de física le ayudó a comprender?
4. ¿Cómo valora la experiencia con robótica educativa para la enseñanza de la física?

Duración: 1 hora

Anexo C. Formato de preguntas diagnóstico.

Actividad diagnóstica: A continuación, se presentan algunas preguntas relacionadas con robótica. Se tiene como propósito realizar una serie de cuestionamientos que se puedan argumentar de forma breve y oral, para socializar grupalmente desde lo que cada estudiante pueda saber sobre robótica y de llegar a una sola conclusión a partir de la negociación de significados.

1. ¿Qué creen que la robótica?
2. ¿Cuáles son las posibles aplicaciones que tiene la robótica?
3. En el colegio Santa Clara de Asís la mayoría de las clases de educación física se están desarrollando dentro del aula de clase, debido a la contaminación del aire que se presenta en Medellín. Teniendo en cuenta esta problemática formule y planifique una posible solución por medio de un prototipo robótico, el cual se pueda ejecutar como un proyecto en la institución.
4. ¿Qué áreas creen que aportarían al desarrollo del proyecto con robótica?
5. Si en sus manos está diseñar un robot. ¿Cuál sería y qué problema solucionaría?

COLEGIO SANTA CLARA DE ASÍS

Práctica experimental

Objetivo: Identificar los conocimientos previos de los estudiantes sobre electrónica y robótica.

Actividad inicial #1: Se representa una secuencia experimental, en donde el docente desempeña el papel de guía en la construcción de un circuito en serie y paralelo, con el fin de que las estudiantes logren identificar algunos conceptos importantes de la electrónica a partir de la negociación de significados que surge por la interacción de los grupos y el docente, el cual son necesarios tenerlos en cuenta a la hora de hacer el ensamblaje del prototipo robótico. Finalmente, durante el proceso se realizarán una serie de preguntas relacionadas al experimento que las estudiantes deberán responder de forma oral.

Materiales

- Protoboard.
- Leds.
- Cables jumper (macho – macho).
- Resistencias.
- Batería de 9v.
- Porta batería.

Procedimiento: Las estudiantes deben conformar grupos de tres integrantes para el montaje del circuito con los materiales previamente mencionados. Luego se hace una breve explicación del manejo de la Protoboard como el siguiente: se ha encontrado que se tienen diferentes términos para definir el mismo concepto; en este caso la Protoboard se divide en dos partes diferentes: buses y pistas, los buses tienen conexión a lo largo de toda la sección roja mientras que las pistas (sección azul) deben ser conectadas adicionalmente por un cable jumper para poder electrizar esta zona.

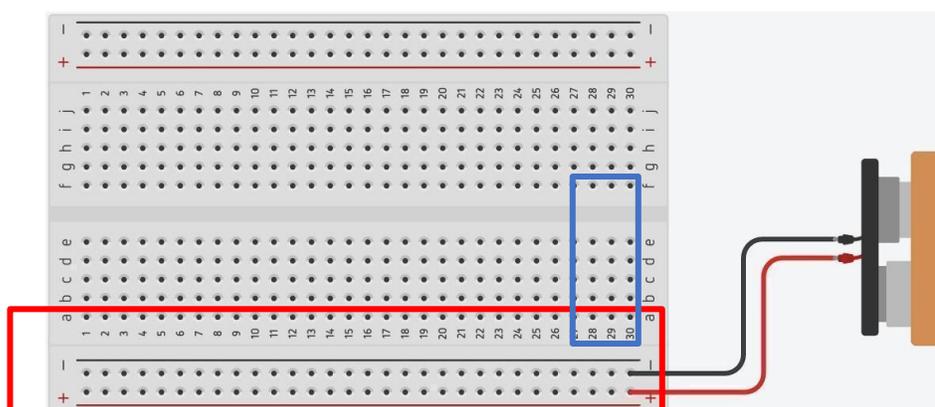
Ahora bien, se pide a las estudiantes que conecten la batería a la Protoboard, teniendo en cuenta que el cable rojo está conectado a la parte positiva (+) y el cable negro conectado en la parte negativa (-). Luego se debe ubicar el led y la resistencia de tal forma que la corriente circule y se complete el circuito haciendo uso de la batería e identificando la importancia de su utilización; en este punto es importante que las estudiantes dialoguen entre sí sobre el procedimiento adecuado de realizar las conexiones; después de tener esto

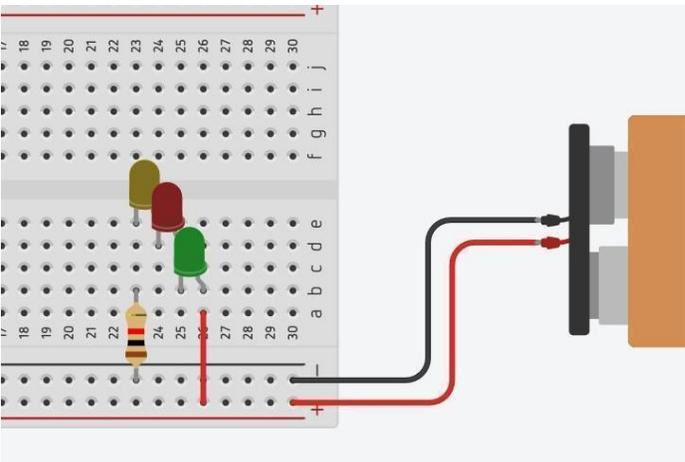
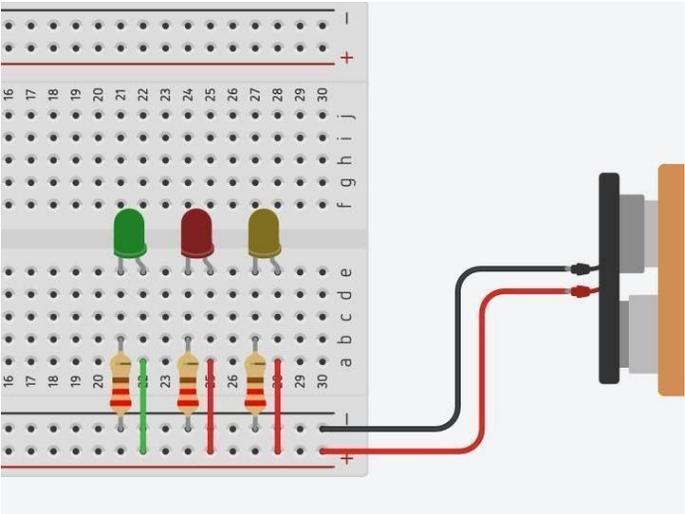
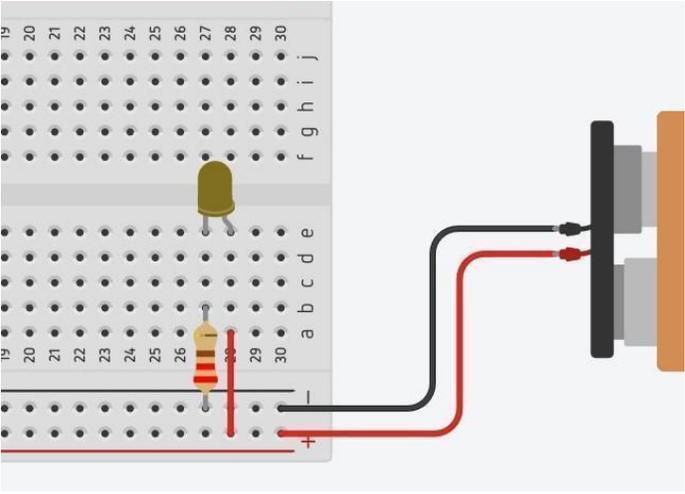
listo se pide realizar el mismo proceso, pero utilizando más leds con el fin de establecer diferencia y característica cuando estos están en serie y en paralelo.

Importante que el docente mientras las estudiantes hacen el respectivo montaje realice preguntas y planteamientos tales como:

- Desde el punto de vista físico, ¿qué creen que sucede en el fenómeno?
- Teniendo en cuenta lo observado, mencionar una definición con sus propias palabras de lo que sería entonces un circuito.
- ¿Cuál es la diferencia que establecen entre circuito en serie y paralelo?
- ¿Qué papel representa el voltaje y la corriente en el circuito elaborado?
- ¿De qué creen que depende la intensidad lumínica de los leds?
- Dentro de cada uno de los equipos identifiquen los posibles escenarios reales donde se logre observar el fenómeno que se realizó y mencionar que otros conceptos de física están relacionados.

A continuación, se anexan algunas imágenes que ayudan como guía al docente para ejemplificar mejor el montaje del circuito.





Anexo E. Formato programación pseudocódigo en PSeInt primer momento.

COLEGIO SANTA CLARA DE ASÍS
Práctica programación con pseudocódigo

Objetivo: Realizar un acercamiento a los conceptos básicos en la lógica de programación.

Actividad inicial #2: En esta etapa el docente debe tomar en gran medida la dirección de la actividad, el cual se torna un poco tradicional, ya que se muestra paso a paso cómo funciona el programa PseInt y resalta la importancia del uso adecuado del lenguaje de signos y símbolos que ofrece la programación. Se presentan una serie de ejercicios relacionados con movimiento rectilíneo, movimiento parabólico, ley de Ohm, entre otros; en donde el docente explica y acompaña los primeros tres problemas y luego se deja que las estudiantes los resuelvan solas.

Materiales

- Computador
- Link para descargar de PseInt (<http://pseint.sourceforge.net/>)

Procedimiento: Se debe tener el programa ya instalado en cada computador, luego de abrir el programa interactúa un poco conociendo las diferentes opciones. Después se hace una breve explicación de todo lo que se puede hacer en PSeInt (en este punto se hace un pequeño resumen explicando para que sirve el programa y sus respectivas opciones de funcionamiento, además se hace referencia a la lógica matemática y la relación con la programación). Posteriormente se plantean los siguientes problemas relacionados con la física y resolverlos a través de códigos de programación y observar lo que sucede dentro del programa.

- a. Describa el paso a paso de la siguiente pregunta, haciendo uso de una secuencia lógica. ¿A qué velocidad debe circular un auto de carreras para recorrer 1200 m en un cuarto de hora?

A continuación, se plantea una situación la cual debe ser resuelta por medio de un algoritmo en el programa PSeInt relacionado con un problema de física en específico.

- b. Calcular la magnitud de la aceleración que produce una fuerza cuya magnitud es de 50 N a un cuerpo cuya masa es de 13000 gramos.

```
1 Algoritmo SegundaLeyNewton
2 Definir fuerza, aceleración, masa Como Real;
3 Escribir "Ingrese la fuerza:"
4 Leer fuerza;
5 Escribir "Ingrese la masa del cuerpo en kilogramos:"
6 Leer masa;
7 aceleración= fuerza/masa;
8
9 Escribir "La magnitud de la aceleración que se produce es:", aceleración, "m/s^2";
10
11 FinAlgoritmo
12
```

- c. Calcula la resistencia atravesada por una corriente con una intensidad de 5 amperios y una diferencia de potencial de 11 voltios.

```
1 Algoritmo LeyDeOhm
2 Definir corriente,resistencia, voltaje Como Real;
3 Escribir "Ingrese el voltaje:"
4 Leer voltaje;
5 Escribir "Ingrese el valor de la corriente:"
6 Leer corriente;
7 resistencia= voltaje/corriente;
8
9 Escribir "El valor de la resistencia es:", resistencia, "Ohm";
10
11 FinAlgoritmo
12
```

Ahora se plantean los siguientes problemas para que las estudiantes resuelvan teniendo lo aprendido previamente y haciendo uso de la lógica y de los pseudocódigos trabajados hasta el momento.

- Un ciclista que está en reposo comienza a pedalear hasta alcanzar los 16.6km/h en 6 minutos. Calcular la distancia total que recorre si continúa acelerando durante 18 minutos más.
- Desde una ventana de un edificio situada a 20 m del suelo se lanza una pelota con una velocidad de 15 m/s formando un ángulo de 60^0 con la horizontal. Determinar: ¿Cuál es la altura máxima alcanzada? y b) calcular el tiempo en alcanzar el suelo y el alcance.

Finalmente, se hace un diálogo entre todas las estudiantes respecto a lo aprendido durante la actividad y la relación que estas encuentren con la robótica y la programación.

Anexo F. Formato presentación tipos de sensores.

COLEGIO SANTA CLARA DE ASÍS

Presentación tipos de sensores

Objetivo: Identificar los tipos de sensores y su aplicación en la cotidianidad para la solución de problemas reales.

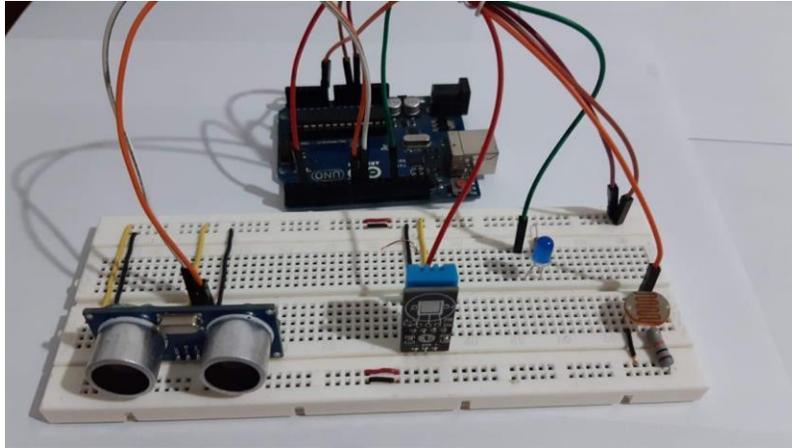
Actividad MOPRE #1: En este punto el docente es quien orienta el desarrollo de la actividad; éste realiza una presentación sobre el uso de los diferentes tipos de sensores que se podrán tener en cuenta para la construcción del prototipo robótico y puntualizar en la funcionalidad de cada uno.

Materiales

- Computador
- Link para descargar de Arduino (<https://www.arduino.cc/en/main/software>)
- Sensor de luz: Fotorresistencia, cables jumper, protoboard, Arduino, led y lcd interfaz I2C.
- Sensor de humedad y temperatura: sensor temperatura, sensor humedad, cables jumper, protoboard, Arduino, led.
- Sensor medidor de distancia: sensor ultrasónico HC-05, cables jumper, protoboard, Arduino, led.
- Sensor seguidor de línea: sensor infrarrojo seguidor de línea o sensor óptico seguidor de línea, cables jumper, protoboard, Arduino.

Descripción: El docente hace una presentación mostrando cada uno de los sensores que se pueden tener en cuenta para la construcción del prototipo robótico. Primero describe cómo se realiza el montaje en la protoboard de cada sensor; posteriormente, el docente realiza una breve explicación del código de programación que se tuvo en cuenta para cada uno de los sensores y la aplicación que tiene este en la realidad, con el fin de que las estudiantes comprendan mejor su funcionamiento; finalmente orienta de forma general cuáles son los posibles proyectos que se pueden llevar a cabo con cada uno de los sensores.

Montaje de los sensores:



De ser necesario el docente abre un espacio para resolver dudas particulares para cada uno de los equipos que ayude a ubicar a los estudiantes en la construcción final del prototipo.

Luego se pide a las estudiantes que por medio del diario de campo respondan a los siguientes cuestionamientos relacionados con la elección del tipo de sensor que cada grupo realizará para el robot; además, la planificación, diseño y explicación del funcionamiento que este tendrá. Aquí el docente asesora a cada uno de los grupos y explica que las preguntas se pueden responder por medio de argumentos breves, dibujos o imágenes que les ayudarán en la etapa de construcción del robot. Luego que los grupos terminen se hace una socialización general de lo realizado.

Preguntas diario de campo:

1. ¿Cuál sería el prototipo robótico que se desea realizar en el proyecto?
2. ¿Cuál sería la funcionalidad del robot?
3. ¿Qué tipo de sensor elegirían para el prototipo robótico? ¿por qué?
4. ¿Qué conceptos de física se aplican en el robot?

COLEGIO SANTA CLARA DE ASÍS
Planificación y organización del proyecto

Objetivo: Indagar el comportamiento de fenómenos existentes, planificar las estrategias para modificar o diseñar una nueva forma de aplicación al contexto.

Actividad MOPRE #2: Iniciar la planificación del proyecto mediante el diario campo, donde se describa el tipo de sensor a elegir, el diseño para el prototipo robótico y su funcionalidad.

Descripción: Para el desarrollo de esta actividad es fundamental que las estudiantes se agrupen con las personas que realizarán el proyecto, y por ende en la construcción del prototipo. Luego se les pide que por medio del diario de campo describan el proceso que se llevará a cabo para el diseño, construcción y funcionalidad del robot. Para esto deberán tener en cuenta lo trabajado hasta el momento y realizar búsqueda de información de ser necesario. Además, se explica que a las preguntas le pueden dar respuesta por medio de argumentos breves, dibujos o imágenes que les ayudarán en la siguiente etapa. Luego que los grupos terminen se hace una socialización general de lo realizado.

En el diario de campo se debe tener lo siguiente:

- Realice un boceto del robot.
- Materiales que se necesitan para la construcción el robot.
- Describa de manera breve cómo sería la construcción del robot.
- Describa el papel que desempeñaría cada uno de los miembros del equipo.
- Enuncie posibles dificultades que se podrían presentar a la hora de construir y de programar el robot y cómo se podrían resolver.

Duración: 2 horas

Anexo H. construcción y programación del prototipo robótico.

COLEGIO SANTA CLARA DE ASÍS

Construcción y programación de robot

Objetivo: Fortalecer el trabajo en equipo a través del análisis de diversas opciones para construir el prototipo robótico y estimular habilidades psicomotoras y psicológicas.

Explorar y aprender el funcionamiento del software con nuevas posibilidades de usar el lenguaje de programación.

Actividad MOPRE #3 y #4: Ensamble del prototipo robótico y realización del código de programación para el funcionamiento del robot utilizando la plataforma Arduino.

Materiales: (depende de los sensores elegidos y del proyecto a ejecutar por las estudiantes, se muestran algunos ejemplos)

- **Robot guiado por Bluetooth:**

Portapilas

Módulo bluetooth

Cargador de batería

Robot circular

Puente H

Cables macho-hembra (10-15)

Batería de litio (2)

- **Robot esquivador de obstáculos:**

Portapilas

Base para sensor ultrasonido

Cargador batería

Robot car

Puente H

Cables macho-hembra (10-15)

Batería de litio (2)

Kit de tornillos

Ultrasonido

- **Robot seguidor de línea:**

Porta baterías

Cargador batería

Robot 4 llantas

Puente H

Cables macho-hembra (10-15)


```

#define MOTOR_I_ADELANTE digitalWrite(PIN_A_MOTOR_I, HIGH); digitalWrite(PIN_B_MOTOR_I, LOW)

#define MOTOR_I_ATRAS    digitalWrite(PIN_A_MOTOR_I, LOW); digitalWrite(PIN_B_MOTOR_I, HIGH)

#define MOTOR_I_PARAR    digitalWrite(PIN_A_MOTOR_I, HIGH); digitalWrite(PIN_B_MOTOR_I, HIGH)

#define MOTOR_D_ADELANTE digitalWrite(PIN_A_MOTOR_D, HIGH); digitalWrite(PIN_B_MOTOR_D,
LOW)

#define MOTOR_D_ATRAS    digitalWrite(PIN_A_MOTOR_D, LOW); digitalWrite(PIN_B_MOTOR_D, HIGH)

#define MOTOR_D_PARAR    digitalWrite(PIN_A_MOTOR_D, HIGH); digitalWrite(PIN_B_MOTOR_D, HIGH)

//Se combinan los movimientos de modo que se crean los comandos de movimiento enteros

#define CARRO_PARAR      MOTOR_I_PARAR;  MOTOR_D_PARAR //Para detener el carro se paran el
motor izquierdo y derecho

#define CARRO_ADELANTE  MOTOR_I_ADELANTE; MOTOR_D_ADELANTE

#define CARRO_ATRAS     MOTOR_I_ATRAS;  MOTOR_D_ATRAS

#define CARRO_IZQUIERDA MOTOR_D_ADELANTE; MOTOR_I_ATRAS //Para girar, un motor se dirige
adelante y el otro atras

#define CARRO_DERECHA   MOTOR_I_ADELANTE; MOTOR_D_ATRAS

//La energia que reciben los PIN EN son la que dicta la velocidad

//Esta es el cambio de velocidad y es enteramente cambiabile el numero para modificarle la velocidad deseada

#define LENTO_VELOCIDAD analogWrite(PIN_EN_MOTOR_I, 85); analogWrite(PIN_EN_MOTOR_D, 85

//Para que gire en unas lineas muy cortantes por ejemplo se pone la velocidad en 80, y si son loineas simples se sube la
velocidad, es solo cosmetico subirle la velocidad

#define MEDIA_VELOCIDAD analogWrite(PIN_EN_MOTOR_I, 210); analogWrite(PIN_EN_MOTOR_D, 210)
//El giro debe ser mas rapido que el aceleras ya que detiene el momento que consiguio, pero no muy alto porque se sale
de la linea

#define TODA_VELOCIDAD analogWrite(PIN_EN_MOTOR_I, 255); analogWrite(PIN_EN_MOTOR_D, 255)
//Esta no la uso pero esta ahi si se quiere ir a todo el potencial

int BOTON = A15;int sal = 0;

int IZQUIERDA=0; int DERECHA=0; int CENTRO=0; //No usé el sensor del centro

void setup() {

  Serial.begin(115200);

```

```

pinMode(PIN_EN_MOTOR_I, OUTPUT);

pinMode(PIN_A_MOTOR_I, OUTPUT);

pinMode(PIN_B_MOTOR_I, OUTPUT);

pinMode(PIN_EN_MOTOR_D, OUTPUT);

pinMode(PIN_A_MOTOR_D, OUTPUT);

pinMode(PIN_B_MOTOR_D, OUTPUT);

pinMode(A0,INPUT); pinMode(A1,INPUT); pinMode(A2,INPUT); MEDIA_VELOCIDAD;

//Es el codigo del boton para iniciar el carro, se puede quitar solo borrando esta parte
pinMode(52,OUTPUT); pinMode (A15,INPUT); digitalWrite(52,HIGH);

BOTON = digitalRead (A15);

while(BOTON==LOW){

BOTON = digitalRead (A15);

}

}

void loop(){ //Uso el WHILE en vez del IF, porque el WHILE es instantaneo y el IF hace que se demore unos
milisegundos en dejar de girar

analogWrite (12,10);

DERECHA=analogRead (A2);

while(DERECHA<35){

CARRO_DERECHA;

MEDIA_VELOCIDAD;

DERECHA=analogRead (A2);

}

IZQUIERDA=analogRead (A0);

while(IZQUIERDA<35){

CARRO_IZQUIERDA;

MEDIA_VELOCIDAD;

IZQUIERDA=analogRead (A0);

}

CARRO_ADELANTE;

LENTO_VELOCIDAD; //Gira en media velocidad y avanza mas lento para que pueda reaccionar

```

```
}
```

Bluetooth

```
//UNCOMMENT THE LINE OF THE ARDUINO WHAT YOU ARE USING

#define USING_ARDUINO_MEGA

#define USING_ARDUINO_UNO

#include "SerialManager.h"

#ifndef USING_ARDUINO_UNO
#include <SoftwareSerial.h>
#endif

/*****Definitions*****/

#define serial_pc Serial

#ifdef USING_ARDUINO_UNO
SoftwareSerial serial_bluetooth(10, 11); // RX, TX
#else
#define serial_bluetooth Serial1
#endif

#define BAUD_RATE 9600

#define PIN_LED 13

#define PIN_EN_MOTOR_L 6
#define PIN_A_MOTOR_L 7
#define PIN_B_MOTOR_L 5
#define PIN_EN_MOTOR_R 3
#define PIN_A_MOTOR_R 2
#define PIN_B_MOTOR_R 4

#define MOTOR_L_FORWARD digitalWrite(PIN_A_MOTOR_L, HIGH); digitalWrite(PIN_B_MOTOR_L, LOW)
#define MOTOR_L_BACKWARD digitalWrite(PIN_A_MOTOR_L, LOW); digitalWrite(PIN_B_MOTOR_L, HIGH)
```

```

#define MOTOR_L_STOP    digitalWrite(PIN_A_MOTOR_L, HIGH); digitalWrite(PIN_B_MOTOR_L, HIGH)

#define MOTOR_R_FORWARD digitalWrite(PIN_A_MOTOR_R, HIGH); digitalWrite(PIN_B_MOTOR_R, LOW)

#define MOTOR_R_BACKWARD digitalWrite(PIN_A_MOTOR_R, LOW); digitalWrite(PIN_B_MOTOR_R, HIGH)

#define MOTOR_R_STOP    digitalWrite(PIN_A_MOTOR_R, HIGH); digitalWrite(PIN_B_MOTOR_R, HIGH)

#define CAR_STOP        MOTOR_L_STOP; MOTOR_R_STOP

#define CAR_FORWARD     MOTOR_L_FORWARD; MOTOR_R_FORWARD

#define CAR_BACKWARD    MOTOR_L_BACKWARD; MOTOR_R_BACKWARD

#define CAR_LEFT        MOTOR_R_FORWARD; MOTOR_L_BACKWARD

#define CAR_RIGHT       MOTOR_L_FORWARD; MOTOR_R_BACKWARD

#define HALF_SPEED      false

#define HIGH_SPEED      true

#define CAR_HALF_SPEED  current_speed = HALF_SPEED; analogWrite(PIN_EN_MOTOR_L, 125);
analogWrite(PIN_EN_MOTOR_R, 125)

#define CAR_HIGH_SPEED  current_speed = HIGH_SPEED; analogWrite(PIN_EN_MOTOR_L, 255);
analogWrite(PIN_EN_MOTOR_R, 255)

boolean current_speed;

SerialManager serial_manager_pc(serial_pc);

SerialManager serial_manager_bluetooth(serial_bluetooth);

void stringReceptorPC(String string_received);

void stringReceptorBluetooth(String string_received);

void setup() {

    serial_pc.begin(BAUD_RATE);

    serial_bluetooth.begin(BAUD_RATE);

    serial_manager_pc.attachRxCallback(stringReceptorBluetooth);

    serial_manager_bluetooth.attachRxCallback(stringReceptorBluetooth);

    pinMode(PIN_LED, OUTPUT);

    pinMode(PIN_EN_MOTOR_L, OUTPUT);

```

```

pinMode(PIN_A_MOTOR_L, OUTPUT);

pinMode(PIN_B_MOTOR_L, OUTPUT);

pinMode(PIN_EN_MOTOR_R, OUTPUT);

pinMode(PIN_A_MOTOR_R, OUTPUT);

pinMode(PIN_B_MOTOR_R, OUTPUT);

CAR_HIGH_SPEED;
}

void loop() {

//No delete, mandatory lines

serial_manager_pc.onSerialRxEvent();

serial_manager_bluetooth.onSerialRxEvent();

}

void stringReceptorPC(String string_received) {  /*This function is called when a string finished with "\r\n" is
received

Serial.println("String received PC: " + string_received);

}

void stringReceptorBluetooth(String string_received) {  /*This function is called when a string finished with "\r\n" is
received

Serial.println("String received Bluetooth: " + string_received);

if (string_received.equals("FORWARD")) {

CAR_FORWARD;

} else if (string_received.equals("BACKWARD")) {

CAR_BACKWARD;

} else if (string_received.equals("LEFT")) {

CAR_LEFT;

} else if (string_received.equals("RIGHT")) {

CAR_RIGHT;

} else if (string_received.equals("STOP")) {

CAR_STOP;

} else if (string_received.equals("SPEED")) {

if(current_speed == HIGH_SPEED){

CAR_HALF_SPEED;

}else{

```

```
CAR_HIGH_SPEED;

}

}

}
```

Ultrasonido

```
#define PIN_EN_MOTOR_L 6

#define PIN_A_MOTOR_L 7

#define PIN_B_MOTOR_L 5

#define PIN_EN_MOTOR_R 3

#define PIN_A_MOTOR_R 2

#define PIN_B_MOTOR_R 4

#define MOTOR_L_FORWARD digitalWrite(PIN_A_MOTOR_L, LOW); digitalWrite(PIN_B_MOTOR_L, HIGH)

#define MOTOR_L_BACKWARD digitalWrite(PIN_A_MOTOR_L, HIGH); digitalWrite(PIN_B_MOTOR_L, HIGH)

#define MOTOR_L_STOP digitalWrite(PIN_A_MOTOR_L, HIGH); digitalWrite(PIN_B_MOTOR_L, HIGH)

#define MOTOR_R_FORWARD digitalWrite(PIN_A_MOTOR_R, LOW); digitalWrite(PIN_B_MOTOR_R, HIGH)

#define MOTOR_R_BACKWARD digitalWrite(PIN_A_MOTOR_R, HIGH); digitalWrite(PIN_B_MOTOR_R, LOW)

#define MOTOR_R_STOP digitalWrite(PIN_A_MOTOR_R, HIGH); digitalWrite(PIN_B_MOTOR_R, HIGH)

#define CAR_STOP MOTOR_L_STOP; MOTOR_R_STOP

#define CAR_FORWARD MOTOR_L_FORWARD; MOTOR_R_FORWARD

#define CAR_BACKWARD MOTOR_L_BACKWARD; MOTOR_R_BACKWARD

#define CAR_LEFT MOTOR_R_FORWARD; MOTOR_L_BACKWARD

#define CAR_RIGHT MOTOR_L_FORWARD; MOTOR_R_BACKWARD

#define HALF_SPEED analogWrite(PIN_EN_MOTOR_L, 125); analogWrite(PIN_EN_MOTOR_R, 125)

#define HIGH_SPEED analogWrite(PIN_EN_MOTOR_L, 255); analogWrite(PIN_EN_MOTOR_R, 255)

const int pinecho = 8;

const int pintrigger = 9;

const int pinled = 13;

// VARIABLES PARA CALCULOS
```

```

unsigned int tiempo, distancia;

void setup() {
  pinMode(PIN_EN_MOTOR_L, OUTPUT);
  pinMode(PIN_A_MOTOR_L, OUTPUT);
  pinMode(PIN_B_MOTOR_L, OUTPUT);
  pinMode(PIN_EN_MOTOR_R, OUTPUT);
  pinMode(PIN_A_MOTOR_R, OUTPUT);
  pinMode(PIN_B_MOTOR_R, OUTPUT);
  //digitalWrite(PIN_EN_MOTOR_L, HIGH);
  //digitalWrite(PIN_EN_MOTOR_R, HIGH);
  HIGH_SPEED;

  // CONFIGURAR PINES DE ENTRADA Y SALIDA
  pinMode(pinecho, INPUT);
  pinMode(pintrigger, OUTPUT);
  pinMode(13, OUTPUT);
}

void loop() {
  // ENVIAR PULSO DE DISPARO EN EL PIN "TRIGGER"
  digitalWrite(pintrigger, LOW);
  delayMicroseconds(2);
  digitalWrite(pintrigger, HIGH);
  // EL PULSO DURA AL MENOS 10 uS EN ESTADO ALTO
  delayMicroseconds(10);
  digitalWrite(pintrigger, LOW);

  // MEDIR EL TIEMPO EN ESTADO ALTO DEL PIN "ECHO" EL PULSO ES PROPORCIONAL A LA
  DISTANCIA MEDIDA
  tiempo = pulseIn(pinecho, HIGH);

  // LA VELOCIDAD DEL SONIDO ES DE 340 M/S O 29 MICROSEGUNDOS POR CENTIMETRO
  // DIVIDIMOS EL TIEMPO DEL PULSO ENTRE 58, TIEMPO QUE TARDA RECORRER IDA Y VUELTA UN
  CENTIMETRO LA ONDA SONORA
  distancia = tiempo / 58;
}

```

```
if(distancia >=20 ){  
  CAR_FORWARD;  
  }else{  
    CAR_STOP;  
  }  
  delay(2000);  
  CAR_RIGHT;  
  delay(500);  
}
```

Duración: 4 horas

Anexo I. Socialización del proyecto.

COLEGIO SANTA CLARA DE ASÍS
Socialización del proyecto

Objetivo: Identificar el aprendizaje logrado en el desarrollo de la intervención en la construcción del robot comparado con el primer momento.

Actividad final: Exposición por grupos en forma de carrusel del proyecto robótico.

Descripción: Las estudiantes deberán organizar un lugar donde muestren el robot que se construyó y explicar el proyecto elaborado con las características que este tiene y la funcionalidad del prototipo realizado; además deben decir los conceptos de física que se aprendieron durante la investigación y mostrar el diario de campo que cada grupo realizó para aquellas personas que estén interesados en conocer más detalladamente el proyecto. Las estudiantes podrán utilizar cualquier otro medio que les ayude a ilustrar mejor lo que quieren transmitir o mostrar a los miembros invitados, tales como, diapositivas, carteleras, póster entre otros.

Duración: 2 horas

Anexo J. Explicación código de programación.

A continuación, se enseñan las líneas de código más relevantes a cada uno de los equipos de acuerdo con el tipo de sensor y que se relacionaba con algún concepto de física. A modo de aclaración, donde inician las dos barras (//) es porque en el código se agrega un comentario de aclaración o explicación que facilita el entendimiento de ellas:

Código seguidor de línea:

```
#define LENTO_VELOCIDAD    analogWrite(PIN_EN_MOTOR_I,
85);    analogWrite(PIN_EN_MOTOR_D, 85

//Para que gire de una mejor forma se inserta la velocidad a 80
para que tenga mejor ángulo de giro y a la vez pueda leer los datos
desde el sensor.

#define MEDIA_VELOCIDAD    analogWrite(PIN_EN_MOTOR_I,
210);    analogWrite(PIN_EN_MOTOR_D, 210) //El giro debe ser más
rápido que el aceleras ya que detiene el momento que consiguió,
pero no muy alto porque se sale de la línea.

#define TODA_VELOCIDAD    analogWrite(PIN_EN_MOTOR_I,
255);    analogWrite(PIN_EN_MOTOR_D, 255) //Esta no la uso pero está
ahí si se quiere ir a todo el potencial
```

En esta parte del código seguidor de línea se controlará toda la parte de la velocidad del prototipo robótico, teniendo así tres tipos de velocidades. Estos motores estarán conectados a un pin analógico, el cual puede tener valores entre 0 y 255, siendo 0 el nivel más bajo de velocidad y 255 el valor máximo; como el robot será un seguidor de línea este debe tener una potencia media para que este no se salga de la línea que debe seguir.

```
void loop(){ //Uso el WHILE en vez del IF, porque el WHILE es
instantaneo y el IF hace que se demore unos milisegundos en dejar de
girar

    analogWrite (12,10);
    DERECHA=analogRead (A2);
    while(DERECHA<35){
        CARRO_DERECHA;
        MEDIA_VELOCIDAD;
        DERECHA=analogRead (A2);
    }
    IZQUIERDA=analogRead (A0);
    while(IZQUIERDA<35){
        CARRO_IZQUIERDA;
        MEDIA_VELOCIDAD;
        IZQUIERDA=analogRead (A0);
    }
    CARRO_ADELANTE;
    LENTO_VELOCIDAD; //Gira en media velocidad y avanza más lento para que
pueda reaccionar}
```

Esta es la parte del código más importante, ya que es donde se calibran los sensores seguidores de línea; es importante ponerlas en las superficies oscuras y claras para poder

tomar los valores por medio del sensor e ir registrando desde el Monitor Serial del IDE de Arduino, donde por medio de una comunicación serial con el computador se pueden observar valores entre 0 y 1023. En este caso, si ponemos el sensor en una parte oscura el reflejo de la luz va a dar un valor de 1023, mientras que si la ponemos en una parte clara como es el baldosín blanco se notará que los valores se reducen en aproximadamente 35. Estos valores serán de referencia para poder calibrar el carro dependiendo del color que debe seguir, teniendo ambos motores conectados a los pines analógicos A2 y A0. En esa parte se puede trabajar todo lo que tiene que ver con la reflexión de la luz, pero en ese caso todo medido desde los sensores que se les dio, midiendo distintos valores en diferentes regiones, por ejemplo, si ponemos el sensor en una parte café va a tomar un valor, si se pone en una parte oscura tomará otro valor y si se pone en una parte blanca como es el baldosín de laboratorio tomará otro valor. Todo esto funciona mediante ondas electromagnéticas las cuales serán de referencia para el color que estos deben seguir.

Código carro Bluetooth:

```
#ifndef USING_ARDUINO_UNO
SoftwareSerial serial_bluetooth(10, 11); // RX, TX
#else
#define serial_bluetooth Serial1
#endif

#define BAUD_RATE 9600
```

En esta parte del código se establecerá conexión tanto con el prototipo robótico como con el celular por medio del módulo Bluetooth, el cual funciona mediante ondas electromagnéticas, transmitiendo información a cortas distancias las cuales no pueden abarcar más de 10 metros; es decir, en el momento en que se establezca conexión con el carro Bluetooth éste no puede irse más de 10 metros ya que la información que se suministre del celular no le alcanzará a llegar. Dentro del espectro electromagnético se pueden encontrar distintos tipos de señales dependiendo de las características de la onda, ustedes consultan podrán observar todo el espectro electromagnético dentro de las normas que se establecen teniendo en cuenta la salud del ser humano. Aparte de eso para lograr una comunicación entre el Arduino y el módulo Bluetooth hay una velocidad de información la cual se denomina baudios, los baudios se consideran como la cantidad de información que pasa en un segundo; es algo así parecido como la frecuencia (Hertz). Podemos notar que la velocidad de información dentro del pc puede estar a 9600 como la cantidad mínima de información y como la cantidad máxima información que le podemos suministrar a un dispositivo dentro del monitor serial es de 115200. Esta velocidad de comunicación también se puede ver en el monitor serial de Arduino IDE (interfaz de programación de Arduino).

```
#define HALF_SPEED false

#define HIGH_SPEED true
#define CAR_HALF_SPEED current_speed =
HALF_SPEED; analogWrite(PIN_EN_MOTOR_L, 125);
analogWrite(PIN_EN_MOTOR_R, 125)
```

```

#define CAR_HIGH_SPEED      current_speed =
HIGH_SPEED; analogWrite(PIN_EN_MOTOR_L, 255);
analogWrite(PIN_EN_MOTOR_R, 255)
boolean current_speed;

```

En esa parte del código se establecerá los pines digitales usados en la construcción del Arduino en el prototipo robótico establecerán la velocidad de este en donde cero es el valor mínimo de velocidad y 255 es el valor máximo de velocidad. Para tener más control del carro se establecen pins en analógicos teniendo una velocidad de 125 . Se debe tener en cuenta que si el carro va a ser un movimiento rectilíneo o se comparará velocidades con otro carro es necesario ponerlo en el valor máximo de velocidad de 255, procurando tener las baterías muy cargadas ya que a mayor velocidad mayor es el gasto de la batería.

Código sensor ultrasónico:

```

#define HALF_SPEED          analogWrite(PIN_EN_MOTOR_L, 125);
analogWrite(PIN_EN_MOTOR_R, 125)
#define HIGH_SPEED          analogWrite(PIN_EN_MOTOR_L, 255);
analogWrite(PIN_EN_MOTOR_R, 255)

const int pinecho = 8;
const int pintrigger = 9;
const int pinled = 13;

// VARIABLES PARA CALCULOS
unsigned int tiempo, distancia;

```

Dentro de esta parte del código se establecen las velocidades a las cuales deben de ir los motores siendo 0 la mínima velocidad y 255 la máxima velocidad. Es importante establecer una velocidad media ya que como se va a trabajar con el sensor ultrasónico este no puede ir a mucha velocidad para que pueda detectar los obstáculos rápidamente y poder tomar decisiones. Ideal que se establezca una velocidad media en los pines analógicos de 125. Por otro lado el sensor ultrasónico funciona por medio de ondas, tanto en la emisión como en la recepción; por tanto se establecen los pines analógicos, el pin de recepción está conectado directamente al pin 8, el pin de emisión está conectado al pin 9 del Arduino y se hace una conexión de manera aleatoria al pin 13 conectando un led que pueda dar información de un obstáculo, es decir que cuando el LED se conecte al pin 13 y encuentre un obstáculo este se prenderá.

```

// CONFIGURAR PINES DE ENTRADA Y SALIDA
pinMode(pinecho, INPUT);
pinMode(pintrigger, OUTPUT);
pinMode(13, OUTPUT);

```

Activamos el pin trigger como salida emitiendo una onda y el pin que recibe la onda y lo establecemos como input (recepción de la señal). Estos valores se establecen como altos y bajos output o input respectivamente).

```

void loop() {

```

```

        // ENVIAR PULSO DE DISPARO EN EL PIN "TRIGGER"
        digitalWrite(pintrigger, LOW);
        delayMicroseconds(2);
        digitalWrite(pintrigger, HIGH);
        // EL PULSO DURA AL MENOS 10 uS EN ESTADO ALTO
        delayMicroseconds(10);
        digitalWrite(pintrigger, LOW);

        // MEDIR EL TIEMPO EN ESTADO ALTO DEL PIN "ECHO" EL PULSO ES
        PROPORCIONAL A LA DISTANCIA MEDIDA
        tiempo = pulseIn(pinecho, HIGH);

```

En esta parte se mide el tiempo en estado alto y se agrega como pin de entrada o encendido y el tiempo que dure el pulso se administra en la variable tiempo.

```

        // LA VELOCIDAD DEL SONIDO ES DE 340 M/S O 29 MICROSEGUNDOS POR
        CENTIMETRO
        // DIVIDIMOS EL TIEMPO DEL PULSO ENTRE 58, TIEMPO QUE TARDA RECORRER
        IDA Y VUELTA UN CENTIMETRO LA ONDA SONORA
        distancia = tiempo / 58;

```

En esa parte se tiene en cuenta que aproximadamente la velocidad del sonido en condiciones normales es de 340 m/s y que viaja en el espacio tomando sólo 9 microsegundos por centímetro. Ahora se divide el tiempo del pulso entre 58 y ese es el tiempo que tarda en recorrer la honda, ida y vuelta (tiempo desde la emisión hasta la recepción).

```

        if(distancia >=20 ){
            CAR_FORWARD;
        }else{
            CAR_STOP;

```

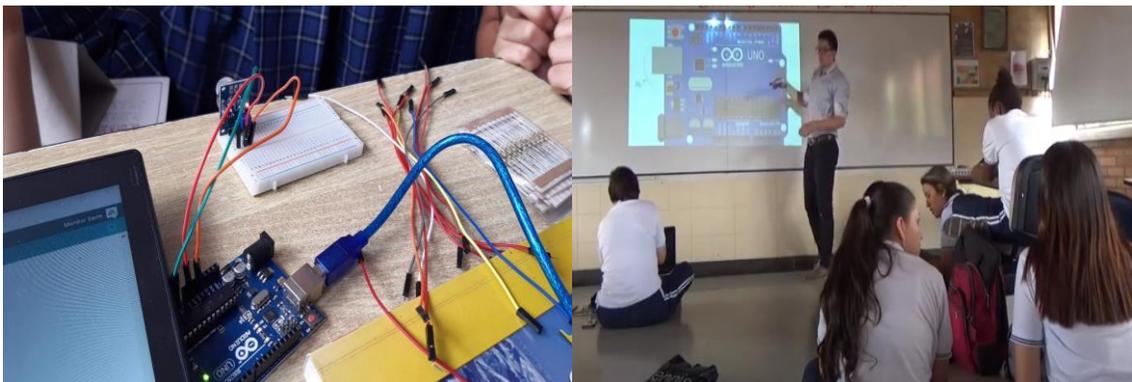
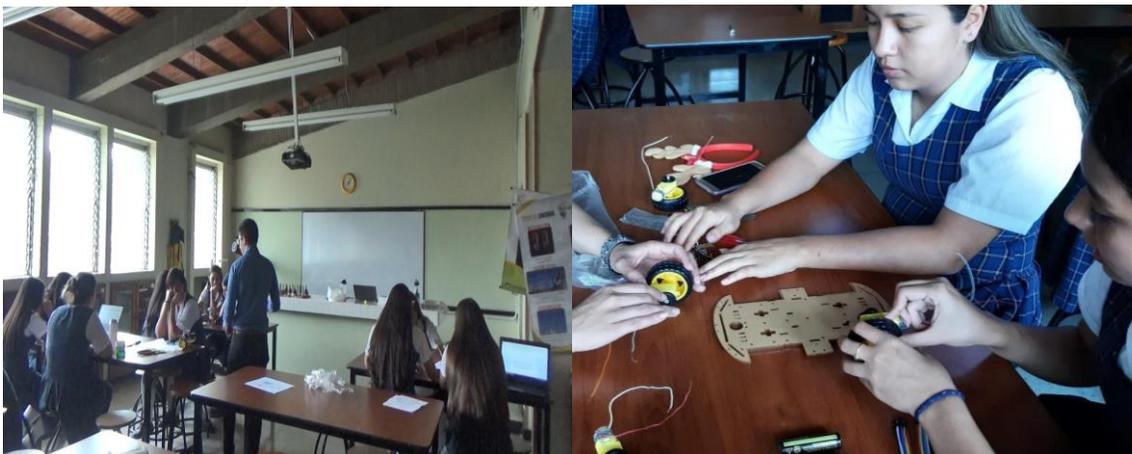
Por último, se establece la condición, teniendo en cuenta la distancia o el cálculo que se hace cada 10 microsegundos detecte un obstáculo. Si el obstáculo está a más de 20 cm el carro seguirá adelante, pero de lo contrario el carro realizará un pare. Esa es la parte más importante del código ya que aquí es donde se establecen los pines como altos o bajos. El pulso entre cada emisión y cada recepción dura al menos 10 microsegundos en estado alto; esto quiere decir que cada 10 microsegundos se estará enviando y recibiendo una señal o en otras palabras, el sensor estará advirtiendo de un obstáculo cada 10 microsegundos.

Anexo K. Registro fotográfico de la implementación del modelo MOPRE.

Primer momento: Formulación del problema de la realidad y planificación del proyecto.



Segundo momento: Construcción y programación del robot.



Tercer momento: Socialización del proyecto.



Anexo L. Registro fotográfico acercamiento a la programación desde PSeInt y Arduino



Anexo M. Registro fotográfico familiarización con los elementos base de la robótica

