



**Escuela Superior  
Politécnica del Litoral**

## Aplicativo móvil para reciclaje automático de basura

**PARALELO: 1 GRUPO: 6**

**INTEGRANTES:**

JHONSTON HERNAN BENJUMEA MASACHE

ANDRE DANIEL ICAZA GONZALEZ

ANGIE ROMINA LEÓN PEÑAFIEL

DIANA ESTEFANÍA ONCE BRAVO

# ÍNDICE

<b>Resumen ejecutivo</b> .....	3
<b>Introducción y descripción del problema</b> .....	3
<b>Objetivos del Proyecto</b> .....	3
<b>¿Cómo funciona la solución?</b> .....	4
<b>¿Qué van a construir para resolver el problema?</b> .....	4
<b>Recursos de hardware y software</b> .....	5
<b>Implementación del proyecto</b> .....	6
Diseño de la interfaz del usuario .....	8
<b>Diagramas del proyecto</b> .....	11
Diagrama de casos UML .....	11
Diagrama del modelo entidad-relación .....	11
Diagrama de red .....	12
Diagramas de circuitos .....	12
<b>Código Arduino</b> .....	15
<b>Código Raspberry PI</b> .....	18
<b>Código SQL</b> .....	19
<b>Análisis de presupuesto</b> .....	25
<b>Conclusiones</b> .....	26
<b>Referencias bibliográficas</b> .....	27

## Resumen ejecutivo

El proyecto presente plantea el diseño y prototipado de un sistema de reciclaje automático de basura, así como una aplicación móvil que permita llevar un control acerca de los niveles de cada bote. Esto ayuda a la separación eficiente de los residuos y a un mejor manejo del vaciado de los contenedores.

## Introducción y descripción del problema

La ciudad de Guayaquil cuenta con aproximadamente 2.7 millones de habitantes de los cuales sólo el 40% recicla en sus hogares [1]. Guayaquil al ser un sector mayormente urbano, cuenta con una alta cantidad de desperdicios producidos al día que contienen materiales plásticos, metálicos, orgánicos, entre otros, que terminan en las avenidas y calles de la ciudad. A pesar de que existen botes de basura para un tipo de material específico, es común notar que los ciudadanos depositan sus desechos en ellos de manera indiferente, sin separar lo plástico de lo orgánico, por ejemplo.

Al no existir un reciclaje consciente dentro de los hogares, es complicado el impulsar el reciclaje en las calles y parques de la ciudad, donde el primer argumento es la falta de tiempo para separar los residuos. El problema radica en que los guayaquileños no separan sus residuos en los contenedores correspondientes al momento de desecharlos.

## Objetivos del Proyecto

### Objetivo general

- Diseñar y construir un sistema de recolección de desechos que permita la separación de estos de manera automática en tres grupos y el control de sus niveles de llenado a través de una aplicación móvil.

### Objetivos específicos

- Diseñar un modelo 3D del sistema de recolección, incluyendo sus compartimentos.
- Realizar un circuito eléctrico con elementos y dispositivos que permitan el manejo y sensado de los desechos para su correcta clasificación.
- Implementar el uso de bases de datos SQL en conjunto con una aplicación móvil para el mejor manejo de la información recopilada.

## ¿Cómo funciona la solución?

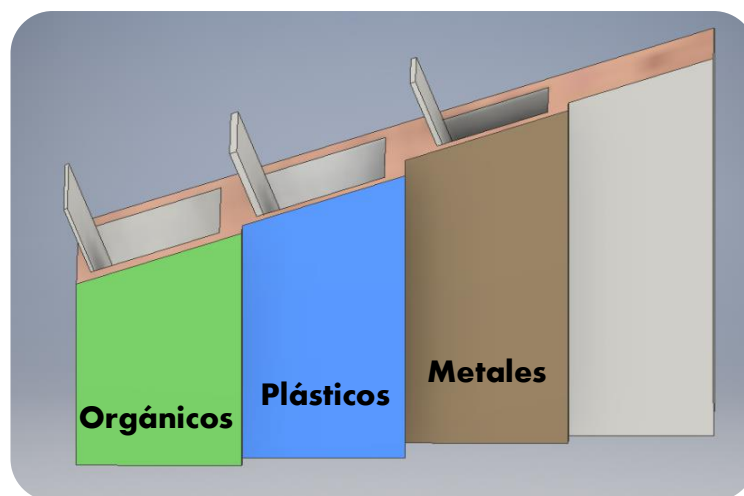
La solución propuesta permite que, a través de sensores ubicados a la entrada del bote de basura, se identifique la categoría a la cual el desperdicio ingresado pertenece, para luego ser depositado de manera automática en su compartimento correspondiente. Los tres tipos de materiales por separar son plásticos, metales y orgánicos. Cada uno posee su propio compartimento y tapa, la cual es levantada mediante un micro servomotor después de que se haya realizado la identificación del tipo de desecho.

Estas categorías se encuentran definidas previamente en la aplicación móvil para que a medida que ingresa un elemento, la información se envíe a una base de datos en phpMyAdmin, y se pueda visualizar en tiempo real el incremento en el nivel de basura del bote.

Las aplicaciones móviles estarán instaladas en los Smartphones de los encargados de limpieza del lugar donde se encuentren los botes inteligentes, y al iniciar sesión con su usuario y contraseña, podrán observar cuál bote de basura está listo para ser vaciado, registrar cuando lo hayan hecho, y conocer cuáles otros encargados han retirado la basura.

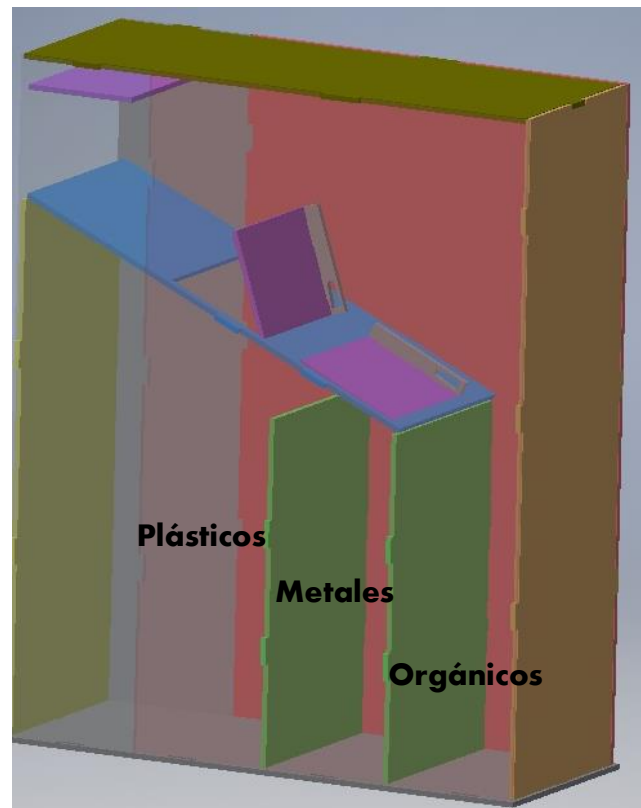
## ¿Qué van a construir para resolver el problema?

Lo que se planea construir es un bote de basura de 3 compartimentos que posea una entrada empinada la cual permite el desplazamiento del desperdicio ingresado por acción de la gravedad, y a su vez esta tiene dos aperturas que se accionan con servomotores. En la figura siguiente se observa el primer prototipo del bote. Originalmente este poseía una compuerta por cada tipo de residuo y una entrada inclinada que avanzaba hasta el compartimento de orgánicos.



*Ilustración 1: Primer diseño del prototipo*

El diseño final, a diferencia del primer prototipo, posee una rampa de entrada que avanza sólo hasta el segundo compartimento, evitando así el tener una puerta rotaria para los componentes orgánicos. En su lugar, se colocó una puerta a la entrada de los desechos, la cual retiene a estos hasta que el sistema identifique a la categoría a la cual pertenece. La figura siguiente ilustra la forma final del proyecto.



*Ilustración 2: Diseño final del prototipo*

Con respecto a la aplicación móvil, se planea que esta muestre la información de cada uno de los contenedores, como el nivel de desechos y el nombre de la última persona que lo vació, todo en tiempo real. Además, enviar notificaciones al celular del encargado cuando un contenedor ha sido categorizado como lleno.

## Recursos de hardware y software

### Hardware:

- 1 Arduino Genuino Uno
- 1 Cable de Arduino
- 1 Raspberry PI

- 1 Sensor inductivo
- 4 Módulos de sensores infrarrojos IR FC-51
- 1 Fococelda
- 1 Módulo láser KY-008
- 3 Micro Servomotores SG90
- 1 Laptop
- 1 Módulo de alimentación 5V DC
- 4 Baterías de 1.5 V DC
- Cables jumpers

#### Software:

- Software de Arduino
- Android Studio
- Raspbian
- phpMyAdmin
- Python 2.0

## Implementación del proyecto

Antes de detallar el proceso de armado del proyecto, se necesita conocer el funcionamiento de los sensores empleados en la solución.

#### Sensor inductivo:

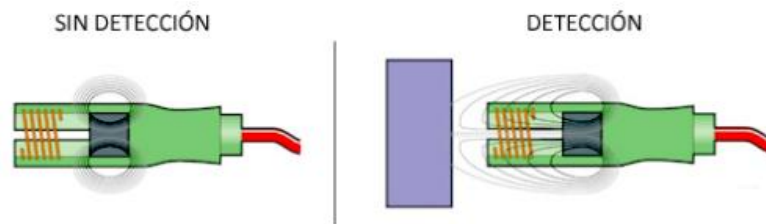
Este sensor tiene un voltaje de operación entre 6 y 36 V DC, y un rango de detección de hasta 4 mm, con  $\pm 10\%$  de error. Es capaz de detectar materiales de hierro o aleaciones de acero. Posee tres salidas, un pin VCC que va conectado a la fuente, un pin GND que va conectado a tierra, y un pin OUT [2] que va conectado en este caso a una entrada digital del Arduino.



*Ilustración 3: Conexiones del sensor inductivo*

El sensor posee un generador de campo magnético y una bobina que detecta a este campo generado por sí mismo. Al existir un objeto metálico cerca, el campo inducido se ve modificado, por

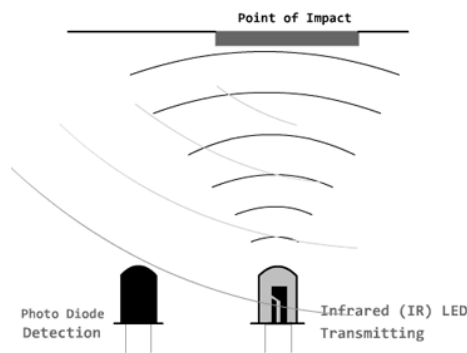
lo que la corriente inducida en la bobina sensora aumenta [3]. La salida del sensor puede tomar dos valores, HIGH cuando detecta a un metal, y LOW cuando no se ha detectado un material metálico.



*Ilustración 4: Forma de detección del sensor inductivo*

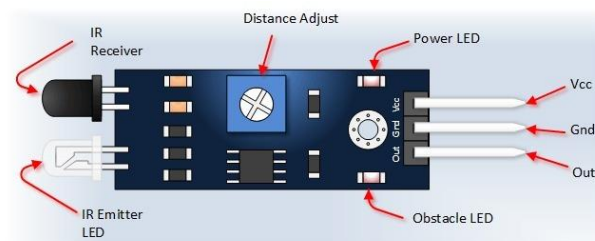
#### Sensor infrarrojo:

El módulo sensor de infrarrojos tiene un voltaje de operación entre 3.3 y 5 V DC. Es un tipo de sensor de proximidad básico para evitar obstáculos, compuesto de un transmisor y un receptor. El transmisor emite la energía de infrarrojos, y el receptor detecta esta energía cuando se refleja ante la presencia de un obstáculo frente al módulo.



*Ilustración 5: Método de trabajo del sensor infrarrojo*

Posee un rango de detección que va desde los 2 hasta los 30 cm, con un ángulo de cobertura de 35° [4]. Al igual que el sensor inductivo, posee tres pines: un pin Vcc, un pin GND, y un pin de salida, el cual se conecta al Arduino. La salida del sensor da como resultado dos valores, HIGH cuando detecta un objeto, y LOW cuando no existe algún objeto frente a él.



*Ilustración 6: Conexiones del módulo infrarrojo*

## Diseño de la interfaz del usuario

La aplicación fue diseñada principalmente para los encargados de la recolección de la basura, con el fin de que puedan visualizar de manera interactiva los niveles de cada tacho.



*Ilustración 7: Pantalla de inicio de sesión*

Al momento de iniciar la aplicación, aparece la pantalla de inicio de sesión, donde el encargado debe ingresar su usuario y contraseña previamente otorgados por nosotros para poder hacer uso de la misma. El login le da acceso a la siguiente ventana, es decir que una persona que no esté registrada no podrá revisar la aplicación.



*Ilustración 8: Niveles de los contenedores*

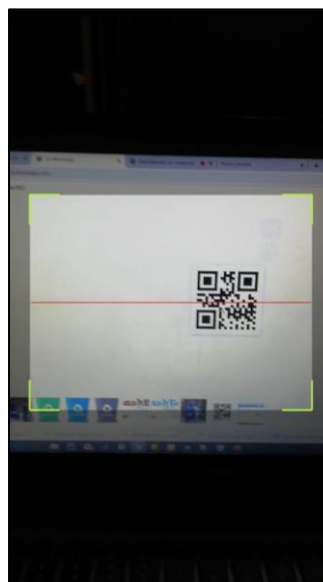


En esta pantalla se observa la función más importante de la aplicación, el mostrar cada tacho con sus niveles de basura. A medida que un objeto es identificado en el contenedor, el valor del porcentaje incrementa hasta llegar al 100%.



*Ilustración 9: Contenedores llenos*

Esta pantalla muestra a todos los contenedores con niveles de “lleno”, lo cual permite ir a la siguiente pestaña de retiro de basura.



*Ilustración 10: Lectura del código QR*

Al tocar el símbolo del código QR, se cambia la pantalla a un lector QR para leer el número identificador del contenedor. Cuando la lectura del contenedor es satisfactoria, se muestra la siguiente pantalla.

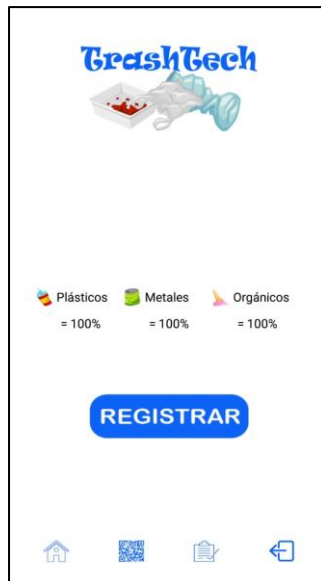


Ilustración 11: Pantalla de registro del retiro

En esta pantalla se permite al usuario registrar los niveles de los contenedores al momento de vaciarlos y colocarlos nuevamente en 0%.

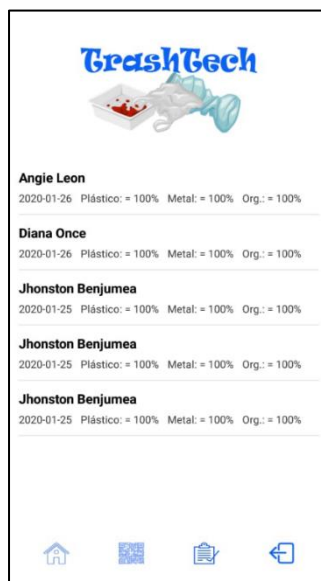


Ilustración 12: Listado de personas que han retirado la basura

Finalmente, es posible acceder a una pestaña donde se ve el listado de las últimas personas que han retirado basura de los contenedores.

## Diagramas del proyecto

### Diagrama de casos UML

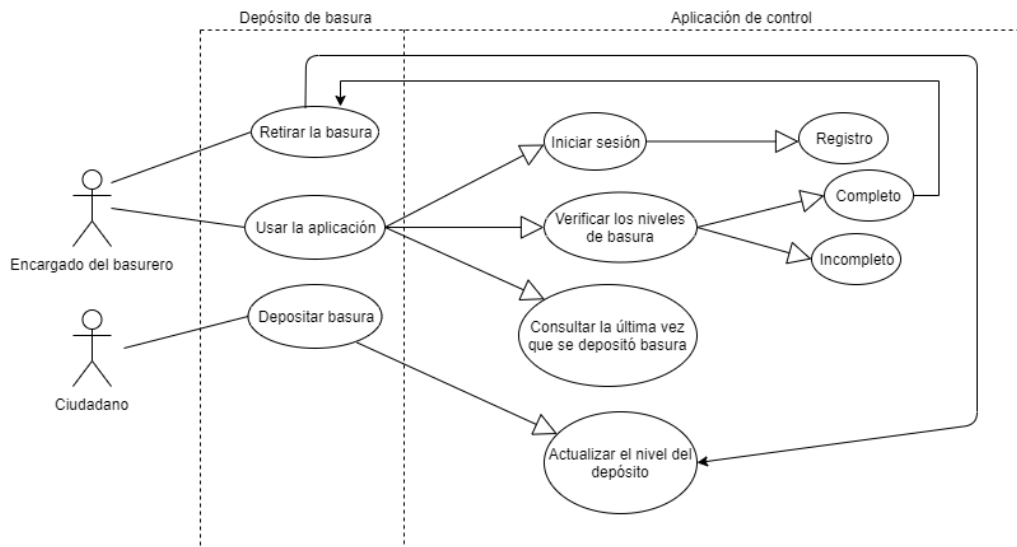


Ilustración 13: Diagrama de casos de uso

En el diagrama de casos de usos existen dos actores principales, el encargado del basurero y el ciudadano que deposita la basura. Cuando el ciudadano ingresa sus desechos, en la aplicación se ve reflejado inmediatamente el aumento del nivel en el contenedor, el cual es monitoreado por el encargado. Este último a través de la aplicación puede iniciar sesión con su usuario y contraseña, verificar los niveles de basura, y registrar cuando un bote ha sido vaciado por él.

### Diagrama del modelo entidad-relación

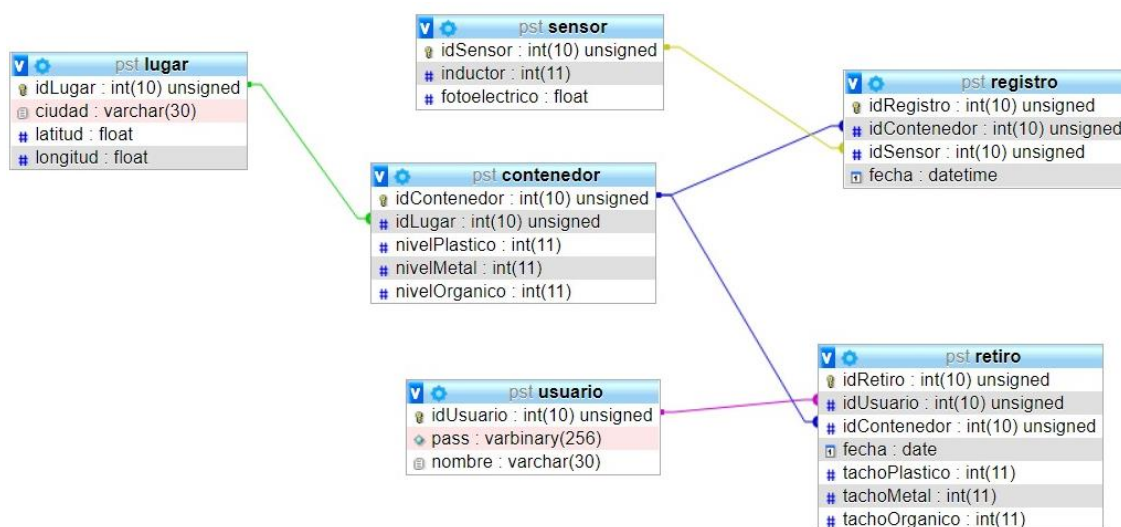
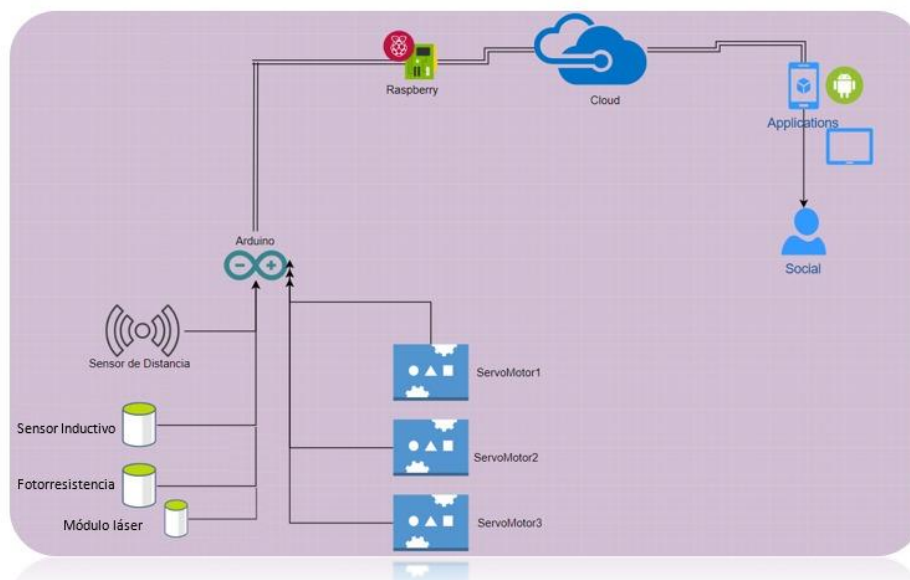


Ilustración 14: Modelo entidad-relación

En este modelo entidad-relación se plantearon las principales entidades tomadas en cuenta para la creación de la base de datos. Entre ellas está el usuario, que hace referencia al encargado que usará la aplicación; los sensores, que permiten conocer el tipo de residuo desechado; la actividad de retiro de basura y su registro en el sistema; y el contenedor al que se está analizando.

### Diagrama de red



*Ilustración 15: Diagrama de red del Proyecto*

El proyecto constituye en un Arduino Uno al cual se conectan los sensores infrarrojos, el sensor inductivo, el módulo láser, los servomotores y la fotocelda. Este, se comunica con la Raspberry PI, permitiendo así el envío de la información a la nube para que sea almacenada en la base de datos, y visualizada por el encargado en su aplicación móvil.

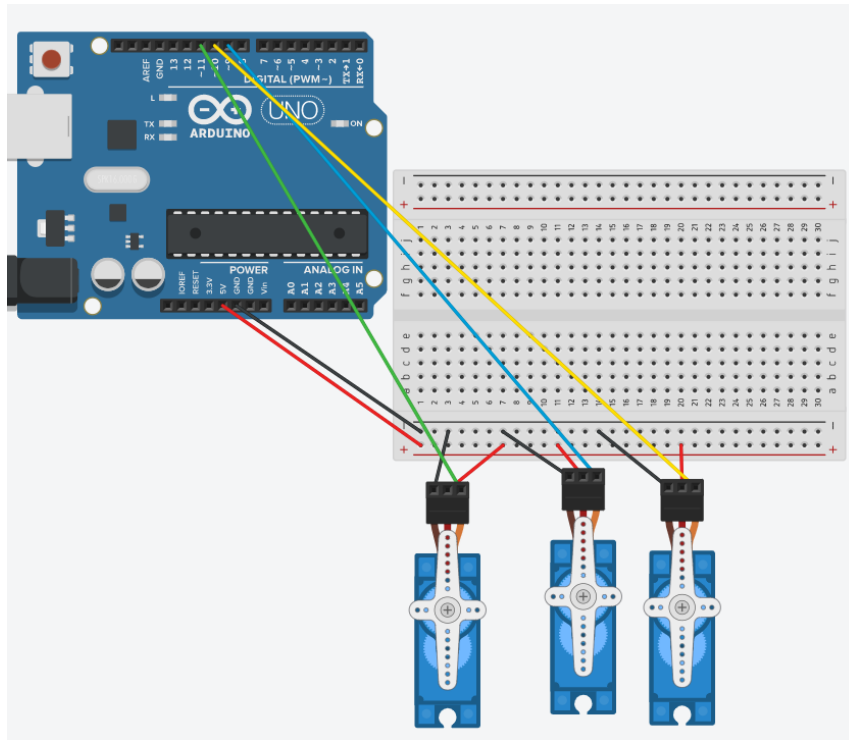
### Diagramas de circuitos

En esta sección se van a detallar las conexiones realizadas de los distintos elementos empleados. Es necesario acotar que debido a la complejidad de este y para evitar el cruce de cables, se separaron las ilustraciones esquemáticas por grupos de componentes, pero todos se encuentran conectados al mismo Arduino Uno.

Nota: Los diagramas fueron realizados con el software de TinkerCAD.

#### Grupo 1: Servomotores

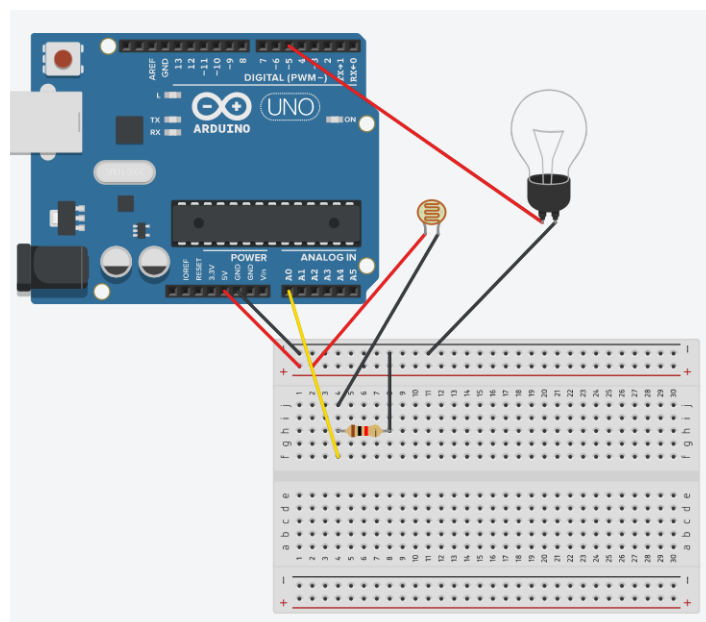
Los tres servomotores están conectados a la fuente de 5V y al GND del Arduino. Para el servo de la puerta principal del bote se usó el pin digital #9, para el de la puerta del metal el pin digital #10, y para el plástico el pin digital #11.



*Ilustración 16: Conexión de los servomotores*

## Grupo 2: Sensor de plástico

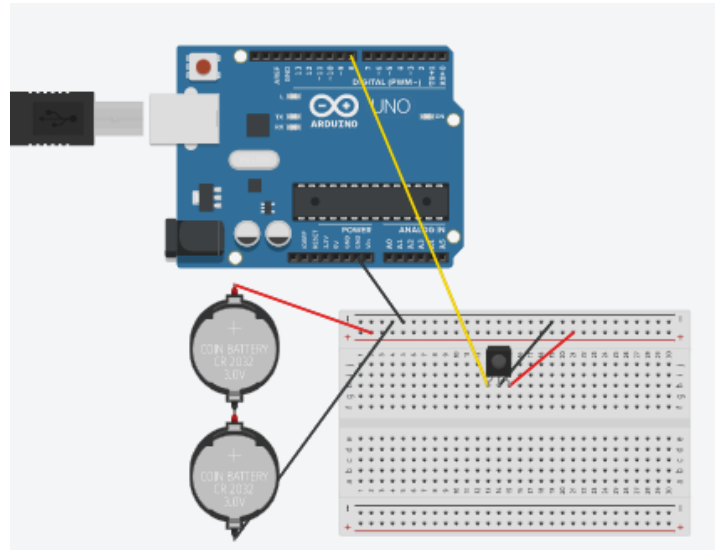
La bombilla en el diagrama representa al módulo láser usado para la detección del plástico, el cual fue conectado directamente al pin digital #5 y al GND del Arduino. En complemento con el láser, también se usó una fotocelda en circuito divisor de voltaje. La fotocelda está conectada a los 5V del Arduino y a una resistencia que a su vez está conectada al GND. Entre la fotocelda y la resistencia nace un cable que se conecta al pin analógico A0 del Arduino para enviar las señales de luz recibida por él.



*Ilustración 17: Conexión de la fotocelda y el módulo láser*

### Grupo 3: Sensor de metal

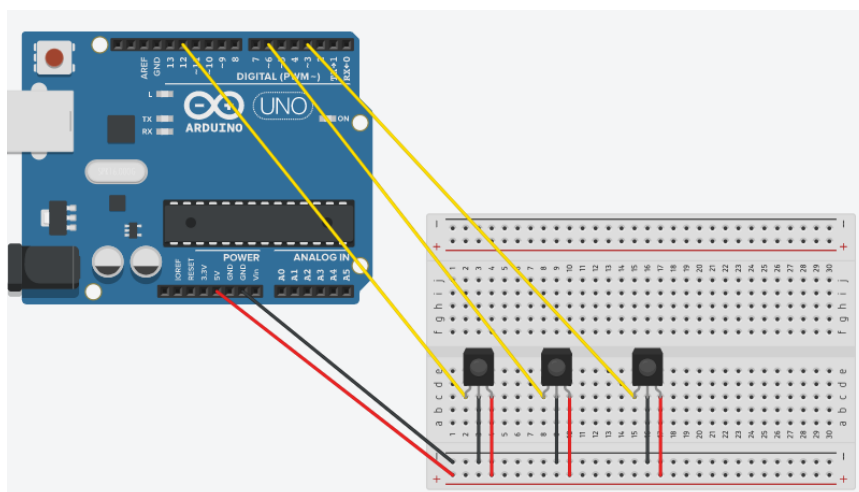
El sensor inductivo no se encuentra en el catálogo de TinkerCAD, pero su funcionamiento es similar al de un sensor infrarrojo o de proximidad, por lo que se usará este último sólo de manera representativa. El sensor inductivo necesita alimentación de 6V o más, que proviene en este caso de baterías externas en serie. El lado positivo del sensor va conectado directamente a la fuente, mientras que su lado negativo con el GND del Arduino y el negativo de la fuente. La salida del sensor está conectado al pin digital #8 del Arduino.



*Ilustración 18: Conexión del sensor inductivo*

### Grupo 4: Sensores de proximidad

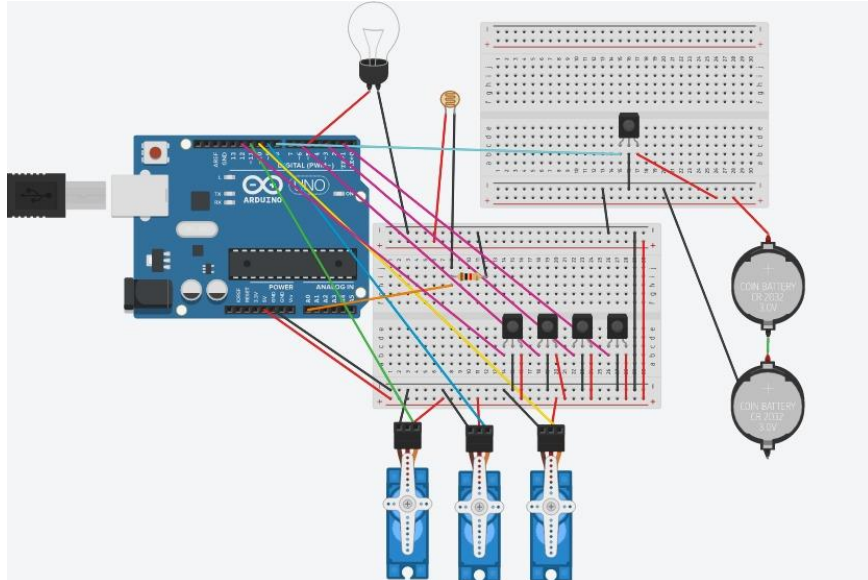
En el diagrama se encuentran los tres sensores infrarrojos, donde cada uno corresponde a un compartimento del bote en específico. Los tres sensores están conectados a los 5V y al GND del Arduino, y sus puertos libres a los pines digitales #3, #6 y #12 respectivamente, que permiten enviar lo sensado.



*Ilustración 19: Conexiones de los sensores infrarrojos*

## Circuito completo

En el circuito final se añade otro módulo infrarrojo, siendo en total 4, el cual estará ubicado a la entrada del contenedor con el fin de detectar cuando un objeto ha sido insertado.



*Ilustración 20: Circuito completo del proyecto*

## Código Arduino

A continuación, se muestra el código usado en el Arduino para el funcionamiento del contenedor automático.

```
#include <Servo.h> //Se incluye la librería
//Creación de las variables de los Servos
Servo servoEntrada; //Está ubicado a la entrada del contenedor
Servo servoPlastico; //Ubicado a la entrada del compartimento plástico
Servo servoMetal; // Ubicado a la entrada del compartimento metal

//Variables sensores de material
int sensorPlastico = A0; //Entrada fotocelda
float valorLuz = 0; //Valores que lee la fotocelda
int pinLaser = 5; //Salida láser
int pinInductivo = 8;

//Variables sensores de proximidad
int pinInfraPlastico = 3;
int pinInfraMetal = 6;
int pinInfraPrincipal = 12;

//Variables arduinos
int estadoPlastico = 0;
int estadoMetal = 0;
int estadoOrganico = 0;

int isMetal = 0;
```

```

float isPlastico = 0;

void setup() {
  Serial.begin(9600);
  servoEntrada.attach(9);
  servoPlastico.attach(11);
  servoMetal.attach(10);

  //Se colocan en sus posiciones de cerrado
  servoEntrada.write(90); //Se abre horario a 90
  servoPlastico.write(180); //Se abre antihorario a 90
  servoMetal.write(180); //Se abre antihorario a 90

  pinMode(sensorPlastico, INPUT); //Fotocelda recibe
  pinMode(pinLaser, OUTPUT); //Láser emite
  pinMode(pinInductivo, INPUT);
  pinMode(pinInfraPlastico, INPUT);
  pinMode(pinInfraMetal, INPUT);
  pinMode(pinInfraPrincipal, INPUT);
}

void loop() {
  estadoPlastico = 0;
  estadoMetal = 0;
  estadoOrganico = 0;

  isMetal = digitalRead(pinInductivo);
  isPlastico = 0;

  //Los sensores usados son normalmente abiertos, es decir que su salida es
  HIGH si no existe objeto en frente
  if (digitalRead(pinInfraPlastico) == LOW) {
    estadoPlastico = 1; //Significa que está lleno el bote
  }

  if (digitalRead(pinInfraMetal) == LOW) {
    estadoMetal = 1;
  }

  if (digitalRead(pinInfraPrincipal) == LOW) { //Detecta si un objeto ha
    sido insertado

    if (isMetal == HIGH) { //El sensor inductivo detectó un metal

      Serial.print("ES METAL");
      Serial.print("-");
      Serial.print(estadoPlastico);
      Serial.print("-");
      Serial.print(estadoMetal);
      Serial.print("-");
      Serial.print(estadoOrganico);
      Serial.print("-");

      Serial.print(isMetal);
      Serial.print("-");
      Serial.println(isPlastico);

      servoMetal.write(90); // Se abre el metal

      delay(1000);
      servoEntrada.write(180); //Se abren las puertas
    }
  }
}

```



```

    delay(3000);
    servoEntrada.write(90); //Se cierran las puertas
    servoMetal.write(180);

    delay(2000);
}

else { //Esto sólo ocurre si no detectó metal, verifica si es plástico

    digitalWrite(pinLaser, HIGH);
    delay(1000);
    valorLuz = analogRead(sensorPlastico);

    digitalWrite(pinLaser, LOW);

    if (valorLuz > 850) { //Si deja pasar luz, el objeto es transparente y
    por ende plástico

        isPlastico = 1;

        Serial.print("ES PLASTICO");
        Serial.print("-");
        Serial.print(estadoPlastico);
        Serial.print("-");
        Serial.print(estadoMetal);
        Serial.print("-");
        Serial.print(estadoOrganico);
        Serial.print("-");

        Serial.print(isMetal);
        Serial.print("-");
        Serial.println(isPlastico);

        servoPlastico.write(90); // Se abre el plastico

        delay(1000);
        servoEntrada.write(180); //Se abren las puertas

        delay(3000);
        servoEntrada.write(90); //Se cierran las puertas
        delay(500);
        servoPlastico.write(180);

        delay(2000);
    }

    else { //Si ya no detecta nada, es considerado como orgánico

        Serial.print("ES ORGANICO");
        Serial.print("-");
        Serial.print(estadoPlastico);
        Serial.print("-");
        Serial.print(estadoMetal);
        Serial.print("-");
        Serial.print(estadoOrganico);
        Serial.print("-");

        Serial.print(isMetal);
        Serial.print("-");
        Serial.println(isPlastico);
    }
}

```

```

servoEntrada.write(180); //Se abren las puertas

delay(3000);
servoEntrada.write(90); //Se cierran las puertas

delay(2000);
}
}
}
delay(1000);
}

```

El código primero realiza la medición de los estados de los contenedores, donde el valor de 0 indica que están vacíos, y un valor de 1, que están llenos. Esta respuesta es dada por el sensor infrarrojo colocado encima del recipiente de cada material. La clasificación empieza cuando el sensor infrarrojo de la entrada detecta un objeto, comenzando primero por la verificación de estado metálico mediante el sensor inductivo, donde si la respuesta es afirmativa, se abren las compuertas del contenedor metálico y de la entrada, dejando caer al material.

Si la respuesta es negativa, se verifica entonces si se trata de plástico, con los mismos casos del anterior. Si ya no es detectado como ninguno de los dos, se asume que es orgánico y sólo se activa la compuerta de entrada y el objeto cae al último recipiente.

Las variables de niveles de botes, así como las lecturas de los sensores son de suma importancia ya que son enviadas a la Raspberry para ser subidas a la base de datos, y que esto se pueda reflejar en la aplicación.

## Código Raspberry PI

```

import requests, serial, time

arduino = serial.Serial('/dev/ttyACM0', 9600)

time.sleep(1)

txt = ''

while True:

    inductor = 1
    fotoelectrico = 0
    tPlastico = 0
    tMetal = 0
    tOrganico = 0

    valores = []

    while arduino.inWaiting() > 0:
        txt = str(arduino.readline())[2:-5]
        print(txt)

```

```

valores = txt.split("-")

inductor = valores[4]
fotoelectrico = valores[5]
tPlastico = valores[1]
tMetal = valores[2]
tOrganico = valores[3]

params =
{'inductor':inductor,'fotoelectrico':fotoelectrico,'tPlastico':tPlastico,'t
Metal':tMetal,'tOrganico':tOrganico}
response =
requests.post('http://192.168.43.130/PST/query/addRecord.php', data =
params)
print(response.text)

```

Dentro de Python hay una librería "Pyserial" que permite la comunicación Arduino-Python. Esta librería hace la conexión con el Serial que envía el Arduino, lo agarra como String. Dentro del mismo, se utiliza el código respectivo de comunicación Rasp-Base de datos. Finalmente, este código se mantiene ejecutando hasta que el Arduino mande un serial, lo cual ocurre al sensar la basura que ingresa.

## Código SQL

```

DELIMITER //
CREATE PROCEDURE getPercen(IN idCont int)
BEGIN
/*Selecciona los niveles de cada tacho dependiendo del contenedor*/
SELECT nivelPlastico,nivelMetal,nivelOrganico FROM Contenedor
WHERE idContenedor = idCont;
END//
DELIMITER ;

```

Este query permite obtener los niveles de cada tacho del contenedor en cuestión. Cada contenedor tiene un id que es ingresado como parámetro para obtener su nivel en porcentajes.

```

DELIMITER //
CREATE PROCEDURE addRecord(IN inductor int,IN fotoelectrico float,IN
tPlastico int,IN tMetal int,IN tOrganico int)
BEGIN
/*Añade un registro de sensor*/
INSERT INTO sensor VALUES(null,inductor,fotoelectrico);

/*Selecciona el id del último sensor registrado*/
SELECT MAX(idSensor) into @idSens from sensor;

/*Selecciona los niveles de cada tacho dependiendo del contenedor y
los almacena en variables*/

```

```

SELECT nivelPlastico into @nPla from contenedor where
idContenedor=1;
SELECT nivelMetal into @nMet from contenedor where idContenedor=1;
SELECT nivelOrganico into @nOrg from contenedor where
idContenedor=1;

/*Si el objeto es metálico*/
IF (inductor>=1) THEN
    /*Aumenta el nivel del tacho de metal*/
    UPDATE contenedor
    SET nivelMetal=@nMet+1 WHERE idContenedor=1;

/*Si el objeto es plástico*/
ELSEIF (fotoelectronico>=1) THEN
    /*Aumenta el nivel del tacho de plástico*/
    UPDATE contenedor
    SET nivelPlastico=@nPla+1 WHERE idContenedor=1;

/*Si el objeto es orgánico*/
ELSE
    /*Aumenta el nivel del tacho de orgánico*/
    UPDATE contenedor
    SET nivelOrganico=@nOrg+1 WHERE idContenedor=1;
END if;

/*Si el tacho plastico esta lleno*/
IF (tPlastico>=1 OR @nPla>=5) THEN
    /*Actualizar el nivel de plastico a su maximo valor (5)*/
    UPDATE contenedor
    SET nivelPlastico=5 WHERE idContenedor=1;
END if;

/*Si el tacho metal esta lleno*/
IF (tMetal>=1 OR @nMet>=5) THEN
    /*Actualizar el nivel de metal a su maximo valor (5)*/
    UPDATE contenedor
    SET nivelMetal=5 WHERE idContenedor=1;
END if;

/*Si el tacho organico esta lleno*/
IF (tOrganico>=1 OR @nOrg>=5) THEN
    /*Actualizar el nivel de organico a su maximo valor (5)*/
    UPDATE contenedor
    SET nivelOrganico=5 WHERE idContenedor=1;
END if;

/*Agregar registro con el id del sensor*/
INSERT INTO registro VALUES(null,1,@idSens,now());

END//
DELIMITER ;

```

Este query permite actualizar los niveles de los tachos en función de los desechos ingresados. Dependiendo de la información que los sensores detecten, si es metal, plástico u orgánico, se añaden los registros en la base de datos. También permite cambiar el estado de un contenedor de vacío a lleno en el caso que 5 objetos hayan sido sensados como el mismo tipo de material.

```
DELIMITER //
CREATE PROCEDURE getRetiro(IN idCont int)
BEGIN
    /*Obtenemos la lista de todos los retiros */

    SELECT retiro.idRetiro as id,
           retiro.fecha,
           retiro.tachoPlastico,
           retiro.tachoMetal,
           retiro.tachoOrganico,
           usuario.nombre
    FROM retiro, usuario
    WHERE usuario.idUsuario=retiro.idUsuario;

END//
DELIMITER ;
```

Este query permite obtener la lista de todas las veces que un contenedor y sus tachos han sido vaciados en función del usuario y la fecha de retiro.

```
DELIMITER //
CREATE PROCEDURE addRetiro(IN tachoPlastico int, IN tachoMetal
int, IN tachoOrganico int)
BEGIN

INSERT INTO sensor VALUES(null, inductor, fotoelectrico);

SELECT MAX(idSensor) into @idSens from sensor;

SELECT nivelPlastico into @nPla from contenedor where
idContenedor=1;
SELECT nivelMetal into @nMet from contenedor where idContenedor=1;
SELECT nivelOrganico into @nOrg from contenedor where
idContenedor=1;

    IF (tachoPlastico==1) THEN
        UPDATE contenedor
        SET tachoPlastico=0 WHERE idContenedor=1;
    END if;

    IF (tachoMetal==1) THEN
        UPDATE contenedor
        SET tachoMetal=0 WHERE idContenedor=1;
    END if;

END//
DELIMITER ;
```

```

END if;

IF (tachoOrganico==1) THEN
    UPDATE contenedor
    SET tachoOrganico=0 WHERE idContenedor=1;
END if;

INSERT INTO registro VALUES (null,1,@idSens,now());

END//
DELIMITER ;

```

Este query permite añadir un retiro de basura al sistema. Dependiendo del id del contenedor y el usuario conectado, se añade a la base de datos la información de la fecha, hora, contenedor y tacho vaciado.

A continuación, se muestran los archivos .php usados.

1:

```

<?php

require("../config.inc.php");

if (!empty($_POST))
{
    $query = "CALL getPercen(:idContenedor) ";
    $query_params = array(

        ':idContenedor' => urldecode($_POST['idContenedor'])
    );
    try {
        $stmt = $db->prepare($query);
        $result = $stmt->execute($query_params);
        $row = $stmt->fetch();
        if ($row)
        {
            $response["nivelPlastico"] =
$row['nivelPlastico'];
            $response["nivelMetal"] = $row['nivelMetal'];
            $response["nivelOrganico"] =
$row['nivelOrganico'];
            $response["success"] = 1;
            $response["message"] = "Datos obtenidos";
            die(json_encode($response));
        }
    }
    catch (PDOException $ex) {
        $response["success"] = 0;
        $response["message"] = $ex->getMessage();
        die(json_encode($response));
    }
}

```

```

    }
}
else
{
    ?>
    <html>
    <head>
    <h1>Login</h1>
    <form action="index.php" method="post">
        Username:<br />
        <input type="text" name="username"
placeholder="username" />
        <br /><br />
        Password:<br />
        <input type="password" name="password"
placeholder="password" value="" />
        <br /><br />
        <input type="submit" value="Login" />
    </form>
    <a href="register.php">Register</a>
    <?php
}
?>

```

2:

```

<?php

require("../config.inc.php");

if (!empty($_POST))
{
    $query = "CALL
addRecord(:inductor,:fotoelectrico,:tPlastico,:tMetal,:tOrganico) ";
    $query_params = array(
        ':inductor' => urldecode($_POST['inductor']),
        ':fotoelectrico' => urldecode($_POST['fotoelectrico']),
        ':tPlastico' => urldecode($_POST['tPlastico']),
        ':tMetal' => urldecode($_POST['tMetal']),
        ':tOrganico' => urldecode($_POST['tOrganico'])
    );

    try {
        $stmt = $db->prepare($query);
        $result = $stmt->execute($query_params);
        $response["success"] = 1;
        $response["message"] = "Datos ingresados";
        die(json_encode($response));
    }
    catch (PDOException $ex) {
        $response["success"] = 0;
    }
}

```

```

        $response["message"] = "No se pudo subir los datos".$sex-
>getMessage();

        die(json_encode($response));

    }

}
else
{
    ?>
    <html>
        <head>
            <h1>Login</h1>
            <form action="index.php" method="post">
                Username:<br />
                <input type="text" name="username"
placeholder="username" />
                <br /><br />
                Password:<br />
                <input type="password" name="password"
placeholder="password" value="" />
                <br /><br />
                <input type="submit" value="Login" />
            </form>
            <a href="register.php">Register</a>
        <?php
    }
?>

```

3:

```

<?php

require("../config.inc.php");

if (!empty($_POST))
{
    $query = "CALL getRetiro(:idContenedor) ";
    $query_params = array(

        ':idContenedor' => urldecode($_POST['idContenedor'])
    );
    try {
        $stmt = $db->prepare($query);
        $result = $stmt->execute($query_params);
        $row = $stmt->fetch();

        if($row)
        {
            $response["success"] = 1;
            $response["message"] = "Datos Obtenidos";
            $i=0;
            while($row)
            {

```



```

        $i+=1;
        $array_id[$i]=$row['id'];
        $array_nombre[$i]=$row['nombre'];
        $array_fecha[$i]=$row['fecha'];
        $array_tPlastico[$i]=$row['tachoPlastico'];
        $array_tMetal[$i]=$row['tachoMetal'];
        $array_tOrganico[$i]=$row['tachoOrganico'];
        $row = $stmt->fetch();
    }
    $response["id"]=$array_id;
    $response["nombre"]=$array_nombre;
    $response["fecha"]=$array_fecha;
    $response["tPlastico"]=$array_tPlastico;
    $response["tMetal"]=$array_tMetal;
    $response["tOrganico"]=$array_tOrganico;
    die(json_encode($response));
}
}
catch (PDOException $ex) {
    $response["success"] = 0;
    $response["message"] = $ex->getMessage();

    die(json_encode($response));
}
}
else
{
    ?>
    <html>
        <head>
        <h1>Login</h1>
        <form action="index.php" method="post">
            Username:<br />
            <input type="text" name="username"
placeholder="username" />
            <br /><br />
            Password:<br />
            <input type="password" name="password"
placeholder="password" value="" />
            <br /><br />
            <input type="submit" value="Login" />
        </form>
        <a href="register.php">Register</a>
    <?php
}
?>

```

## Análisis de presupuesto

Con respecto a los materiales usados y a partir de la lista de hardware mostrada en secciones previas, se tiene el siguiente reporte de gasto.

CANTIDAD	NOMBRE	PRECIO UNIT.	PRECIO TOTAL
3	Servomotor	\$2.50	\$7.50
4	Módulo infrarrojo	\$2.00	\$8.00
1	Módulo láser	\$2.00	\$2.00
1	Fotocelda	\$0.75	\$0.75
1	Sensor inductivo	\$25	\$25
1	Arduino Uno	\$12	\$12
1	Módulo de alimentación	\$3.50	\$3.50
1	Raspberry PI***	\$130.00	\$130.00
TOTAL			\$58.75

\*\*\* El elemento no será tomado en cuenta debido a que no pertenece netamente a un contenedor.

El costo total de la parte electrónica del contenedor está en aproximadamente \$60. Comparando con la reducción de costos operativos en función de mano de obra encargada de separar la basura, existe una ganancia real. El componente de mayor precio es el sensor inductivo, debido a que su funcionalidad ya es de tipo industrial y bastante específica.

El resto de los elementos tienen precios razonables, sin embargo, es posible buscar por componentes más específicos que resultarían más caros, como para identificar por ejemplo, el tipo de plástico ingresado, mejorando considerablemente el proceso de separación de basura.

## Conclusiones

- La planificación inteligente de recolección requiere menos camiones, por lo tanto, menos combustible y a su vez tiempo, reduciendo así los costos operativos hasta en un 80%.
- Favorece en un medioambiente más verde. Elimina los contenedores desbordados y minimiza las emisiones nocivas dando como resultado ciudades más limpias.
- Reduce la huella de ecológica generada por la empresa lo que favorece reduciendo su impacto ambiental.

## Referencias bibliográficas

- [1] «Más familias ya reciclan en Guayaquil, pero falta reforzar el acopio,» *El Universo*, 16 may 2019.
- [2] Naylamp Mechatronics, «Sensor de proximidad inductivo LJ12A3-4-Z/BX,» [En línea]. Available: <https://naylampmechatronics.com/sensores-proximidad/136-sensor-de-proximidad-inductivo-lj12a3-4-zbx-npn.html>.
- [3] L. Llamas, «DETECTOR DE METALES CON ARDUINO Y SENSOR INDUCTIVO,» 23 October 2016. [En línea]. Available: <https://www.luisllamas.es/detector-de-metales-con-arduino-y-sensor-inductivo/>.
- [4] J. L. Ahedo y A. Ahedo, «Módulo sensor de Infrarrojos IR FC-51,» [En línea]. Available: <https://www.web-robotica.com/arduino/conceptos-basicos-arduino/como-usar-el-modulo-sensor-de-infrarrojos-ir-fc-51-para-evitar-obstaculos-con-robot-arduino-genuino>.

## Anexos



*Ilustración 21: Prototipo ensamblado en físico*