



Universidad Nacional de La Matanza

**Departamento de Ingeniería e Investigaciones Tecnológicas**  
**Ingeniería en Informática**

Sistemas Operativos Avanzados

Sistemas Embebidos - Arduino y Android

**“Smart Trash”**

Segundo Cuatrimestre 2018 - Año 2018

**Docentes:**

- Lic. Graciela de Luca
- Ing. Waldo Valiente
- Ing. Esteban Carnuccio
- Ing. Mariano Volker
- Ing. Sebastián Barillaro
- Ing. Gerardo Garcia

**Integrantes:**

- |                            |              |
|----------------------------|--------------|
| • ESCOBAR, Fabián Norberto | DNI 39210915 |
| • ESTIGARRIBIA, Emmanuel   | DNI 38531035 |
| • FERRARESE, Martín        | DNI 36073729 |
| • TEJERINA, Ezequiel Luis  | DNI 39268074 |
| • VERA, Cristian           | DNI 33786091 |

## Índice

Introducción .....	3
Descripción general del sistema .....	3
Alcance del sistema .....	4
Sistema embebido Arduino.....	4
Aplicación Android.....	5
Componentes .....	5
Software utilizado .....	5
Hardware utilizado.....	5
Sistema embebido Arduino .....	5
Aplicación Android .....	6
Circuito .....	6
Implementación.....	7
Sistema embebido - Android .....	7
Módulo Bluetooth .....	7
PIR.....	7
IR .....	8
Fotorresistor.....	8
Pulsador .....	8
Piezo Buzzer.....	8
LED.....	9
RGB LED .....	9
Micro Servo Motor.....	9
Aplicación Android.....	9
Activities .....	9
Sensores.....	11
Modo de uso .....	13
Descripción de los componentes .....	13
Arduino Uno .....	13
Fuente de alimentación .....	15
Módulo Bluetooth HC-05 .....	15
Sensor PIR HC-SR501 .....	16
Sensor IR .....	17
Fotoresistor (LDR) .....	17
Pulsador .....	18
Piezo Buzzer .....	18
LEDs .....	19
LED RGB .....	19
Micro Servomotor SG-90.....	20
Diagramas .....	21
Diagrama físico .....	21
Diagrama lógico .....	22
Diagrama funcional.....	23

Diagrama de hardware .....	24
Diagrama de software .....	26
Imágenes del prototipo.....	27
Problemas presentados y mejoras a futuro .....	31

## Introducción

A lo largo de este informe se explica nuestro proyecto para Sistemas Operativos Avanzados, en el que pudimos experimentar en el mundo de los sistemas embebidos y de las aplicaciones mobile. Nos vimos obligados a alejarnos un poco de lo que veníamos acostumbrados a trabajar, es decir, solo software, para poder dedicarnos también al hardware. Llevamos a cabo el desarrollo completo de un proyecto constituido por dos sistemas desde el surgimiento de una idea que nos motive, el planeamiento, el modelado, la adquisición de los componentes necesarios y el entrenamiento para poder trabajar con aquello que no conocíamos, hasta la puesta en marcha de la idea, la simulación del sistema embebido, la prueba de cada componente, su disposición y su conexión, el armado de la estructura, el desarrollo de los sistemas de software, la comunicación entre ellos y el testeo para comprobar que todo funcione adecuadamente.

Primero se describe en detalle la idea y la motivación por la que la hemos elegido, el objetivo de cada uno de los sistemas y todo lo que abarcan. Luego, contamos cuáles son los componentes que se usaron, mostramos un diagrama del circuito con la conexión dispuesta, aclaramos el uso que se le da a cada uno y adjuntamos una descripción técnica y del funcionamiento de cada elemento. Finalmente, se muestran los diagramas usados para el modelado, se cuentan cuáles son los problemas con los que nos hemos topado y se deja una propuesta de mejora a futuro.

## Descripción general del sistema

Smart Trash es una idea que surge de nuestras ganas de desarrollar un producto que brinde comodidades a las personas. Entre otras ideas, esta fue la que prevaleció ya que fue la que más nos gustó e interesó concretar.

Smart Trash es una papelera inteligente que permite arrojar papeles para mantener ordenado y limpio nuestro ambiente, ya sea en el ámbito laboral, de estudio o simplemente en el hogar, con la facilidad de no tener que abrir el tacho ni levantarse del asiento ya que este se abrirá automáticamente cuando detecte que se lo utilice. Esto nos motivó mucho ya que nos suele ocurrir que estamos ocupados realizando otras tareas y ante la necesidad de tirarlos y el tener que evaluar suspender por un momento nuestras actividades, suele optarse por no tirar los bollos de papel y simplemente ir colocándolos a un costado hasta que se forme una montaña de papel o hasta la próxima vez que tengamos que levantarnos y pasar cerca del cesto para poder arrojarlos.

Ya que esta idea ya fue implementada anteriormente, decidimos agregar algunas funcionalidades para distinguirla como información del estado de llenado del tacho con alertas de luces y sonido, y control de la luminosidad de la habitación en la que se encuentre. Además, se brinda la posibilidad de comunicación con una aplicación mobile para Android desde la cual se puede abrir y cerrar la papelera sin la necesidad de estar tan cerca del tacho para que pueda detectar que se desea usarlo. También se puede ordenar el vaciado. A modo de entretenimiento, se ofrece la emisión de una simpática melodía como lo es la Marcha Imperial de Star Wars para distenderse por unos instantes, un modo que

llamamos Tacho Loco en que la tapa abre y cierra el tacho rápidamente y de forma indeterminada, y también se puede cambiar a amarillo el color del led RGB.

Para su uso, usted debe conectar la papelera a la red eléctrica mediante la ficha de su transformador, o también alimentarlo por cable USB y colocarlo en un lugar que le sea cómodo y que quede de frente al lugar en el que se encuentre para que detecte cuando requiera que se abra. No debe colocar nada sobre la tapa del producto que impida que se abra cuando corresponda incluso atentando contra su integridad. Además, debe contar con un smartphone con sistema operativo Android y por supuesto instalar la aplicación de Smart Trash, para poder usarla cuando usted quiera. Cabe aclarar que debe tener encendido el Bluetooth del dispositivo, de lo contrario, se le pedirá que lo encienda.

## **Alcance del sistema**

### **Sistema embebido Arduino**

El sistema embebido Arduino es el que provee el funcionamiento de la papelera inteligente. Este sistema reside en la placa de prototipado Arduino con la que cuenta el cesto, la que se encuentra detrás de la papelera, dentro de la cajita.

El tacho enciende los LEDs que se encuentran en su frente dando información acerca del nivel de llenado: el LED verde está encendido mientras el tacho no está lleno, es decir, está vacío o aún tiene lugar disponible, y el LED rojo está encendido cuando el tacho está lleno; ambos LEDs se mantienen encendidos durante el vaciado; el LED azul se enciende cuando el tacho está en modo Bluetooth para ser manejado desde la aplicación Android. El LED transparente se apaga o enciende mostrando diferentes colores de acuerdo a la situación, diferenciando cuando en el ambiente hay mucha luz o poca luz, o cuando se está emitiendo la marcha imperial, o cuando se indica un cambio de color desde la aplicación.

Cuando se agita el celular, la tapa se levanta para que se pueda arrojar un bollo de papel y se baja cuando se vuelve a agitar. Cuando se activa el sensor de proximidad del celular, se inicia el modo Tacho Loco, abriendo y cerrando el tacho sin parar. Y la tapa se mantiene levantada cuando se ordena el vaciado del tacho al ser presionado el pulsador amarillo que se encuentra en el perfil derecho o cuando se indica desde la aplicación.

Adelante y sobre la tapa de la papelera, se encuentran los sensores de proximidad que controlan el comportamiento de la tapa. Adentro, está colocado el sensor IR que se encarga de controlar si el tacho se llena. En el perfil izquierdo, se encuentra el buzzer que emite la marcha imperial y el sonido de alerta indicando que el tacho está lleno. El buzzer también emite un sonido tras ser alimentado para indicar que está disponible para ser utilizado. Por último, en la arista delantera izquierda se encuentra el sensor que capta la luminosidad.

## **Aplicación Android**

La aplicación mobile para Android es el sistema que provee la comunicación del smartphone con la papelera inteligente, la que se establece mediante conexión vía Bluetooth. La app tiene una interfaz muy sencilla para facilidad del usuario. Primero que nada muestra los dispositivos Bluetooth emparejados y ofrece agregar y quitar conexiones. Permite abrir y cerrar el tacho como alternativa a la detección de proximidad, así como permite cambiar el color que se emite en el led RGB, ordenar el vaciado, y habilitar la marcha imperial y el tacho loco. Hace uso de tres sensores del framework de Android: sensor de proximidad, acelerómetro y giróscopo. En base a sus mediciones, ofrece comandos según si el smartphone es agitado, si es colocado de costado o si su sensor frontal detecta proximidad.

## **Componentes**

### **Software utilizado**

- IDE Arduino Genuino 1.8.7 (para el desarrollo del sistema embebido para la placa Arduino Uno)
- IDE Android Studio 3.2.1 (para el desarrollo de la aplicación mobile para Android).
- Android SDK 28.0.1 (software development kit para Android)
- ISO Android 4.0.3 Ice Cream Sandwich (para la simulación de la aplicación mobile para Android)
- Tinkercad Autodesk 2018 (para la simulación previa implementación del sistema embebido Arduino)

### **Hardware utilizado**

#### *Sistema embebido Arduino*

- 1 placa de prototipado Arduino Uno
- 1 cable USB puerto serial Arduino Uno 30cm
- 1 mini Protoboard 830 puntos
- 1 kit de jumpers con sockets
- 1 tira de headers macho-macho
- 6 resistores de 220 $\Omega$  5%
- 1 resistor de 1K $\Omega$  5%
- 1 módulo Bluetooth HC06
- 1 LED rojo 5mm
- 1 LED verde 5mm
- 1 LED azul 5mm
- 1 módulo LED RGB
- 1 pulsador Tact Switch 6mm x 6mm x 5mm

- 1 micro servo motor Tower Pro Sg90 1.2kg
- 1 Piezo Buzzer activo 5v
- 1 fotorresistor Ldr 5mm
- 2 sensor PIR HC Sr501
- 1 sensor IR Pcb A0106
- 1 papelera
- 1 fuente transformador 9v 1a

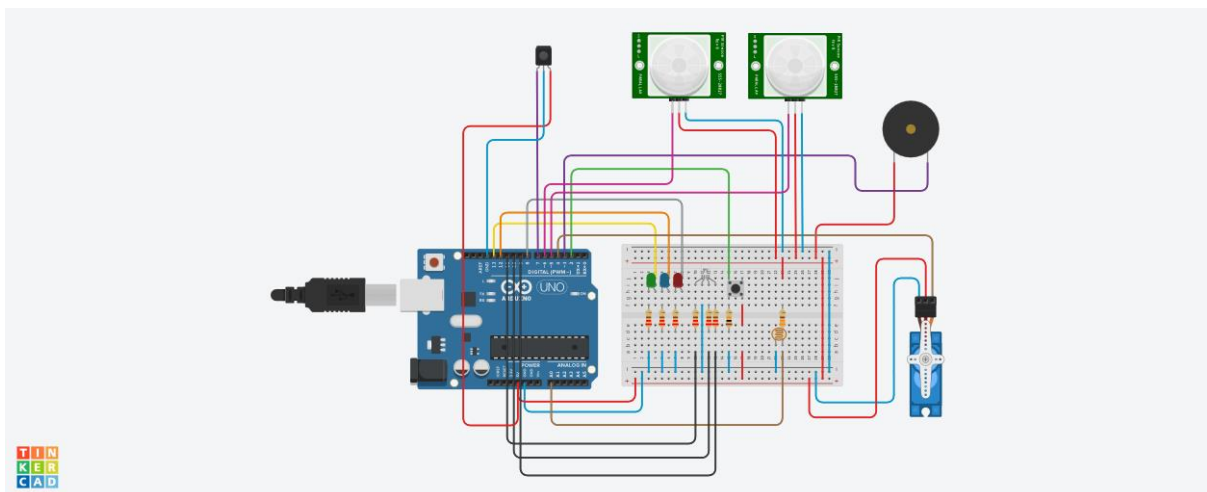
#### *Aplicación Android*

- Smartphones con sistema operativo Android (para usar la app)
- Sensor de proximidad
- Acelerómetro
- Giroscopio

## Circuito

El circuito fue diseñado y simulado desde el comienzo usando la aplicación online Tinkercad de Autodesk para asegurarnos de cómo se conectan los componentes y que de esa forma funcionan correctamente. También lo usamos como precaución para no correr el riesgo de quemar ningún elemento y en un principio entrenamos aquí para empezar a entrar en contacto con el mundo Arduino.

El proyecto puede encontrarse en <https://www.tinkercad.com/things/dwYeEtnybhy>, donde puede ser simulado y copiado para ser modificado si alguien lo desea. Allí también puede verse el código con el que fue programado. El circuito simulado se muestra en la siguiente imagen:



La única diferencia con la realidad es el uso del módulo Bluetooth el cual no está disponible en este simulador y por lo tanto no pudimos incluir en el diagrama del circuito. El mismo tiene pines positivo y negativo, y dos pines de señal que están conectados a las entradas digitales 0 y 1. El pin Rx del módulo Bluetooth está conectado al pin Tx de la placa Arduino y el pin Tx del módulo Bluetooth está conectado al pin Rx de la placa Arduino.

## Implementación

### Sistema embebido - Android

#### *Módulo Bluetooth*

Haciendo uso de la librería “SoftwareSerial.h”, desarrollamos la comunicación del módulo Bluetooth integrado en el Arduino con la aplicación Android. Es una pieza clave en el producto, ya que, a partir de dicha comunicación, se comparten datos que, en base a qué valores tengan, se realizarán distintas acciones (tanto en el sistema embebido Arduino como en la aplicación Android).

Se utilizan las siguientes funciones:

- SoftwareSerial: crea una nueva instancia del objeto SoftwareSerial.
- available: obtiene el número de bytes disponibles para la lectura de un puerto serie software.
- begin: se configura la velocidad en baudios para la comunicación serie.
- read: devuelve un carácter que se recibió en el pin RX del puerto serie de software.
- write: envía los comandos al dispositivo Android.

Básicamente, los datos enviados por la aplicación Android, indican:

- apertura y cierre del tachó.
- modo marcha: emisión de la marcha imperial.
- modo tachó loco: el tachó se abre y cierra.
- cambio de color del led RGB: apagado y color alternativo.

#### *PIR*

Para su implementación, lo único que hicimos es obtener los valores del sensor a través de la función digitalRead. Se le debe configurar la sensibilidad tanto de distancia como de tiempo. Lo configuramos para que tenga un buen alcance para brindar más comodidades a las personas que lo usen.

El sensor PIR viene con un lente de Fresnel que tiene el objetivo de permitir una mayor sensibilidad, pero también amplía el ángulo de detección, por lo que decidimos cubrir parcialmente el lente para que sense solo el rango que nos interese. Uno de los sensores PIR está colocado en el frente del tachó para detectar movimiento adelante y el otro sensor se encuentra en la tapa para que detecte movimiento arriba del tachó.



Si los sensores detectan movimiento, la papelera se abre, y si los sensores no detectan movimiento, la papelera se cierra.

### *IR*

Para su implementación, de la misma forma que con los sensores PIR, lo único que hicimos es obtener los valores del sensor a través de la función `digitalRead`. Este sensor se encuentra ubicado dentro del tacho, apuntando desde la cara trasera hacia la cara delantera para cubrir la apertura. Si el sensor detecta una obstrucción dada por un objeto, en este caso papel, entonces se comprende que el tacho está lleno.

### *Fotorresistor*

El fotorresistor es usado para detectar el nivel de luz que hay en el ambiente y para eso se obtiene el valor de su lectura a través de la función `digitalRead`. Este sensor se encuentra ubicado en la arista izquierda del frente de la papelera, posición estratégica pensando que el tacho seguramente esté colocado de espaldas a una pared. Allí, la luz va a ser detectada sin inconvenientes.

### *Pulsador*

El pulsador se utiliza para indicar el vaciamiento del tacho. Para esto, se hace uso de la función `digitalRead`. Cuando se detecta que fue presionado, se activa el vaciamiento del tacho, por lo que la tapa se queda abierta hasta que se termine de quitar los papeles del interior. Cuando deja de estar presionado el pulsador, la tapa se baja. El pulsador está ubicado en el perfil derecho.

### *Piezo Buzzer*

El Piezo Buzzer se usa para la emisión de los sonidos. Para reproducir los tonos, se usa la función `tone`, a la que se indica la frecuencia y la duración del tono, y para no reproducir un tono se usa la función `noTone`. El speaker se encuentra en el perfil izquierdo.

Por una parte, reproduce la alarma que indica que la papelera está llena, la que consta de dos pitidos que se repetirán hasta que no se vacíe el tacho. Por otra parte, reproduce la marcha imperial de Star Wars, la melodía que elegimos para la funcionalidad de entretenimiento. La marcha imperial está formada por una cantidad mayor de tonos con diversidad de frecuencias y duraciones. Además, se usa para emitir un sonido una vez que el dispositivo está listo para usarse tras haber sido conectado.

Para la correcta formación de la canción de la marcha imperial, es necesaria la utilización de la función `delay` provista por Arduino para la implementación de las pausas. Y si bien es desalentado su uso debido al parate que implica en el desarrollo de tareas, no queda más que usarla porque de lo contrario la marcha no se puede emitir adecuadamente.

### *LED*

Los leds son uno de los elementos más sencillos de utilizar ya que se encienden y apagan con la función `digitalWrite`, variando el parámetro de estado entre HIGH y LOW, según corresponda. Todos los leds se encuentran en el frente de la papelera, en la esquina superior derecha.

El led rojo se enciende cuando el tacho está lleno, el led verde se enciende cuando el tacho está vacío o no está vacío pero aún tiene lugar, ambos están encendidos durante el vaciado y el led azul se enciende cuando el modo Bluetooth está activado.

### *RGB LED*

Este led RGB tiene un uso similar al de un led común, pero para la formación de un color se da un valor a cada componente con la función `digitalWrite`, como si se estuviese manejando a tres leds diferentes.

Este led se encuentra en el frente de la papelera, en la esquina superior izquierda. Le asignamos varios usos a este led. Toma distintos colores o se apaga según la ocasión, variando en las siguientes: mucha luz en el ambiente, poca luz en el ambiente, alarma por tacho lleno, marcha Imperial y cambio de color.

### *Micro Servo Motor*

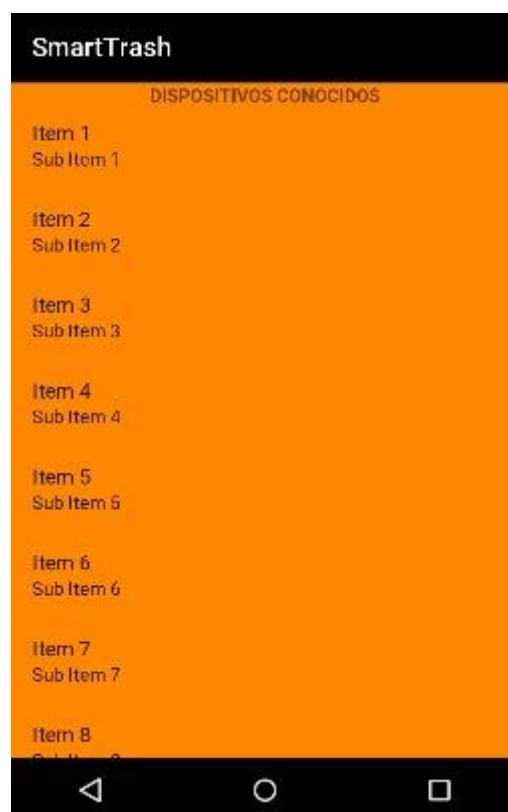
El servomotor tiene una implementación sencilla. Simplemente se maneja con la función `write` con la que se indica el ángulo de posición en el que se debe colocar el servo. Está colocado sobre el borde izquierdo del perfil izquierdo del tacho, pegado a la bisagra de la tapa. Solamente se usan dos posiciones: 180 grados para poner el tacho en abierto y 90 grados para poner el tacho en cerrado.

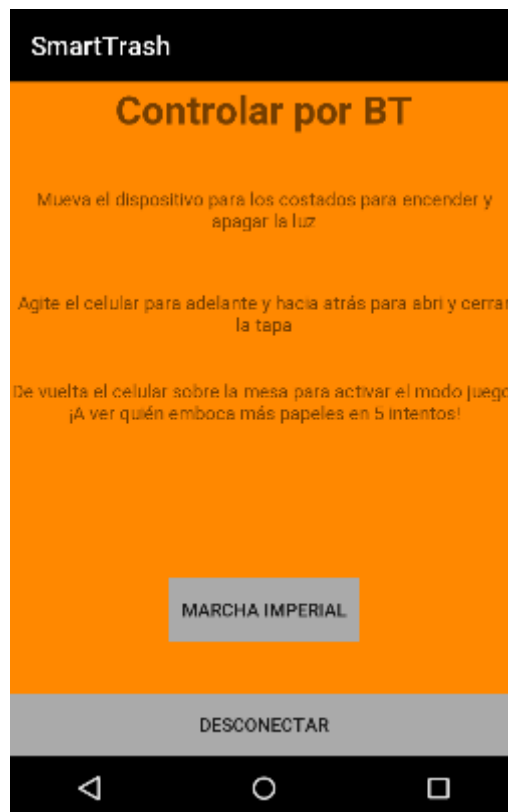
Además interviene obviamente en el modo tacho loco en el que se mueve la tapa sin parar. Para que la apertura y cierre tenga tiempo de realizarse completa, fue necesario el uso de la función `delay` para generar un retraso

## **Aplicación Android**

### *Activities*

A continuación, se muestran las activities que forman la aplicación:





## Sensores

### Giróscopo

El giróscopo, giroscopio o gyro, como se conoce en inglés, mide la velocidad de rotación del dispositivo alrededor de los ejes x , y, y z, y sus propiedades se resumen en esta tabla:

Sensor	Tipo	Datos de SensorEvent (rad/s)	Descripción
Giroscopio	TYPE_GYROSCOPE	values[0] values[1] values[2]	Velocidad de rotación alrededor del eje x Velocidad de rotación alrededor del eje y Velocidad de rotación alrededor del eje z

Este sensor lo usamos de la siguiente manera:

- cuando se detecta que el smartphone es girado hacia la izquierda, se envía un comando al sistema embebido Arduino para que se encienda el led RGB en su combinación amarilla;
- cuando se detecta que el smartphone es girado hacia la derecha, se envía un comando al sistema embebido Arduino para que se cambie el color del led rgb.

*Acelerómetro*

El sensor TYPE\_ACCELEROMETER mide la aceleración aplicada al dispositivo y sus propiedades se resumen en esta tabla:

Sensor	Tipo	Datos de SensorEvent (m/s <sup>2</sup> )	Descripción
Acelerómetro	TYPE_ACCELEROMETER	values[0] values[1] values[2]	Aceleración en el eje x Aceleración en el eje y Aceleración en el eje z

Este sensor lo usamos de la siguiente manera:

- cuando se detecta que el smartphone es agitado hacia adelante y hacia atrás, se envía un comando al sistema embebido Arduino para que se abra o cierre la papeleras, según corresponda, de acuerdo al estado en el que se encuentra la tapa en ese instante.

*Sensor de proximidad*

El sensor TYPE\_PROXIMITY proporciona la distancia entre el dispositivo y otro objeto y puede por ejemplo usarse para determinar si la persona lo sostiene cerca de su cara al hacer una llamada o simplemente pasa su mano cerca de él; y sus propiedades se resumen en esta tabla:

Sensor	Tipo	Datos de SensorEvent	Descripción
Proximidad	TYPE_PROXIMITY	values[0]	Distancia de un objeto en cm. Algunos sensores de proximidad solo declaran un valor booleano para indicar si el objeto se encuentra lo suficientemente cerca.

Este sensor lo usamos de la siguiente manera:

- cuando se detecta que el sensor de proximidad del smartphone está tapado, se envía un comando al sistema embebido Arduino para activar modo juego, el cual abre y cierra la tapa automáticamente unas 5 veces.

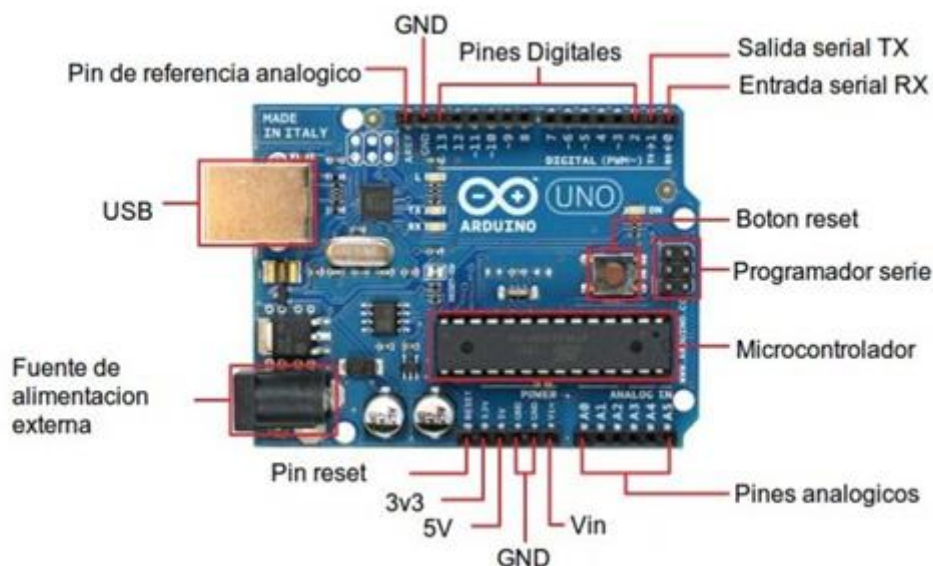
#### *Modo de uso*

Es muy sencillo utilizar la app, básicamente pueden realizarse las siguientes:

- previamente, se debe emparejar la papelera con el dispositivo Android por medio de los ajustes del propio sistema.
- abrir la aplicación SmartTrash.
- presionar el botón “Conectar Bluetooth” para ver las papeleras emparejadas.
- presionar la papelera emparejada a la que se desee controlar.
- realizar alguna de las acciones allí mencionada. ¡Hay muchas opciones!
- presionar el botón desconectar la papelera cuando quiera finalizar su uso.

## Descripción de los componentes

### Arduino Uno



Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de sistemas embebidos a los que la comunidad de creadores puede darles diferentes tipos de uso. Esto es, en gran parte, gracias a la alta compatibilidad con un vasto conjunto de sensores y actuadores de los más variados tipos.

En particular, en nuestro proyecto usamos la placa Arduino UNO R3, la cual es la más documentada de toda la familia de placas Arduino, según la propia página web oficial de Arduino. Esta, al igual que la mayoría, puede ser alimentada de varias formas, con un cable USB conectado al ordenador o con una fuente externa. El lenguaje de programación del microcontrolador está basado en C y C++.

Características técnicas:

Microcontrolador	ATmega328P
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Digital pines I/O	14 (de los cuales 6 proporcionan una salida PWM)
PWM digital pines I/O	6
Pines de entrada analógica	6
Corriente DC por Pin I/O	20mA
Corriente DC para Pin 3.3V	60mA
Memoria flash	32KB ATmega328P de los que 0,5 KB son utilizados por el gestor de arranque.
SRAM	2KB ATmega328P
EEPROM	1KB ATmega328P
Velocidad de reloj	16 MHz
Longitud	68,6 mm

Anchura	53,4 mm
Peso	25 g

### **Fuente de alimentación**



La fuente de alimentación es el dispositivo que convierte la corriente alterna, en una o varias corrientes continuas, que alimentan los distintos circuitos del aparato electrónico al que se conecta.

En nuestro caso, usamos una fuente de 9v, para alimentar la placa Arduino, junto con los sensores, actuadores, y el módulo bluetooth, para hacer al embebido independiente de tener que estar siempre conectado a una computadora y poderlo alimentar directamente con un tomacorrientes convencional.

### **Módulo Bluetooth HC-05**

Es el módulo que usamos para conectar el sistema embebido a un celular inalámbricamente. Se configura mediante comandos AT y tiene la posibilidad de hacerlo funcionar tanto en modo maestro como esclavo. Esto quiere decir que además de recibir conexiones desde una PC o celular, también es capaz de generar conexiones hacia otros dispositivos bluetooth.

El módulo Bluetooth HC-05 puede alimentarse con una tensión de entre 3.3 y 6V (normalmente 5V), pero los pines TX y RX utilizan niveles de 3,3V por lo que no se puede conectar directamente a placas de 5V, sino que hay que usar resistencias para limitar la misma.





**Sensor PIR HC-SR501**

Los detectores PIR (Passive Infrared Receiver) o receptor pasivo Infrarrojo, reaccionan ante el calor del cuerpo humano o animales. Estos captan la presencia detectando la diferencia entre el calor emitido por el cuerpo humano y el espacio alrededor. Son llamados pasivos debido a que no emiten radiaciones, sino que solamente las reciben.

Su componente principal son los sensores piroeléctricos. Se trata de un componente electrónico diseñado para detectar cambios en la radiación infrarroja recibida. Generalmente dentro de su encapsulado incorporan un transistor de efecto de campo

que amplifica la señal eléctrica que genera cuando se produce dicha variación de radiación recibida. El sensor está dividido en dos mitades. La razón de ello es que se desea detectar movimiento (cambio) y no los niveles promedio de IR. Las dos mitades están cableados de manera que se anulan entre sí. Si uno ve la mitad con más o menos radiación IR que el otro, la salida se moverá a alta o baja.

Además, cuenta con un lente de Fresnel, el cual es un encapsulado semiesférico hecho de polietileno de alta densidad cuyo objetivo es permitir el paso de la radiación infrarroja y concentrar la energía en la superficie de detección del sensor, permitiendo una mayor sensibilidad del dispositivo.

El HC-SR501 permite ajustar el tiempo de disparo de la señal de alarma de movimiento, con un mínimo de 3 seg y un máximo de 5 min. Así mismo, se puede ajustar la distancia de detección, con un mínimo de 3 mts y un máximo de 7 mts.

Especificaciones técnicas:

Voltaje de alimentación	5 a 12
Consumo promedio	Menor a 1mA
Distancia de detección	3 a 7 metros
Ángulo de detección	Cono de 110°
Tiempo de inicialización	1 minuto antes de que inicie su operación normal

Temperatura de operación	-15° a +70° C
--------------------------	---------------

### **Sensor IR**



Formado por un emisor y un receptor IR, es lo que usamos para detectar si el tacho está lleno.

El emisor es un led, que funciona con una longitud de onda de 940nm. Mientras que el receptor es un fotodiodo, un dispositivo que conduce una cantidad de corriente eléctrica proporcional a la cantidad de luz que lo ilumina o incide.

En la implementación establecimos un nivel (la altura donde ubicamos el sensor), y evaluamos si la recepción se corta o no. Este mecanismo nos sirve para contar la cantidad de veces que se tira algo dentro de la papelera, además de la función de detectar si está llena (por haber alcanzado la altura del sensor).

### **Fotoresistor (LDR)**

El LDR (Light-Dependent Resistor, resistor dependiente de la luz) es un resistor que varía su valor de resistencia eléctrica dependiendo de la cantidad de luz que incide sobre él. Se le llama, también, fotorresistor o fotorresistencia. El valor de resistencia eléctrica de un LDR es bajo cuando hay luz incidiendo en él (en algunos casos puede descender a tan bajo como 50 ohms) y muy alto cuando está a oscuras (puede ser de varios megaohms).



El uso que le dimos en nuestro proyecto es detectar el nivel de iluminación del ambiente, para prender el led RGB y hacer a la papelera más sencilla de ubicar. Además de poder controlarse desde la aplicación en Android.

### **Pulsador**



Un pulsador es básicamente un interruptor con dos estados, presionado o libre. Cuando se presiona, dentro del interruptor, dos cables son unidos, lo que permite a la corriente fluir y envía un estado HIGH a la placa. Cuando se libera, los dos cables son desunidos y corta el flujo de la corriente, significando un estado LOW de señal.

Usamos un pulsador en el proyecto para poder abrir la papelera cuando esta queda bloqueada (al detectar que se llenó), y cerrarla nuevamente para volver a activar los sensores.

### **Piezo Buzzer**

Un buzzer, en español también llamado zumbador, es un transductor electroacústico que produce un sonido o zumbido continuo o intermitente de un mismo tono.

La piezoelectricidad (de ahí el nombre de piezo buzzer) es un fenómeno que ocurre en determinados cristales que, al ser sometidos a tensiones mecánicas, adquieren una polarización eléctrica y aparece una diferencia de potencial y cargas eléctricas en su superficie que generan una tensión eléctrica. Es decir, que son materiales que al someterlos a una tensión eléctrica variable (como una señal PWM) vibran.



Utilizamos el buzzer con diversas funciones, desde alertar que la papelera está llena o que ya ha sido inicializada para usarse, hasta reproducir una melodía en el modo marcha imperial

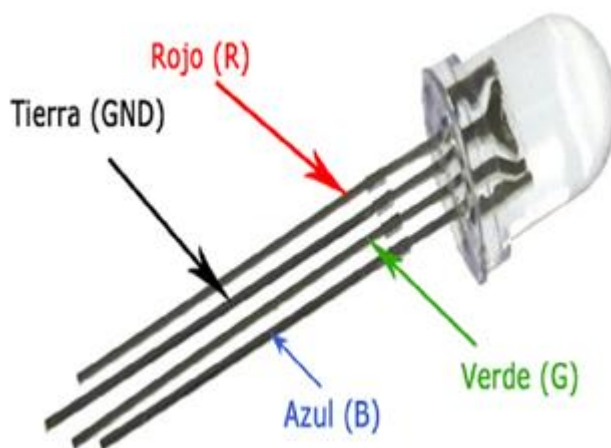
## LEDs



Un diodo emisor de luz (también conocido por la sigla LED, del inglés light-emitting diode) es una fuente de luz constituida por un material semiconductor dotado de dos terminales. Se trata de un diodo de unión p-n, que emite luz cuando está activado. Si se aplica una tensión adecuada a los terminales, los electrones se recombinan con los huecos en la región de la unión p-n del dispositivo, liberando energía en forma de fotones.

Usamos 3 leds: rojo, verde y azul. El rojo para indicar que la papelera está llena, el verde para indicar que sigue teniendo espacio, y el azul para indicar conexión con un dispositivo mediante bluetooth.

## LED RGB



Es un tipo de led especial, el cual permite crear combinaciones de los colores primarios en la iluminación (rojo, verde y azul), para mostrar muchos más colores.

Para eso, en el código se envían valores entre 0 y 255 para cada componente de color, representando la intensidad de estos, formando así una gran variedad de colores secundarios y terciarios.

En nuestro proyecto, el led RGB es usado para dar iluminación a la papelera en caso de baja iluminación del entorno, y para acompañar el modo marcha imperial.

**Micro Servomotor SG-90**

Un servomotor (también llamado servo) es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. Tiene incorporado un sistema de regulación que puede ser controlado tanto en velocidad como en posición.

El micro servo SG-90 tiene requerimientos de energía bastante bajos lo que permite alimentarlo con la misma fuente de alimentación que el resto del circuito.



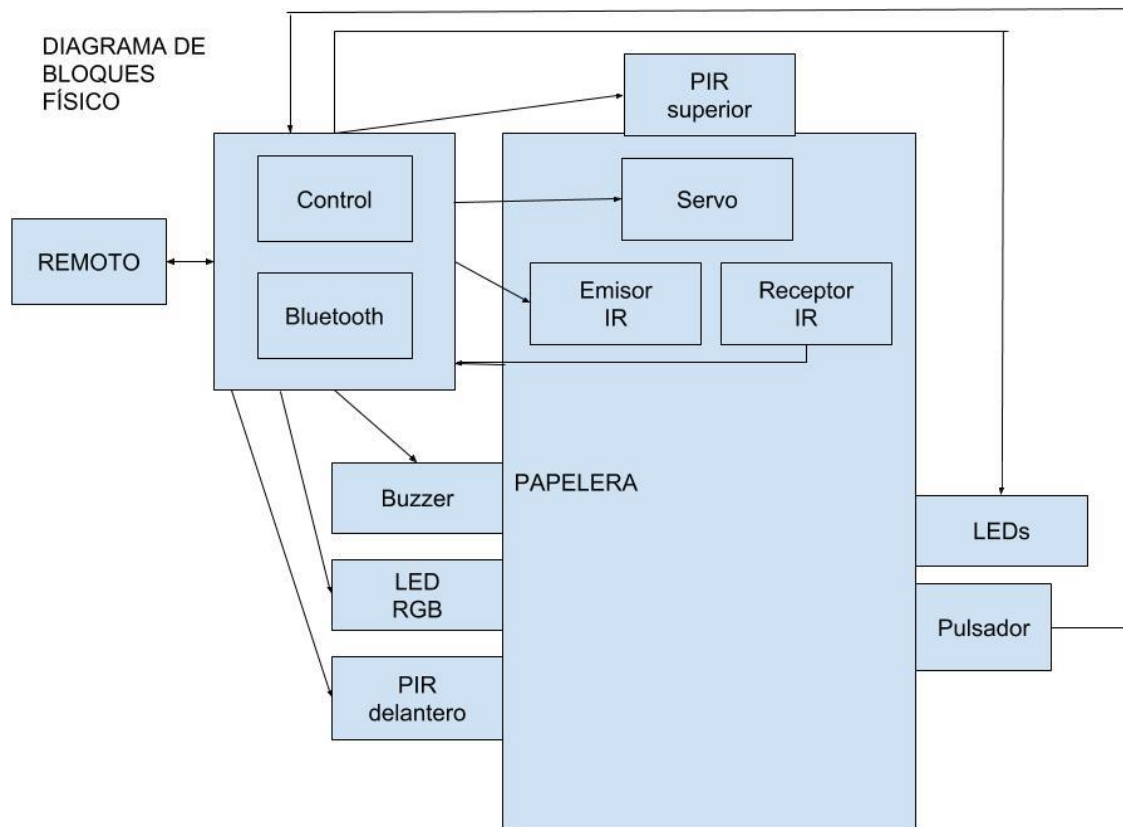
Características técnicas:

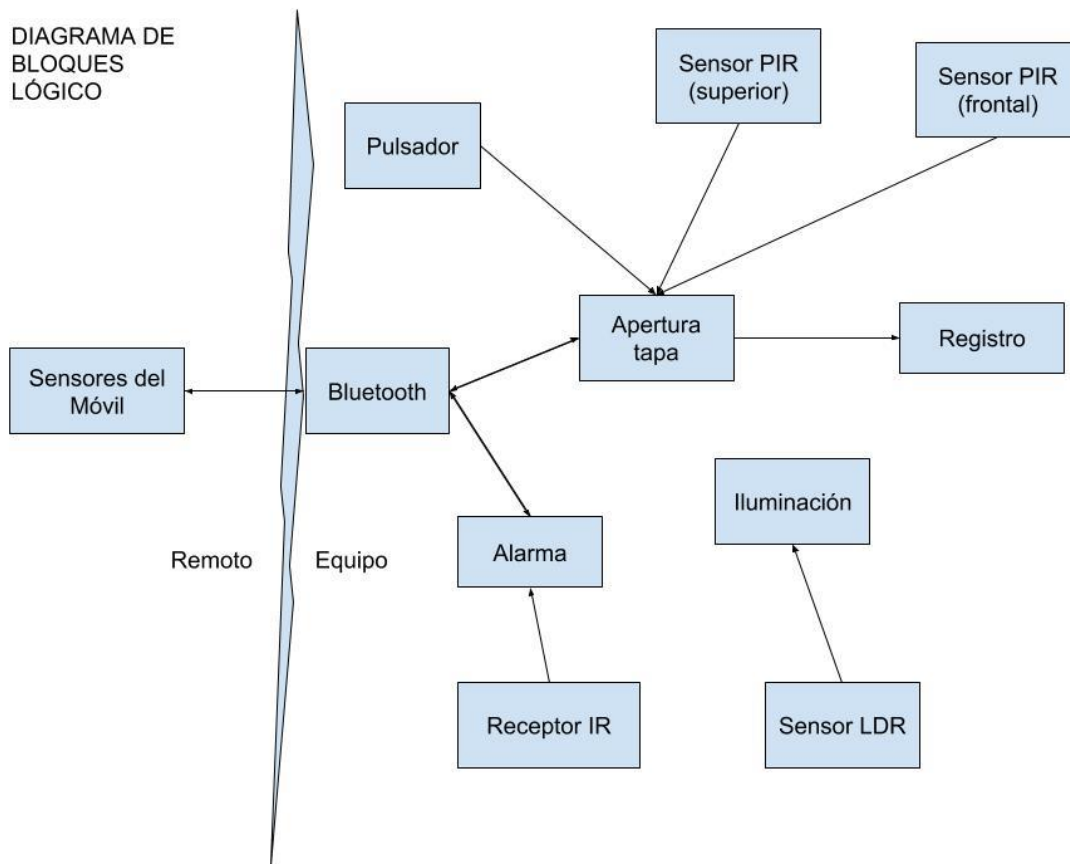
Velocidad	0.10 sec/60° @ 4.8V
Torque	1.8 Kg-cm @ 4.8V
Voltaje de funcionamiento	3.0-7.2V
Temperatura de funcionamiento	-30 °C a 60 °C
Ángulo de rotación	180°
Ancho de pulso	500-2400 $\mu$ s

En nuestro proyecto, el micro servo es el que mueve la tapa de la papelera, en función de lo que detecten los sensores, o del comando que el embebido reciba del celular.

## Diagramas

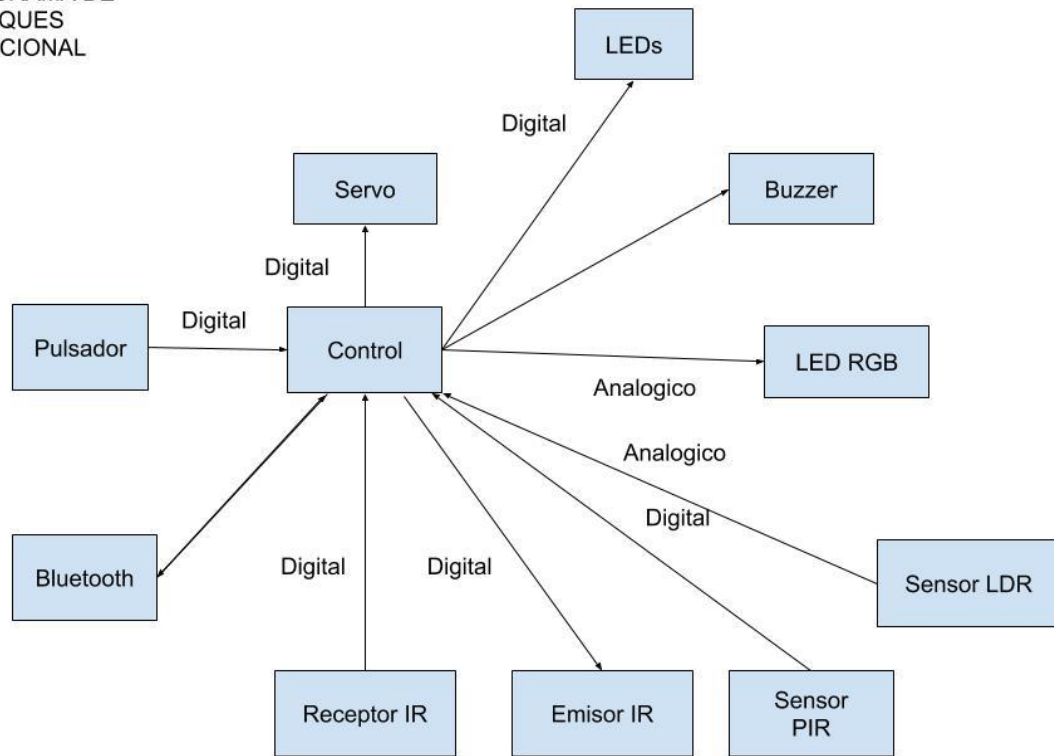
### Diagrama físico



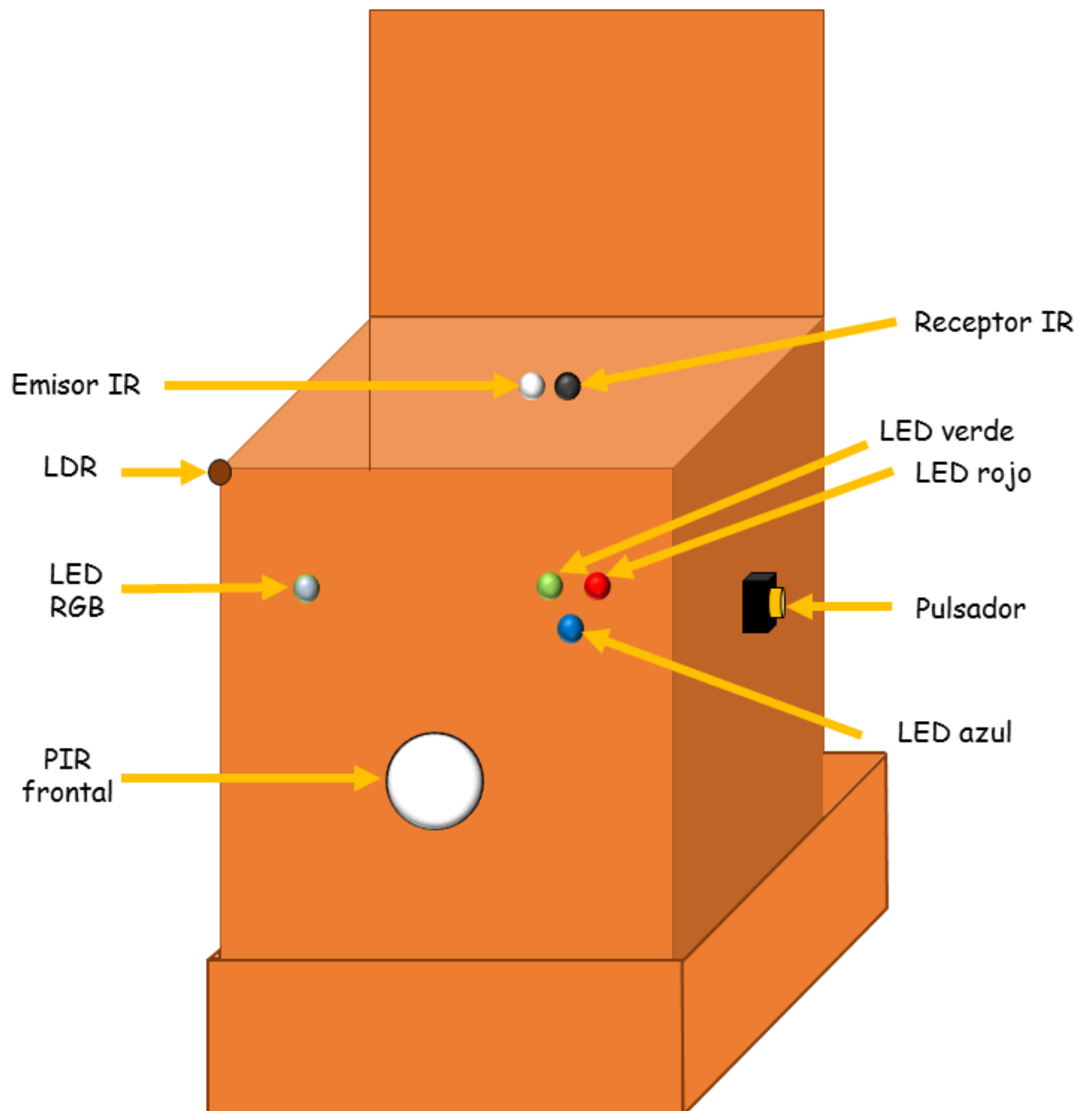
**Diagrama lógico**

**Diagrama funcional**

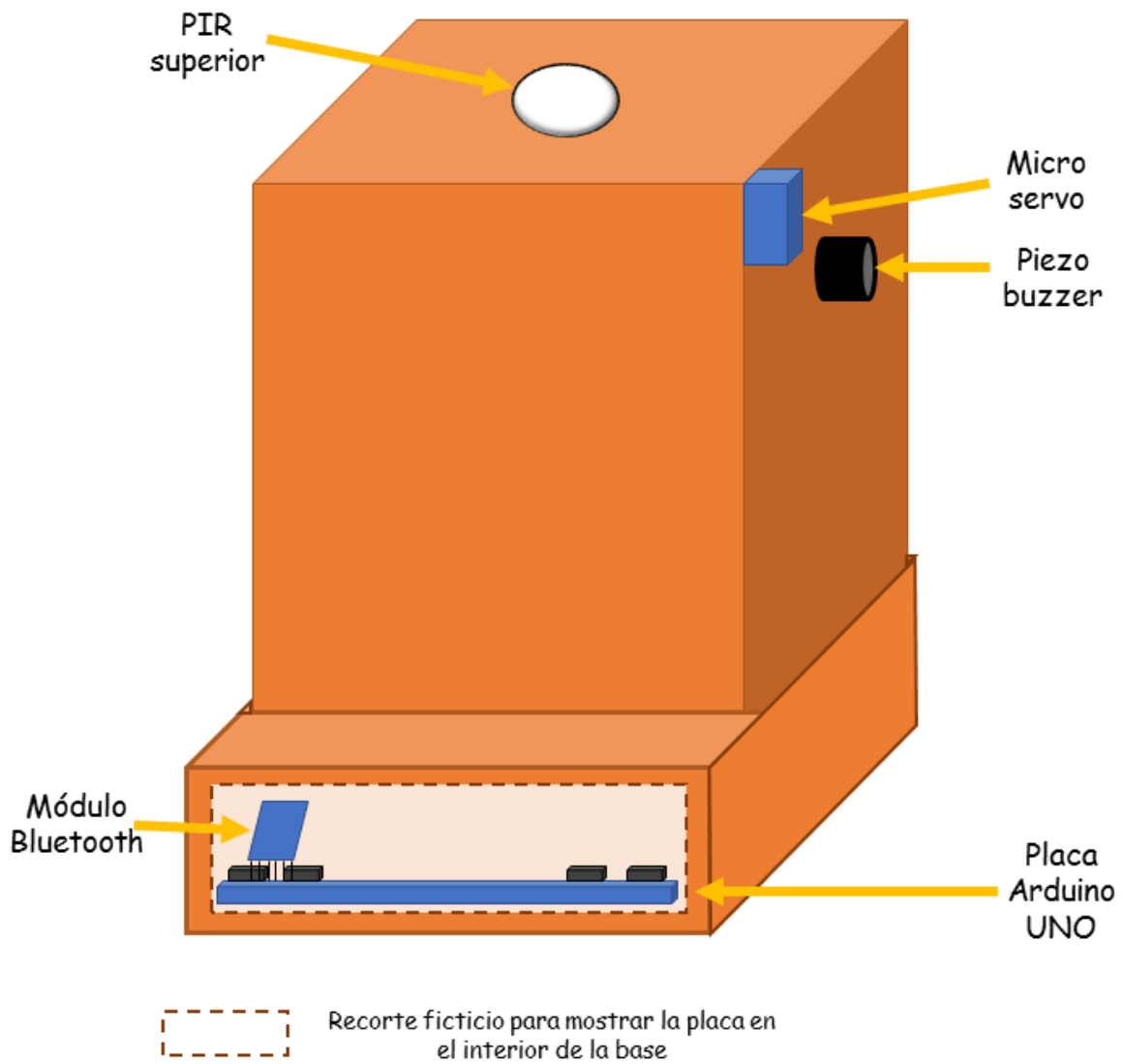
DIAGRAMA DE  
BLOQUES  
FUNCIONAL





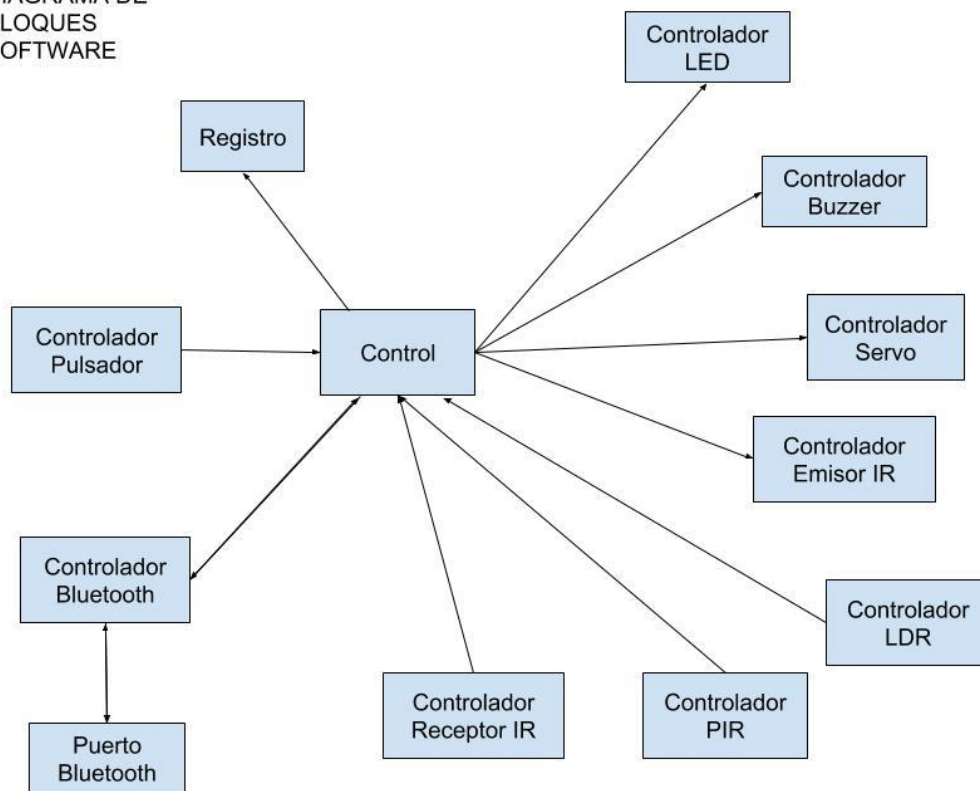
**Diagrama de hardware****Vista frontal**

## Vista trasera



**Diagrama de software**

DIAGRAMA DE  
BLOQUES  
SOFTWARE



## Imágenes del prototipo



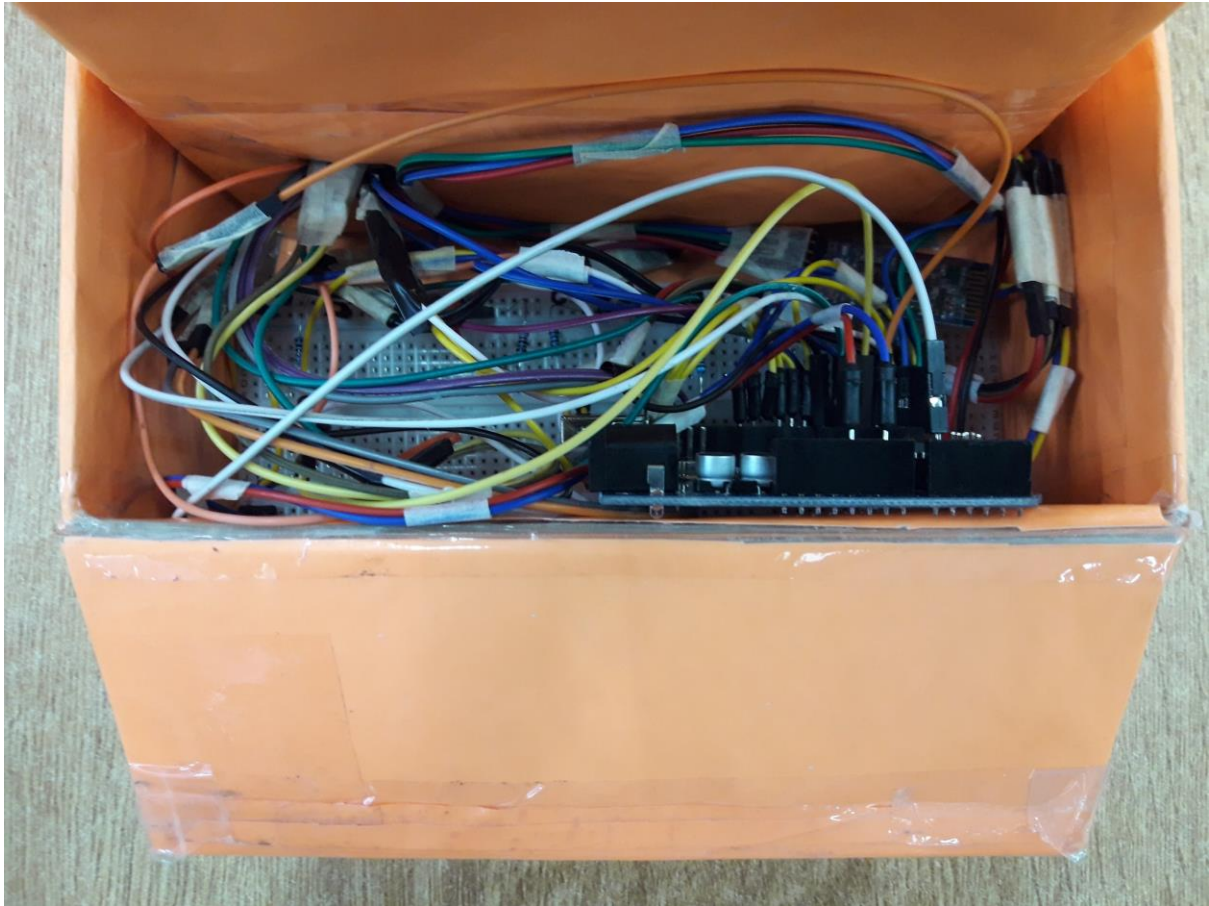












## Problemas presentados y mejoras a futuro

Los problemas principales o más significativos que tuvimos a la hora de realizar este proyecto fueron: la calibración de los sensores PIR, el uso de los sensores PIR e IR, la colocación y afirmación del servo para que maneje la tapa, y la imposibilidad de paralelismo para desarrollar más de una tarea al mismo tiempo. Una de las cosas que queríamos hacer es que la marcha imperial se emita al mismo tiempo que se activaba el tacho loco. A futuro se podría simular el paralelismo implementando concurrencia.

La mejora principal que se podría agregar es la inclusión de ruedas al tacho para que pueda moverse hasta donde uno se encuentre para solucionar el problema de que se encuentre demasiado lejos para arrojar bollos de papel. Nos gustaría poder conducir el tacho desde la aplicación y que incluso sepa esquivar obstáculos. Este fue un deseo que tuvimos en un principio pero que dejamos de lado debido al poco tiempo con el que contamos para realizar el proyecto.