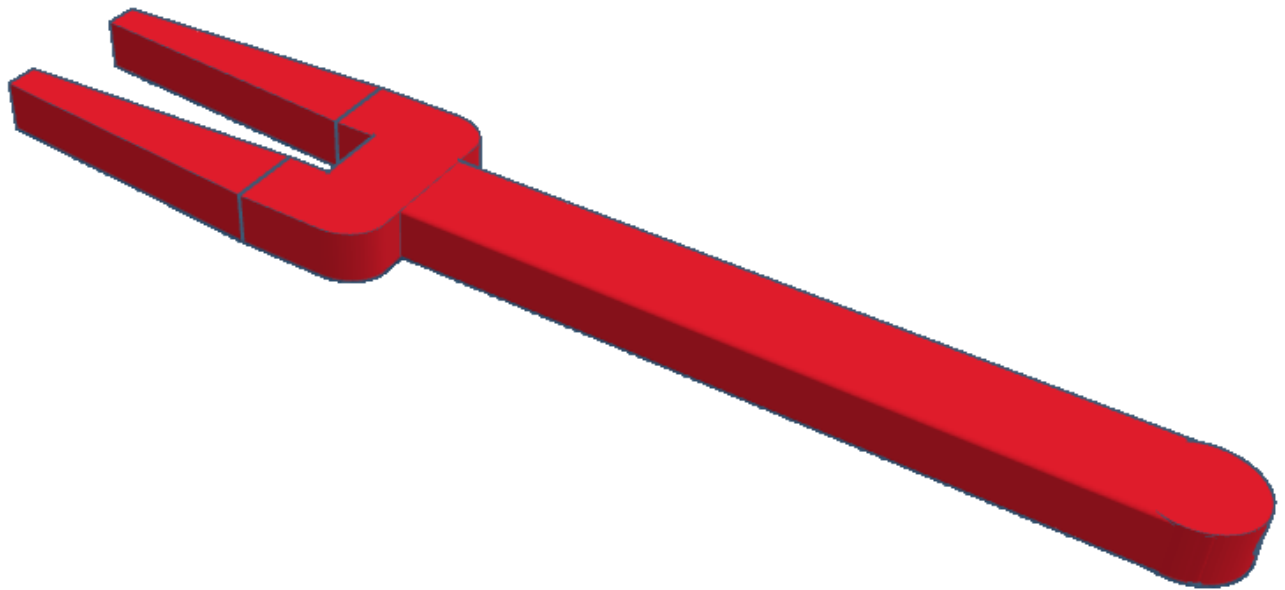


FOOD DETECTOR



Candela Esparrica Torrecilla
Pablo Cuadrado León
IES Vicente Aleixandre 4º ESO A

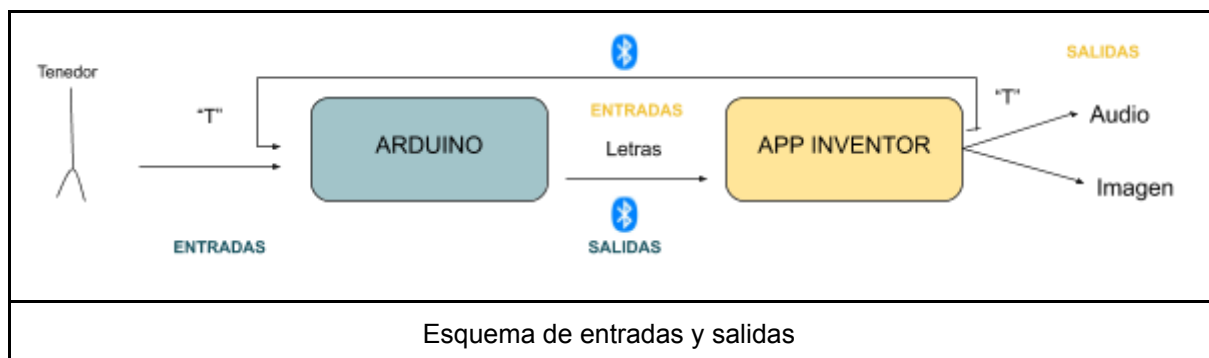
Índice

1. Finalidad del proyecto	3
2. Búsqueda de información	4
3. Planificación	5
3.1 Presupuesto hoja de cálculo	5
3.2 Proceso de fabricación	5
4. Diseño de la maqueta	6
4.1 Perspectiva de conjunto	6
4.2 Vistas de conjunto	7
5. Subsistemas	8
5.1 Hardware	8
5.2 Código Arduino	9
5.3 App Inventor	10
5.4 Diseño 3D	11
6. Hardware	12
6.1 Esquema protoboard	12
6.2 Esquema electrónico	13
7. Códigos	14
7.1 Arduino IDE	14
7.2 App Inventor	15
8. Fotos y Vídeos	17
9. Análisis del proyecto	18
9.1. Análisis funcionamiento del proyecto	18
Anexo	18
Evaluación funcionamiento del equipo	18

1. Finalidad del proyecto

La finalidad del proyecto es que al pinchar el tenedor en un alimento, se nos indique qué alimento es en función de su resistencia, mediante una aplicación en el móvil; el proyecto podría ser de utilidad para personas con discapacidad visual o con alergias a ciertos alimentos.

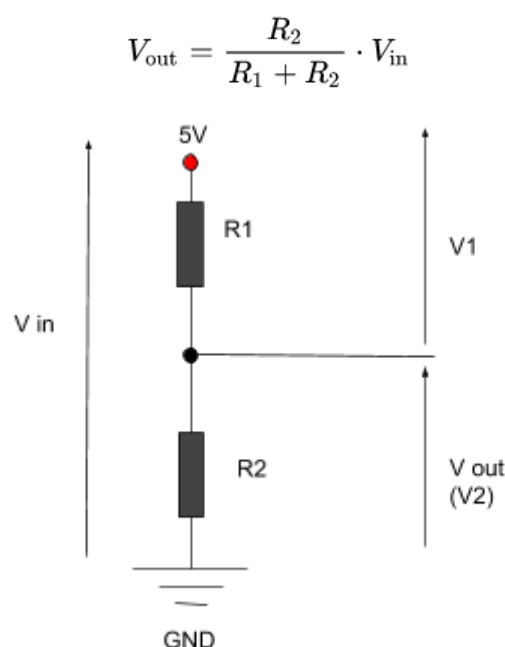
Para realizar ampliaciones hemos añadido en la aplicación del móvil diseñada con App Inventor mejoras, como añadirle un audio de voz informando del alimento.



2. Búsqueda de información

- [Food Detector](#)- hackster.io
Nos ha servido de base. Nos hemos basado en la información de los códigos de Arduino IDE y nos ha dado una idea general del proyecto.
- [Info. App inventor](#)- tecnopujol
Aquí encontramos los códigos para la conexión Bluetooth, tanto en los bloques de código de App Inventor como en las librerías que debíamos añadir en el software de Arduino IDE.
- [Imagen App inventor](#)- kio4.com
Encontramos esta página que nos sirvió de base para subir imágenes a App inventor, aunque tuvimos que hacer cambios y simplificarlo bastante.
- [Reproducir audios App Inventor](#)- kio4.com
Gracias a esta página pudimos tener una idea de cómo reproducir audios en App Inventor, y nos fue de utilidad para después añadir los bloques de audio finales.
- [Divisor de tensión](#)- Wikipedia
Un Divisor de Tensión o Divisor de Voltaje es un circuito que divide la tensión de entrada en el circuito en otras dos diferentes y más pequeñas de salida.
$$V_{in} = V_1 + V_{out}$$

Utilizamos el divisor de tensión en el Hardware ya que nuestro tenedor mide R_1 , pero Arduino no lee resistencias sino tensiones: Arduino lee V_2 , que es proporcional a R_1 , siendo R_1 la resistencia de las frutas. La relación entre R_1 y V_2 se basa en la siguiente fórmulas:



3. Planificación

3.1 Presupuesto hoja de cálculo

COMPONENTES				
Nº	Descripción	Cantidad	Precio Unitario	Precio Total
1	Arduino UNO	1	7,5	7,5
2	Módulo Bluetooth	1	8,49	8,49
3	Resistencia	1	0,10 €	0,10 €
4	Protoboard	1	1,57 €	1,57 €
5	Cables	1	1,16 €	1,16 €
			TOTAL	18,82 €

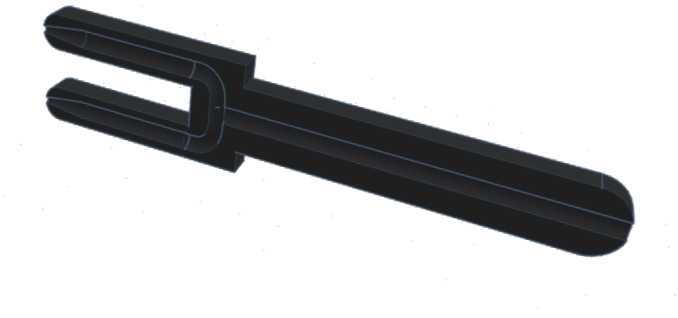
3.2 Proceso de fabricación

1. Realizar el hardware
2. Realizar el software (Arduino IDE)
3. Desarrollar aplicación con App inventor
4. Diseñar tenedor conTinkercad
5. Fabricar el tenedor en la impresora

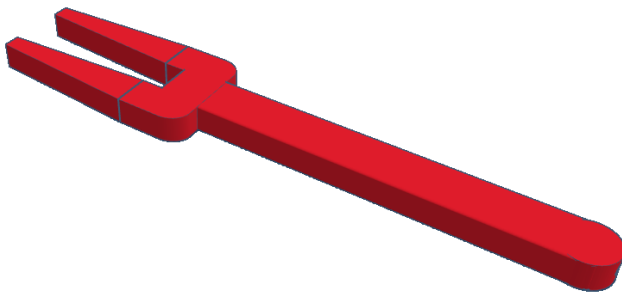
4. Diseño de la maqueta

4.1 Perspectiva del tenedor

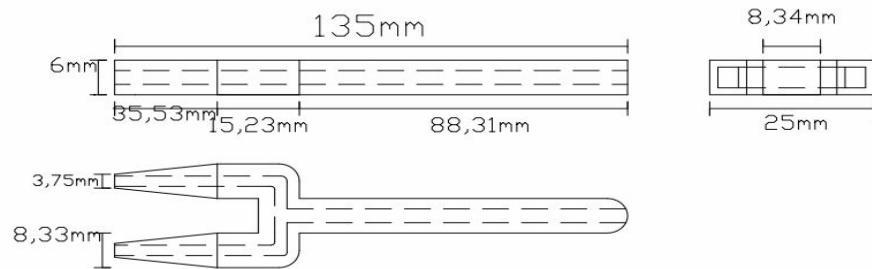
- Primer modelo



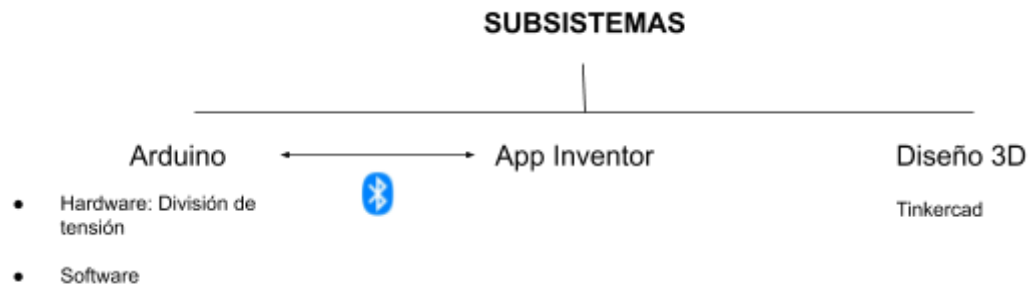
- Segundo modelo



4.2 Vistas del tenedor

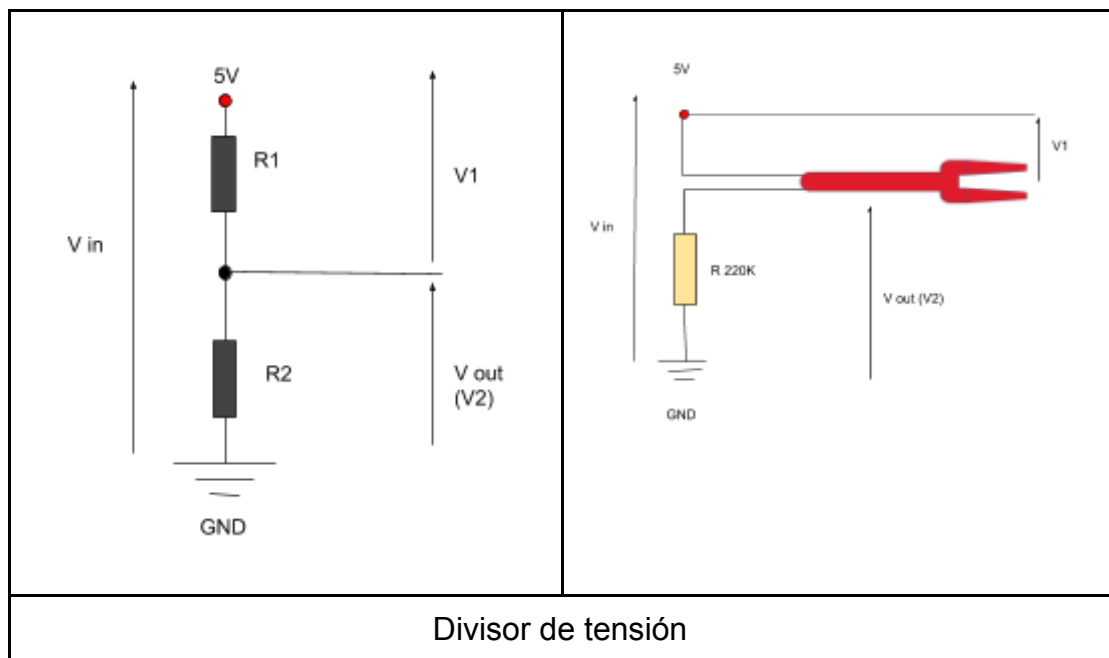


5. Subsistemas



5.1 Hardware

El hardware consiste en el divisor de tensión. Las dos resistencias son R1 (las frutas) y R2 (la resistencia que le añadimos al circuito).



Medimos las diferentes frutas con el polímetro y calculamos una media, eligiendo así la resistencia que se ajuste más a esta media para poder medir la mayor cantidad de alimentos posibles.

Frutas	Valores (K Ω)
Manzana	175
Mandarina	185
Plátano	200
Sandía	210
Pepinillo	230

La media de estos valores es de 200k Ω , las resistencias más acertadas son las de 220k Ω y la de 180k Ω , pero elegimos la de 220k Ω .

5.2 Código Arduino

Copiamos el código de la página mencionada en Búsqueda de información y realizamos algunos cambios, eliminando códigos que no eran útiles para el proyecto. Añadimos los valores que habíamos medido con las frutas que utilizamos.



5.3 App Inventor

Para el desarrollo de la app copiamos códigos de las presentaciones de clase ([tecnopujol](#)). Los bloques de códigos para la conexión Bluetooth son un ejemplo.

Para la comunicación entre Arduino y la app nos sirvió de base un proyecto ya desarrollado por José Pujol, nuestro profesor.

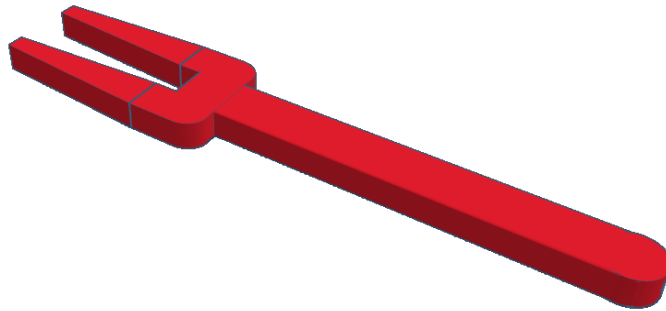
Buscamos en internet los códigos que nos hicieron falta para reproducir un audio y añadir una imagen y terminamos de desarrollarlos.



Así es como aparece en la pantalla del dispositivo Android

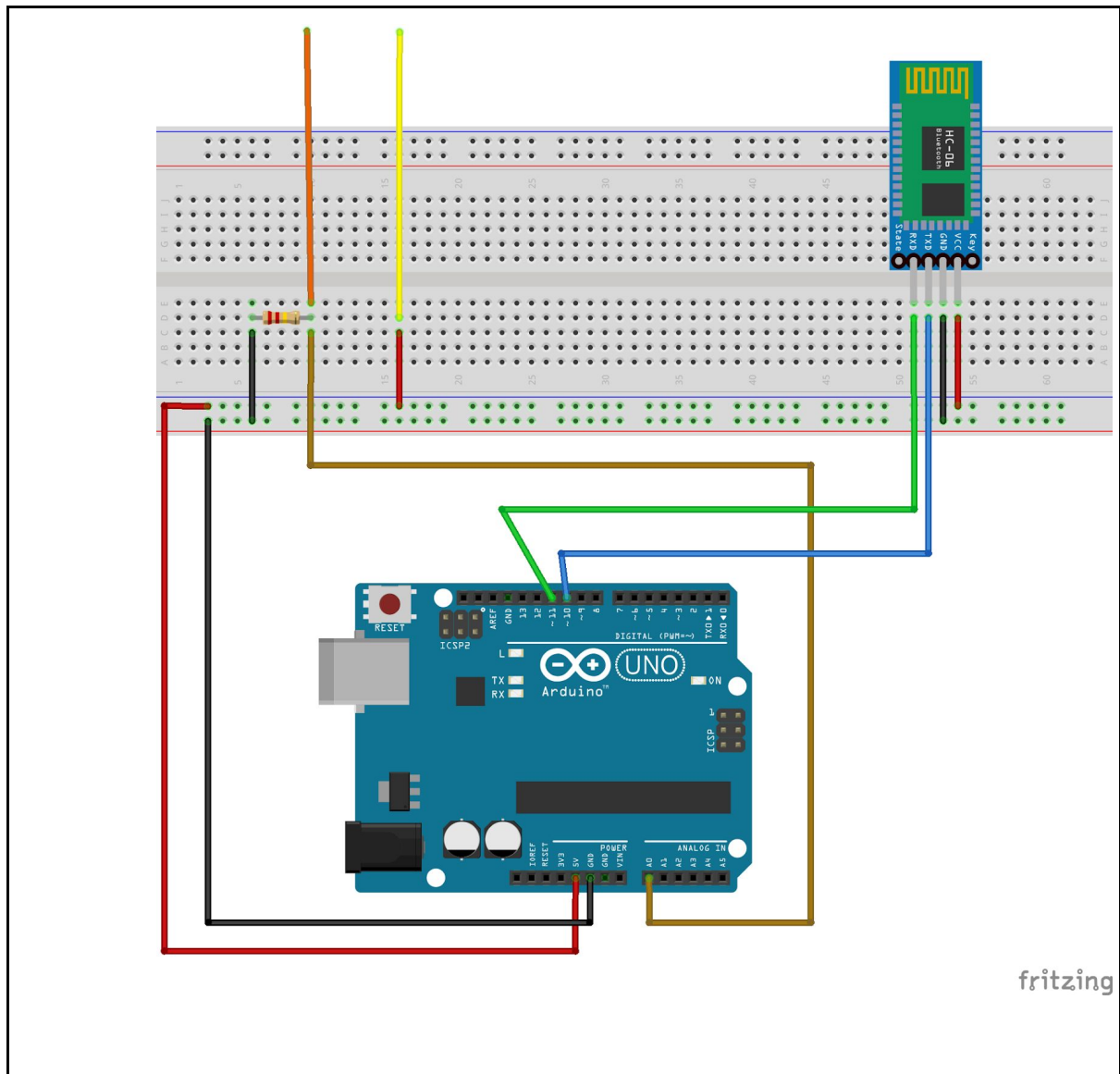
5.4 Diseño 3D

Del diseño del tenedor en Tinkercad se encargó nuestro compañero Fabián Cuevas. Realizó modelos con pequeñas modificaciones hasta conseguir el tenedor que mejor se adaptaba a la forma de los cables. Después de algunos cuantos modelos y modificaciones conseguimos un modelo final que imprimimos en la impresora 3D, parando la impresión a la mitad y añadiéndole los cables, para después continuar con la impresión.

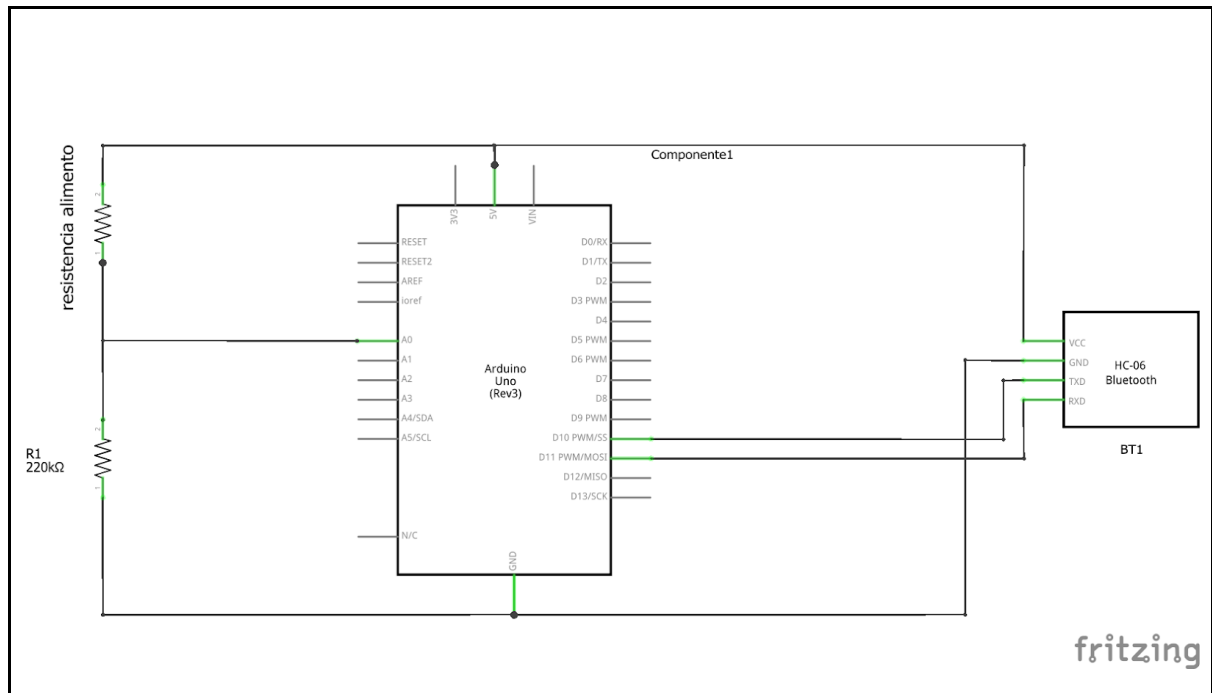


6. Hardware

6.1 Esquema protoboard



6.2 Esquema electrónico



7. Códigos

7.1 Arduino IDE

```
#include <SoftwareSerial.h>
SoftwareSerial I2CBT(10, 11);
// El TX del módulo BT va al pin 10 del Arduino
// El RX del módulo BT va al pin 11 del Arduino

int incomingByte;    // variable para leer los bytes de entrada
// variables temperatura
const int fruitSense = A0; // pin conexion
int fruitResistance = 0; // variable almacenar lectura

void setup() {
  // inicializamos la comunicacion serie
  Serial.begin(9600);
  // inicializamos la comunicacion serie BT
  I2CBT.begin(9600);
}

void loop() {

  // leemos el sensor
  // comprobamos si hay datos de entrada
  if (I2CBT.available() > 0) {
    // lectura del byte mas antiguo del buffer serial
    incomingByte = I2CBT.read();
    // si el byte es T envia dato de temperatura
    if (incomingByte == 'T') {

      fruitResistance = analogRead(fruitSense);
// clasificamos según la fruta que sea
      Serial.print("Resistance:");
      Serial.print(fruitResistance);

// clasificamos según la fruta que sea
      if (fruitResistance > 40 & fruitResistance < 400) { // si resistencia entre 400 y 550 se imprime y se envía el
dato
        Serial.println("Humano");
        I2CBT.write("H");
      }

      if (fruitResistance > 700 & fruitResistance < 790) { // si resistencia entre 400 y 550 se imprime y se envía el
dato
        Serial.println("Mandarina");
        I2CBT.write("N");
      }

      if (fruitResistance > 790 & fruitResistance < 950) { // si resistencia entre 600 y 740 se imprime y se envía el
dato
```

```

    Serial.println("Plátano");
    I2CBT.write("A");

}

if (fruitResistance > 950 & fruitResistance < 1050) { // si resistencia entre 950 y 1000 se imprime y se envía
el dato

    Serial.println("Pepinillo");
    I2CBT.write("P");
}

// si el valor no está en esos rangos se informa
if (fruitResistance > 0 & fruitResistance < 40) {
    Serial.println ("Sin comida");
    I2CBT.print("S");
}
}
}
}

```

7.2 App Inventor

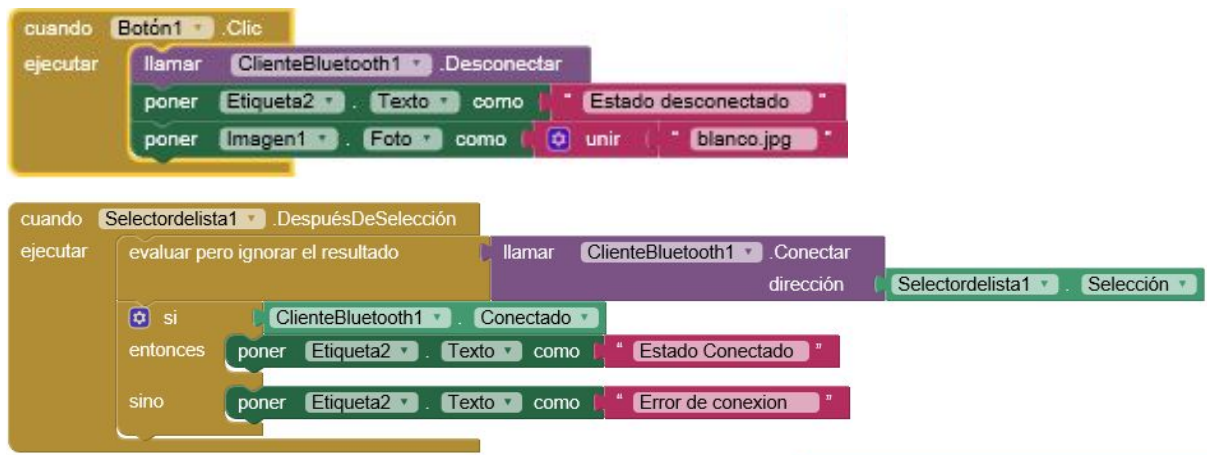
Variable

inicializar global datos_entradaBT como 

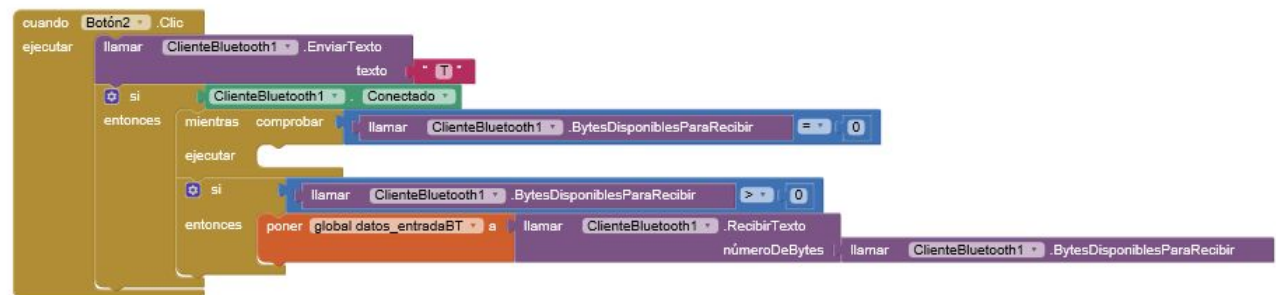
Inicialización y conexión Bluetooth

cuando Screen1 . Inicializar
ejecutar
poner Imagen1 . Foto como  unir blanco.jpg
poner Etiqueta2 . Texto como Estado desconectado

cuando Selectordelista1 . AntesDeSelección
ejecutar
poner Selectordelista1 . Elementos como ClienteBluetooth1 . DireccionesYNombres



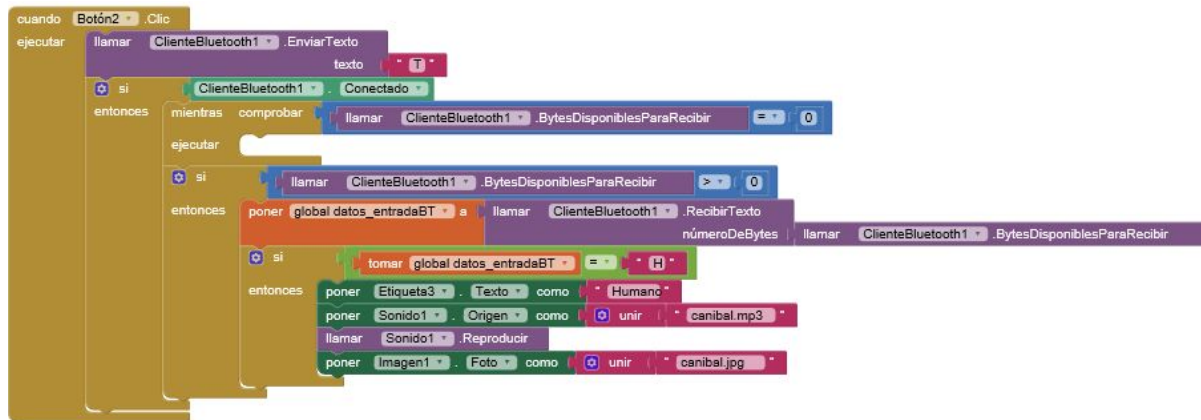
Información enviada a Arduino



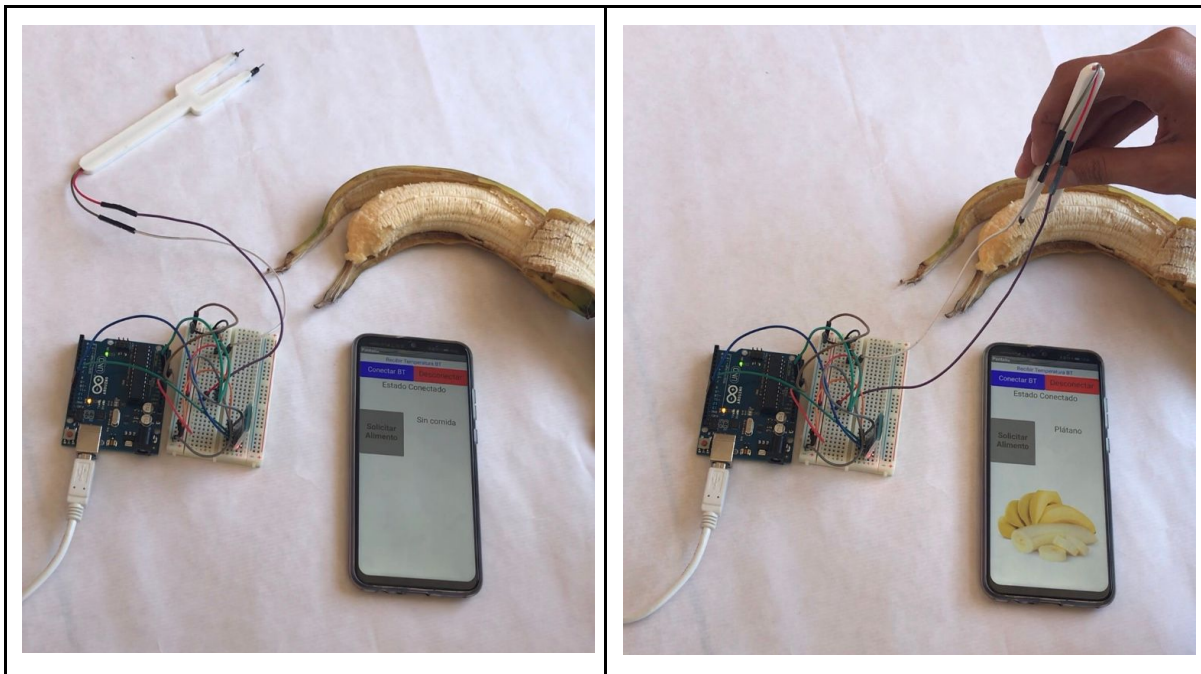
Información recibida de Arduino



Conexión con Arduino (códigos unidos)



8. Fotos y Vídeos



[Demostración del proyecto](#)

9. Análisis del proyecto

9.1. Análisis funcionamiento del proyecto

El proyecto finalmente funciona sin problemas:

- La comunicación entre Arduino y App Inventor no tiene fallos, al igual que los códigos de ambas aplicaciones.

Mejoras

En general no encontramos muchas mejoras posibles:

- Se podría realizar un tenedor que tuviera una mayor precisión en cuanto a medir alimentos, y así poder añadir más frutas sin que se solaparan.

Problemas

- No todas las frutas del mismo tipo tienen la misma resistencia, debido a su nivel de maduración u otros factores; por ejemplo, no todas las manzanas tienen la misma resistencia.
- Al conectar algunos móviles al bluetooth, a veces se desconecta sin razón (que nosotros sepamos).

Pero a pesar de estos fallos, creemos que nuestro proyecto está bien resuelto, ya que en la gran mayoría de casos funciona sin inconvenientes.

Anexo

Evaluación funcionamiento del equipo

No han habido inconvenientes en cuanto al funcionamiento del equipo. Los dos nos hemos preocupado por el proyecto y nos hemos repartido más o menos bien las tareas. El resultado final nos ha parecido adecuado y en nuestra opinión ha alcanzado nuestras expectativas. Hemos tenido un buen ambiente de trabajo todo el tiempo y así nos ha sido más ameno el desarrollo del Food Detector.