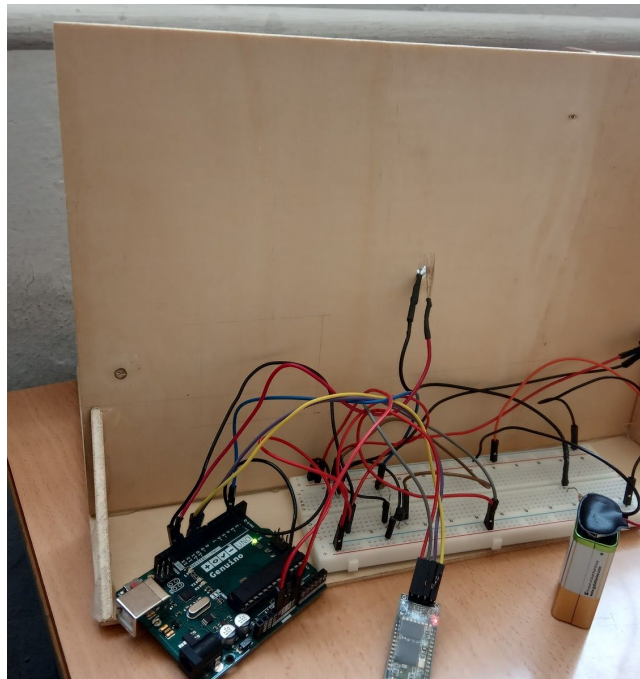




# ASISTENTE VIRTUAL CANCO



Francisco Fernández-Llebrez Acedo  
Fernando Gallardo Hidalgo  
Pablo Pérez Vida

1ºA BACHILLERATO

IES VICENTE ALEIXANDRE  
9/6/2019

# ÍNDICE

|                           |   |
|---------------------------|---|
| 1.- Finalidad del sistema | 1 |
| 2.- Hardware              | 2 |
| 3.- Software              | 6 |
| 4.- Funcionamiento        | 7 |
| 5.- Evaluación            | 7 |
| <b>BIBLIOGRAFÍA</b>       | 9 |

# ASISTENTE VIRTUAL CANCO

## **1.- Finalidad del sistema**

Con este proyecto pretendemos diseñar un asistente virtual que nos permita realizar algunas funciones domésticas. El proyecto final será una maqueta en la que controlemos un LED y un ventilador con nuestra voz, usando App Inventor, Machine Learning, Bluetooth y Arduino. El proceso será:

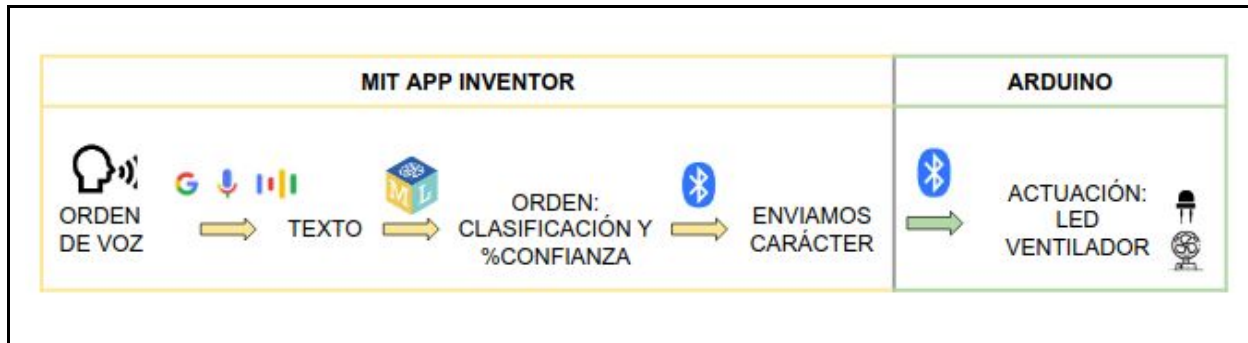
1. Orden de voz con el reconocimiento de voz de App Inventor
2. Clasificar e identificar con Machine Learning la orden
3. Enviar un carácter a Arduino via Bluetooth
4. Arduino envía la orden al actuador

Este proyecto nos sirve para iniciarnos en el mundo del Machine Learning, es decir, el aprendizaje automático de las máquinas, seguir empleando App Inventor y para seguir programando con Arduino. Otro factor importante es el hecho de haber enlazado las diferentes tecnologías que hemos empleado . Sería un proyecto muy útil si consiguiéramos pasarlo a una habitación real.

Debemos obtener un producto que sea fácil de utilizar y que mejore la comodidad de las personas ya que sería empleado en casas o pisos con personas de diferentes edades que no tienen por qué saber sobre informática para usarlo.

## 2.- Hardware

### a) Esquema de entradas y salidas.



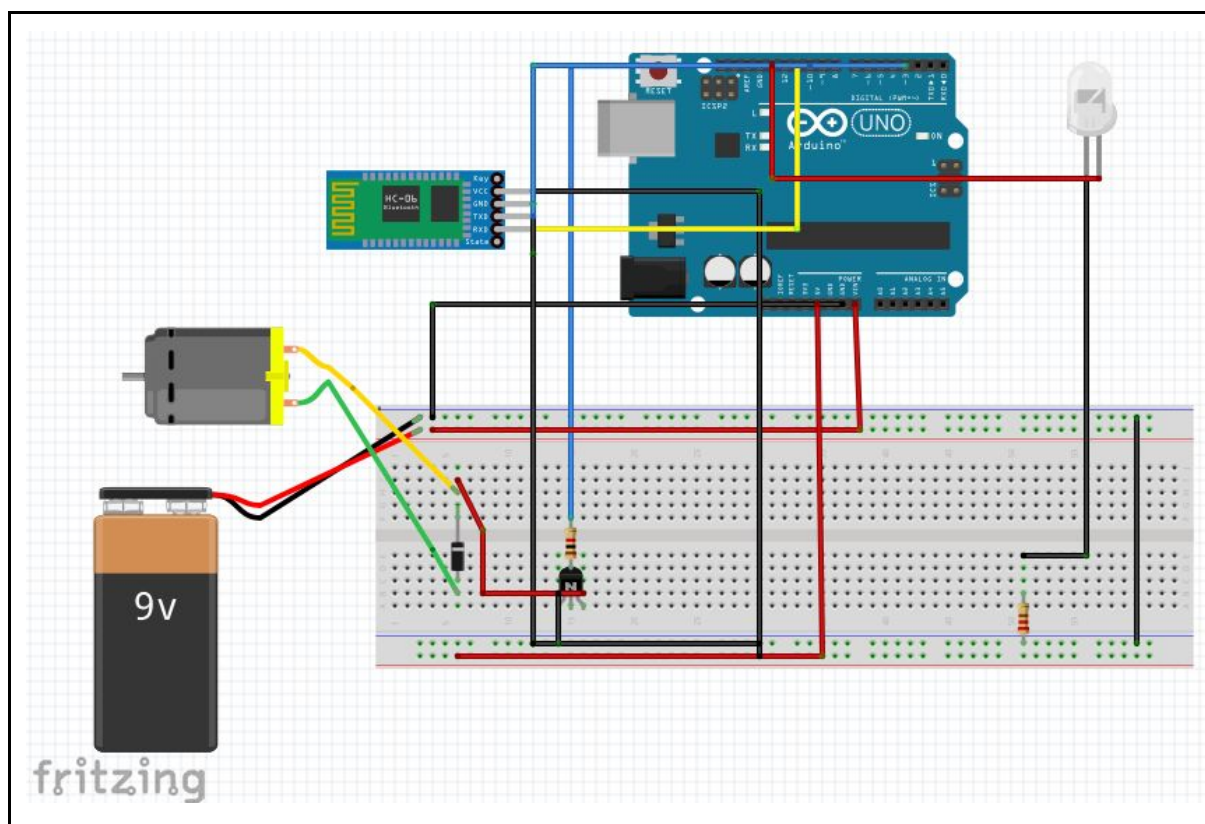
### b) Lista de materiales ( Hardware )

| <i>Número</i> | <i>Descripción</i>  | <i>Cantidad</i> | <i>Precio</i> |
|---------------|---------------------|-----------------|---------------|
| 1             | Arduino Uno         | 1               | 20€           |
| 2             | Protoboard          | 1               | 5.90€         |
| 3             | LED                 | 1               | 0.20€         |
| 4             | Ventilador          | 1               | 5€            |
| 5             | Diodo               | 1               | 0.10€         |
| 6             | Módulo bluetooth    | 1               | 8.50€         |
| 7             | Transistor bc547    | 1               | 1.50€         |
| 8             | Pila                | 1               | 2.00€         |
| 9             | Resistencia 220/1KΩ | 1/1             | 0.03/0.03€    |
| TOTAL         |                     |                 | 43.26€        |

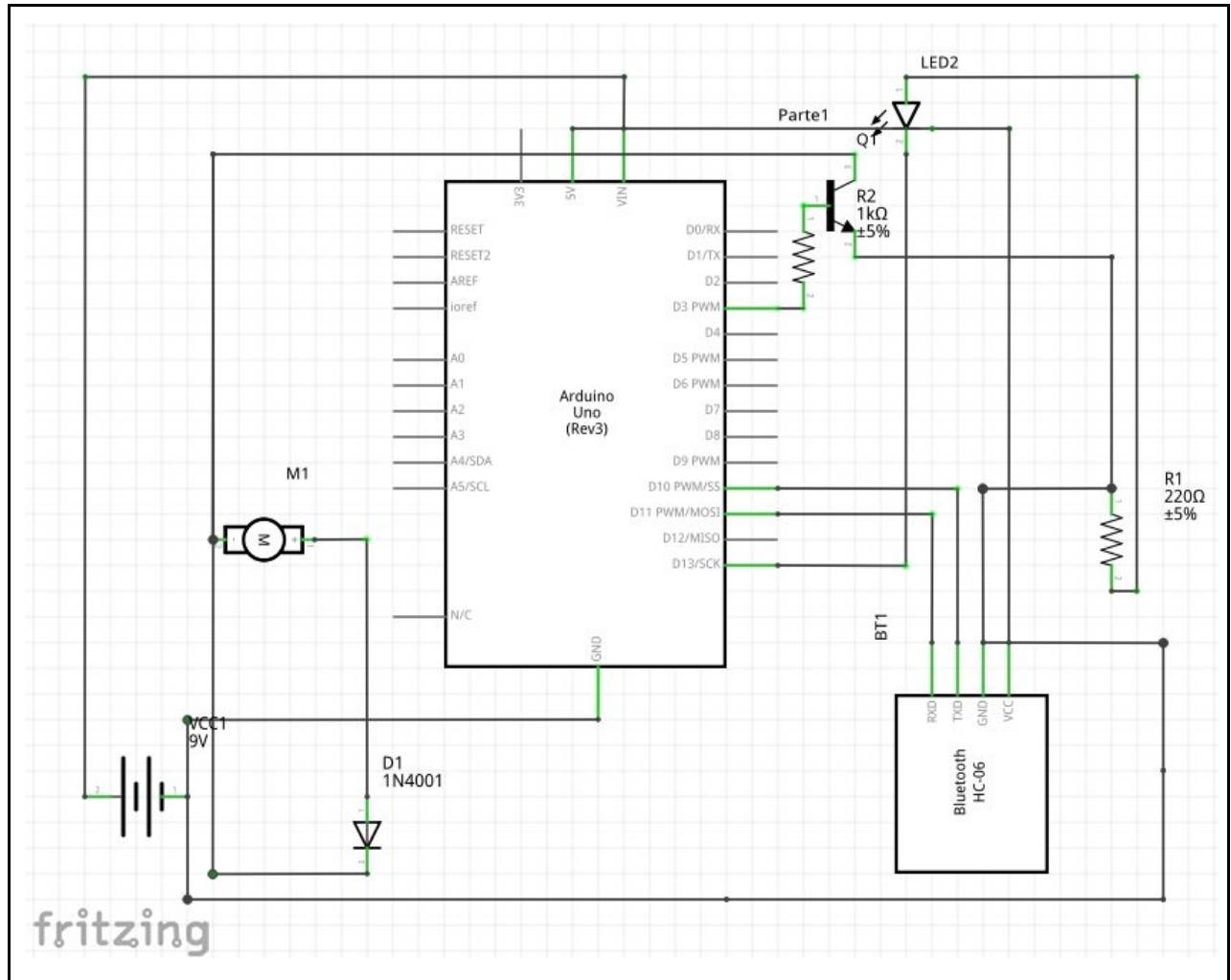
## Lista de materiales ( Maqueta )

| <i>Número</i> | <i>Descripción</i>        | <i>Cantidad</i> | <i>Precio</i> |
|---------------|---------------------------|-----------------|---------------|
| 1             | Madera<br>20x30/12,5x20cm | 2/1             | 2,69€         |
| 2             | Miniaturas                | 4               | 20€           |
| 3             | Lámpara 3D                | 1               | 4,40€         |
| TOTAL         |                           |                 | 27,09€        |

## c) Esquema de la Protoboard



#### d) Esquema electrónico



## 3.- Software

### Código Arduino

```
#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

const int ledPin = 13; // el LED se conecta al pin digital 13
const int motPin = 3; // la base del transistor se conecta al pin digital 3
int incomingByte; // variable que se usa para los datos que introducimos en el serial buffer

void setup() {
  I2CBT.begin(9600); // inicializa la comunicación con el módulo Bluetooth
  Serial.begin(9600); // inicializa la comunicación con el puerto serie
  pinMode(motPin, OUTPUT); // inicializa el motor como una salida
  pinMode(ledPin, OUTPUT); // inicializa el LED como una salida
}

void loop() {

  // comprobamos si hay datos de entrada
  if (I2CBT.available() > 0) {
    // lectura del byte más antiguo del módulo Bluetooth
    incomingByte = I2CBT.read();

    // si recibe una H mayúscula, enciende el LED
    if (incomingByte == 'H') {
      digitalWrite(ledPin, HIGH);
    }
    // si recibe una L mayúscula, apaga el LED
    if (incomingByte == 'L') {
      digitalWrite(ledPin, LOW);
    }
    // si recibe una E mayúscula, enciende el motor
    if (incomingByte == 'E') {
      digitalWrite(motPin, HIGH);
    }
    // si recibe una A mayúscula, apaga el motor
    if (incomingByte == 'A') {
      digitalWrite(motPin, LOW);
    }
  }
}
```

## Código App Inventor

```
cuando reconocedor1.Click
ejecutar
  llamar ReconocimientoDeVoz1.ObtenerTexto

cuando Button1.Click
ejecutar
  llamar ML4K1.ClassifyText
  data ReconocimientoDeVoz1.Resultado
```

```
cuando ListPicker1.AntesDeSelección
ejecutar
  poner ListPicker1.Elementos como BluetoothClient1.DireccionesYNombres

cuando ListPicker1.DespuésDeSelección
ejecutar
  evaluar pero ignorar el resultado
  llamar BluetoothClient1.Conectar
  dirección ListPicker1.Selección

  si BluetoothClient1.Conectado
  entonces
    poner Label3.Texto como "Bluetooth conectado"
  sino
    poner Label3.Texto como "Error de conexión Bluetooth"
```

```
cuando ML4K1.GotClassification
data classification confidence
ejecutar
  poner Label1.Texto como tomar classification
  poner Label2.Texto como tomar confidence

  si tomar classification = encender_luz
  entonces
    si tomar confidence > 70
    entonces
      llamar BluetoothClient1.EnviaTexto
      texto "H"
    si no, si
      tomar classification = apagar_luz
      entonces
        si tomar confidence > 70
        entonces
          llamar BluetoothClient1.EnviaTexto
          texto "L"

  si tomar classification = encender_ventilador
  entonces
    si tomar confidence > 70
    entonces
      llamar BluetoothClient1.EnviaTexto
      texto "E"
    si no, si
      tomar classification = apagar_ventilador
      entonces
        si tomar confidence > 70
        entonces
          llamar BluetoothClient1.EnviaTexto
          texto "A"
```



## 4.- Funcionamiento

### Vídeo

## 5.- Evaluación

### a) Qué funciona bien y qué se puede mejorar.

Este proyecto funciona correctamente de acuerdo con el problema planteado aunque hemos tenido algunos fallos a lo largo del proyecto que han frenado bastante el ritmo de trabajo.

### b) Problemas que hemos tenido y soluciones.

Para empezar, comenzamos el proyecto siguiendo un tutorial muy complejo de cómo encender un LED mediante una orden directa y al reproducirlo no nos funcionó. No conseguimos encender el LED de con una orden directa hasta que no lo hicimos con el ejemplo Physical Pixel de Arduino con el que habíamos trabajado anteriormente.

Nuestro siguiente problema fue con Machine Learning. A la hora de introducir Machine Learning en App Inventor nos sabíamos cómo hacerlo pero buscando información descubrimos que teníamos que introducir una “llave” para enlazar nuestro modelo de Machine Learning con App Inventor. Esta “llave” se la dieron a nuestro profesor cuando se creó la cuenta en Machine Learning for Kids.

Con App Inventor y Machine Learning enlazados tuvimos problemas ya que apenas disponíamos de información sobre cómo usar Machine Learning en App Inventor. Escribimos algunos correos a Machine Learning for Kids y pedimos ayuda a nuestro profesor pero seguimos sin conseguirlo por lo que buscamos una alternativa. Lo que hicimos fue poner listas en App Inventor para poder encender el LED con varias órdenes distintas.

Nos apartamos de Machine Learning y nos pusimos con el ventilador. Tuvimos un problema en el hardware al montar el circuito de potencia y alimentamos un módulo Bluetooth con 9 voltios por lo que lo fundimos. Tras esto, montamos de nuevo el circuito de potencia cuidadosamente y programamos el ventilador para poder encenderlo con la voz.

Después de esto, volvimos a retomar el Machine Learning. A partir de un correo de Machine Learning for Kids, nuestro profesor nos enseñó cómo conseguir la confidencia y la clasificación en App Inventor. Lo que hicimos fue incluir un botón en la aplicación que pulsamos, tras el reconocimiento de voz, para identificar y clasificar la orden de voz. Con esto conseguimos solucionar un problema que hemos tenido presente casi desde que empezamos el proyecto.

Tras enlazar Machine Learning con App Inventor, nos dimos cuenta de que teníamos que entrenar el modelo de Machine Learning siempre antes de usar la aplicación. Buscando en la web de Machine Learning for Kids vimos que los modelos se borraban automáticamente en un día ya que esta web está pensada para trabajos de un día y no para proyectos. También vimos que el tiempo límite de borrado podía ser cambiado por el profesor, aunque lo máximo que se puede es 1 semana y 3 días.

Por último, un problema que no hemos conseguido resolver es por qué cuando descargamos la aplicación en nuestro móvil no funciona Machine Learning y tenemos siempre que abrir la aplicación desde la web de App Inventor.

### **c) Propuestas de mejora y ampliación del proyecto**

Como ampliación podríamos haber añadido una persiana, la cual no hemos añadido por falta de tiempo. Además de la persiana, no podríamos haber añadido muchas más cosas ya que nuestro proyecto es a pequeña escala. Podríamos también combinar nuestra App y nuestro código con otros proyectos que se han realizado en nuestra clase.

### **d) Conclusiones**

Para este proyecto deberíamos haber seguido algunas estrategias con las que hubiésemos sido más eficientes y productivos. Por ejemplo:

- Apuntar en un documento todos los problemas que nos surgen y documentarlos para solucionarlos o que nos ayuden a solucionarlos de una forma más eficiente.
- Estructurar y planificar las tareas a realizar en cada clase para conseguir una mayor efectividad en el trabajo.
- Guardar y copiar en un documento todos los cambios y mejoras realizados en el código para que si el nuevo paso que demos en el código falla podamos volver al paso anterior.
- Simplificar. Uno de nuestros problemas principales en cuanto a resolución ha sido intentar dar los nuevos pasos sobre el código completo y no haberlo hecho por separado.

# BIBLIOGRAFÍA

- [Fritzing](#)
- [Cambio html a código normal](#)
- [Arduino](#)
- [App Inventor](#)
- [Machine Learning for Kids](#)
- [GitHub](#)