

**2020**

**gekk**   
Tu ayuda tecnológica

 [gekk85.cms.webnode.es/](http://gekk85.cms.webnode.es/)

 [proyectogekko@gmail.com](mailto:proyectogekko@gmail.com)

 +54 9 11 2373-6033

 @proyectogekko

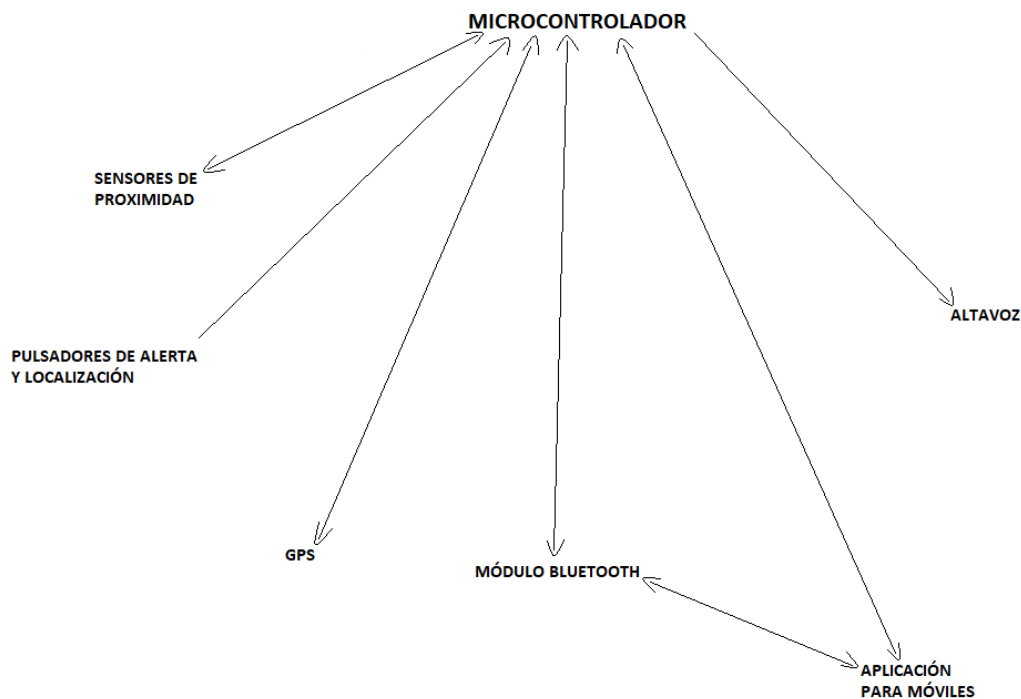
**Carpeta de campo**

### Semana del 19 al 24 de abril:

- Se realizó una búsqueda de proyectos disponibles.  
[https://docs.google.com/document/d/1LZ2ffHP7zFI0\\_WXvjuVVuS2lhBsqSJgq7y5wuCDEFA8/edit?usp=sharing](https://docs.google.com/document/d/1LZ2ffHP7zFI0_WXvjuVVuS2lhBsqSJgq7y5wuCDEFA8/edit?usp=sharing)
- Nos pusimos de acuerdo y decidimos desarrollar Gekko.

### Semana del 24 al 29 de mayo:

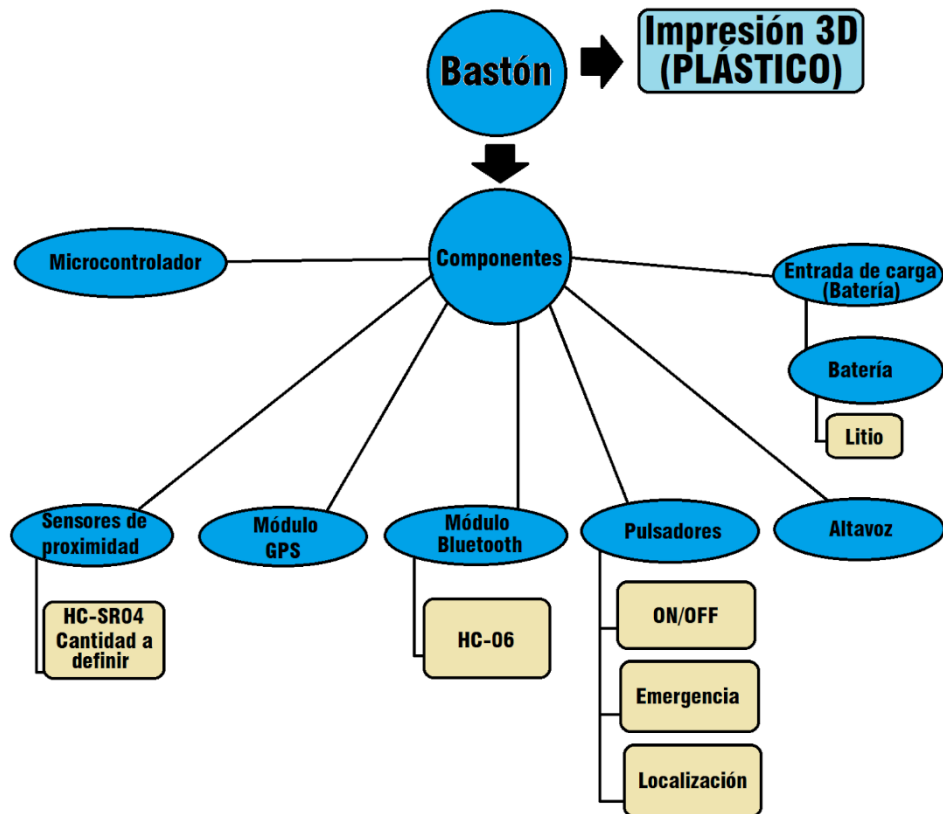
- Creamos el anteproyecto  
[https://docs.google.com/document/d/1Et\\_al9DliUvXYjag04nkPs-xts4ccxrmYAvca4g1PCI/edit?usp=sharing](https://docs.google.com/document/d/1Et_al9DliUvXYjag04nkPs-xts4ccxrmYAvca4g1PCI/edit?usp=sharing)
- Definimos en diagramas en bloques los programas de Gekko.



<https://docs.google.com/document/d/1Vq5vnE2Fy2eSLYGxGWqRqXkED4N5cAbeA7HkloNGVqA/edit?usp=sharing>

- Buscamos fundaciones en Buenos Aires que traten con personas no videntes.

- Definimos en diagramas en bloques el diseño de Gekko.



- Definimos los problemas y soluciones que nos íbamos a encontrar durante la realización del proyecto.  
<https://docs.google.com/document/d/1ISal8hRxs5Q8vdNxt3UfLgVGNzi2tCx8pVjr3wd40EA/edit?usp=sharing>

### Semana del 14 al 19 de junio:

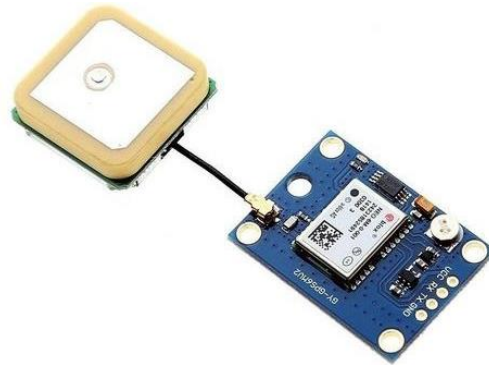
- Definimos los sensores que utilizamos para medir distancia.  
 HC-SR04: (Sensor ultrasónico)



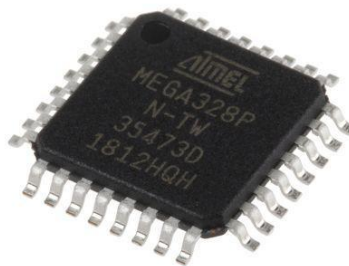
- Definimos el módulo para reproducir audio que utilizamos.  
DFPlayer Mini: (Módulo reproductor de audio MP3).



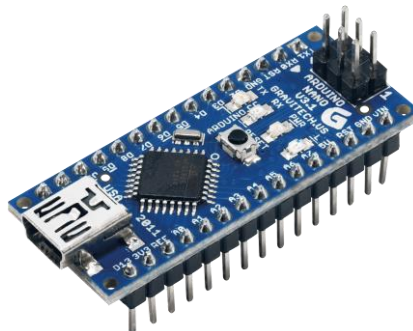
- Definimos el módulo GPS que utilizamos (este fue removido posteriormente).  
NEO-6M:(Módulo GPS).



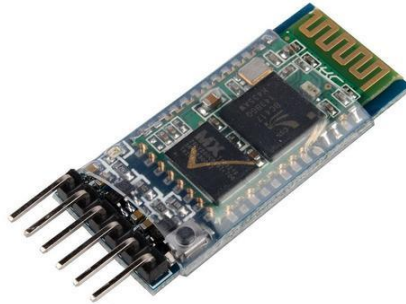
- Definimos el microcontrolador que utilizamos.  
Atmega328p:



- Definimos la placa de desarrollo que utilizamos.  
Arduino Nano:



- Definimos el módulo bluetooth que utilizamos.  
HC-05:

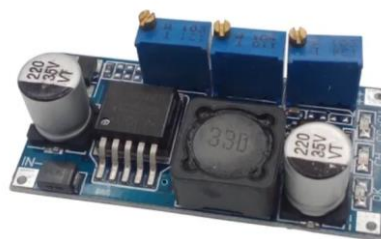


**Semana del 28 de junio al 3 de Julio:**

- Definimos las baterías que utilizamos.  
LNR18650-25R:



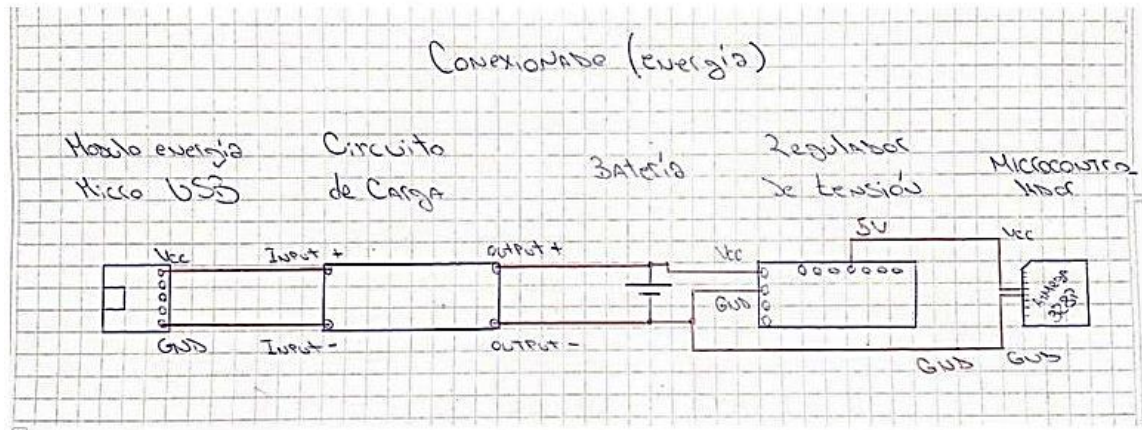
- Definimos el circuito de carga de la batería que utilizamos.  
LM2596-ADJ-CURRENT:



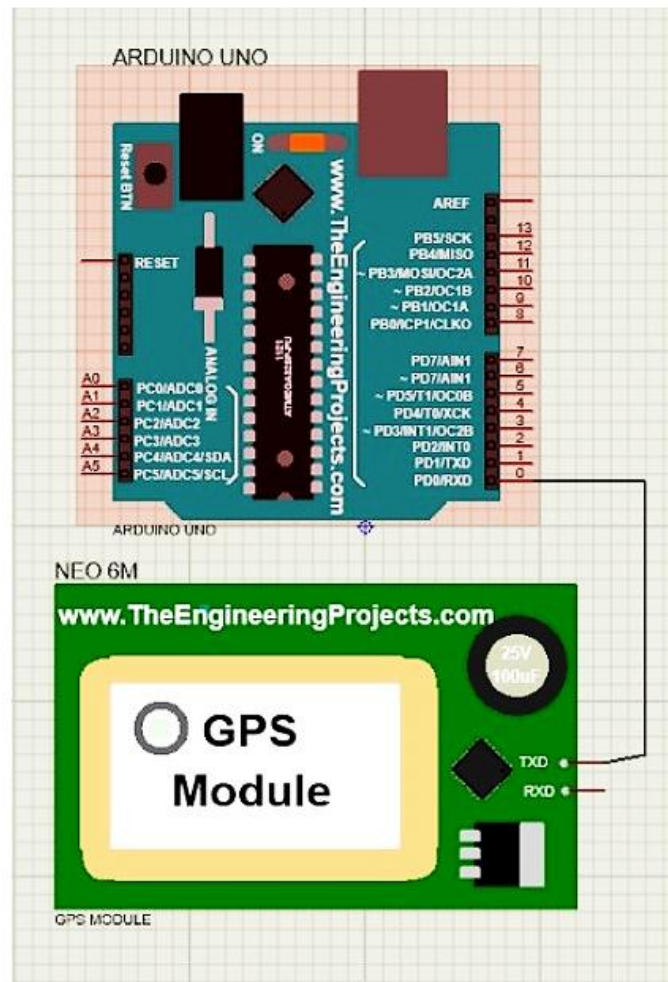
- Definimos el regulador de tensión que utilizamos.



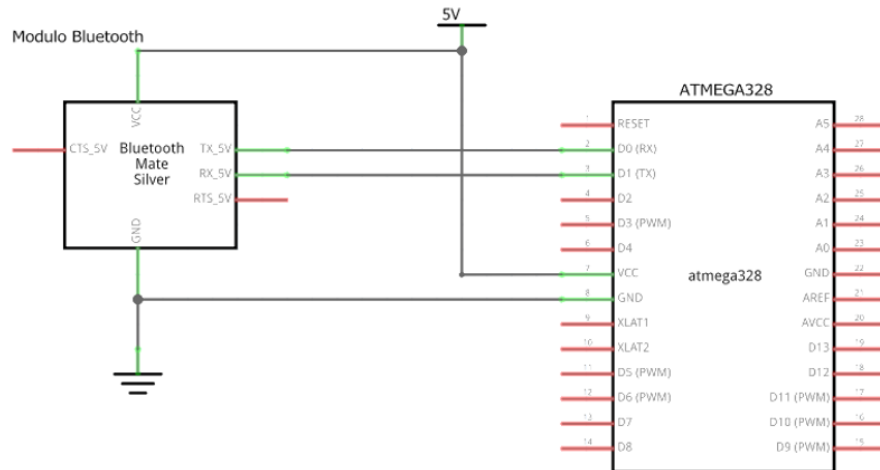
- Se realizó el conexionado del regulador de tensión entre la batería y el microcontrolador.



- Se realizó el conexionado del GPS con el microcontrolador.



- Se realizó el conexionado del módulo bluetooth con el microcontrolador.



Conexionado de modulo Bluetooth HM 10

Proyecto GEKKO

\*Filename Esquemático HM 10

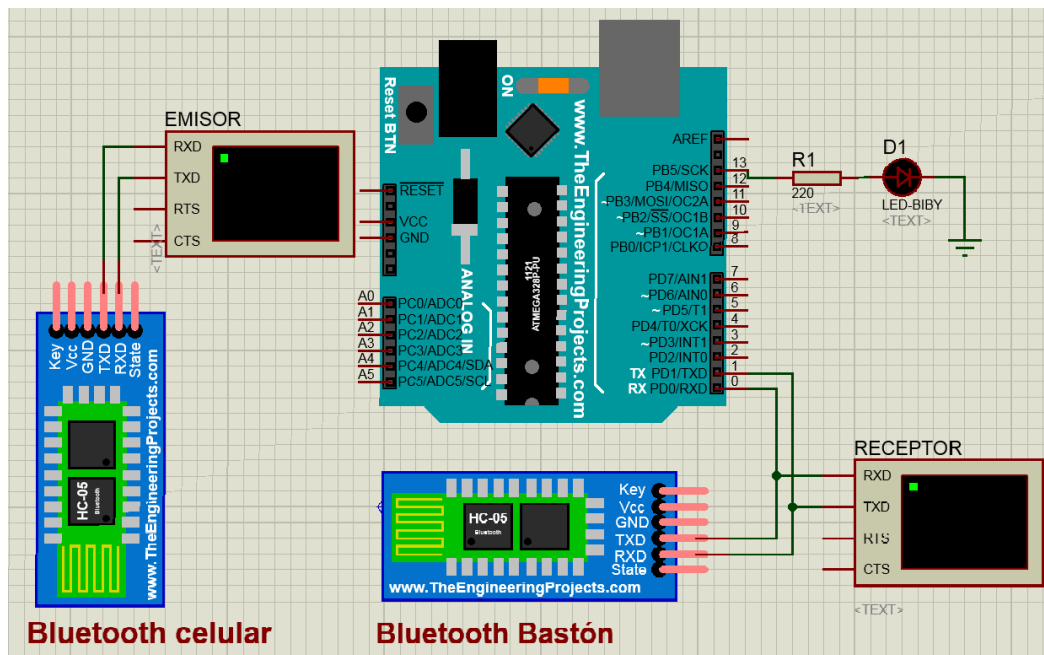
Rev

Fecha 03 jul 2020 15:27:26

Hoja 1/1

### Semana del 5 al 10 de Julio:

- Simulamos el funcionamiento del microcontrolador con Bluetooth (pines definidos) en Proteus.



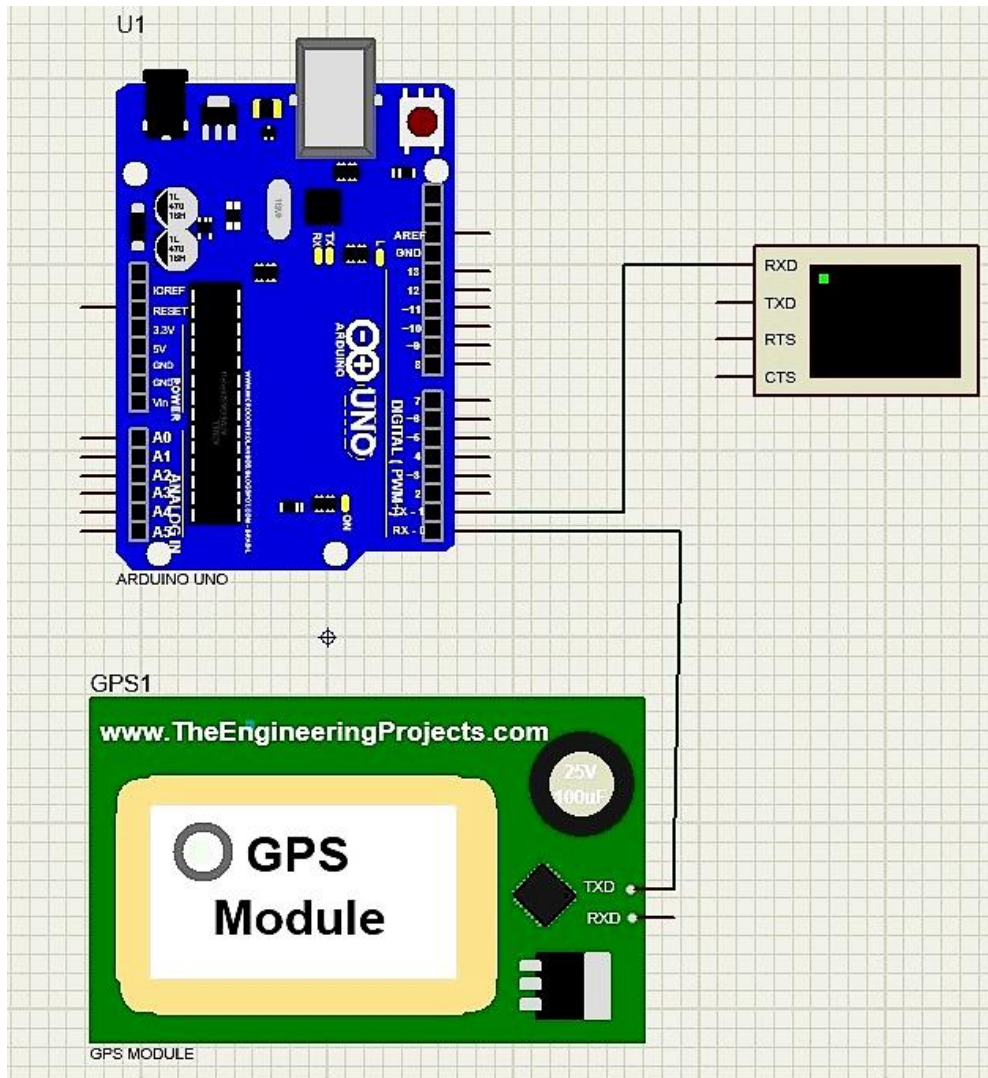






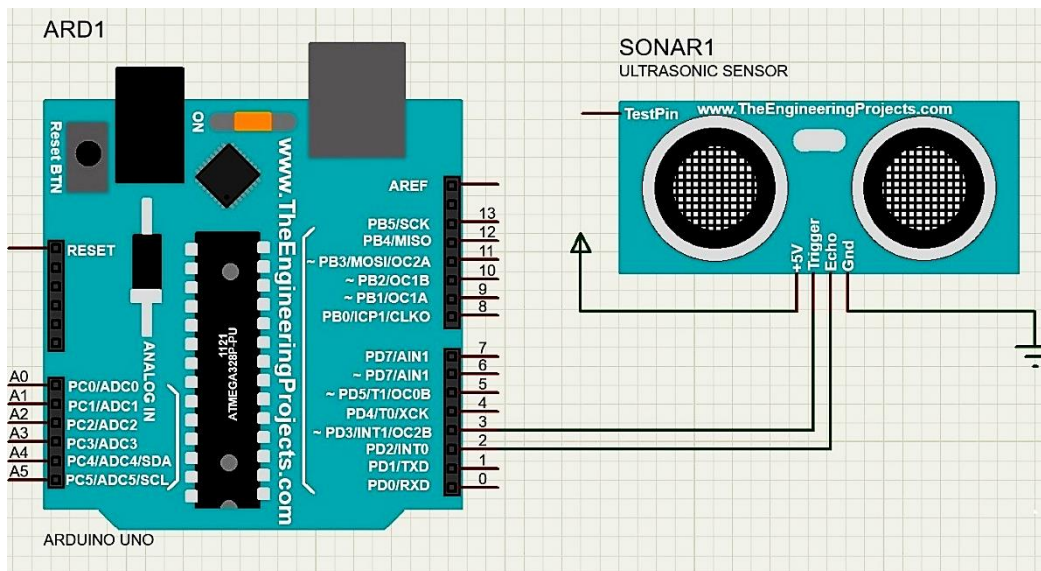
**Semana del 12 al 17 de Julio:**

- Simulamos el funcionamiento del microcontrolador con el GPS (pines definidos) en Proteus.

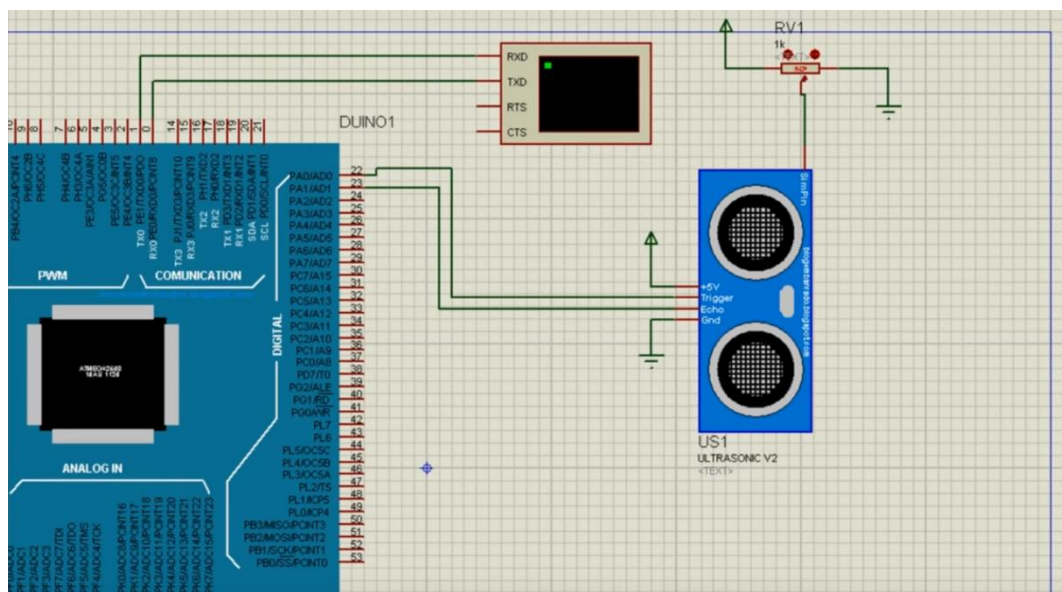


## Semana del 19 al 24 de Julio:

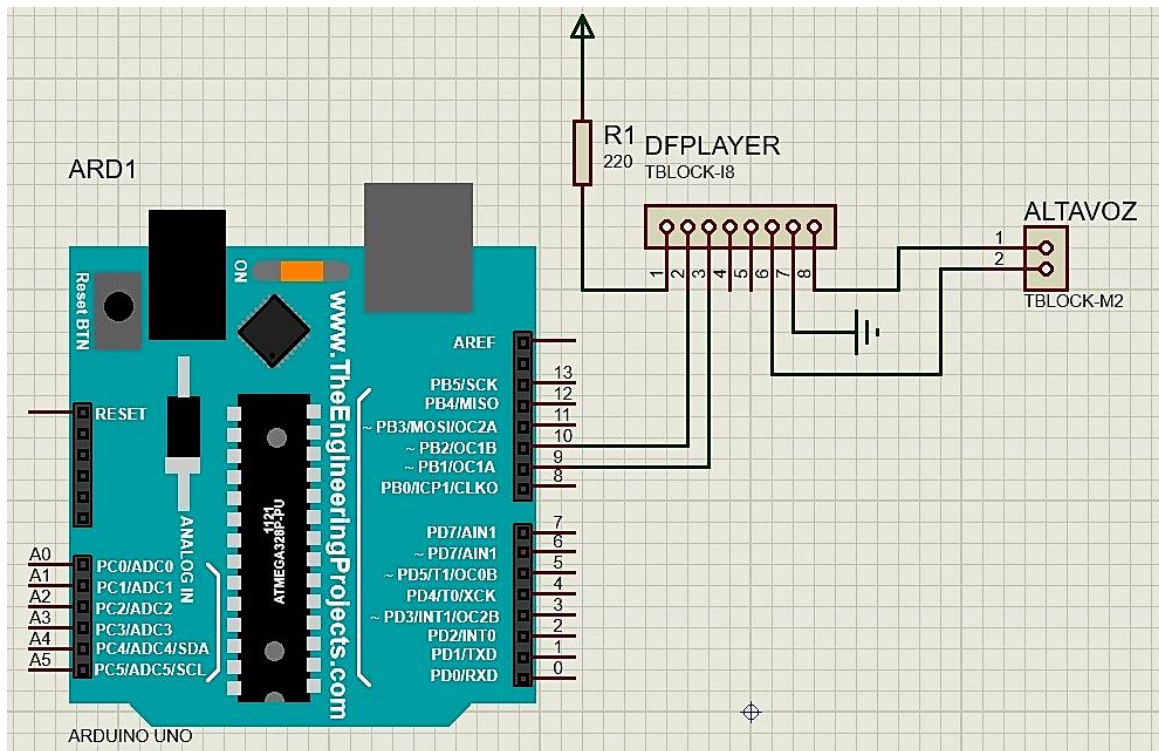
- Investigamos sobre modelados 3D de bastones.
- Conectamos los sensores HC-SR04 con el microcontrolador (Esquemático).



- Simulamos el microcontrolador con los sensores HC-SR04 (pines definidos) en Proteus.

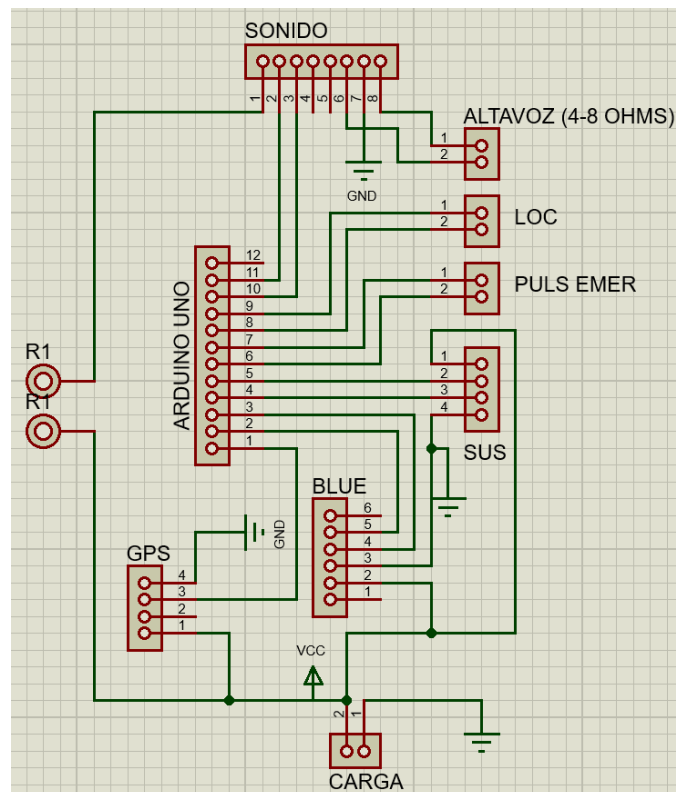


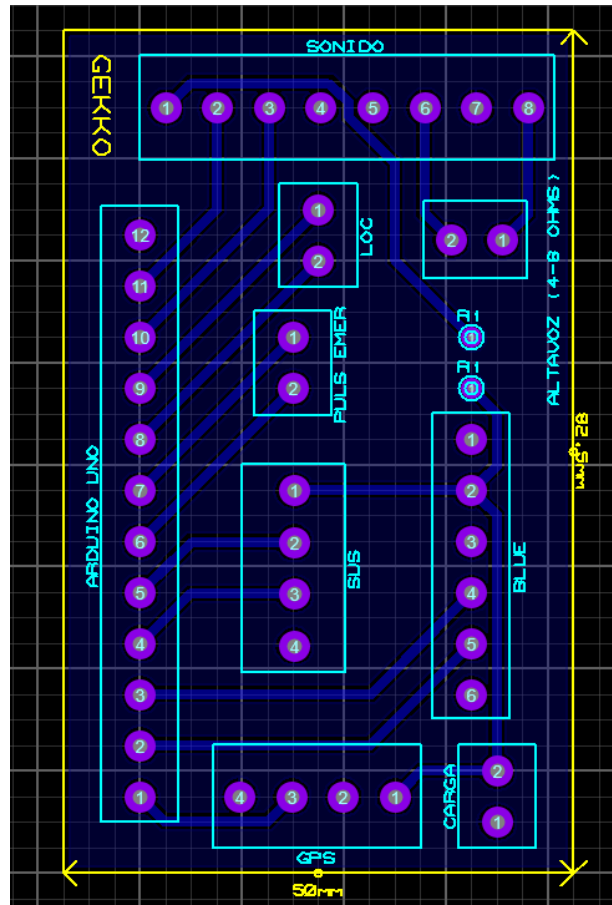
- Conectamos el módulo de audio con el microcontrolador (Esquemático).



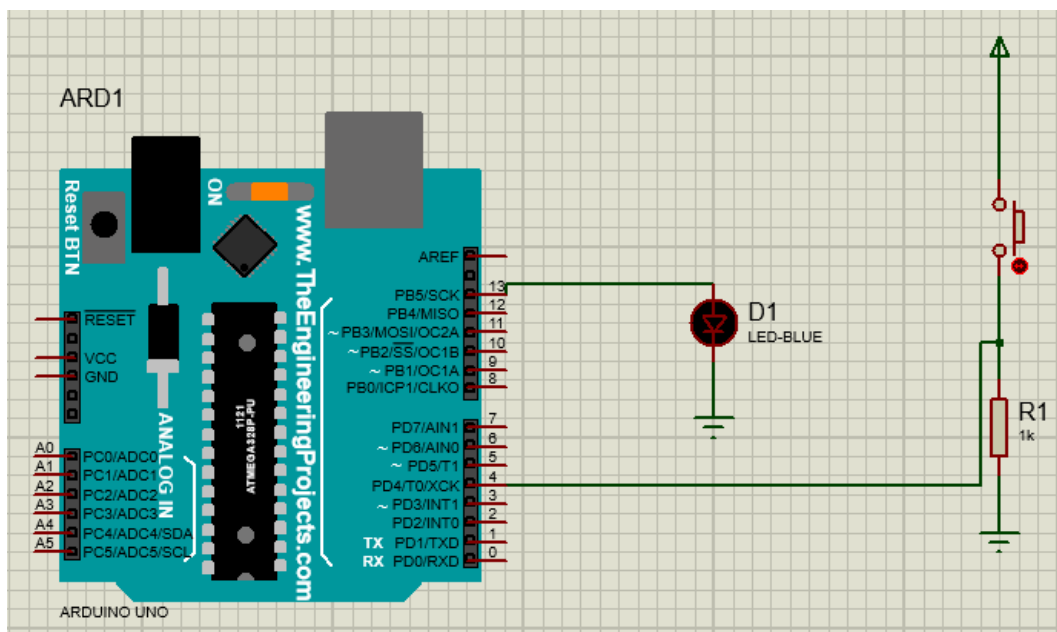
### Semana del 26 al 31 de julio:

- Diseñamos la primera versión de la placa (esquemático y PCB).





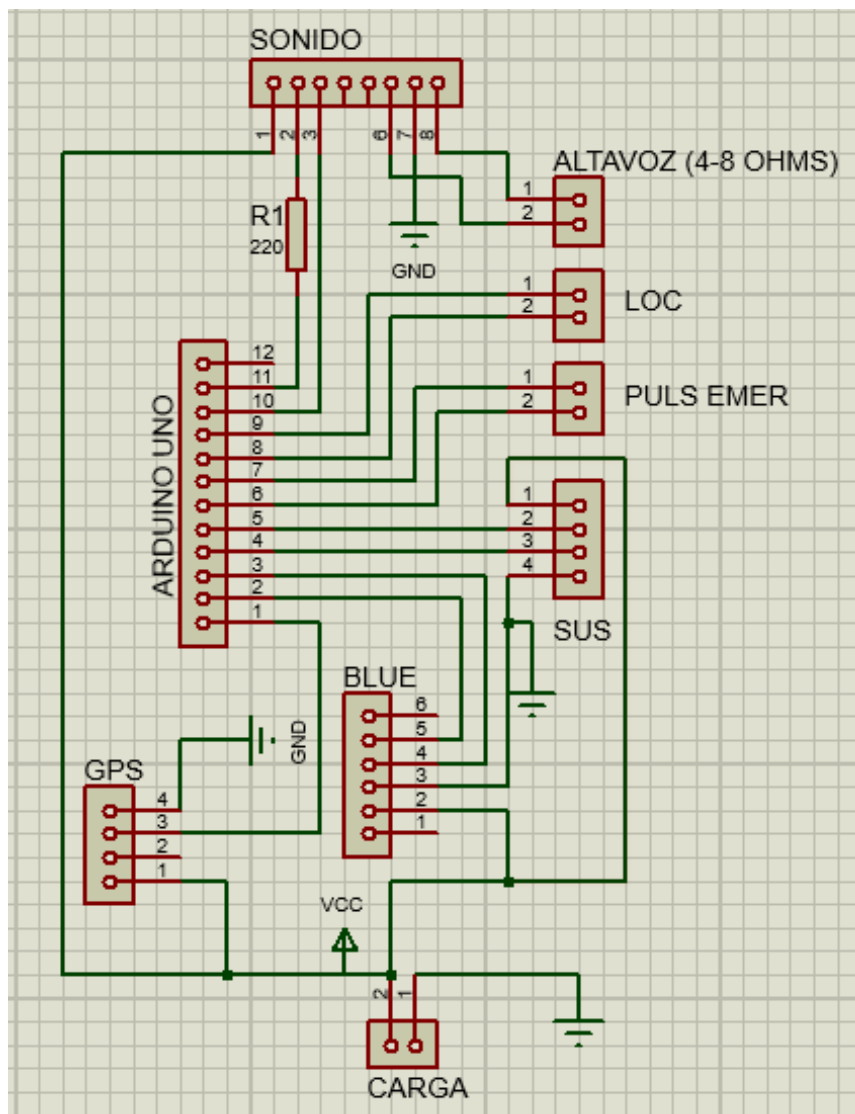
- Se realizó el conexionado de los pulsadores (localización y emergencia) al microcontrolador.
- Simulamos el microcontrolador con los pulsadores en Proteus.

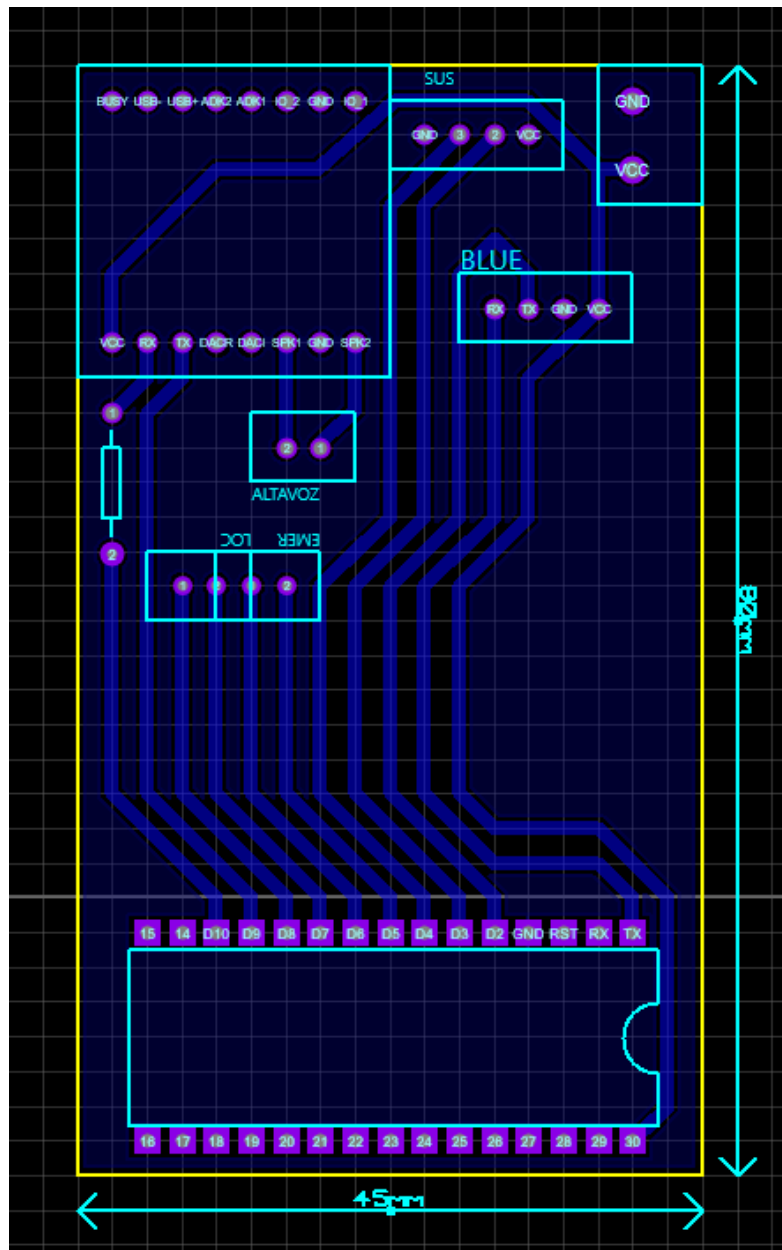


- Se diseñó la primera página Web.

**Semana del 9 al 14 de agosto:**

- Diseñamos la segunda versión de la placa (esquemático y PCB).





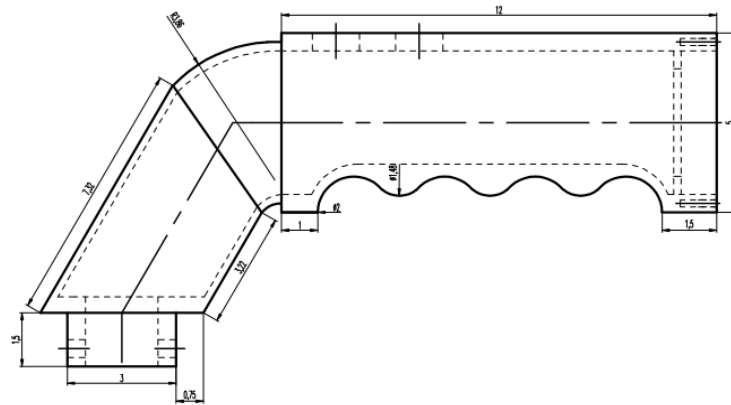
- Investigamos sobre páginas Web.  
<https://docs.google.com/document/d/1rvr2cZbw-wyl-aD5sBONv5N93q1NxLY-9eWi91F-WUk/edit?usp=drivesdk>
- Investigamos sobre “SERIAL por Software” en Arduino.  
[https://docs.google.com/document/d/1XmN\\_6ywRpWIKmG37RcPmGJx6PZc1d-wT-U7YGCuxoSc/edit?usp=drivesdk](https://docs.google.com/document/d/1XmN_6ywRpWIKmG37RcPmGJx6PZc1d-wT-U7YGCuxoSc/edit?usp=drivesdk)
- Investigamos sobre comandos de voz por bluetooth.  
[https://www.youtube.com/watch?v=q7gzRyam\\_o0](https://www.youtube.com/watch?v=q7gzRyam_o0)



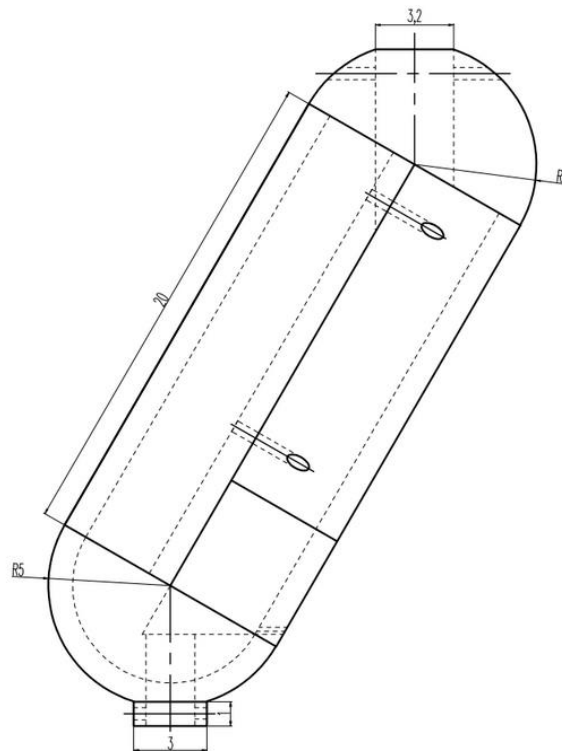
- Buscamos información sobre el DFPlayer Mini y su conexión con el módulo Bluetooth HC-05  
<http://www.infotronicblog.com/2017/09/controlar-dfplayer-mini-con-android-por-8.html>

**Semana del 16 al 21 de agosto:**

- Se diseñó el plano de la pieza N° 1 de la estructura.

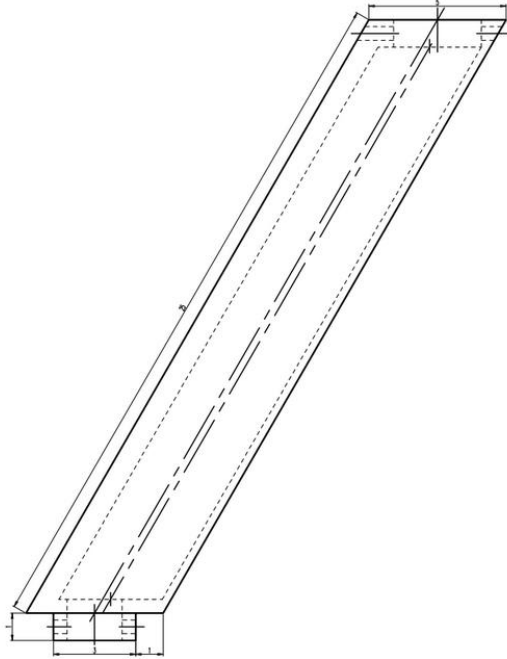


- Se diseñó el plano de la pieza N° 2 de la estructura.

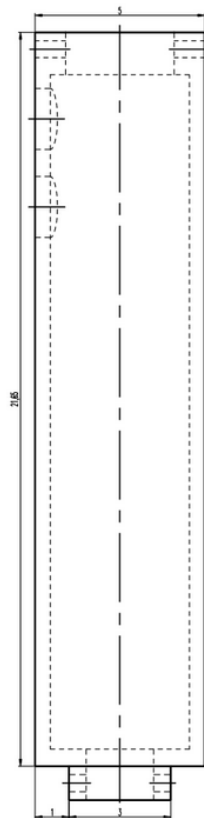




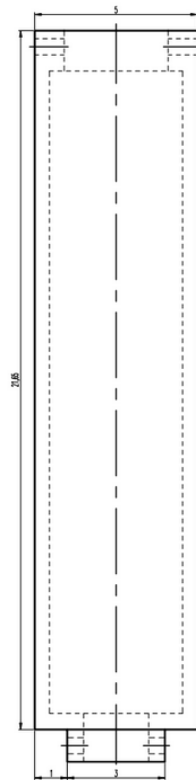
- Se diseñó el plano de la pieza N° 3 de la estructura.



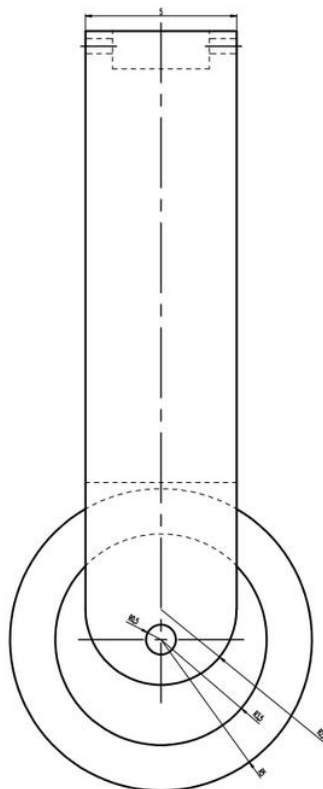
- Se diseñó el plano de la pieza N° 4 de la estructura.



- Se diseñó el plano de la pieza N° 5 de la estructura.



- Se diseñó el plano de la pieza N° 6 de la estructura.



Semana del 23 al 28 de agosto:

- Se creó el logotipo de Gekko.



- Se diseñó una nueva página Web, mejorando el diseño anterior, y utilizando Webnode.

Página Web: [www.gekko85.webnode.es](http://www.gekko85.webnode.es)



Tu ayuda tecnológica

[Inicio](#) [Información técnica](#) [Sobre nosotros](#) [Contacto](#)

## AYUDANDO A LA SOCIEDAD NO VIDENTE CON NUESTRO PROYECTO

*Desarrollamos un bastón electrónico el cual detecta y notifica al usuario mediante avisos sonoros la existencia de un objeto frente a él, logrando con esto, evitar posibles accidentes.*

Gekko es un bastón equipado con la electrónica necesaria para poder ayudar a personas no videntes, y especialmente pensado para disminuir la cantidad de accidentes.

Además, Gekko cuenta con un diseño esencialmente ergonómico y de fácil adaptación, lo que lo hace un bastón sumamente cómodo y accesible.



- Se diseñó el modelo 3D de la pieza N° 1 de la estructura.



- Se diseñó el modelo 3D de la pieza N° 2 de la estructura.



- Se diseñó el modelo 3D de la pieza N° 3 de la estructura.



**Semana del 6 al 11 de septiembre:**

- Se investigó sobre cómo usar la Api de Google Maps para JavaScript.  
<https://docs.google.com/document/d/1QRI-8FjMYEoEv-CLapjx7vqNSSSqYuPVQxKNZaHIRVU/edit?usp=drivesdk>
- Se diseñó el modelo 3D de la pieza N° 4 de la estructura.



- Se diseñó el modelo 3D de la pieza N° 5 de la estructura.



- Se diseñó el modelo 3D de la pieza N° 6 de la estructura.



### Semana del 13 al 18 de septiembre:

- Se creó el programa para medir distancia en Arduino.

```
int pinTrig = 2;
int pinEcho = 3;
int led1 = 4;
int led2 = 5;
int led3 = 6;
int led4 = 7;
int led5 = 8;
float tiempo_de_espera;
float distancia;

void setup() {
  Serial.begin (9600); // establemos la comunicacion serial
  pinMode (pinTrig, OUTPUT); // declarmos el pin 2 como salida
  pinMode (pinEcho, INPUT); // declaramos el 3 como entrada
  pinMode (led1, OUTPUT); // declaramos el 4 como salida
  pinMode (led2, OUTPUT); // declaramos el 5 como salida
  pinMode (led3, OUTPUT); // declaramos el 6 como salida
  pinMode (led4, OUTPUT); // declaramos el 7 como salida
  pinMode (led5, OUTPUT); // declaramos el 8 como salida
}

void loop() {
  digitalWrite (pinTrig, LOW); // ponemos en bajo el pin 2 durante 2 microsegundos
  delayMicroseconds (2);
  digitalWrite (pinTrig, HIGH); // ahora ponemos en alto pin 2 durante 10 microsegundos;
  delayMicroseconds (10); // pues este el momento en que emite el sonido durante 10 segundos
  digitalWrite (pinTrig, LOW); // ahora ponemos en bajo pin 2

  tiempo_de_espera = pulseIn (pinEcho, HIGH); // pulseIn, recoge la señal del sonido que emite el pinTrig

  distancia =(tiempo_de_espera/2)/29.15; // formula para hallar la distancia

  Serial.print (distancia); // imprimimos la distancia en cm
  Serial.println ("cm");
  delay (1000);

  if (distancia>=81 && distancia<100){
    digitalWrite(led1, HIGH);
  }
  if (distancia>=61 && distancia<80){
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
  }
  if (distancia >=41 && distancia<60){
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, HIGH);
  }
}
```

- Se creó el programa en Arduino para enviar un 1 cuando se toca el botón vía bluetooth.

```
int Enviados[] = {0,0}; //Hacemos un arreglo para los datos a enviar
int led13 = 13; //Declaramos el pin del LED
char entrada; //Declaramos una variables para los datos de entrada

//El tiempo que se demora en enviar un nuevo dato a la aplicacion
unsigned long TiempoAhora = 0; //Variable para determinar el tiempo transcurrido

void setup() {
  pinMode(led13, OUTPUT); //Declaramos el pin del LED como salida
  Serial.begin(9600); //Declaramos el puerto Serie
}

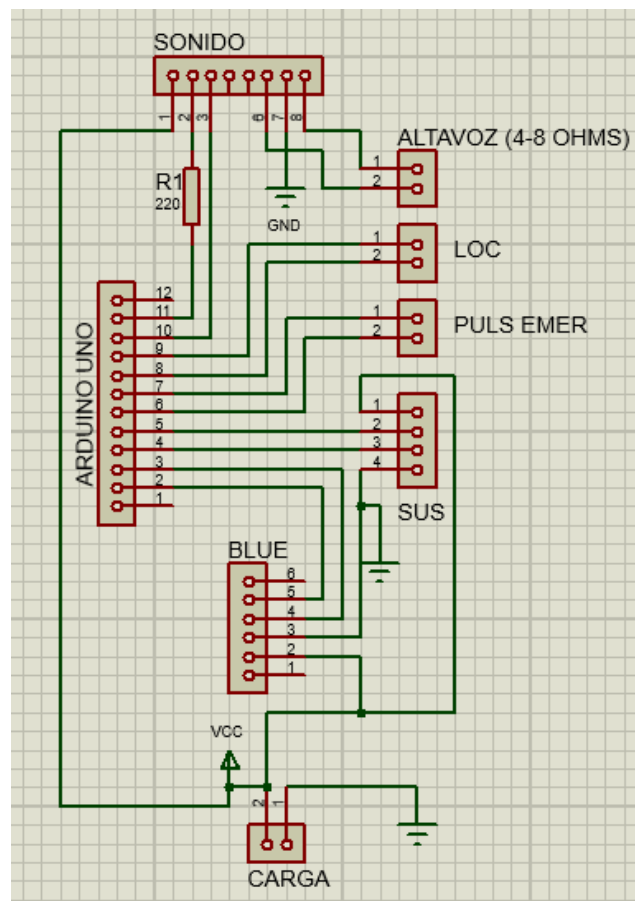
void loop() {

  if (Serial.available() > 0) { //Si hay datos enviados por la aplicacion
    entrada = Serial.read(); //Leemos los datos recibidos

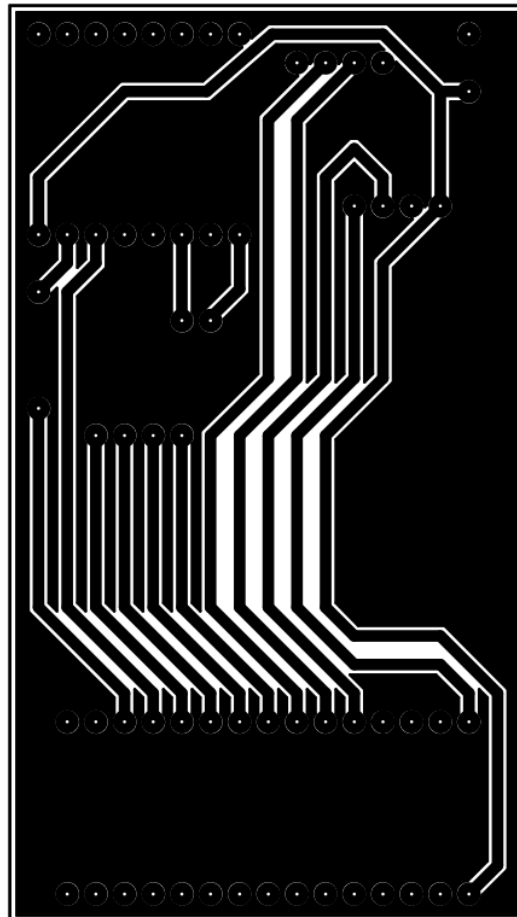
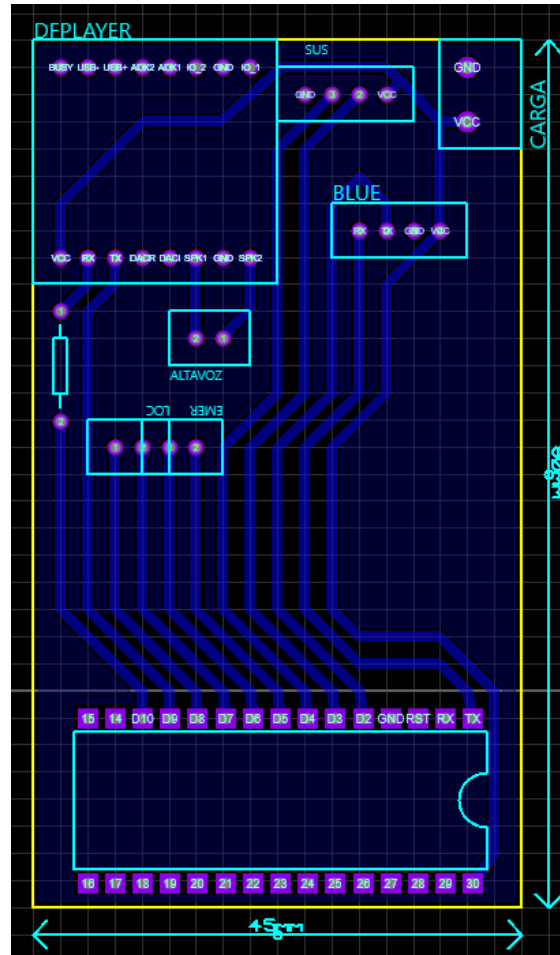
    if (entrada == 'A') { //Si el dato recibido es A, se enciende el led
      digitalWrite(led13, HIGH);
      Enviados[0] = 1; //Modificamos en el la posicion 1 del arreglo el estado del led con 0 o 1
    }

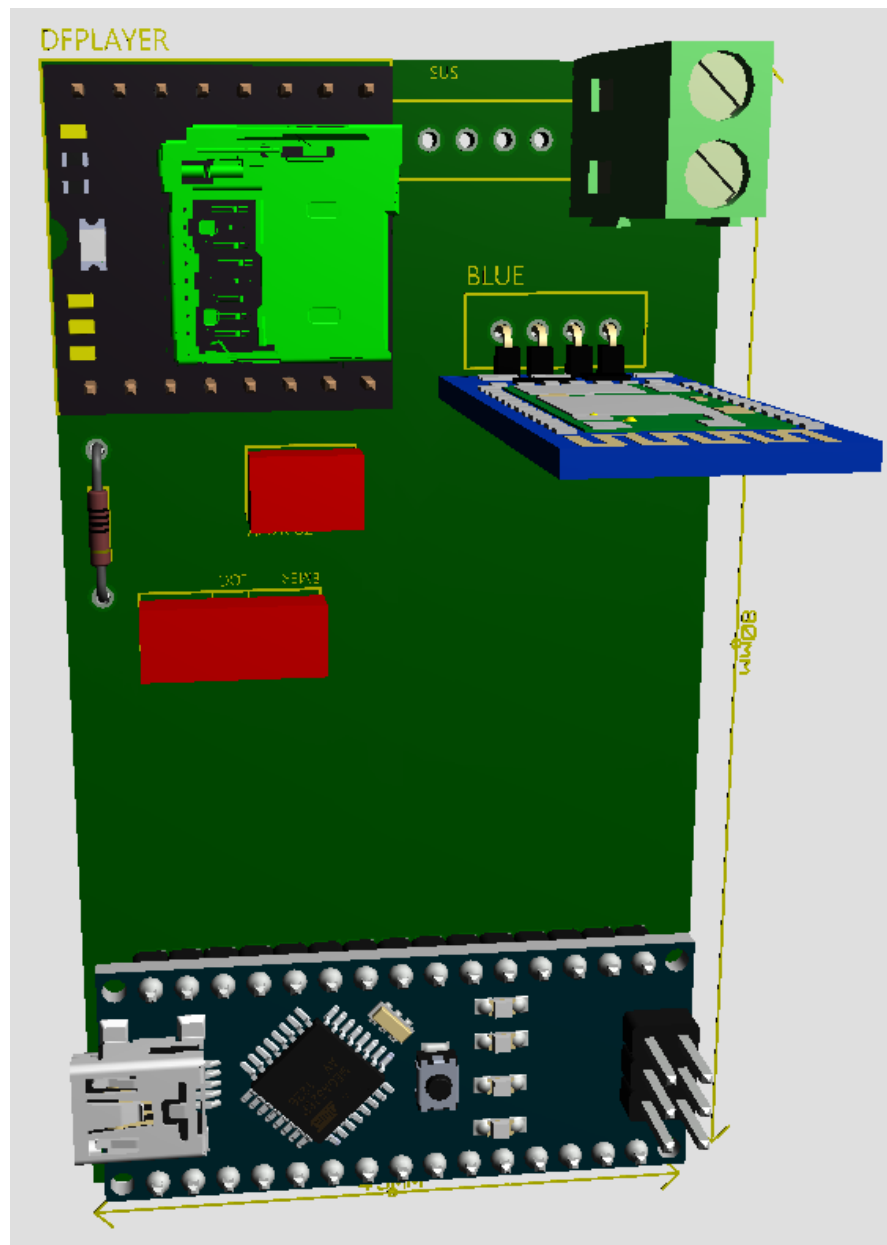
    if (entrada == 'B') { //Si el dato recibido es B, se apaga el led
      digitalWrite(led13, LOW);
      Enviados[0] = 0; //Modificamos en el la posicion 0 del arreglo el estado del led con 0 o 1
    }
  }
}
```

- Se diseñó la tercera versión del PCB en Proteus. Removiendo el GPS, reduciendo la Placa e introduciendo componentes 3D.





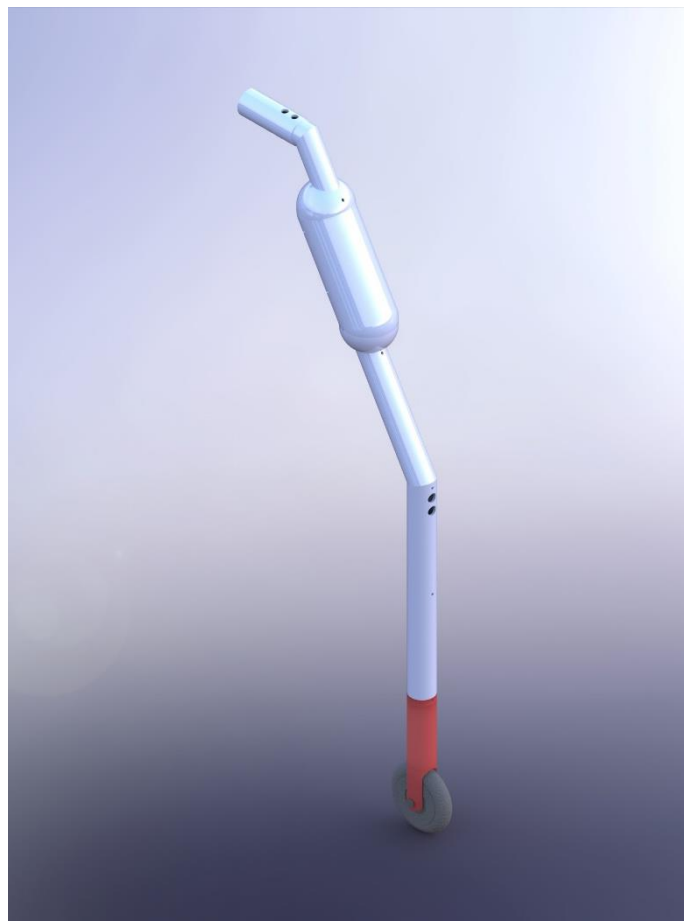




- Se creó el template de la app en Ionic.

**Semana del 27 de septiembre al 2 de octubre:**

- Se realizaron los primeros dos renders del bastón en SolidWorks.

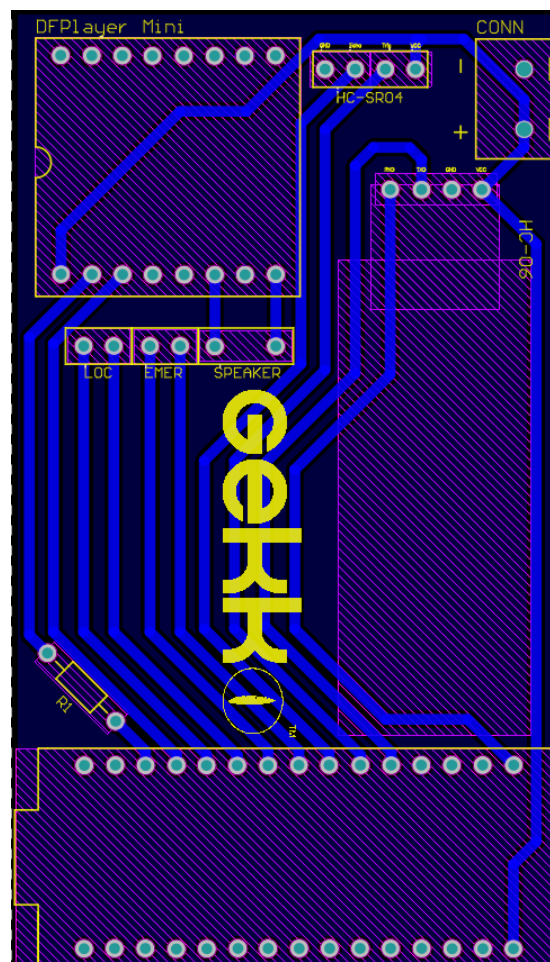
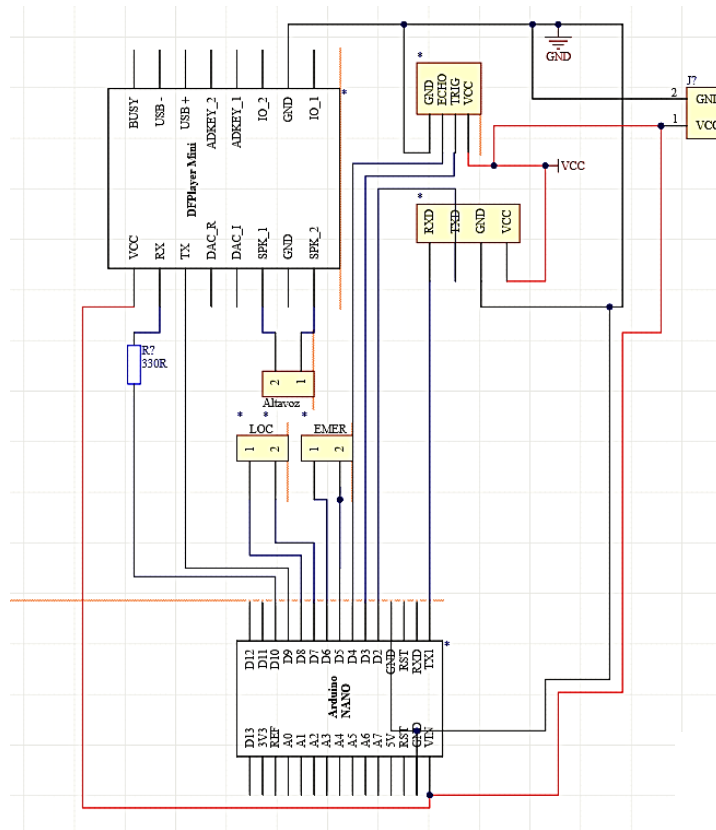


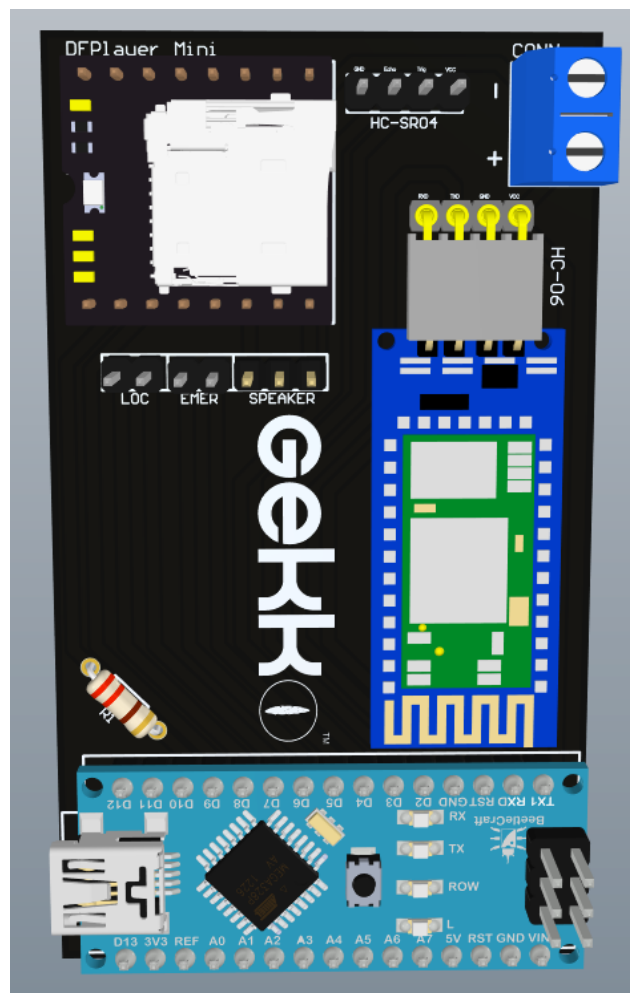
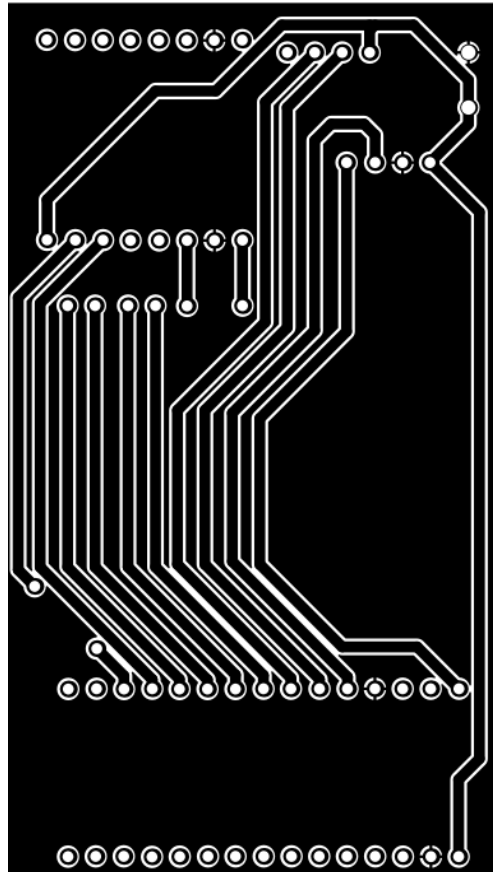
**Semana del 4 al 9 de octubre:**

- Se realizó el tercer render del bastón en SolidWorks.



- Se cambió de Proteus a Altium Designer, y se finalizó el PCB y su modelo 3D.





**Semana del 11 al 16 de octubre:**

- Se realizó el 4to y 5to render del bastón en KeyShot 9.





## Semana del 18 al 23 de octubre:

- Se subieron los planos y modelos 3D de la estructura de forma ordenada.
- Se realizó la primera presentación Power Point de Gekko.
- Se creó el usuario de Instagram de Gekko.  
[@proyectogekko](https://www.instagram.com/proyectogekko)
- Se realizó la primera publicación en Instagram.

## Semana del 1 al 6 de noviembre:

- Se comenzó la programación de la app en Android.

```
#include "SoftwareSerial.h"           // Para el Bluetooth.
#include "DFRobotDFPlayerMini.h"      // Para la musica.
SoftwareSerial mySoftwareSerial(10, 11); // Para la musica.
DFRobotDFPlayerMini myDFPlayer;      // Para la musica.
int pinTrig = 2;                      // Pin 2 del Arduino para el Trigger del sensor.
int pinEcho = 3;                      // Pin 3 del Arduino para el Echo del sensor.
float tiempo_de_espera;               // Tiempo de espera del sensor.
float distancia;                      // Distancia que mide el sensor.
const int pinon = 4;                  // Botón pulsado.
int estaon = HIGH;                    // Variables estado de los botones.
int estaoff = HIGH;                   // Variables estado de los botones.
int estado=0;

//-----

void setup()

{
  mySoftwareSerial.begin(9600);        // El serial del audio empieza a 9600.
  if (!myDFPlayer.begin(mySoftwareSerial)) // Si se prende el dfplayer.
  {
    myDFPlayer.volume(30);             // Seteamos el volumen en 30.
  }
  Serial.begin(9600);                  // El serial empieza a 9600.
  pinMode(pinon, INPUT);               // Boton tocado como entrada.
  pinMode (pinTrig, OUTPUT);           // Declaramos el pin 2 como salida del sensor.
  pinMode (pinEcho, INPUT);            // Declaramos el 3 como entrada del sensor.
}

//-----

void loop()
{
  digitalWrite (pinTrig, HIGH);        // El pin Trig esta HIGH.
  digitalWrite (pinTrig, LOW);         // El pin Trig esta LOW.
  tiempo_de_espera = pulseIn (pinEcho,HIGH); // Esto es para la señal del TRIG.
  distancia =(tiempo_de_espera/2)/29.15; // Mide la distancia.
  Serial.print (distancia);             // Imprimimos la distancia en cm.
  Serial.println ("cm");                // Imprimimos la distancia en cm.

  if (distancia>=200 && distancia<250)   // Si la distancia es mayor o igual a 200 y menor a 250:
  {
    myDFPlayer.play(1);                 // Reproduce el primer aviso.
  }
  if (distancia>=150 && distancia<199)   // Si la distancia es mayor o igual a 150 y menor a 199:
  {
    myDFPlayer.play(2);                 // Reproduce el segundo aviso.
  }
}
```

```

    myDFPlayer.play(2);                // Reproduce el segundo aviso.
}
if (distancia >=100 && distancia<149)    // Si la distancia es mayor o igual a 100 y menor a 149:
{
    myDFPlayer.play(3);                // Reproduce el tercer aviso.
}
if (distancia>=50 && distancia<99)        // Si la distancia es mayor o igual a 50 y menor a 99:
{
    myDFPlayer.play(4);                // Reproduce el cuarto aviso.
}
if (distancia>=0 && distancia<49)        // Si la distancia es mayor o igual a 0 y menor a 49:
{
    myDFPlayer.play(5);                // Reproduce el quinto aviso.
}
if (distancia>251)                      // Si la distancia es mayor a 251:
{
    myDFPlayer.pause();                // Pausa el audio.
}
return distancia;

//-----Aca termina lo del sensor ultrasonico-----

estaon = digitalRead(pinson);           // Lee estado del boton.
if(Serial.available()>0)
{
    estado=Serial.read();                // Lee el estado del puerto serial.
}

if (estaon == LOW)                      // SE OPRIMIO EL BOTON DE EMERGENCIA?:
{
    estado=1;                            // Estado cambia a 1.
    if(estado=='1')                      // Si el estado cambia a 1:
    {
        myDFPlayer.play(6);             // Pone audio de OBSTACULO.
        estado=3;                       // Estado cambia a 3.
    }
}
if (estaoff == LOW)                     // SE OPRIMIO EL BOTON DE LOCACION?:
{
    estado=2;                            // Estado cambia a 2.
    if(estado=='2')                      // Si el estado cambia a 1:
    {
        myDFPlayer.play(7);             // Pone audio de LOCACION.
        estado=4;                       // Estado cambia a 4.
    }
}
}
}

```

- Se realizó el programa para cambiar el nombre del bluetooth HC-05.

```

#include <SoftwareSerial.h>

/* Programa el modulo bluetooth HC-06 con un nuevo:
NOMBRE (Nombre de 20 caracteres)
PIN (Clave de cuatro numeros)
BFS (Velocidad de conexion en baudios)

Tienda donde se compro el modulo: http://dinastiatecnologica.com/producto/modulo-bluetooth-hc-05/
By: http://elprofegarcia.com

CONEXIONES:
ARDUINO  BLUETOOTH
SV       VCC
GND      GND
PIN 2    TX
PIN 3    RX
*/

SoftwareSerial blue(2, 3); //Crea conexion al bluetooth - PIN 2 a TX y PIN 3 a RX

char NOMBRE[21] = "Gekko"; // Nombre de 20 caracteres maximo
char BFS = '4'; // 1=1200 , 2=2400, 3=4800, 4=9600, 5=19200, 6=38400, 7=57600, 8=115200
char PASS[5] = "1234"; // PIN O CLAVE de 4 caracteres numericos

void setup()
{
    blue.begin(9600); // inicialmente la comunicacion serial a 9600 Baudios (velocidad de fabrica)

    pinMode(13,OUTPUT);
    digitalWrite(13,HIGH); // Enciende el LED 13 durante 4s antes de configurar el Bluetooth
    delay(4000);

    digitalWrite(13,LOW); // Apaga el LED 13 para iniciar la programacion

    blue.print("AT"); // Inicializa comando AT
    delay(1000);

    blue.print("AT+NAME"); // Configura el nuevo nombre
    blue.print(NOMBRE);
    delay(1000); // espera 1 segundo

    blue.print("AT+BAUD"); // Configura la nueva velocidad
    blue.print(BFS);
    delay(1000);

    blue.print("AT+PIN"); // Configura el nuevo PIN
    blue.print(PASS);
    delay(1000);
}

void loop()
{
    digitalWrite(13, !digitalRead(13)); // cuando termina de configurar el Bluetooth queda el LED 13 parpadeando
    delay(200);
}

```

- Se añadieron rótulos a los planos del bastón.
- Se realizó la segunda publicación en Instagram.
- Se finalizó el render del PCB.



- Se finalizó la página web.  
<https://gekko85.webnode.es/>
- Investigación flutter.  
<https://ionicframework.com/docs/intro/cli>
- Prueba APK flutter.
- Se utilizó IONIC.

**Semana del 8 al 13 de noviembre:**

- Se finalizó el render del bastón en KeyShot 9.

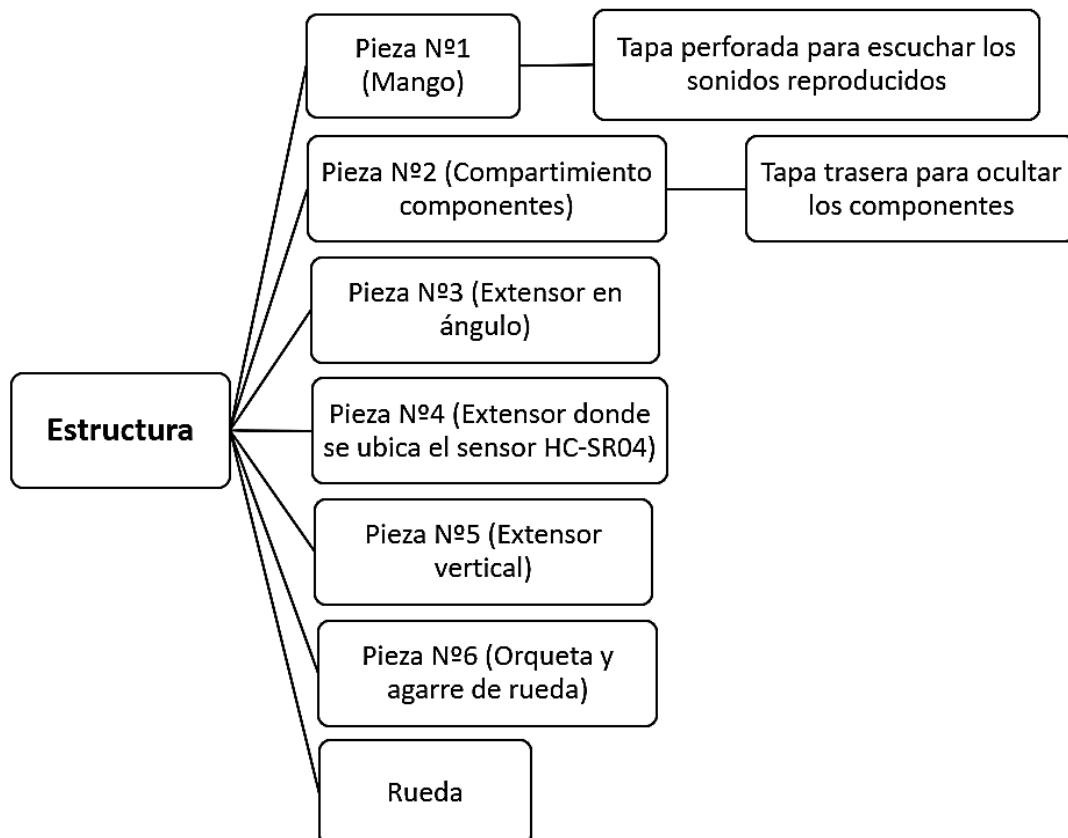


- Se agregaron los diseños a la página Web.
- Se realizó la presentación en Power Point para la exposición.  
<https://drive.google.com/file/d/1DkVcqHYwxIBh6nMYcTGWT28KHICQso/cm/view?usp=sharing>
- Se adaptó el diseño del logo para la muestra.

- Se realizó la tercera publicación en Instagram.
- Se realizó la cuarta publicación en Instagram.

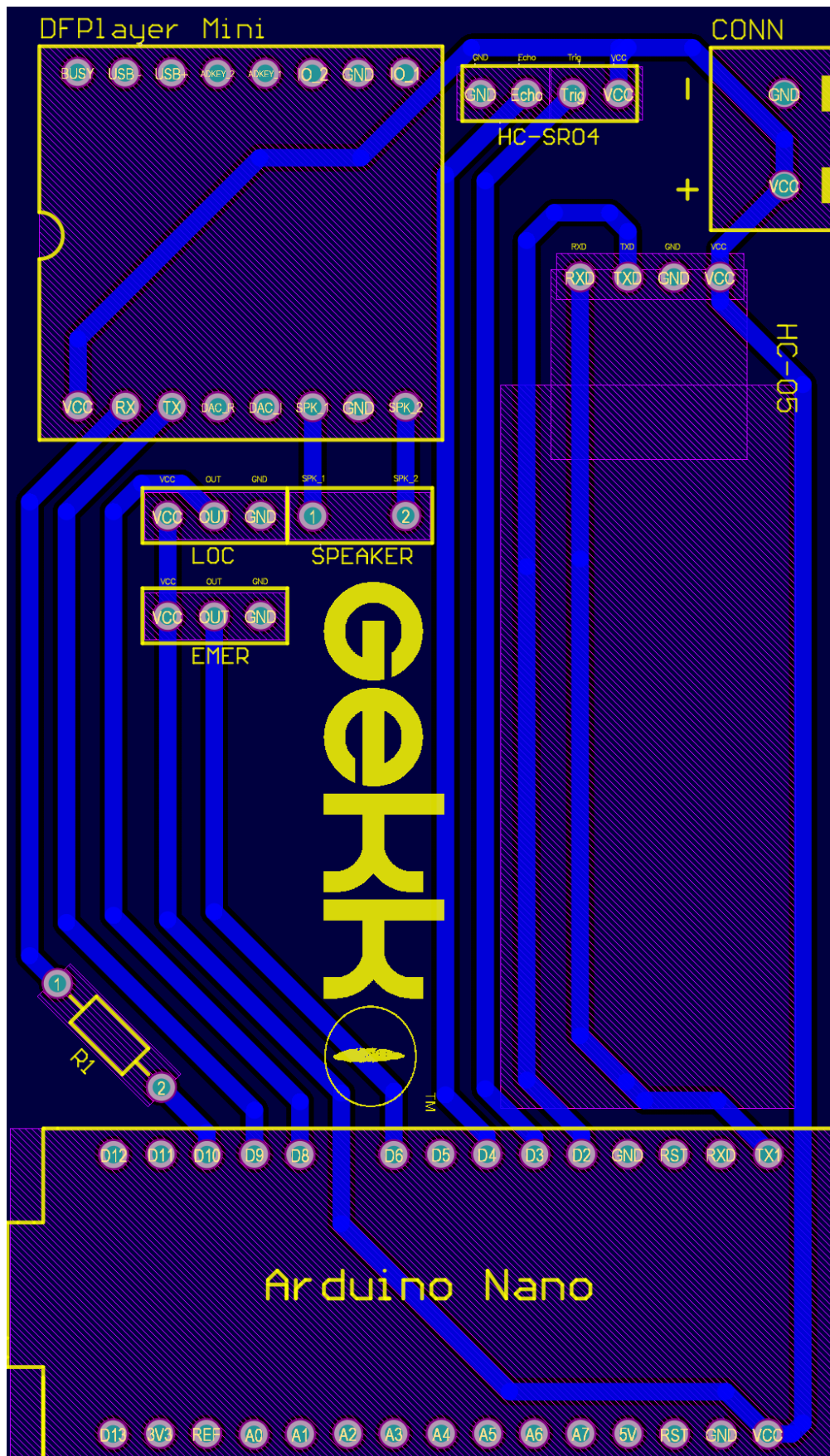
**Semana del 15 al 20 de noviembre:**

- Comenzamos a realizar la carpeta técnica.
- Se realizó la quinta publicación en Instagram.
- Se realizó la sexta publicación en Instagram.
- Se realizó el anexo de “Factores humanos”.  
<https://docs.google.com/document/d/1f6RJz6FIHRnoTzImvMXGc5x76mhqgnwueQJcxWehgCY/edit?usp=sharing>
- Se realizó un nuevo diagrama en bloques de diseño.

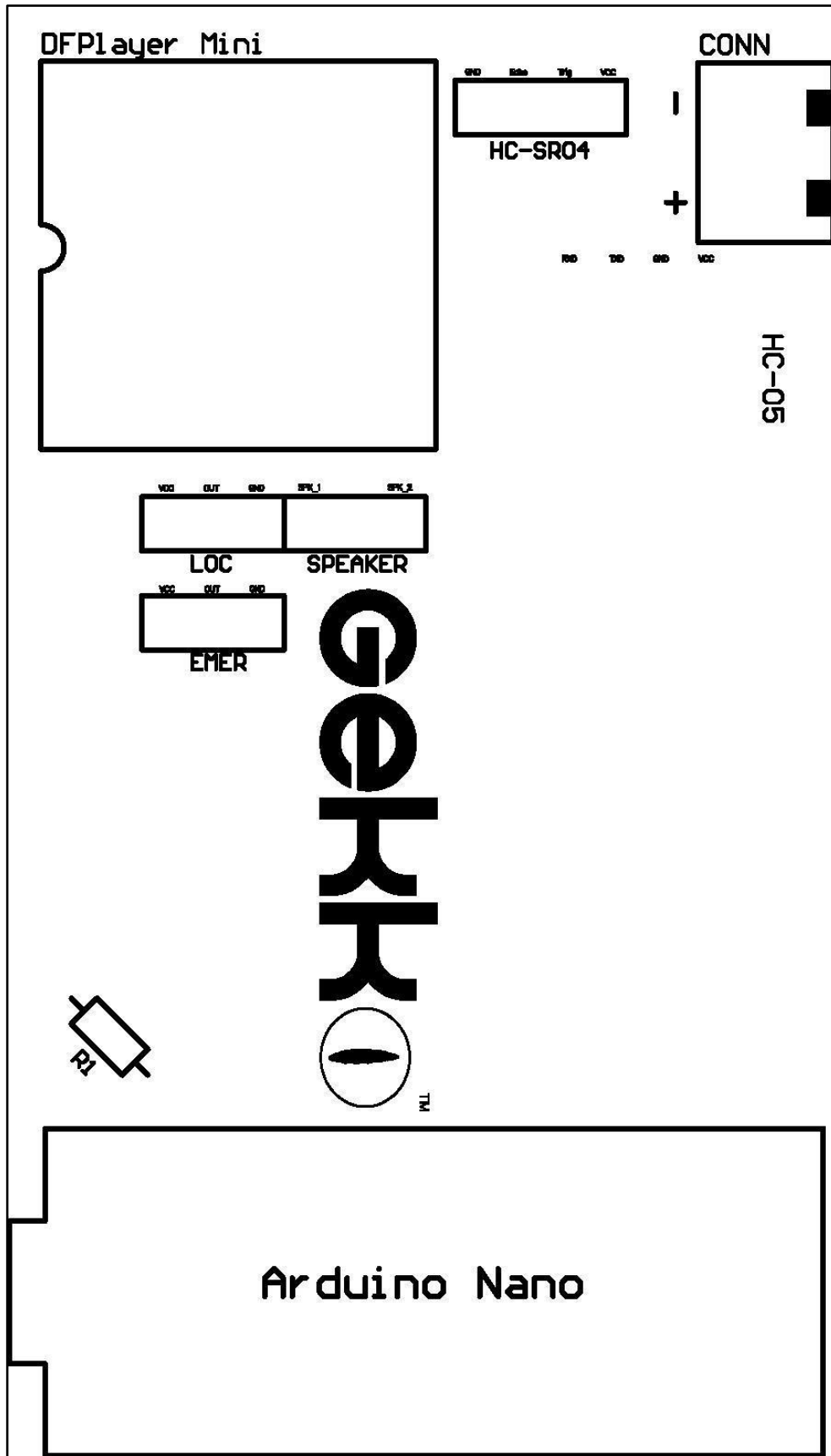




- Se corrigió el PCB y el modelo 3D.

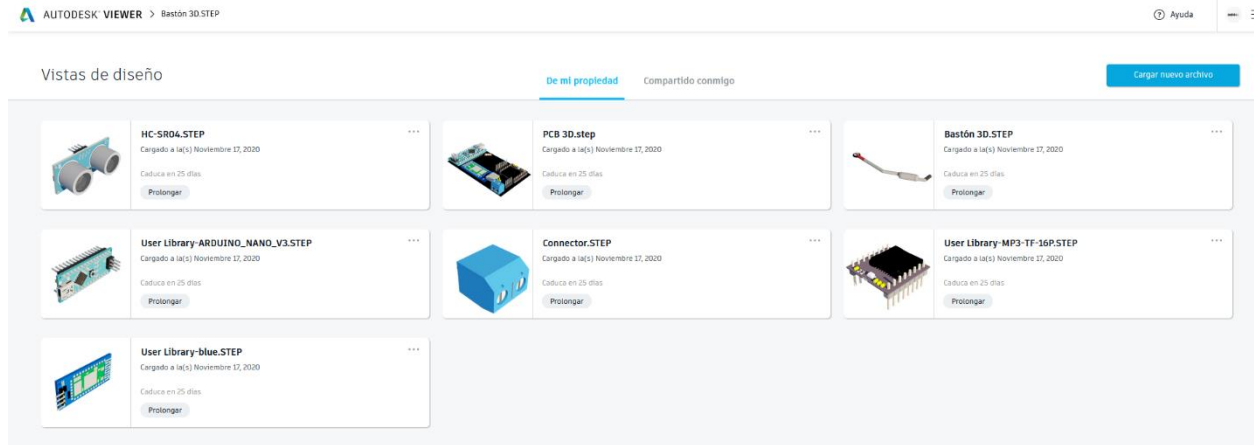








- Se subieron nuevamente los modelos 3D a Autodesk Viewer.



**HC-SR04:** <https://autode.sk/33uLrfF>

**Arduino Nano:** <https://autode.sk/376VoAz>

**HC-05:** <https://autode.sk/2UZ0slc>

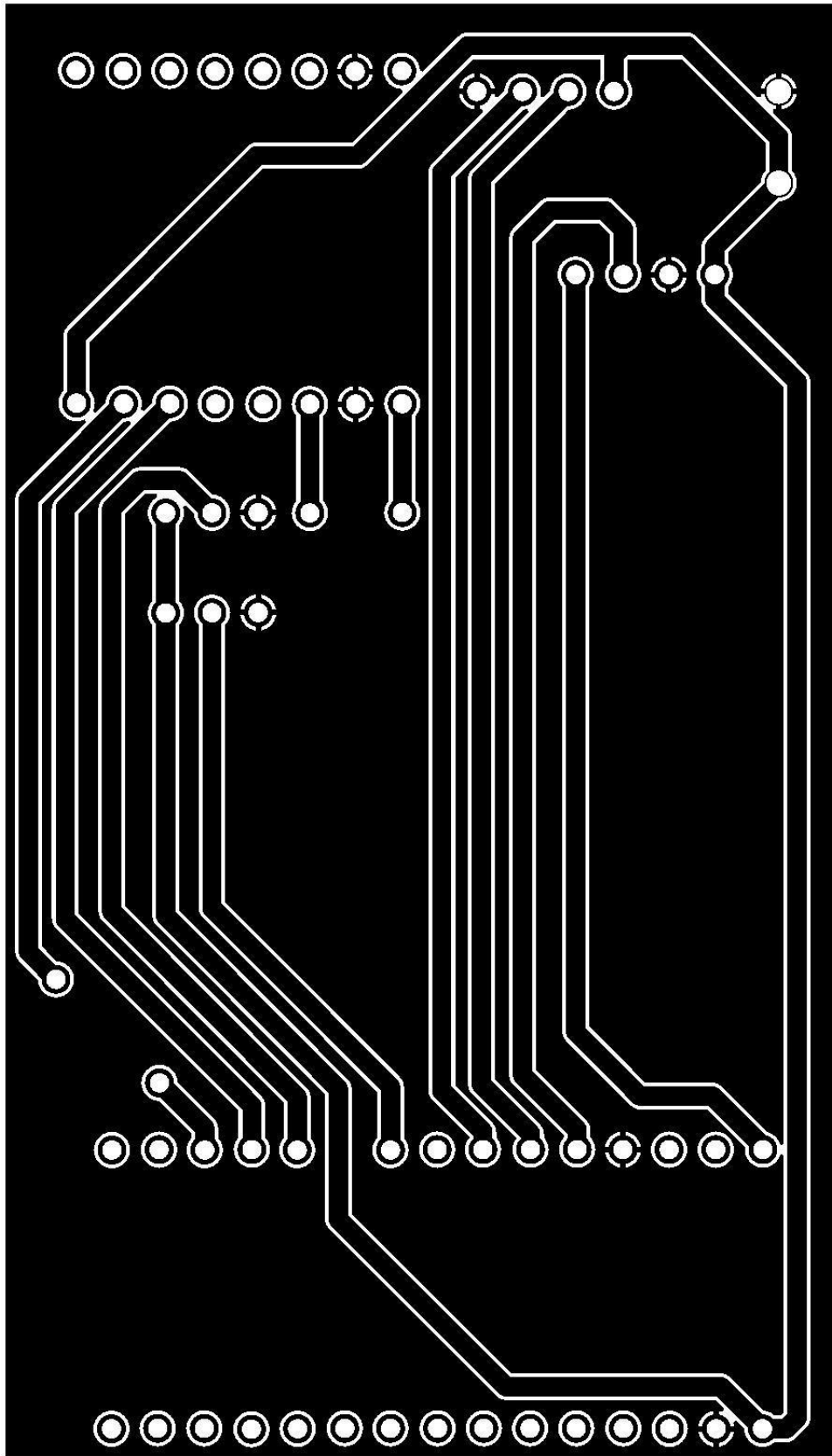
**PCB:** <https://autode.sk/3pXHDNB>

**Bornera:** <https://autode.sk/3kVppIE>

**DFPlayer Mini:** <https://autode.sk/2J4nHYB>

**Estructura:** <https://autode.sk/35LB4Wd>

- Se realizó el modelo de impresión del PCB.



- Creación código Arduino.

```
#include "SoftwareSerial.h"           // Para el Bluetooth.
#include "DFRobotDFPlayerMini.h"      // Para la musica.
SoftwareSerial BT1(4,2);              // RX, TX Del Bluetooth.
SoftwareSerial mySoftwareSerial(10, 11); // Para la musica.
DFRobotDFPlayerMini myDFPlayer;      // Para la musica.
const int Trigger = 2;               // Pin digital 2 para el Trigger del sensor.
const int Echo = 3;                 // Pin digital 3 para el Echo del sensor.
float distancia;                     // Distancia que mide el sensor.
const int pinemergencia = 4;         // Botón pulsado emergencia.
const int pinlocacion = 5;          // Botón pulsado locacion.
int BotonEmergencia = HIGH;          // Variables estado de los botones.
int BotonLocacion = HIGH;           // Variables estado de los botones.
int estado=0;                        // Estado es 0.

//-----Void Setup-----

void setup()

{
  Serial.begin(9600);                // El serial empieza a 9600.
  mySoftwareSerial.begin(9600);      // El serial del audio empieza a 9600.
  if (!myDFPlayer.begin(mySoftwareSerial)) // Si se prende el dfplayer.
  {
    myDFPlayer.volume(30);          // Pone el volumen a 30.
  }
  pinMode(pinemergencia, INPUT);     // Boton tocado como entrada.
  pinMode(pinlocacion, INPUT);       // Boton tocado como entrada.
  pinMode(Trigger, OUTPUT);          // Declaramos el pin 2 como salida del sensor.
  pinMode(Echo, INPUT);              // Declaramos el 3 como entrada del sensor.
}

//-----Void Loop-----

void loop()
{
  long t;                            //Tiempo que demora en llegar el eco.
  long d;                            // Distancia en centimetros.
  digitalWrite (Trigger, HIGH);      // El prin Trig esta HIGH.
  digitalWrite (Trigger, LOW);       // El prin Trig esta LOW.
  Serial.print (distancia);          // Imprimimos la distancia en cm.
  t = pulseIn (Echo, HIGH);          // Obtenemos el ancho del pulso.
  d = t/59;                          // Escalamos el tiempo a una distancia en cm.

  if (distancia>=200 && distancia<250) // Si la distancia es mayor a 200 y menor a 250:
  {
    Serial.print("Distancia: ");
    Serial.print(d);
    myDFPlayer.play(1);              // Reproduce el primer aviso de Obstaculo.
  }
  if (distancia>=150 && distancia<199) // Si la distancia es mayor a 150 y menor a 199:
  delay(2600);
}
```

```

if (distancia>=150 && distancia<199)          // Si la distancia es mayor a 150 y menor a 199:
delay(2600);
{
  myDFPlayer.play(2);                          // Reproduce el segundo aviso.
  delay(2300);
}
if (distancia >=100 && distancia<149)          // Si la distancia es mayor a 100 y menor a 149:
{
  myDFPlayer.play(3);                          // Reproduce el Tercer aviso.
  delay(2600);
}
if (distancia>=50 && distancia<99)             // Si la distancia es mayor a 50 y menor a 99:
{
  myDFPlayer.play(4);                          // Reproduce el Cuarto aviso.
  delay(2300);
}
if (distancia>=0 && distancia<49)              // Si la distancia es mayor a 0 y menor a 49:
{
  myDFPlayer.play(5);                          // Reproduce el quinto aviso.
  delay(2300);
}
if (distancia>251)                             // Si la distancia es mayor a 251:
{
  myDFPlayer.pause();                          // Pausa el audio.
}
return distancia;

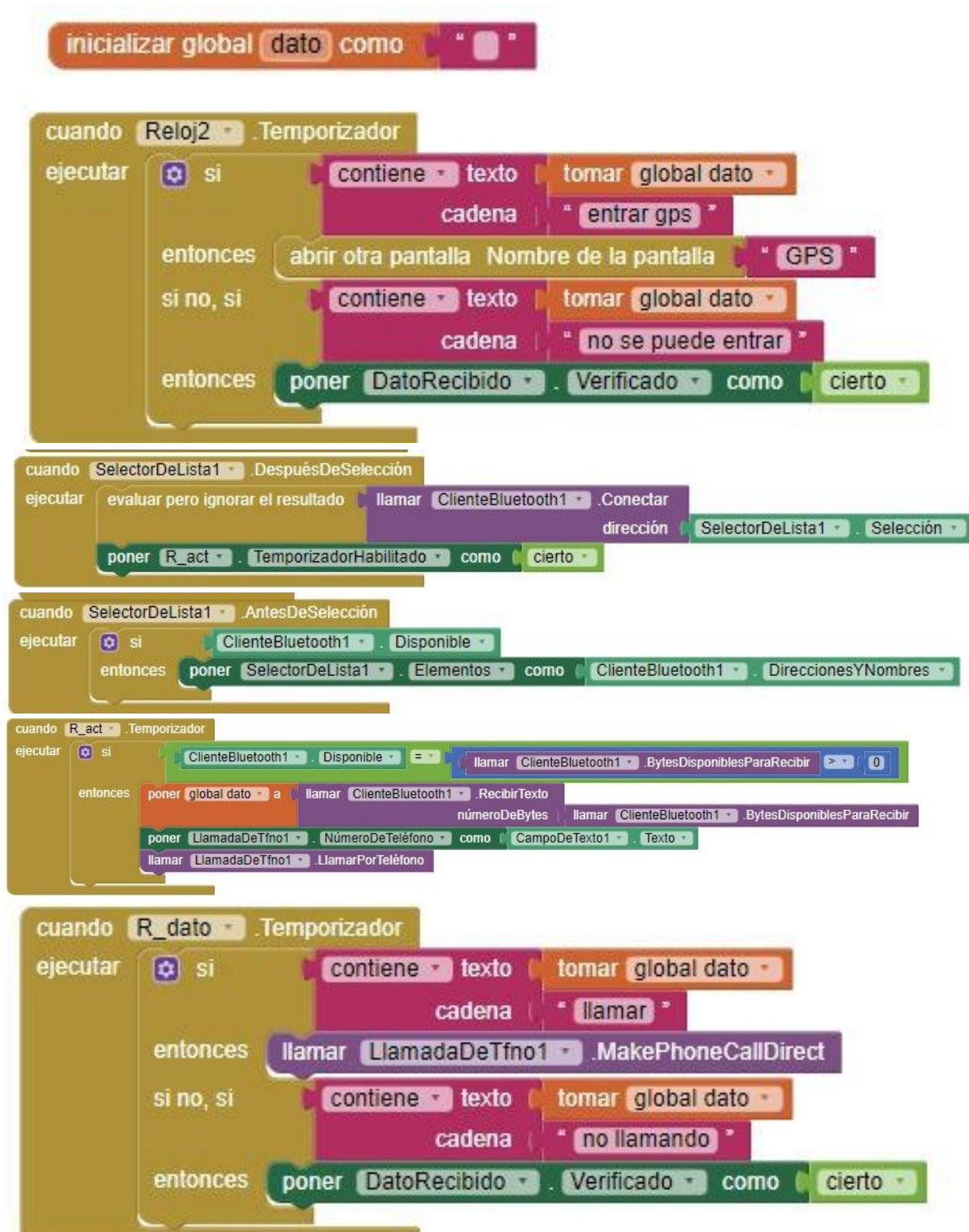
//-----Aca termina lo del sensor ultrasonico-----

BotonEmergencia = digitalRead(pinemergencia); // Lee estado del boton de emergencia.
if(Serial.available()>0)                      // Si hay un serial disponible, entonces...
{
  estado=Serial.read();                      // Lee el estado del puerto serial.
}
BotonLocacion = digitalRead(pinlocacion);      // Lee estado del boton de locacion.
if(Serial.available()>0)                      // Si hay un serial disponible, entonces...
{
  estado=Serial.read();                      // Lee el estado del puerto serial.
}
if (BotonEmergencia == LOW)                   // SE OPRIMIO EL BOTON DE EMERGENCIA?:
{
  BT1.println("1");                          // Se manda un 1 a la Aplicacion.
}
if (BotonLocacion == LOW)                     // SE OPRIMIO EL BOTON DE LOCACION?:
{
  BT1.println("0");                          // Se manda un 1 a la Aplicacion.
}
}

```



- Código APP Inventor.



- Presupuesto.

| Partes Del Proyecto          | Precio      |
|------------------------------|-------------|
| Modulo de audio DFPlayerMini | \$ 460      |
| Parlante 3W 4Ohms            | \$ 420      |
| Sensor Ultrasonico HC-SR04   | \$ 180      |
| Arduino Nano                 | \$ 522      |
| Tarjeta Micro SD             | \$ 1,090    |
| Plaqueta                     | \$ 282      |
| Estaño                       | \$ 110      |
| Pulsadores x2                | \$ 110      |
| Modulo Bluetooth HC-06       | \$ 650      |
| Impresión 3D                 | Consultando |
| Borneras                     | \$ 220      |
| Resistencia                  | \$ 2.50     |
| Pines                        | \$ 86       |
| Soldador de Estaño           | \$ 490      |

#### Semana del 22 al 27 de noviembre:

- Se finalizó la carpeta técnica.
- Se finalizó la carpeta de campo.