

## PROYECTO PUENTE LEVADIZO MANDO INALÁMBRICO.

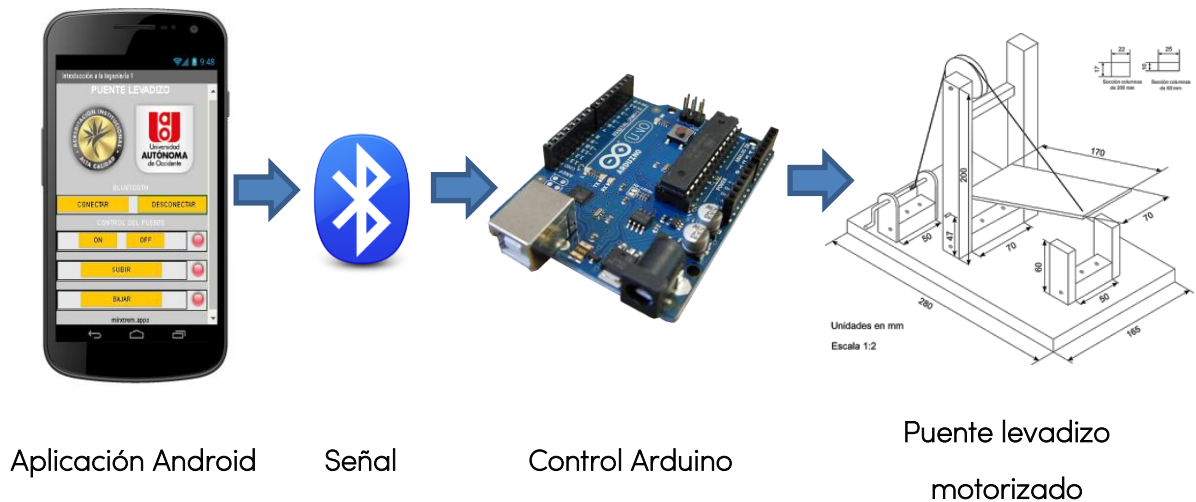
Este proyecto se desarrollado para la clase de introducción a la ingeniería 1 por Gustavo Arango de mirxtrem.apps y es de código abierto para su libre uso y se debe

## Resumen.

Se trata de un puente hecho en madera donde se requiere que suba y baje sus plataformas por medio de un motor dc y poleas; este se activará a través de un celular por medio de una aplicación y usando comunicación bluetooth al control del motor.

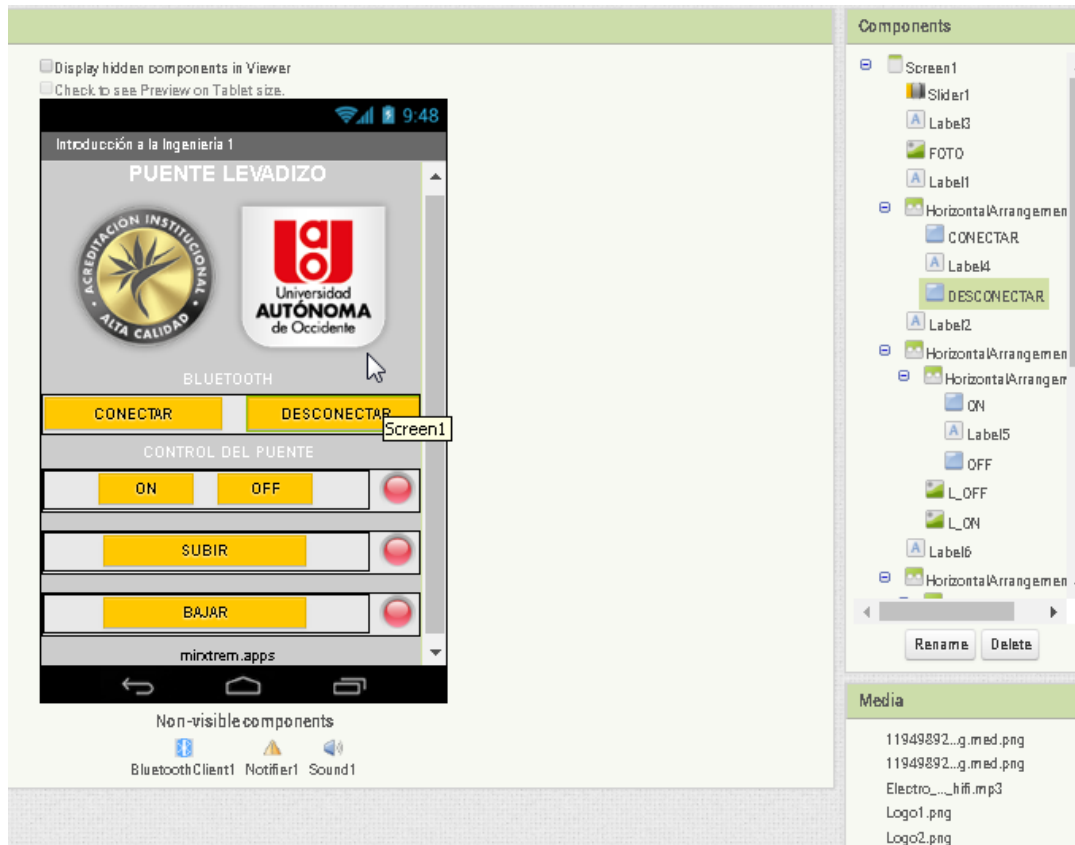
La plataformas tendrán un micro-interruptores finales de carrera para detener el motor y la plataforma cuando lleguen al límite superior e inferior. También se tiene dos led que serán el semáforo.

## Estructura de comunicación.



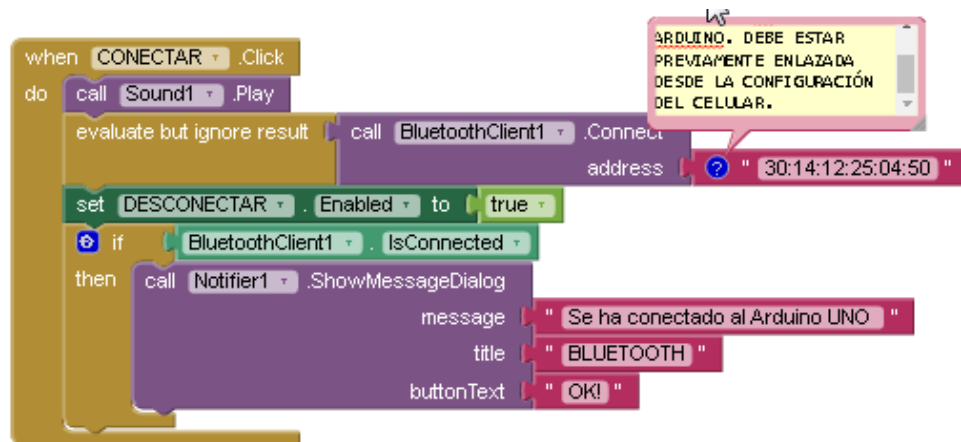
Aplicación móvil android.

Desarrollada en App Inventor 2.

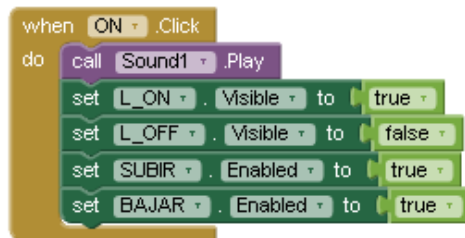


Código AppInvetor2.

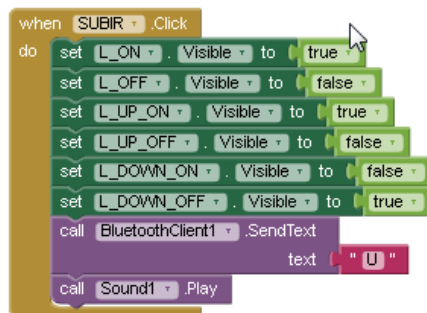
Botón CONECTAR: para establecer conexión con Arduino y notifica la conexión.



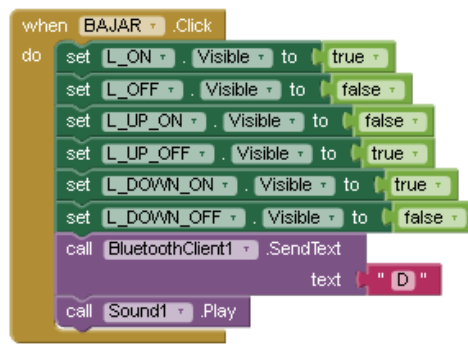
Botón ON: enciende los leds indicadores y habilita los botones SUBIR y BAJAR. Al presionar el botón genera un sonido.



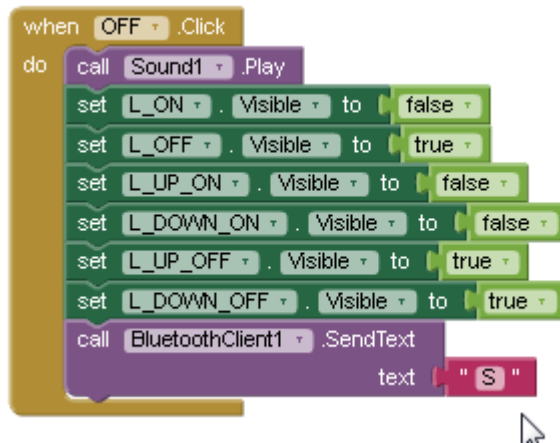
Botón SUBIR: enciende los leds indicadores y envia el text 'U' al Arduino a través de bluetooth. Al presionar el botón genera un sonido.



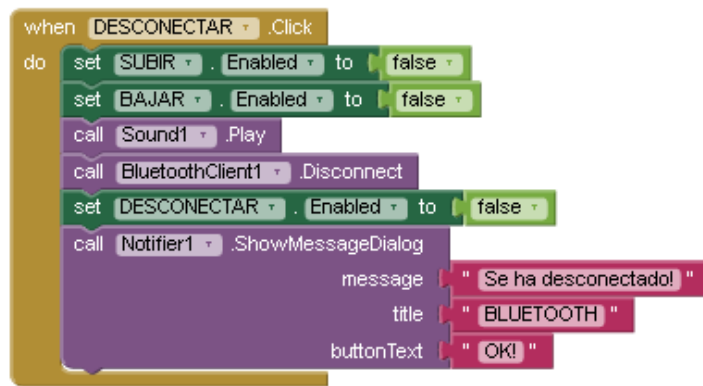
Botón BAJAR: enciende los leds indicadores y envia el text 'D' al Arduino a través de bluetooth. Al presionar el botón genera un sonido.



Botón OFF: enciende los leds indicadores y envía el text 'S' al Arduino a través de bluetooth. Al presionar el botón genera un sonido.



Botón DESCONECTAR: desconecta la conexión con Arduino y notifica la desconexión. Al presionar el botón genera un sonido.



## Sketch Arduino.

// Primero creamos las variables del programa

```
int microSup = 6;
int microInf = 7;
int subir = 13;
int bajar = 12;
int botonUp = 8;
int botonDown = 9;
int ledVerde = 10;
int ledRojo = 11;
int estadoMicroU = 0;
int estadoMicroD = 1;
int estadoUp = 0;
int estadoDown = 0;
```

```
void setup() {
```

// Declaramos cuales son las salidas y las entradas y sus estados

```
pinMode(botonUp, INPUT);
pinMode(botonDown, INPUT);
pinMode(microSup, INPUT);
```

```
pinMode(microInf, INPUT);
pinMode(subir, OUTPUT);
pinMode(bajar, OUTPUT);
pinMode(ledVerde, OUTPUT);
pinMode(ledRojo, OUTPUT);
digitalWrite(subir, LOW);
digitalWrite(bajar, LOW);
digitalWrite(ledVerde, HIGH);
digitalWrite(ledRojo, LOW);
```

```
}
```

```
void loop() {
```

```
    // Acción A: La rampa sube a la señal de subir
```

```
    estadoUp=digitalRead(btonUp);
```

```
    if(estadoUp == HIGH){
        digitalWrite(ledVerde, LOW);
        digitalWrite(ledRojo, HIGH);
        delay(2000);
        digitalWrite(subir, HIGH);
    }
```

```
    // Acción B: El motor se detiene para que la rampa deje de subir
```

```
    estadoMicroU=digitalRead(microSup);
```

```
    if(estadoMicroU == HIGH) {

        digitalWrite(subir, LOW);
```

```
// Acción C: La rampa baja a la señal de bajar
```

```
}
```

```
estadoDown=digitalRead(btonDown);
```

```
if(estadoDown == HIGH){
```

```
digitalWrite(bajar, HIGH);
```

```
}
```

```
// Acción D: La rampa baja y el semaforo cambia
```

```
estadoMicroD=digitalRead(microInf);
```

```
if(estadoMicroD == HIGH) {
```

```
digitalWrite(bajar, LOW);
```

```
delay(2000);
```

```
digitalWrite(ledVerde, HIGH);
```

```
digitalWrite(ledRojo, LOW);
```

```
}
```

```
}
```

Esquema de conexión.

