



UNIÓN DE ASOCIACIONES  
DE INGENIEROS TÉCNICOS  
INDUSTRIALES Y GRADUADOS  
EN LA INGENIERÍA DE LA  
RAMA INDUSTRIAL DE ESPAÑA

# UNIÓN DE ASOCIACIONES DE INGENIEROS TÉCNICOS INDUSTRIALES Y GRADUADOS EN INGENIERÍA DE LA RAMA INDUSTRIAL DE ESPAÑA (UAIITIE)

“CONVOCATORIA 2023”

## VIII PREMIO NACIONAL DE INICIACIÓN A LA INVESTIGACIÓN TECNOLÓGICA

### Cuna Mecedora Inteligente

AUTOR/ES:

Hugo Prieto, Samuel Jiménez, Mateo Orozco, Adrián Pérez

BLOQUE TEMÁTICO:

Robótica aplicadas al entorno doméstico.

NIVEL EDUCATIVO:

1º Bachillerato

COORDINADOR:

María Sol Martín

Marzo 2023



## Resumen

Este proyecto consiste en una cuna mecedora inteligente cuya finalidad es controlar de forma automática el sueño de los bebés. Hecha de madera, esta cuna tiene la capacidad de mecer: sin embargo esta no es su única funcionalidad, en realidad esta cuna tiene la capacidad de realizar diversas funciones las cuales explicaremos a continuación. Son:

1. Capacidad de medir la temperatura ambiente a la que está expuesta la cuna.
2. Capacidad de activar una pequeña manta térmica en caso de que la temperatura disminuya por debajo de los 20° C.
3. Capacidad de medir la temperatura haciendo uso de un sensor LM35.
4. Tira de leds integrados que se pueden activar y desactivar en remoto si se considera que es necesario.
5. Integración bluetooth que permite controlar las variaciones de los datos especificados y accionar el mecanismo que mece la cuna, además de activar y desactivar los leds todo ello desde una app móvil.
6. Detección de movimiento con sensor de ultrasonidos que activa el mecanismo de mecido si se detecta que el bebé se está moviendo.

## Palabras Clave

- Cuidado
- Infantil
- Cuna
- Mecedora
- Bluetooth



# Índice

<b>Resumen</b>	<b>2</b>
<b>Palabras Clave</b>	<b>2</b>
<b>Índice</b>	<b>3</b>
<b>Desarrollo</b>	<b>5</b>
1. Introducción	5
2. Objetivos	5
2.1 Objetivos relativos al resultado	5
2.2 Objetivos relativos al proceso	5
3. Metodología	6
3.1 <i>Proceso</i>	6
3.2 <i>Estructura</i>	7
3.2.1 <i>Cuna</i>	7
Figura 1: Alzado	8
Figura 2: Planta	8
Figura 3: Perfil	9
3.2.2 <i>Soporte</i>	9
3.2.3 <i>Caja</i>	9
3.3 <i>Código</i>	10
3.3.1 <i>Arduino</i>	10
Figura 4: Código 1	11
Figura 6: Código 3	13
3.3.2 <i>App móvil</i>	13
Figura 7: Código de AppInventor	14
3.4 <i>Circuitos</i>	14
Figura 8 : Simulación en tinkercad	15
4. Resultados	16
5. Conclusiones	16
<b>Referencias</b>	<b>17</b>



# Desarrollo

## 1. Introducción

La tarea de ser padre es una de las tareas más importantes y demandantes del mundo. Compatibilizar el cuidado de los hijos con el trabajo, la salud y otras actividades puede resultar complicado, especialmente si estos son muy activos y despiertan a los padres por la noche. Para facilitar la tarea hemos considerado la creación de una cuna mecedora capaz de llevar a cabo múltiples que en otro caso llevarían tiempo a los padres y que pueden facilitar el proceso de mantener a un hijo y hacerlo lo más ameno, sencillo y feliz posible con la comodidad de poder controlar dichos procesos desde un dispositivo que llevamos con nosotros a todos lados.

## 2. Objetivos

### 2.1 Objetivos relativos al resultado

1. Facilitar la tarea de ser padres disminuyendo la cantidad de tareas a realizar mediante la automatización de procesos.
2. Crear un prototipo de cuna mecedora automatizada mediante un procedimiento sencillo que pueda ser replicado.
3. Verificar la seguridad del prototipo para el cuidado del bebé mediante la utilización de materiales y uniones resistentes manteniendo la estética del mismo.

### 2.2 Objetivos relativos al proceso

1. Desarrollar habilidades colaborativas y de trabajo en equipo.
2. Poner en práctica los conocimientos adquiridos en la asignatura para afianzarlos.



## 3. Metodología

### 3.1 Proceso

Antes de comenzar con la explicación del procedimiento, queremos hacer mención de los roles y tareas desarrollados por cada integrante del grupo. Hugo Prieto tomaba el rol de coordinador y su tarea principal era desarrollar código y aplicación. Samuel Jiménez tenía el rol de cuidado de la maqueta y encargado de taquilla y su tarea principal era la planificación y construcción de la maqueta. Mateo Orozco tenía el rol de encargado de herramientas y su tarea principal era el desarrollo de la parte de circuito. Por último, Adrián Pérez, con rol de limpieza de la zona de trabajo en el taller compartía con Mateo y Samuel las tareas de desarrollo de circuito y construcción de estructura. Cabe destacar que si bien cada miembro del grupo tenía una función principal, durante el desarrollo, todos los integrantes dedicaron una gran parte de tiempo a la construcción de la maqueta, tarea que tomaba más tiempo que el resto por su complejidad y lento proceso.

La primera fase del proceso fue la concepción de una idea de lo que íbamos a hacer. Fueron varias las ideas que ocuparon el lugar que acabó tomando la cuna inteligente. Algunas ideas planteadas fueron una maceta automatizada y una taza con medidas del consumo de cafeína. La reflexión y la búsqueda de una idea creativa nos llevaron a pensar en la cuna inteligente que hemos desarrollado.

Cuando tuvimos clara la idea de proyecto, empezamos a plantearnos qué material era el óptimo para construir un prototipo resistente y a la vez similar a los de su tipo. Optamos por utilizar la madera, al igual que muchas cunas, en concreto, madera de pino, tipo de madera resistente y fácil de trabajar. A continuación dedicamos un día en su totalidad para adquirir la madera y establecer y dibujar unas medidas muy próximas a las que finalmente utilizaremos.

Con el material ya preparado empezamos a trabajar cortando las láminas de madera limándolas para que queden lisas y sin desniveles y uniéndolas con la ayuda de clavos. El proceso de cortado y lijado de las piezas en conjunto con la unión de la estructura llevó un total aproximado de 8 horas.



Con la estructura ya terminada, desarrollamos el código y el circuito y los metimos en la caja de debajo de la cuna. Llegado este punto y con el funcionamiento esperado ya conseguido, aprovechamos la fase de pruebas de 3 días (3 horas en total) de duración previos a la evaluación del proyecto para buscar un mejor acabado estético, posicionando correctamente la bisagra y colocando correctamente el servo y ocultándolo ligeramente haciendo uso de una caja. También solucionamos errores de circuito y código que provocaron que el Idr funcionase de forma incorrecta. Es con esto que el trabajo quedó terminado.

A continuación les explicamos la maqueta a profundidad. Podemos dividir el proyecto completo en 3 pilares: estructura, código y circuitos.

## **3.2 Estructura**

La estructura es similar a la de una cuna convencional, con la adición de unos ejes a base de tornillos en los extremos superiores de las patas, que permiten el correcto funcionamiento del mecanismo de mecido. El material utilizado es madera, concretamente madera de pino, escogida por su calidad, y las uniones están mayoritariamente hechas con clavos que consideramos eran la mejor opción para aportar resistencia a la estructura en su conjunto. Podemos dividir la estructura en 3 partes: la cuna, el soporte y la caja.

### **3.2.1 Cuna**

La cuna está compuesta por una plancha de madera de pino a la que hemos unido 11 barrotes separados aproximadamente por la misma distancia y que funcionan de barrera para que el bebé no se pueda caer, aumentando la seguridad aplicando la estructura convencional de una cuna. Además, 4 barras de madera se unen a estos 11 barrotes para dar un mejor acabado y asegurar que estos mantienen su posición firme, posibilitando que el prototipo tenga una mayor vida útil. Por encima de la plancha hemos puesto una manta plegada sobre sí misma que funciona como un colchón, permitiendo que el bebé pueda dormir cómodamente.



**Figura 1: Alzado**



**Figura 2: Planta**



**Figura 3: Perfil**



### 3.2.2 Soporte

El soporte es sencillo. Consta de dos patas, que están unidas por encaje a dos planchas de madera por la parte inferior que mantienen la estructura en pie. Ambas patas están unidas por una unión central, construida para añadir firmeza a la estructura. Como ya se ha mencionado con anterioridad, ambas patas tienen en la parte superior una unión con tornillo a los laterales superiores de la cuna que funcionan como ejes y permiten que la cuna pueda mecer.

### 3.2.3 Caja

La caja es una estructura clavada a la plancha de la cuna por debajo. Es el corazón del prototipo. Contiene la placa Arduino que controla los circuitos, una placa protoboard, el módulo de bluetooth, el sensor de temperatura y da salida tanto a los leds, como al motor y la mini manta térmica. Tiene una puerta con unión de bisagra y una goma elástica que permite que esta se quede completamente fija para el transporte de un lado a otro de la cuna. La función de la puerta es que el circuito esté oculto y se mantenga accesible.

## 3.3 Código

### 3.3.1 Arduino

Para el control de los componentes electrónicos hemos utilizado una placa Arduino UNO programada utilizando su IDE. En términos de librerías, sólo hemos utilizado la librería para servomotores, cuyas funciones y métodos necesitábamos para poder controlar el motor que mece la cuna. Por indagar un poco en el código sin ponerse muy técnicos, comenzamos por declarar e inicializar todas las variables y constantes (para los pines) que necesita el código, además de las variables de tipo char que alojan los datos que son enviados desde el módulo de bluetooth a la placa Arduino y la variable estado, la cual recoge las instrucciones enviadas desde el teléfono y mediante un switch luego permite que la cuna realice lo que le es pedido.

Después de declarar e inicializar variables, en el void setup() indicamos el pinMode de todos los pines, iniciamos el puerto serie a 9600 baudios y indicamos el pin signal del servomotor.



```
//Hugo Prieto, Samuel Jiménez, Adrián Pérez y Mateo Orozco

// Servo - Version: Latest
#include <Servo.h> //Incluimos la librería Servo
Servo miservo; //Nombremos al Servo miservo

//Variables del entorno
float temperatura; //Declaramos la variable temperatura, de tipo float.
float volts = 0; //Declaramos e inicializamos la variable volts, utilizada para la lectura del sensor LM35.
float nivel_de_luz; //Declaramos la variable nivel_de_luz que guardará el porcentaje de luz.
bool movimiento; //Declaramos la variable boolean movimiento que indicará si el bebé está en movimiento.
float distancia; //Declaramos una primera variable distancia.
float tiempo; //Declaramos una primera variable tiempo.
float distancia2; //Declaramos una segunda variable distancia.
float tiempo2; //Declaramos una segunda variable tiempo.

//Constantes de los pines
const int ldr = 5; //A5 //Declaramos la cte ldr y la inicializamos al pin A5.
const int tpin = 4; //A4 //Declaramos la cte tpin, para el sensor LM35, y la inicializamos al pin A4.
const int leds = 13; //Declaramos la cte leds, y la inicializamos al pin 13.
const int calor = 12; //Declaramos la cte calor, para el sensor LM35, y la inicializamos al pin 12.
const int trig = 2; //Declaramos la cte trig, para el sensor ultrasónico, y la inicializamos al pin 2.
const int echo = 3; //Declaramos la cte echo, para el sensor ultrasónico, y la inicializamos al pin 3.

//Variables de bluetooth
char buffer[20]; //Declaramos la cadena de caracteres buffer, que es la que enviamos por bluetooth.
char tbuffer[8]; //Declaramos la cadena de caracteres tbuffer, que guarda el valor convertido de la medida de temperatura.
char lbuffer[8]; //Declaramos la cadena de caracteres lbuffer, que guarda el valor convertido de la medida de nivel_de_luz.
int estado; //Declaramos la variable estado que toma los valores que se envían desde la app móvil.

//Void Setup
void setup() {
  Serial.begin(9600); //Iniciamos el monitor serie a 9600 baudios.
  pinMode(leds, OUTPUT); //Indicamos que el pin 13, el de los leds, funciona como salida.
  pinMode(calor, OUTPUT); //Indicamos que el pin 12, el de la manta térmica, funciona como salida.
  pinMode(ldr, INPUT); //Indicamos que el pin A5, el del LDR, funciona como entrada.
  pinMode(tpin, INPUT); //Indicamos que el pin A4, el del LM35, funciona como entrada.
  pinMode(PinTrig, OUTPUT); //Indicamos que el pin 2, del trigger del ultrasonidos, funciona como salida.
  pinMode(PinEcho, INPUT); //Indicamos que el pin 3, del echo del ultrasonidos, funciona como entrada.
  miservo.attach(10); //Indicamos que el pin 10 es el pin signal del servomotor.
}
```

**Figura 4: Código 1**

La parte más extensa del código es el void loop(). Este está estructurado según las diferentes funciones que desempeña la cuna. En resumidas cuentas son:

1. Lectura del nivel de luz.
2. Lectura de la temperatura.
3. Lectura del sensor de ultrasonidos que detecta si el bebé se mueve por comparación de distancias separadas por un breve periodo de tiempo.
4. Envío de datos por el módulo de bluetooth hasta el dispositivo conectado.
5. Recepción de datos captados por el módulo de bluetooth (enviados por el móvil)
6. Switch que hace lectura de los valores enviados y ejecuta instrucciones de encendido, apagado o mecido de la cuna.
7. Encendido de la manta térmica si la temperatura es menor de 20°C.



```
//Void Loop
void loop() {

//Análisis del entorno -- nivel de luz
//La variable nivel de luz es igual al valor leído en el pin A5 dividido entre 10.
nivel_de_luz = (analogRead(1dr)/10);

//Análisis del entorno -- temperatura
//Código propio del sensor LM35 para el calculo de la temperatura en grados ºC.
volts = analogRead(tpin);
volts = 5 * volts;
volts = volts / 1023;
temperatura = volts; // volts/0.02

//Acción en el entorno
//Breve encendido y apagado del trigger para correcto funcionamiento.
digitalWrite(trig, LOW);
delayMicroseconds(4);
digitalWrite(trig, HIGH);
delayMicroseconds(10);
digitalWrite(trig, LOW);
//Cálculo de las distancias a partir del tiempo
tiempo = pulseIn(echo, HIGH)/2;
distancia = tiempo/29.2;
delay(2000);
tiempo2 = pulseIn(echo, HIGH)/2;
distancia2 = tiempo2/29.2;
//Comparación de las distancias. Si la diferencia es muy alta, activamos el mecido.
if (abs(distancia-distancia2)>7) {
  for (int i = 0; i < 5; i++) {
    miservo.write(180);
    delay(1000);
    miservo.write(0);
    delay(1000);
  }
}

//Envío de datos bluetooth
dtostrf(temperatura, 2, 2, tbuffer); //Convertimos a char el float temperatura.
dtostrf(nivel_de_luz, 2, 2, lbuffer); //Convertimos a char el float nivel_de_luz.
sprintf(buffer, "%s,%s", tbuffer, lbuffer); //El char buffer incluye a los anteriores 2 separados por una coma.
Serial.println(buffer); //Enviamos los datos de buffer por bluetooth.
delay(500); //Delay de 0,5 segundos.
}
```

Figura 5: Código 2

```
//Recepción de datos bluetooth
//Si hay datos en el monitor serie, estado guarda esos datos.
if(Serial.available() > 0) {
  estado = Serial.read();
}

//Recepción de datos + acción -- encender y apagar leds + mecer
digitalWrite(leds, LOW);
switch (estado) {
  //Si estado es A, encendemos los leds.
  case 'A':
    digitalWrite(leds, HIGH);
    break;
  //Si estado es B, apagamos los leds.
  case 'B':
    digitalWrite(leds, LOW);
    break;
  //Si estado es C, activamos el mecido.
  case 'C':
    for (int i = 0; i < 3; i++) {
      miservo.write(180);
      delay(1000);
      miservo.write(0);
      delay(1000);
    }
    break;
  default:
    break;
}

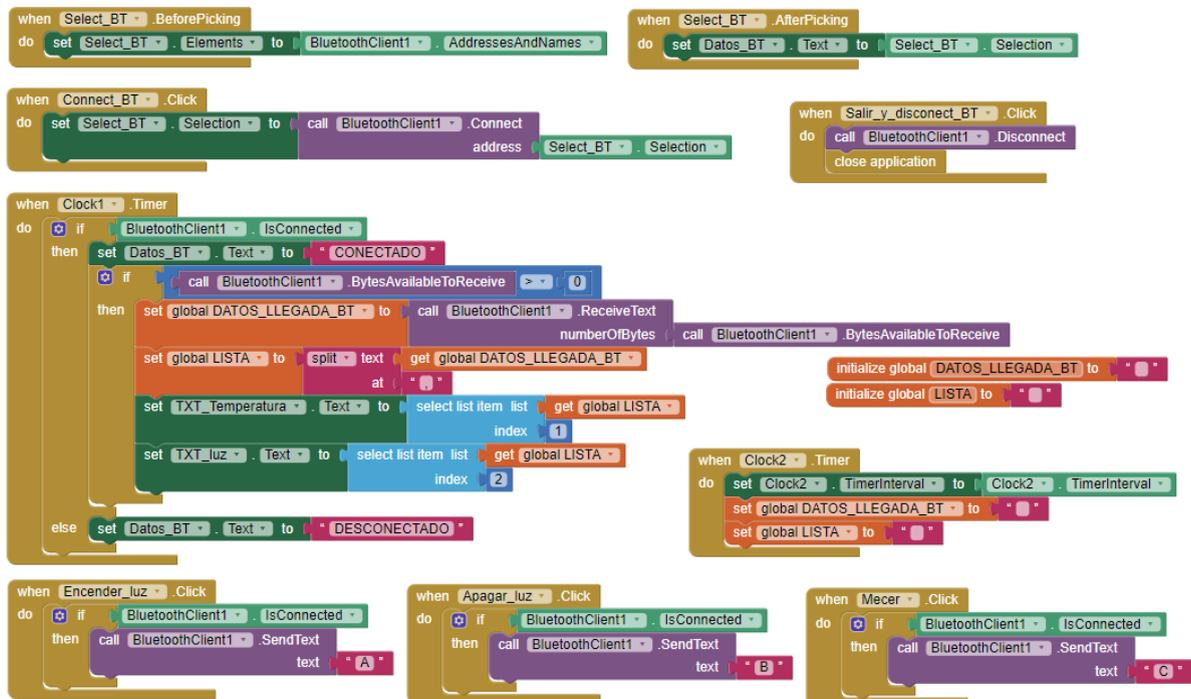
//Acción en el entorno -- temperatura
//Si la temperatura es menor a 20 ºC activamos la manta térmica; de lo contrario, la apagamos.
if (temperatura < 20) {
  digitalWrite(calor, HIGH);
} else {
  digitalWrite(calor, LOW);
}
}
```

Figura 6: Código 3

### 3.3.2 App móvil

La app móvil ha sido desarrollada utilizando la herramienta MIT App Inventor. Hemos desarrollado una APK que tiene capacidad de conectarse al módulo de

bluetooth, recibir datos de temperatura y luz que este le envía para mostrarlos en la interfaz de la aplicación y devolver instrucciones relativas al encendido y apagado de luz y el mecanismo de medido para accionarlos en remoto. La siguiente imagen muestra el código de bloques, que incluye la conexión y envío de datos por bluetooth:



```

when Select_BT.Timer.BeforePicking
do set Select_BT.Elements to BluetoothClient1.AddressesAndNames

when Select_BT.Timer.AfterPicking
do set Datos_BT.Text to Select_BT.Selection

when Connect_BT.Click
do set Select_BT.Selection to call BluetoothClient1.Connect
address Select_BT.Selection

when Salir_y_disconnect_BT.Click
do call BluetoothClient1.Disconnect
close application

when Clock1.Timer
do if BluetoothClient1.IsConnected
then set Datos_BT.Text to "CONECTADO"
if call BluetoothClient1.BytesAvailableToReceive > 0
then set global DATOS_LLEGADA_BT to call BluetoothClient1.ReceiveText
numberOfBytes call BluetoothClient1.BytesAvailableToReceive
set global LISTA to split text get global DATOS_LLEGADA_BT
at " "
set TXT_Temperatura.Text to select list item list get global LISTA
index 1
set TXT_luz.Text to select list item list get global LISTA
index 2
else set Datos_BT.Text to "DESCONECTADO"

initialize global DATOS_LLEGADA_BT to ""
initialize global LISTA to ""

when Clock2.Timer
do set Clock2.TimerInterval to Clock2.TimerInterval
set global DATOS_LLEGADA_BT to ""
set global LISTA to ""

when Encender_luz.Click
do if BluetoothClient1.IsConnected
then call BluetoothClient1.SendText
text "A"

when Apagar_luz.Click
do if BluetoothClient1.IsConnected
then call BluetoothClient1.SendText
text "B"

when Mecer.Click
do if BluetoothClient1.IsConnected
then call BluetoothClient1.SendText
text "C"

```

Figura 7: Código de AppInventor

### 3.4 Circuitos

La parte del circuito es probablemente la más sencilla del proyecto. El cerebro del circuito es, como ya hemos mencionado con anterioridad, una placa Arduino UNO. La mayor parte del circuito se organiza en torno a una placa protoboard o “breadboard”.

Dentro de la caja podemos encontrar el sensor de temperatura y el módulo de bluetooth, además de la salida del LDR que aprovecha un pequeño espacio entre la bisagra y el suelo de la caja para dar al exterior. También sale por un espacio entre bisagra y caja el cableado del sensor de ultrasonidos que detecta distancia y activa medido de forma automática.





## 4. Resultados

Los resultados del proyecto fueron los esperados. Logramos realizar una maqueta estética con los materiales y uniones planeados. El código y el circuito funcionan correctamente y la cuna es capaz de cumplir las funciones esperadas.

Acceso al vídeo explicativo:

<https://drive.google.com/file/d/1wwPmKEYx9E2hrI7xxVopYSr0kbLRmHH2/view?usp=sharing>

Acceso a los recursos multimedia:

<https://drive.google.com/drive/folders/1uNKTe2XeIQR8Stt4gO6JEnoORTg1wC-B?usp=sharing>

## 5. Conclusiones

El proyecto ha sido completado exitosamente pues cumple con las expectativas iniciales y tiene un acabado similar al esperado además de cumplir con las funciones esperadas. Como mejoras para una posible segunda construcción del prototipo estarían las mejoras en el acabado, que podríamos curvar para asemejarse más al acabado convencional de la cuna; la utilización de un mecanismo de cerrado de la puerta distinto al de la goma, que puede resultar poco cómodo si se requiere acceder a la caja con cierta frecuencia y, por último, el desarrollo de una mejor caja de cobertura para el motor, pues si bien la actual es bonita y cumple su función, se podría crear una que se asemeje más a la caja y deje al motor completamente oculto, pero accesible.



# Referencias

- <https://www.electronicwings.com/sensors-modules/lm35-temperature-sensor>
- <https://www.programmingelectronics.com/dtostrf/>
- <https://www.programmingelectronics.com/sprintf-arduino/>
- <https://www.arduino.cc/reference/en/>
- <https://www.rinconingenieril.es/medir-temperatura-con-sensor-lm35/#:~:text=Medir%20temperatura%20con%20Arduino%20y%20sensor%20LM35&text=El%20c%C3%B3digo%20es%20tan%20sencillo,0v%20representa%201%20grado%200Celsius.>

Acceso al vídeo explicativo:

<https://drive.google.com/file/d/1wwPmKEYx9E2hrI7xxVopYSr0kbLRmHH2/view?usp=sharing>

Acceso a los recursos multimedia:

<https://drive.google.com/drive/folders/1uNKTe2XeIQR8Stt4qO6JEnoORTg1wC-B?usp=sharing>