

# Construcción de un dispositivo inteligente de cuidado de bebés

Gomez, Ezequiel; Noseda Maximiliano; Radice, Adrian ; Ron, Lucas ; Terraza, Rodrigo

*DIIT - Universidad Nacional de la Matanza*

g.ezequiel90@gmail.com, radiceadrian@gmail.com, ronlucas86@gmail.com,  
nosedamaxi@gmail.com, rodxte@gmail.com

## Resumen

*En este presente documento se explicará el proceso de diseño y construcción de un dispositivo inteligente para el descanso y traslado de un bebé. El mismo puede obtener mediciones sobre el estado de ánimo del bebé y tratar de calmarlo con la ejecución de movimientos, y sonidos. Adicionalmente como consecuencia del análisis de estado de ánimo del bebé, se podrá obtener información relacionada al sueño y al estado cardíaco del bebé. Además, el dispositivo permite analizar volumen y temperatura del líquido en el biberón del infante, de manera no invasiva. También el dispositivo carrito va a poder desplazarse ejecutando rutas pregrabadas y evitando obstáculos mediante el análisis del entorno. El dispositivo se complementará con aplicación móvil, en la cual se puede cargar las rutas para que el carrito ejecute, acceder al streaming de video de la cuna, activar el mecedor si se desea, enviar sonidos al dispositivo cuna, y leer todas las métricas de los sensores.*

## Palabras Clave

Cuna, PIA, mecedora, machine learning, pytorch, streaming video, webrtc, detección llanto, ritmo cardíaco, sensor proximidad, móvil, firebase, volumen, temperatura, biberón, notificaciones.

## Introducción

PIA (Parent Intelligent Assistant) es nuestra solución integral al cuidado de los bebés. PIA es un nuevo concepto, un sistema que combina hardware y software, un carrito/cuna inteligente.

Como carrito será capaz de seguir nuestros movimientos y ejecutar recorridos prefijados. Como cuna será capaz de transmitirle al niño la calma mediante movimientos y la reproducción de distintos sonidos. PIA

interactúa con los responsables del niño (registrados en el sistema) mediante una aplicación móvil, en la cual podrán saber los estados de ánimo del niño, así como también ver imágenes o video capturadas del mismo por el dispositivo.

El objetivo principal de este proyecto consiste en asistir el descanso y traslado de los bebés. PIA pretende disminuir el esfuerzo físico requerido para empujar un carrito, y por otro lado también intenta disminuir el número de veces que deben interrumpir su descanso las personas al cuidado de un infante para atenderlo. Además, PIA informará estadísticas sobre el descanso y conducta del bebé.



Figura 1 Prototipo PIA

Con el objeto de una fácil comprensión del alcance, entenderemos al sistema PIA como tres subsistemas/módulos: la cuna, el cochecito y la aplicación móvil que interactúa con los dos anteriores.

## Módulo Cuna

El módulo de cuna está construido en fibrofacil por ser ligero y lo suficientemente resistente. La cuna se va a encontrar totalmente recubierta de gomaespuma y un cobertor de tela. Este módulo cuenta con una raspberry pi 3, en la cual ejecutarán los servicios de PIA, el servidor de

streaming y los controladores de los sensores activos de la cuna, cardiaco, de volumen, temperatura, micrófono y cámara.

La cuna también cuenta con un sistema mecedor construido en su totalidad. El sistema mecedor consta de un sistema de elevador de tijeras controlado por un motor paso a paso y una varilla rosada que actúa como actuador lineal.

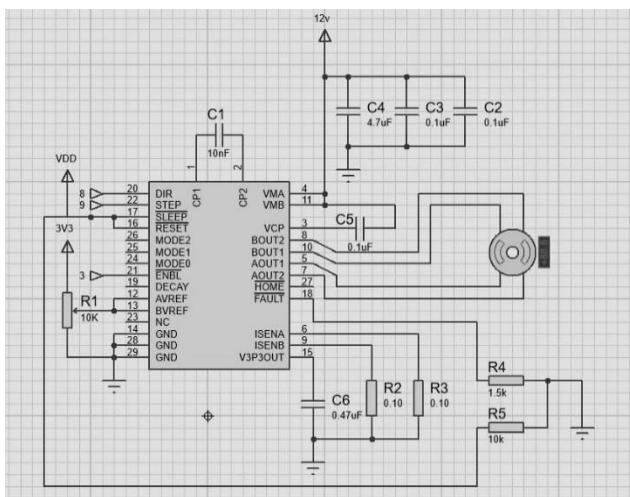


Figura 2 Circuito control motor mecedor

La idea es que cuando sea necesario mecer al bebé la cuna pueda ejecutar rutinas de movimientos para tranquilizar al niño. Estas rutinas son unas series de pasos precargadas dentro del módulo cuna. Estas rutinas se podrán disparar a demanda, mediante la aplicación móvil (la cual se explicará luego), o mediante el sistema de reconocimiento de llanto.

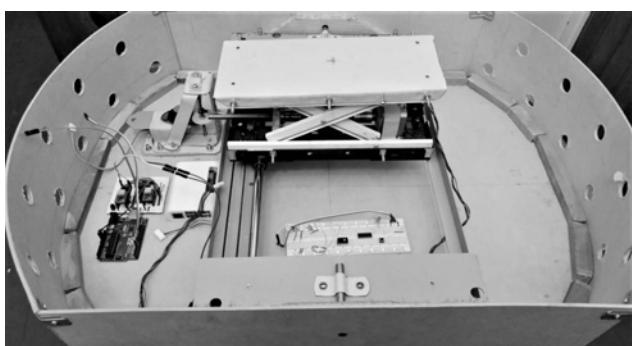


Figura 3 Dispositivo Mecedor

El sistema de reconocimiento de llanto hace uso del micrófono incluido en la cuna. El sonido del bebé es constantemente monitoreado y analizado mediante técnicas de machine learning. El algoritmo que utilizamos ha sido entrenado con numerosas muestras de sonidos de llantos de bebé y nos garantiza un reconocimiento del 85%

de los casos. Este software fue programado utilizando la librería PyTorch y entrenado mediante un gran número de muestras de llantos y luego compilado para que pueda correr en el hardware de PIA (Raspberry pi3).

Una vez que el sistema detecta el llanto, dispara la rutina para generar calma. Esto significa ejecutar la rutina en el dispositivo mecedor, así como reproducir sonidos estimulantes para el bebé, esto podría ser sonidos pregrabados de los padres, música o audiolibros infantiles.

### Detección de Llanto

Detectamos el sonido del ambiente mediante un micrófono para poder analizarlo y poder decidir si el bebé se encuentra llorando o no.

El audio captado se guarda cada 5 segundos en la memoria de la Raspberry Pi. Sobre el mismo se realiza un proceso para poder visualizar la información del audio en forma de imagen [5]. Este proceso consiste en:

- Eliminar las frecuencias inferiores a 1000 Hertz y superiores a 5000 Hertz, ya que el llanto de los bebés no se encuentra en este rango, mejorando la precisión del análisis.
- Generar una imagen del audio construyendo un espectrograma en escala de Mel [6], generando una imagen similar a la siguiente:

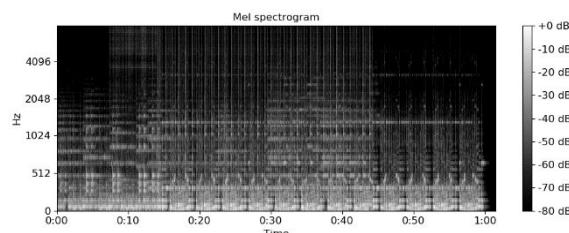


Figura 4 Escala de Mel

- El espectrograma anterior mide la amplitud al cuadrado (potencia). Convertimos el espectrograma a decibeles, generando el siguiente tipo de imagen:

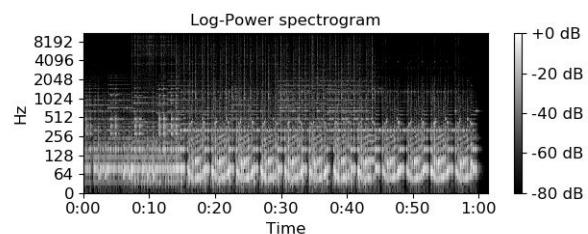


Figura 5 Espectrograma en decibeles

Este proceso nos permite generar una imagen clara para nuestro siguiente proceso: el entrenamiento de la red neuronal. Decidimos utilizar imágenes para analizar el audio para disminuir la cantidad de parámetros que debe

manejar la red, y porque en la actualidad, los algoritmos utilizados para el reconocimiento de imágenes están mucho más desarrollados que los de reconocimiento de audio.

Para poder entrenar la red, obtuvimos audios de bebés llorando de distintas fuentes [2-3], junto con diversos videos de YouTube que fuimos encontrando.

Como siguiente paso, pre-procesamos todos los audios obtenidos, transformándolos en imágenes útiles para el entrenamiento de la red neuronal. Algunos ejemplos de bebés llorando son:

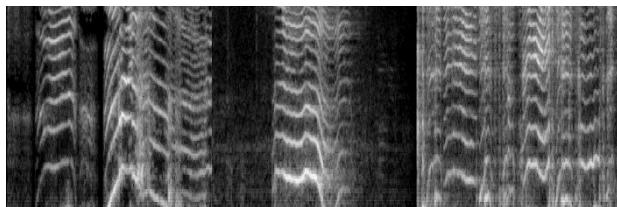


Figura 6 Llantos de bebés

En cambio, audios no pertenecientes a llantos de bebés pueden visualizarse de la siguiente manera:

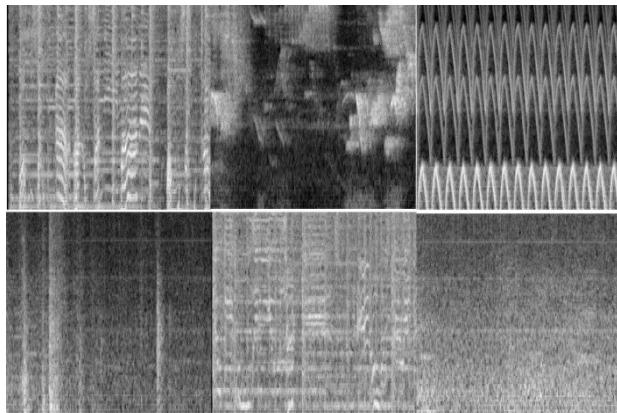


Figura 7 Distintos audios

Una vez preprocesados los diferentes audios ya podemos entrenar la red, separando el 70% de los audios en el set de entrenamiento, apartando un 15% para el set de prueba y el 15% restante para el set de evaluación.

Para la red neuronal utilizamos una red neuronal Convolucional [4], como mencionamos anteriormente, muy utilizada en el reconocimiento de imágenes. Se denomina convolucional porque las neuronas se entrena a través de convoluciones algebraicas, es decir, tomando no solo un valor específico de un píxel de la imagen, sino también procesando los pixeles vecinos.

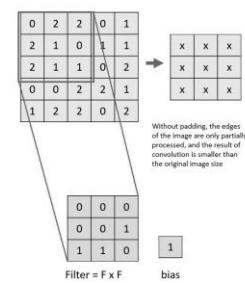


Figura 8 Tablas de convoluciones

Al entrenar la red, las diferentes convoluciones van generando distintos filtros en las distintas capas de la red, que detectan los bordes y formas del espectrograma que corresponde a los bebés.

Como tecnología para ejecutar el reconocimiento de los llantos utilizamos Pytorch, desarrollada por Facebook, con el lenguaje de programación Python.

Nuestra red neuronal tiene definidas unidades convolucionales, consistentes de:

- Una capa convolucional en dos dimensiones, con kernel de tamaño 3.
- Una capa de normalización en dos dimensiones para impedir inconsistencias en los pesos de las diferentes capas.
- La función de activación ReLU.

En cada unidad convolucional ingresan los datos de la imagen en las tres dimensiones de color (rojo, verde, azul).

La topología general de la red consiste en:

- Unidad convolucional (definida anteriormente).
- Pooling en dos dimensiones por el mayor valor (para disminuir el tamaño de la imagen antes de procesarla en la siguiente unidad).
- Unidad convolucional
- Pooling en dos dimensiones por el mayor valor.
- Unidad convolucional
- Pooling en dos dimensiones por valor promedio.

Estas capas se aplican secuencialmente hasta que finalmente, el resultado total es evaluado por una red neuronal lineal, que decide si el espectrograma pertenece o no a un llanto de bebé.

## Ritmo cardiaco

Para calcular el ritmo cardiaco usamos el sensor cardiaco (MAX30100), el cual se puede sujetar cómodamente del pie del infante. Este sensor toma una muestra de las pulsaciones del niño para tomar estadísticas e informar a la aplicación móvil de alguna irregularidad.

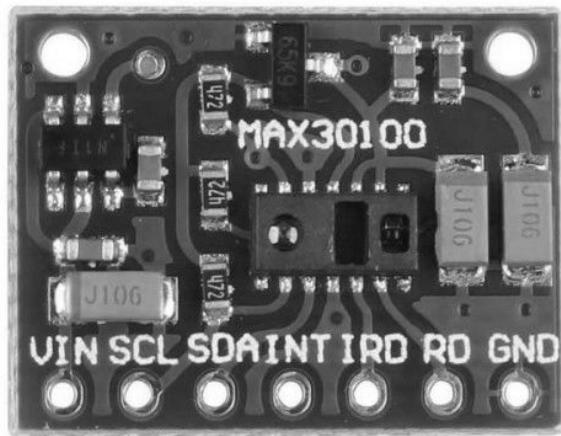


Figura 9 Integrado MAX30100

El integrado MAX30100 emite pulsos de luz infrarroja de 950nm, que se colocan sobre el dedo o sobre el lóbulo de la oreja, lugares donde dicha onda puede penetrar fácilmente.

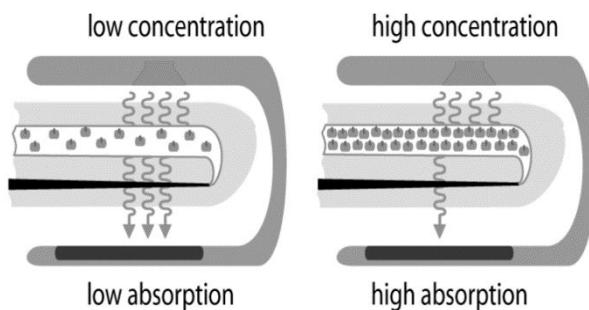


Figura 10 Ejemplo absorción de luz MAX30100

De esta manera se pueden medir los pulsos cardíacos, donde al haber una pulsación aumenta la cantidad de sangre oxigenada en el punto donde se realiza la medición. Midiendo las diferencias de reflexión de la luz infrarroja se pueden deducir las apariciones de las pulsaciones cardíacas.

Al tomar 10000 mediciones con el sensor sobre el dedo repetidas veces obtenemos el siguiente patrón:

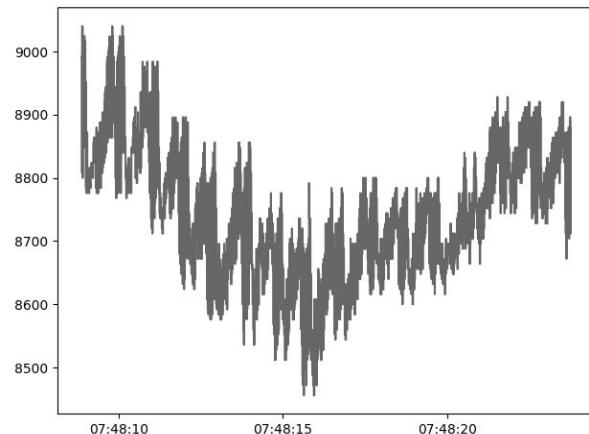


Figura 11 Ejemplo Medición

Este patrón no nos sirve para calcular el pulso cardíaco, sin embargo, se puede aplicar un filtro a la señal denominado DC Removal para poder obtener un resultado menos variante y centrado sobre el eje:

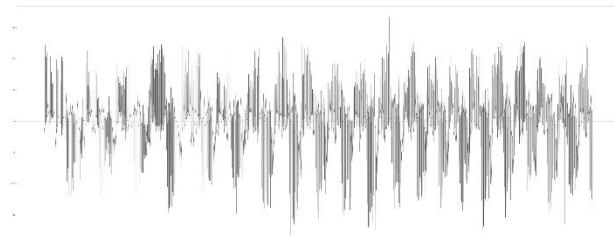


Figura 12 Resultado de aplicar el filtro DC Removal a la medición del MAX30100

Ahora el resultado es mucho más visible y se pueden deducir los momentos en los que hubo un pulso cardíaco. Para mejorar aún más la señal, utilizamos un algoritmo de filtrado de señales denominado Moving Average. Aplicándolo con una ventana de tamaño 150, obtenemos los siguientes resultados:

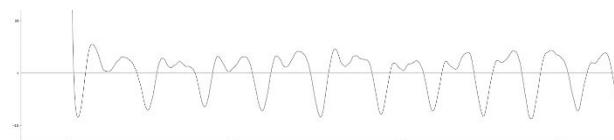


Figura 13 Resultado aplicar el filtro Moving Average.

Ya es muy visible la posición de los latidos, sin embargo, la medición es muy inestable habiendo muchos saltos entre cada medición. Para estabilizarla, tomamos el valor absoluto de la señal para eliminar los picos negativos, obteniendo el siguiente resultado:

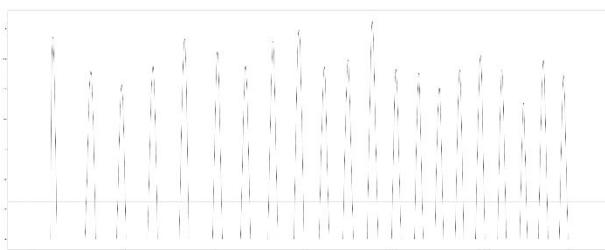


Figura 14 Resultado de tomar el valor absoluto

Como puede observarse, la computadora puede ahora detectar los flancos de subida y bajada de la función para obtener la cantidad de latidos durante la grabación.

La línea roja del medio corresponde a lo que será tomado como un latido si un pico lo sobrepasa.

Al realizar la función obtenemos una especie de función cuadrada que graficamos para visualizar como se detectan los latidos:

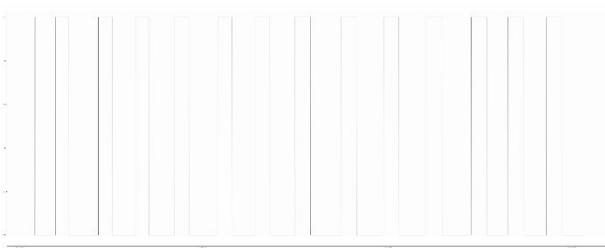


Figura 15 Señal de latidos

Así fácilmente pueden apreciarse la cantidad de latidos, y solo resta obtener la cantidad de latidos por segundo según el tamaño de la grabación, para luego obtener la cantidad por minuto.

El promedio de pulsaciones de un infante recién nacido es de 127 bpm (beats per minute) aumentando a un máximo de 145 bpm a un mes de edad y luego bajando a 113 bpm al llegar a los dos años [1].

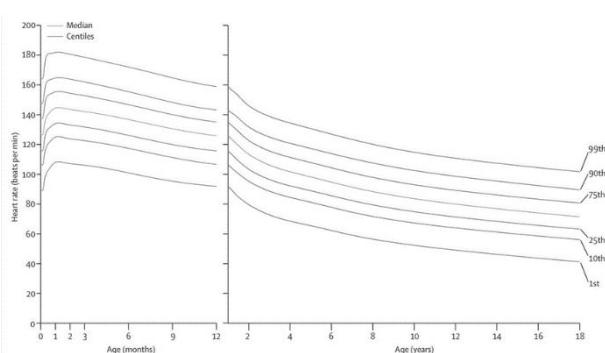


Figura 16 Frecuencia Cardiaca desde el nacimiento hasta los 18 años

Si las pulsaciones tienen reiteradas mediciones fuera del rango saludable promedio, una notificación será

enviada, mostrando los resultados y recomendando hacer una consulta con un profesional de la salud.

### Streaming de video de la cuna

La cuna también está equipada con una webcam integrada en la cuna para poder observar al bebé. Esto también se puede ejecutar bajo demanda desde la aplicación móvil. El sistema de streaming utiliza el protocolo WebRTC, pudiendo armar un canal directo mediante la aplicación móvil y la cuna.

El soporte del biberón está formado por sensor balanza (indicar que tipo de sensor) y un sensor de temperatura (indicar el tipo de sensor), estos sensores hacen un muestreo del peso y temperatura del líquido del biberón y son informados en el servidor interno de PIA. Luego estas novedades son distribuidas a la aplicación móvil para su correspondiente display.

### Módulo biberón

La cuna cuenta con un dispositivo con la capacidad de informar en la aplicación móvil el estado de la bebida contenida en el biberón.

Las variables a analizar el estado de la bebida son:

- Temperatura.
- Volumen restante.
- Indicar si aún es apta para el consumo. (Ejemplo si es un lácteo, evaluar si ya no paso mucho tiempo fuera de la heladera)
- Recomendar si él bebé debiera hidratarse.

El objetivo principal, fue realizar las mediciones mencionadas de una forma no invasiva. Se pretendía por ejemplo que, para medir la temperatura, no se tenga que sumergir un termómetro en la bebida. Por lo tanto, con este objetivo en mente se optó por desarrollar el dispositivo de la siguiente forma.

### Medición de temperatura

Se decidió medir la temperatura del líquido contenido por su emisión infrarroja, de esta forma no tendríamos que introducir peligrosos termómetros de mercurio en contacto con la bebida. Según la ley de Stefan-Boltzmann [7], todo objeto por encima del cero absoluto ( $^{\circ}$ K) emite radiación cuyo espectro es proporcional a su temperatura. Por lo tanto, se buscó un sensor que trabaja dicha propiedad. Se optó por El MLX90614 recoge esta radiación y su salida es una señal eléctrica proporcional a la temperatura de todos los objetos en su campo de visión.

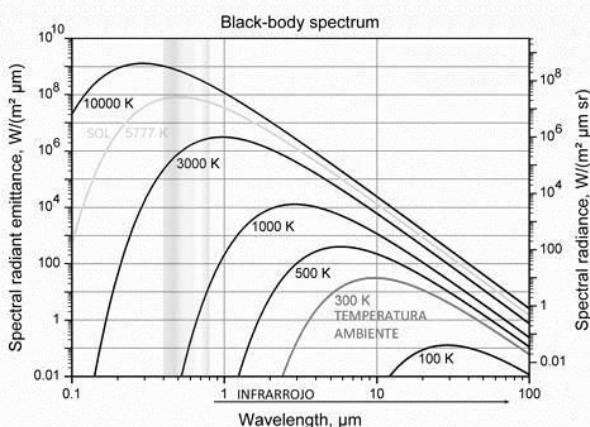


Figura 17 Diagrama de radiación lumínica

La precisión de los MLX90614 depende principalmente del rango de temperatura. Entre 0 °C y 50 °C de temperatura ambiente y 0 °C y 60 °C de temperatura objetivo, el margen de error es de  $\pm 0.5$  °C y aumenta progresivamente hasta llegar a los  $\pm 4$  °C en las condiciones más extremas: una temperatura ambiente superior a 100 °C y una temperatura objetivo superior a 120 °C.

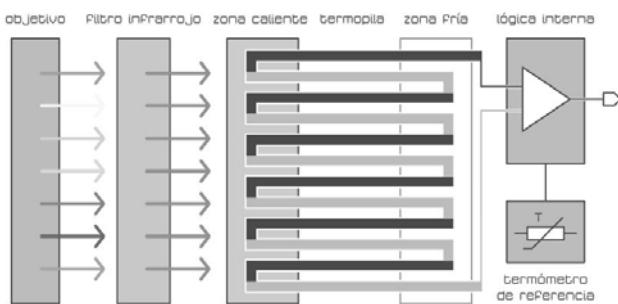


Figura 18 Diagrama general sensor temperatura

Las temperaturas más bajas producen desviaciones algo menores, hasta  $\pm 3$  °C por debajo de cero tanto en la temperatura ambiente como en la temperatura del objeto monitorizado.

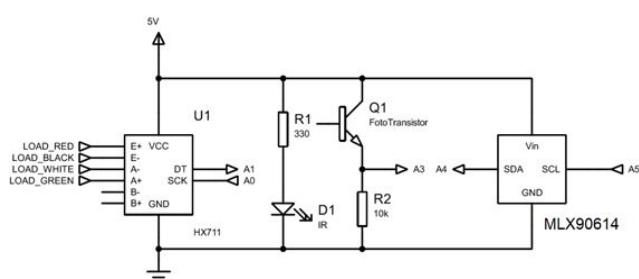


Figura 19 Circuito MLX90614

## Medición de volumen

Se decidió medir el volumen del líquido utilizando el peso de esta, para ello se registra el historial de mediciones de peso del biberón el cual será modificado después de cada consumo.

Para llevar a cabo esto se utilizó el Hx711 y una celda de carga de 5Kg.

El Hx711 es una interfaz entre las celdas de carga (transductor) y el microcontrolador, permitiendo poder leer el peso de manera sencilla. Internamente se encarga de la lectura del puente Wheatstone formado por la celda de carga, convirtiendo la lectura analógica a digital con su conversor A/D interno de 24 bits.

## Optimización de energía

Con el fin de optimizar el uso de las baterías se decidió que estos sensores funcionen de forma pasiva. Para lograr dicha forma de trabajo se colocó una barrera infrarroja que permitirá evaluar la presencia o no del biberón. De este modo, se realizará una única medición de utilidad ante la presencia del biberón en el contenedor.

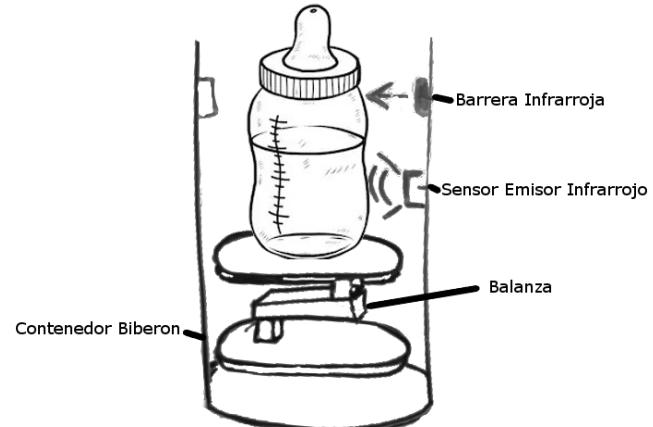


Figura 20 Estructura del módulo biberón

## Modulo Carrito

Este módulo es la base móvil de la cuna. Es el complemento de movilidad, este módulo cuenta con dos motores DC con sus correspondientes cajas reductoras, sensores ultrasónicos para detectar proximidad de objetos y una placa Arduino UNO para controlar la potencia de los motores. Este módulo va a tener dos modos: el modo sígueme y el modo de rutinas.

En el primer modo el carrito va a hacer uso de sus sensores de proximidad [8] para mantener una distancia constante con la persona que se encuentra detrás. De esta forma en vez de empujar el carrito con nuestras manos el carrito se moverá inteligentemente delante de nuestros pasos, disminuyendo el esfuerzo a cero y pudiendo realizar

otras actividades con nuestras manos, cuando nos detenemos el carrito se detendrá también.

En el segundo modo podremos guardar rutinas de desplazamiento y el carrito las podrá ejecutar para moverse dentro de espacios controlados como el hogar. El carrito seguirá la rutina pregrabada y utilizará sus sensores para poder esquivar obstáculos simples [9]. De no poder ejecutar la rutina informará mediante una notificación a la aplicación móvil.

El objetivo del segundo modo es que en caso de que la cuna determine que no puede tranquilizar al infante, al recibirse la alerta en la aplicación móvil, quien la consuma tenga la posibilidad de que se le aproxime el niño sin la necesidad de realizar un esfuerzo físico.

## Aplicación Móvil

Para poder utilizar todas las funcionalidades de PIA, desarrollamos una aplicación móvil Android. La aplicación cuenta con un login y un módulo para asociar a PIA con el usuario. La sincronización de la aplicación y el servidor PIA lo hacemos mediante la plataforma firebase de Google. Esto nos permite tener en tiempo real los cambios que ocurren en los sensores del servidor PIA y poder mostrarlos en la aplicación Android. (Imágenes de la aplicación)

La aplicación móvil fue diseñada de forma que sea sencilla de comprender, fácil de utilizar, de aprender, e intuitiva para el usuario. Este podrá identificar y utilizar rápidamente las funcionalidades provistas por la aplicación y entender fácilmente la información que esta brinda

A continuación, se desarrollarán las diferentes secciones, funciones y datos provistos por la aplicación:

### Automatización

Desde la aplicación de PIA se podrá controlar la automatización de acciones, es decir activar o desactivar el modo automático, donde la cuna se encargará de calmar al bebé ante la detección de llanto e incomodidad de este. En el caso de que el bebé siga llorando aparecerá una notificación en la aplicación y mediante una botonera permitirá al usuario seleccionar las opciones de “intentar calmar de nuevo al bebé”, “hacerse cargo” (cancela el calmado automático) y por último “acercar cuna”, que ordenara al carrito de PIA la acción de acercarse a quien posee la aplicación móvil.

### Reproducción de Audios

Además de esto se posee una sección dedicada al control de audios de la cuna. Está permite seleccionar desde una biblioteca de sonidos precargadas, audiolibros, canciones, y audios grabados. Adicionalmente se incluye un botón del tipo “Presionar para Hablar” que permite el

envío de audios grabados a través del micrófono del dispositivo móvil.

### Mecedora

Se incluye una sección de “Mecedora”, donde manualmente a través de botones del tipo switch se podrán seleccionar la forma de mecer deseada: Horizontal, Vertical, Oblicua. La aplicación permite activar estos switches de forma individual, y en paralelo, permitiendo ejecutar todos los tipos de movimientos a la vez o combinaciones deseadas.

### Streaming de Video

Se incluye un reproductor de streaming de en el que se puede visualizar constantemente las imágenes capturadas por la cámara presente en la cuna de PIA.

### Indicadores

Se podrán observar los distintos indicadores medidos por PIA, a través de iconos representativos con sus respectivos valores numéricos de forma que el usuario pueda identificarlos fácilmente. Los indicadores son los de ritmo cardiaco, grado de felicidad del niño, temperatura del biberón (expresada en caliente, tibia, fría), nivel de líquido(volumen) de biberón.

### Alertas

La aplicación se encarga de informar alertas en el caso de la detección de algún evento como lo puede ser el llanto o la presencia de alguna anomalía en el ritmo cardíaco

### Informes

Se incluye en la aplicación la posibilidad de consultar informes y estadísticas sobre el descanso del bebé

## Conclusión

Construir PIA es un proyecto que involucra muchos conocimientos técnicos de distintas áreas. Es un proyecto con una gran complejidad de hardware, software y comunicación de datos entre múltiples tipos de sensores y clientes. Hemos ganado conocimiento en el área de machine learning y algoritmos genéticos aplicándolos en la detección de llanto y reconocimiento de frecuencia cardíacas. En la construcción del carrito fue necesario tener conocimiento en potencia y torque de motores para garantizar que los movimientos de este sean suaves pero firmes. Cabe destacar que lo que buscamos fue construir un prototipo con la mayoría de las funcionalidades de un dispositivo profesional, siempre acotado en nuestro tiempo de desarrollo y el costo de los materiales.

## Referencias

[1] Normal ranges of heart rate and respiratory rate in children

- from birth to 18 years of age: a systematic review of observational studies, Susannah Fleming, Matthew Thompson, Richard Stevens, Carl Heneghan, Annette Plüddemann, Ian Maconochie, Lionel Tarassenko, David Mant
- [2] Donateacry-corpus An infant cry audio corpus that's being built through the Donate-a-cry campaign -  
<https://github.com/gveres/donateacry-corpus>
- [3] Dataset - The AudioSet dataset is a large-scale collection of human-labeled 10-second sound clips drawn from YouTube videos.  
[https://research.google.com/audioset/dataset/baby\\_cry\\_infant\\_cry.html](https://research.google.com/audioset/dataset/baby_cry_infant_cry.html)
- [4] ImageNet Classification with Deep Convolutional Neural Networks, Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton.  
<http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [5] An Automatic Infants Cry Detection Using Linear Frequency Cepstrum Coefficients (LFCC), Bhagatpatil Varsharani V,V.M.Sardar.  
<http://www.ijtee.org/final-print/feb2015/An-Automatic-Infants-Cry-Detection-Using-Linear-Frequency-Cepstrum-Coefficientslfcc.pdf>
- [6] Mel scale,  
[https://en.wikipedia.org/wiki/Mel-frequency\\_cepstrum](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum)
- [7] Stefan–Boltzmann law,  
[https://en.wikipedia.org/wiki/Stefan%E2%80%93Boltzmann\\_law](https://en.wikipedia.org/wiki/Stefan%E2%80%93Boltzmann_law)
- [8] Raspberry Pi Distance Sensor: How to setup the HC-SR04  
<https://pimylifeup.com/raspberry-pi-distance-sensor/>
- [9] Design and Implementation of Autonomous Car using Raspberry Pi, Gurjashan SinghPannu, MohammadDawud Ansari, Pritha Gupta.  
<https://pdfs.semanticscholar.org/51c0/05ce566e48e95032e690f90938bbcab6ddab.pdf>