

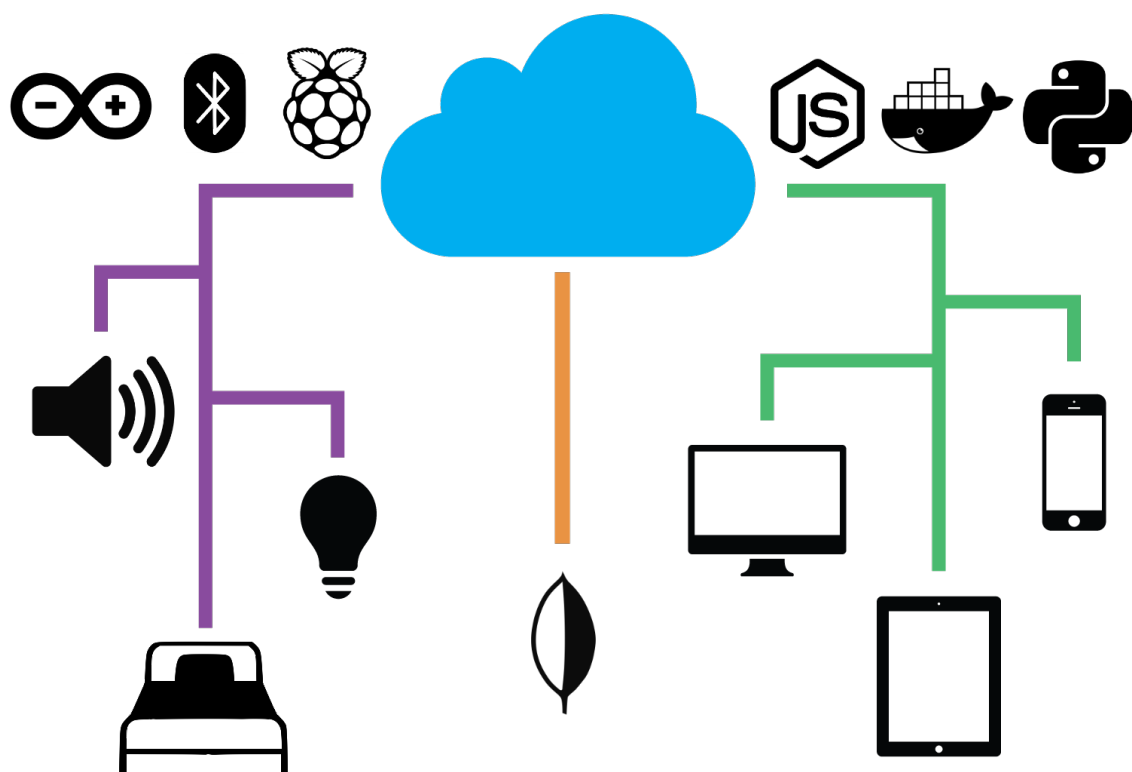
Computación ubicua: Cama inteligente

Juan Casado Ballesteros

Álvaro de las Heras Fernández

Javier Fuentes Fernández

Daniel Mencía Berlinches



Índice

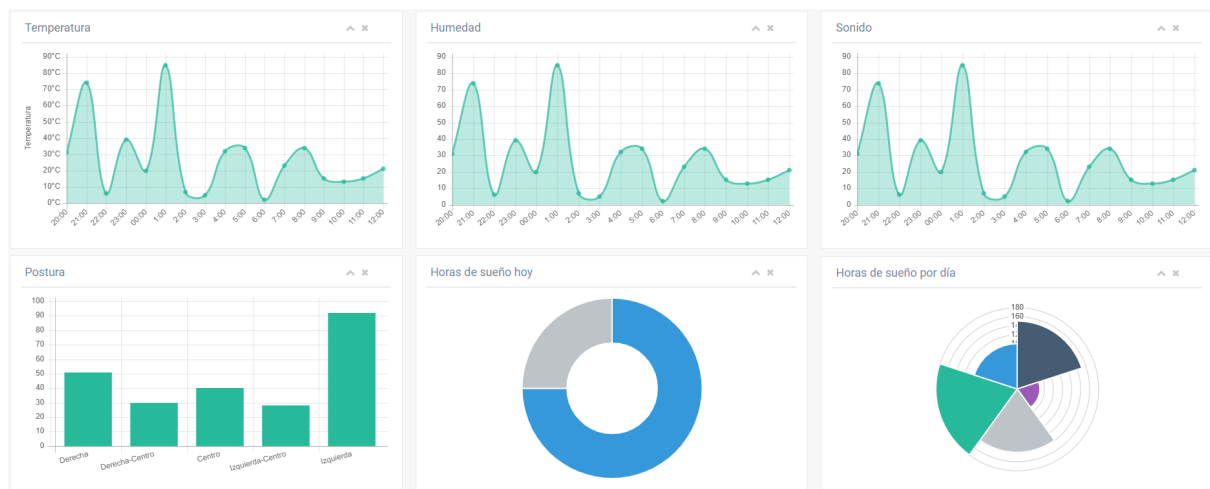
1. Introducción	5
2.Contexto	6
3. Misión y alcance del proyecto seleccionado	8
4. Descripción de ideas descartadas	9
5. Tecnología a utilizar	10
5.1. Obtención de la información	10
5.1.1 Placas controladoras	10
5.1.1.1 Arduino uno	11
5.1.1.2 Arduino due	11
5.1.2 Sensores	12
5.1.2.1 Sensores de flexibilidad	12
5.1.2.2 Sensor de peso	13
5.1.2.3 Sensores de sonido	14
5.1.2.4 Sensores de luz	14
5.1.2.5 Sensor de humedad y temperatura	15
5.1.3 Acceso a internet	15
5.1.3.1 Raspberry PI	15
5.2 Transmisión de la información	16
5.2.1 Sensores -> Controladores	16
5.2.1 Controladores -> Raspberry PI	16
5.2.2 Comunicación con la base de datos	17
5.2.3 Front End -> Usuario	17
5.3 Almacenamiento permanente	17
5.4 Interpretación de la información	18
5.5 Mostrar la información al usuario	19
6. Metodología de desarrollo	21
6.1 Scrum	21

6.2 Kanban	22
6.3 Metodologías pesadas	22
6.4 Adaptación de la metodología a nuestro caso concreto	23
6.4.1 Prueba y aprendizaje	23
6.4.2 Diseño	23
6.4.3 Ejecución	23
6.4.4 Pruebas	23
6.4.5 Reuniones	23
6.4.6 Sprints	23
7. Arquitectura de la aplicación	24
8. Modelo de negocio	25
8.1. Business Model Canvas	25
8.1.1. Propuesta de valor	25
8.1.2 .Segmentos de Cliente	25
8.1.3. Canales	26
8.1.4. Relaciones con clientes	27
8.1.5. Fuentes de ingresos	27
8.1.6. Recursos Clave	28
8.1.7. Actividades clave	28
8.1.8. Socios Clave	29
8.1.9. Estructura de costes	29
9. Plan de marketing	31
9.1. Análisis de la situación actual	31
9.2. Análisis de la competencia	32
9.3. Objetivos	33
9.4. Plan de actuación: estrategias de marketing	34
9.5. Revisión del plan	34
10. Planificación temporal, plan de desarrollo, riesgos, plan de contingencias	35
11. Resumen y conclusiones	36
12. Bibliografía	36

1. Introducción

Se nos asignó el proyecto de realizar una cama inteligente que sea capaz de como mínimo detectar cuántas horas pasamos tumbados y así poder mejorar nuestros hábitos de sueño. Para ello utilizaremos un conjunto de sensores y controladores que se podrán instalar en cualquier cama puesto que el producto será un conjunto de sensores y controladores de pequeño tamaño.

Este producto recogerá la información de la sesión de sueño del usuario y este podrá consultarla mediante la página web del producto. Aquí podrá ver sus datos de forma intuitiva mediante un conjunto de gráficas formadas por cada uno de los datos que se recogen como horas de sueño, temperatura, humedad, sonido y posición en la cama.



2.Contexto

Para realizar el proyecto realizamos una primera búsqueda para conocer el mercado actual de este tipo de tecnologías.

Descubrimos que existen elementos similares, que detectan la ocupación de una cama, pero con un enfoque al cuidado de personas y su supervisión.



En el apartado de sistemas inteligentes encontramos algunas startups o incluso una patente de Apple que se asimila más a la idea que tenemos para el proyecto, una tela que analiza los movimientos que realiza la persona por la noche.



Por último cabe destacar el aumento en el uso de tecnologías “Wearable” como smartbands y smartwatches que realizan análisis del sueño basándose en los movimientos y pulsaciones del usuario. Estos aparatos tienen un gran éxito debido a sus múltiples funciones y en ocasiones su bajo precio.



También hay aplicaciones que están enlazadas con estos wearable y que nos proporcionan información sobre la calidad de sueño. Algunas son incluso de código abierto como:

- <https://github.com/Laity000/SmartBed-Workerman-AngularJS>
- <https://github.com/selfhostedhome/smart-bed-sensor/tree/master/src>
- <https://github.com/aog0036/TFG-SmartBeds>
- <https://github.com/mihaanali/BedSmart>
- <https://github.com/lgo2356/SmartBed>
- <https://github.com/ThornPan/SmartBed>

3. Misión y alcance del proyecto seleccionado

Nuestro objetivo con este proyecto es realizar un producto de bajo coste, fácil instalación y de carácter universal, es decir que se pueda utilizar en cualquier tipo de cama. A su vez queremos que sea intuitivo a la hora de utilizar tanto el producto como la aplicación y sobretodo que ayude al usuario a conocer y mejorar sus hábitos de sueño permitiéndole detectar posibles causas externas que impiden un correcto descanso.

Para ello trataremos de recoger los datos que creemos que pueden tener un mayor impacto sobre la calidad del descanso:

- Ruido: se ha demostrado que incluso ruidos de baja intensidad provocan una mayor actividad cerebral, causando lo que se conoce como micro despertares.
- Temperatura: una temperatura extrema ya sea baja o alta puede llegar a imposibilitar el correcto descanso.
- Humedad: un ambiente muy seco o muy húmedo puede causar problemas respiratorios dando lugar a ronquidos y apneas del sueño.
- Postura: este dato lo recogeremos para saber si el usuario cambia de posición durante la noche lo que podría indicar que el sueño no es profundo y puede estar siendo provocado por algún factor externo.
- Tiempos: nos permitirá saber las horas que pasa el usuario en la cama y así poder comparar los los agentes externos y ver el efecto que estos tienen.

Todos estos datos captados por una serie de sensores de fácil instalación que se explican más adelante. Esto permite que un usuario pueda acoplar el producto a cualquier cama, ya que contará con una tira de sensores de flexibilidad que irá sobre el colchón de forma horizontal, cuatro sensores que se situarán debajo en cada una de las esquinas, y una pequeña caja con el resto de sensores de factores externos que deberá quedar fuera de la cama y al descubierto. Será necesaria una conexión a internet para enviar todos los datos al servidor durante la noche.

En un futuro se debería implementar un modelo de aplicación basado en cuentas personales, de forma que cada usuario disponga de un Usuario y Contraseña únicos y de esta manera solo él pueda ver sus datos. Además podría vincular varias camas a una misma cuenta lo cual puede ser útil para aquellos usuarios que tengan más de un domicilio.

4. Descripción de ideas descartadas

Utilizar sensores de temperatura para detectar al usuario en la cama. descartado debido a la incapacidad de estos sensores de detectar la temperatura del cuerpo humano a través de tejido textil.

Utilizar una matriz de sensores de presión para localizar al usuario dentro de la cama. descartado debido a que consideramos más eficiente utilizar una sola tira de sensores colocada horizontalmente en la cama, esto os daría el resultado necesario a un coste considerablemente mayor ya que por ejemplo se reduce de una matriz de 5x5 a una tira de 5 sensores.

Utilizar el protocolo HTTPS para comunicar los elementos entre sí, en especial el acceso de los usuarios a la página web. Este protocolo requiere que el servidor al que nos queremos conectar tenga un nombre de dominio, adicionalmente deberemos obtener un certificado SSL enlazado con una agente autenticador. Obtener tanto el nombre de dominio como el certificado puede hacerse sin coste alguno aunque lo habitual es que las empresas paguen por ello. No obstante esta idea se descartó tanto por su complejidad como por la falta de necesidad de tales medidas de seguridad en un prototipo. En su lugar se utilizará cifrado de clave privada más sencillo de implementar.

Inicialmente planteamos construir la base de datos con Postgres pero finalmente la hemos hecho sobre MongoDB debido a que tenemos mayor experiencia y nos resultó más sencillo.

Una de las primeras propuestas fue llevar una colchoneta hinchable al taller pero al tratarse de un prototipo decidimos optar por algo más pequeño y manejable y descartamos la idea.

5. Tecnología a utilizar

Para llevar a cabo el proyecto utilizaremos un amplio abanico de tecnologías.

Esto es debido a que IOT es una ciencia multidisciplinar por lo que aunque solo se desee realizar un pequeño prototipo hacerlo supone trabajar en ramas muy diversas dentro de las ciencias de la computación.

Como mínimo para un proyecto genérico será necesario realizar las siguientes tareas:

- Obtener información de distintos sensores o dispositivos.
- Filtrar y procesar dicha información de para evitar ruido en las medidas.
- Almacenar la información de forma segura y accesible en escritura por dispositivos que la generan y en lectura por los que la deban mostrar al usuario.
- Interpretar y procesar dicha información para inferir otra nueva generando conocimiento adicional.
- Mostrar al usuario la información de forma inteligible y accesible.
- Transportar la información de un lugar a otro de forma rápida y segura.

Detallamos a continuación las tecnologías empleadas para realizar cada una de las tareas mencionadas. Las relaciones entre las tareas las expondremos en el apartado de arquitectura.

5.1. Obtención de la información

Esta es la parte de más bajo nivel del proyecto. En ella programaremos para hardware específico sin virtualización en C++ para los controladores y en Python para la Raspberry que actúa como intermediario entre los controladores y la red.

Para llevarla a cabo estas tareas ha sido necesario montar pequeños circuitos que nos permitieran alimentar tanto las placas controladoras como los sensores, además de conectar estos entre sí de modo adecuado. Los circuitos se han montado sobre placas de prototipado de tamaño reducido con la intención de utilizar una placa para cada uno de los sensores y del hardware adicional que cada sensor necesite.

Se ha decidido utilizar esta combinación de placas controladoras arduino con una RaspberryPi como intermediario para abaratar costes. Varios miembros del equipo tenían ya estos dispositivos de modo que se decidió reutilizarlos para crear el prototipo. Si esto no hubiera sido así probablemente habiéramos acabado utilizando los controladores ESP32 ya que con un hardware único habiéramos podido leer los sensores y enviar las medidas captadas por internet directamente.

5.1.1 Placas controladoras

Hemos utilizado dos placas controladoras cuya misión es la de leer los datos de los sensores. Debido a la limitación de entradas analógicas de las placas y la gran cantidad de sensores analógicos que hemos utilizado ha sido necesario utilizar dos.

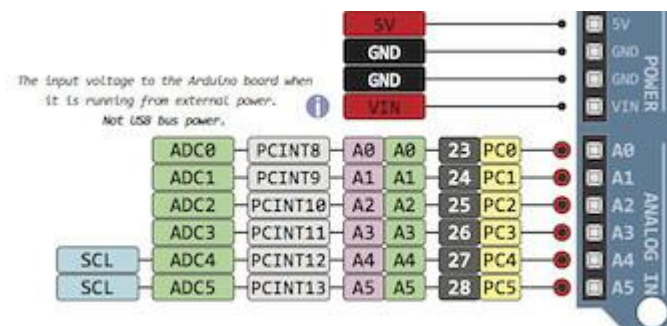
Ambas placas las alimentamos a 5v desde la entrada VIN que se salta la seguridad de la alimentación que las placas contiene. Esto se realiza así pues dicha seguridad ya es proporcionada por una pequeña fuente de alimentación que es la encargada de suministrar energía a todo el circuito. Adicionalmente hacerlo reduce la intensidad que requieren las placas para funcionar permitiendo a la fuente proporcionar intensidad suficiente como para poder alimentar al resto de sensores. Los sensores como hemos dicho se alimentan desde la fuente y no desde la placas arduino, esto se debe a que estas tienen limitada la suma de las intensidades de salida de sus pines a 1A lo cual no es suficiente como para alimentar todos los sensores que utilizamos de forma adecuada.

La única precaución que es necesaria tomar con esta arquitectura reside en que la referencia a tierra de todos los elementos en el circuito debe ser común.

5.1.1.1 Arduino uno

A ella se conectan la mayoría de los sensores. Todos excepto los sensores de flexibilidad que ya no se podían poner debido a la limitación en el número de entradas analógicas; solo tiene seis.

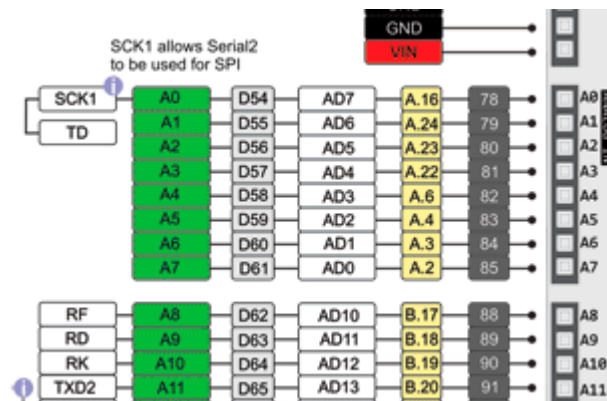
Por simplicidad y modularidad el esquema utilizado para conectar los sensores será incluido junto a cada sensor y no en cada placa.



5.1.1.2 Arduino due

En esta placa conectaremos el resto de sensores de flexibilidad que no podíamos tener en la placa principal por limitación de entradas analógicas.

Esta placa se ha establecido como placa secundaria a pesar de tener una capacidad de cómputo que la anterior de forma provisional. Puede que esto en un futuro cambie. En las primeras pruebas no estamos utilizando todos los sensores de flexibilidad de modo que usar esta placa todavía no ha sido necesaria. Adicionalmente es una placa de mayor tamaño por lo que es más engorrosa de manejar y transportar que la anterior.



5.1.2 Sensores

Hemos utilizado una variada gama de sensores con el fin de recoger la mayor cantidad de información del entorno posible.

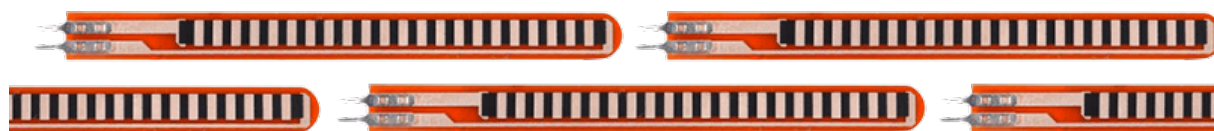
Debido a que estamos construyendo un prototipo hemos decidido hacer esto para poder descubrir qué sensores nos proporcionan mayor información sobre los hábitos de sueño de los usuarios. Posteriormente dependiendo de la calidad de información que cada sensor proporcione podríamos decidir si incluirlo en el producto final o no.

Aunque no se detalla para cada sensor, todas las medidas capturadas son filtradas mediante un filtro pasabajos, un pasaaltos y otro de media con distinto tamaño dependiendo del que se ha observado como el que mejor funcionaba para cada sensor.

5.1.2.1 Sensores de flexibilidad

Estos sensores nos proporcionan la información de cuan doblados están. Dicha información de forma aislada puede resultar vana y poco informativa, no obstante debido a que utilizamos una gran cantidad de sensores en una disposición muy concreta podemos inferir datos de gran utilidad.

Utilizando estos sensores hemos construido un detector de posición que nos permite conocer la posición exacta del usuario cuando está tumbado en la cama. Para ello los hemos dispuesto en dos líneas paralelas de modo que en todo momento el centro de alguno de los sensores quede doblado si alguien se tumbara encima de ellos.

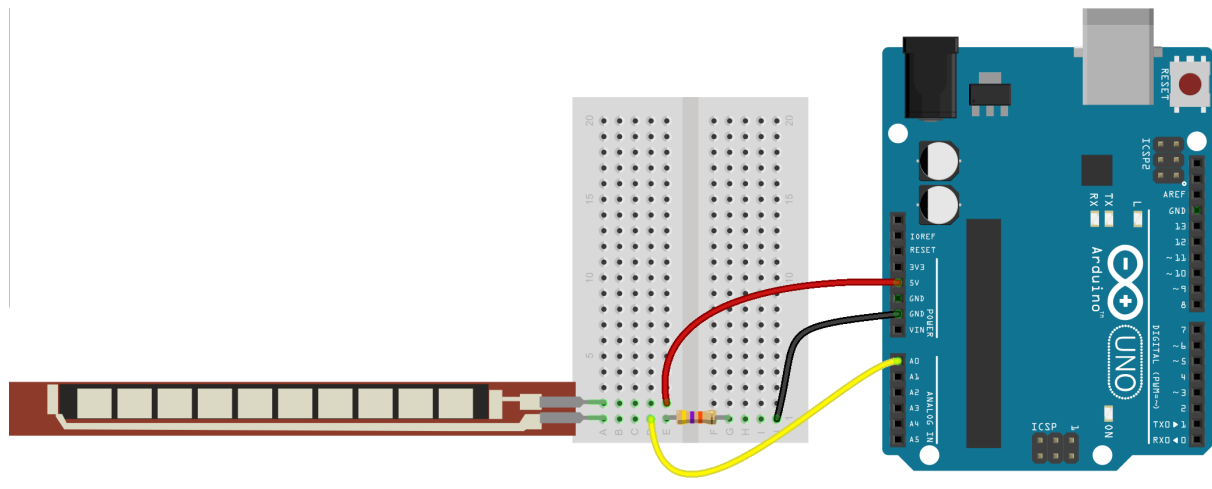


Realizando un seguimiento de esta información somos capaces de detectar movimientos del usuario a partir de los cuales (junto a otros datos) determinamos la calidad del sueño de este. Cuando un sensor esté doblado podremos ubicar un lateral del cuerpo del usuario. Si la posición del cuerpo cambia rápidamente en el tiempo podremos decir que las probabilidades de que la persona esté despierta son altas. Adicionalmente si los

movimientos son lentos aumentarán las probabilidades de que la persona se esté moviendo en sueños.

La implementación de este sensor conjunto no es directa si no que pasa por un proceso de filtrado y amplificación de la señal. La longitud de la cama se discretiza en el doble de segmentos de la cantidad de sensores de que disponemos. Cuando dos sensores próximos indican que están doblados podemos identificar que el lateral del cuerpo del usuario realmente entre ellos en vez de sobre ambos a la vez. Para realizar esto utilizaremos un filtro bayesiano configurado para proporcionar apertura sintética a las medidas.

Tratando la función discreta que generamos somos capaces de dibujar una silueta precisa del cuerpo usuario sobre el sensor.



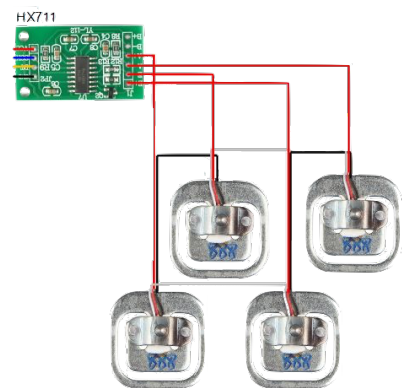
5.1.2.2 Sensor de peso

El uso principal de este sensor es detectar cuando un usuario está encima de la cama.

Gracias a esa información podemos saber el tiempo fuera del horario nocturnos que el usuario pasa tumbado o sentado en la cama. Adicionalmente podremos tener esa misma información a lo largo de la noche para conocer el tiempo de sueño.

Como una característica adicional intentaremos medir la cantidad de peso se ha perdido durante la noche. De media perdemos 1kg principalmente por pérdida de agua.

Estos sensores no son utilizados de forma directa si no a través de un HX711, un convertidor analógico digital diseñado para ser utilizado con sensores de peso. Los registros de este integrado se pueden rellenar directamente mediante la entrada digital que nos proporciona como interfaz de comunicación. Uno de los registros actúa como offset permitiéndonos desplazar en el eje de ordenadas (vertical, peso) las medidas que emitirá a lo largo del eje de abscisas (horizontal, tiempo). El otro



registro que nos expone nos permite establecer una ganancia o factor de escalado por el que las medidas se multiplican antes de ser enviadas.

$\text{medida} = \text{toDigital}(\text{medida de los sensores}, 24) * \text{ganancia} + \text{offset}$

Desde las placas controladoras leeremos este amplificador utilizando una entrada digital para acceder a la información del peso y una salida digital para generar una señal cuadrada que al ritmo de la cual el amplificador nos envía bit a bit la información que contiene.

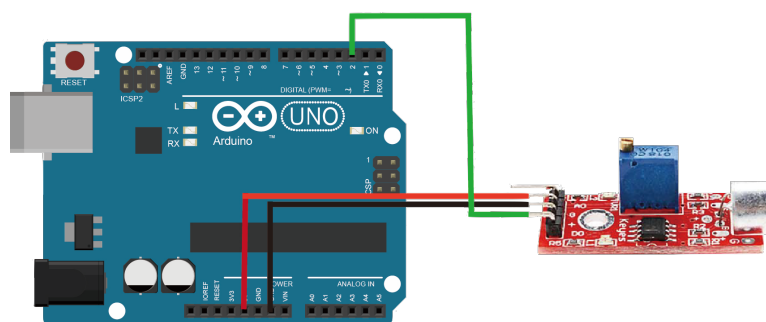
Utilizar este DAC es de gran importancia. Los sensores podrían leerse directamente desde la placa arduino utilizando 4 entradas analógicas, pero en ese caso seríamos nosotros los encargados de filtrar y procesar la señal. Adicionalmente perderíamos precisión en la medida pues el DAC de los controladores Arduino es de 10 bit en las uno y 12 bit en las due (mínimo y máximo que encontramos en estos controladores).

5.1.2.3 Sensores de sonido

Utilizaremos los sensores de sonido para medir el ruido ambiente con la intención de detectar cuando la calidad del sueño del usuario baja. El sensor utilizado es un KY-038.

Para leer estos sensores solo será necesario utilizar una entrada digital que nos indicará cuando el nivel de ruido supera cierto umbral. La principal dificultad para utilizar este sensor consiste en filtrar cuando un nivel de ruido alto tiene sentido para nuestro propósito. Para establecer el umbral debemos ajustar correctamente el potenciómetro incluido en el sensor. Opcionalmente se podría utilizar la configuración de la figura en la que en vez de leer la salida digital del sensor leemos la analógica. En este caso el umbral se establecería en código.

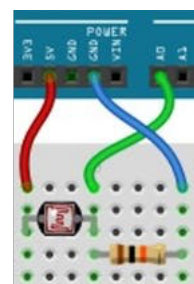
La información proporcionada es de gran calidad pudiendo detectar ronquidos u otros ruidos nocturnos aunque antes de poder utilizarla debemos de habernos asegurado de que el usuario se ha dormido para evitar confundir lo percibido con que el usuario esté hablando.



5.1.2.4 Sensores de luz

Utilizamos estos sensores para detectar la cantidad de luz ambiente con la que el usuario está durmiendo. En base a ella podremos recomendar que procure dormir con una cantidad de luz si fuera muy alta.

El sensor elegido para hacer esto es una LDR que leeremos con una entrada analógica.

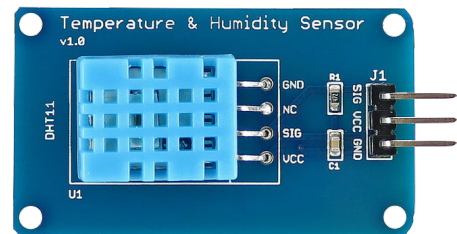


5.1.2.5 Sensor de humedad y temperatura

Hemos adquirido estos dos sensores en un único empaquetado por simplicidad y para reducir costes ya que nos resultaba más barato que comprarlos por separado.

Deseamos adquirir estas medidas con la intención de proporcionar un nivel de calidad al entorno de modo que podamos indicar al usuario si su entorno de sueño es adecuado o no. Un entorno puede ser malo por ser demasiado frío o húmedo o demasiado seco o caliente.

El sensor utilizado es una variante del DHT11 en el que ya se incorporan resistencias para pull-up y un condensador de filtrado entre GND y VCC.



5.1.3 Acceso a internet

De cada sensor leemos la información a la máxima velocidad a la que la placa controladora puede hacerlo con la intención de sobre muestrear esta lo máximo posible. Posteriormente dentro de la propia placa controladora filtramos la muestra con la intención de eliminar ruido.

Dicho proceso es realizado en las placas controladoras reduciendo la frecuencia de envío de mensajes sin afectar drásticamente a la latencia mientras aumentamos la calidad y fiabilidad de la información transmitida. Con esto logramos minimizar el ruido de las señales medidas de forma temprana lo que es más escalable que filtrar el ruido en otro punto de la arquitectura.

La información que se genera en las placas controladoras es enviada por bluetooth a la Raspberry PI donde le añadimos una marca de tiempo antes de enviarla finalmente a la base de datos para ser almacenada.

Hacer esto nos permite reunir la información en paquetes de gran tamaño de modo que nos comuniquemos con la base de datos en una frecuencia menor aprovechando más el ancho de banda disponible. Esto reduce los problemas de saturación de las comunicaciones pero reduce la latencia por lo que no se debe abusar de esta técnica.

5.1.3.1 Raspberry PI

La RaspberryPI como hemos mencionado juega un papel de buffer intermediario entre la base de datos y las placas controladoras. Dicha tarea podría haber sido realizada por otro tipo de placa más simple y barata como una ESP32, no obstante debido a que estamos construyendo un prototipo hemos decidido aprovechar el material del que disponíamos.

La programación de la raspberry ha sido realizada en Python pues es uno de los lenguajes con más soporte en esta plataforma. Mediante las librerías disponibles no solo podremos acceder a toda la funcionalidad que un sistema unix proporciona si no también a la GPIO para acciones a más bajo nivel.

5.2 Transmisión de la información

Comentaremos a continuación las tecnologías utilizadas para transmitir la información que captamos y procesamos a lo largo de nuestro sistema desde los sensores a los usuarios finales.

5.2.1 Sensores -> Controladores

La información se transmite mediante cables dispuestos según los esquemas proporcionados en formato digital o analógico siguiendo el modelo de comunicación que cada sensor define.

El circuito que los cables definen será lo más organizado posible para facilitar su comprensión y mantenimiento. Este ha sido diseñado según las premisas de modularidad hardware expuestas anteriormente.

5.2.1 Controladores -> Raspberry Pi

La comunicación entre ambos será realizada mediante el protocolo bluetooth-serie. Conectaremos el módulo bluetooth HC-05 a uno de los puertos serie de la arduino due o al puerto serie de la uno y emitiremos un stream de bytes.

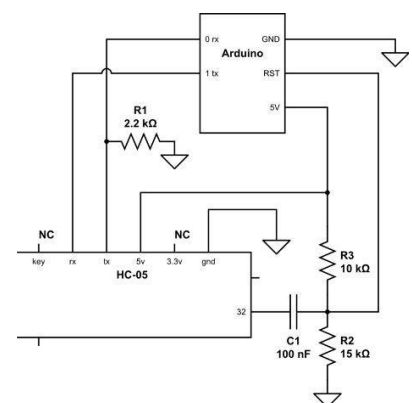
Este método de comunicación por defecto no está cifrado de modo que implementaremos un pequeño cifrado de clave privada suficientemente simple como para correr en las arduino pero suficientemente complejo como para no poder visualizar la información directamente con un sniffer.

La modularidad de este método reside en que la Raspberry podrá escuchar de 1 a N placas controladoras con el mismo código, enlazar todos sus datos para formar un paquete que contenga la información actual de los sensores ya filtrada, añadir una marca de tiempo y enviarla por internet a donde sea necesario que llegue.

La configuración del módulo bluetooth es ligeramente diferente de la que habitualmente se suele utilizar con las placas Arduino. El módulo HC-05 expone un pin de reset que puede controlarse desde el exterior. Conectando este pin con el reset de la Arduino podemos hacer que esta entre en el modo de carga de programa. Con el bootloader que Arduino incluye, al entrar en este modo, podremos almacenar en la memoria de programa todo aquello que llegue a la placa por el puerto serie. Por defecto las ATmega esperan recibir los nuevos binarios por el puerto ICSP.

Mediante el siguiente esquema podremos utilizar el módulo bluetooth para cargar código remoto en las placas controladoras.

Cabe destacar que el pull-down sobre TX no es necesario y que el pull-down sobre el reset puede establecerse a 10k si estamos alimentando el HC-05 de forma externa.



5.2.2 Comunicación con la base de datos

La Raspberry PI tendrá permisos de escritura sobre la base de datos y los usuarios tendrán permisos de lectura en el caso de las medidas de los sensores. Para la configuración de los usuarios esto estará invertido teniendo los usuarios permisos de lectura y escritura y la Raspberry solo de lectura.

El acceso a la base de datos no es directo si no que se realiza a través de una API escrita en Node JS que lo regula y comprueba que se realiza de forma adecuada. La comunicación con dicha API estará cifrada mediante clave privada.

Ya que nuestra base de datos es MongoDB y carece de una estructura fuertemente tipada para los datos utilizaremos Mongoose en la API para comprobar que los datos escritos en la base de datos mantienen la estructura que definamos sobre ellos. Adicionalmente el uso de una API nos facilita las consultas sobre la base de datos exponiendo consultas complejas habituales y reduciendo las dependencias del código a implementar en el cliente.

5.2.3 Front End -> Usuario

Esta es la única comunicación que no estará cifrada pues no contiene datos de los usuarios. Para acceder a la aplicación los usuarios utilizarán el navegador para acceder a nuestro servidor alojado en cloud con máquina virtual donde un programa en python le proporcionará con la página web que solicite.

Dicha página web contendrá scripts en JavaScript que se ejecutarán y accederán a la API que protege a la base de datos (esta comunicación está cifrada, es la que se explicó en el punto anterior).

De este modo solventaremos tener un canal no cifrado de comunicación haciendo que por este no pasen datos de los clientes.

5.3 Almacenamiento permanente

Como ya hemos mencionado utilizaremos la base de datos MongoDB. Esta es una base de datos no relacional que se viene utilizando cada vez más en los últimos años. Hemos elegido este tipo de almacenamiento permanente por su simplicidad. Es una base de datos sencilla de mantener, utilizar y monitorizar. Adicionalmente se integra a la perfección con el lenguaje JavaScript en el que escribiremos la API que hará de proxy para acceder a la base de datos.

Como medidas de seguridad utilizamos Mongoose para limitar la flexibilidad de Mongo restringiendo las inserciones sobre la base de datos a que cumplan unos formatos y especificaciones concretas.

5.4 Interpretación de la información

Expondremos dos alternativas para procesar la información generada por los sensores, la primera que consideramos más escalable y propia de una solución final y la segunda que presentamos pues cabe la posibilidad de que por falta de tiempo acabemos implementando en el prototipo.

La primera alternativa consiste en interpretar la información recogida de los sensores desde el código que los usuarios descarguen cuando accedan a la página web.

Interpretar la información en el cliente tiene la ventaja de que elimina la carga de trabajo a realizar en el back-end. Adicionalmente esta sería repartida de forma justa ya que cada cliente solo procesaría sus propios datos.

A este método solo le hemos encontrado el problema de que exista un cliente que entre en la aplicación con muy poca frecuencia de modo que cuando lo haga el procesamiento de sus datos tarde mucho en ser realizado. Esto lo solventaremos procesando los datos en fragmentos independientes que procesaremos por separado y que posteriormente uniremos en un procesamiento final.

El código de la página web obtendrá los intervalos de tiempo en los que quedan datos sin procesar. Irá procesando estos intervalos desde el más reciente al más cercano para poder mostrar la información reciente antes al usuario. Tras procesar cada intervalo almacenará la información procesada en la base de datos.

Finalmente cuando toda la información haya sido procesada en orden inverso a su recogida ahora se procesará esta nueva información generada y ya almacenada en la base de datos en el orden adecuado produciendo información de más alto nivel como tendencias de sueño o resúmenes mensuales o semanales.

La segunda alternativa consiste en procesar dicha información directamente en la Raspberry Pi cuando es recogida. En este caso el procesamiento de la información sería más sencillo pues solo debemos almacenar los datos necesarios para la inferencia y actualizarlos con cada nueva medida. El problema de este procesamiento es su baja responsividad. Si un usuario entra en la página web a mediados de mes espera ver la cantidad de horas que ha dormido en lo que va de mes o el ruido medio en su habitación en lo que va de día aunque estos no hayan terminado. Si fuera el código del cliente el que está realizando el procesamiento de datos solo tendría que tomar los datos que hubiera desde la fecha adecuada y sumarlos o calcular su media para mostrársela al cliente y almacenarla en la base de datos de modo que solo se calcule una vez.

Por el contrario si el procesamiento se hiciera en la Raspberry deberíamos preguntarnos cuándo debemos actualizar la información en la base de datos. Si lo hiciéramos con una alta frecuencia podríamos saturar la base de datos innecesariamente dependiendo de la cantidad de información que tuviéramos que presentar. No obstante hacerlo con una

frecuencia baja podría hacer que el usuario se fijara en que existen inconsistencias entre los datos procesados y sin procesar.

5.5 Mostrar la información al usuario

Como ya hemos dicho los usuarios accederán a una página web donde todos sus datos serán mostrados de forma amigable y fácil de entender. Utilizaremos Bootstrap como proveedor de estilos CSS para que la página se vea bonita. La librería Chart.js nos proporcionará la capacidad de dibujar gráficas mediante las que mostrar la información al usuario. El repositorio favicons nos permitirá utilizar iconos en la web para crear una interfaz moderna y amigable.

Finalmente todos estos elementos serán unidos con React para crear una interfaz reutilizable, mantenible y modular que proporcione buenos tiempos de respuesta ante las interacciones del usuario con los elementos y ante las actualizaciones de la información mostrada.

Para hacer que la interfaz se actualice de forma dinámica cuando nuevos datos lleguen a la API que hace de proxy delante de la base de datos utilizaremos un sistema de subscripción mediante websockets de modo que el código del cliente no necesite realizar polling de forma constante para refrescar la información que se muestra en la página web.

5.6 Virtualización

Tal y como hemos ido explicando las tecnologías utilizadas podemos ya empezar a diferenciar tres partes claras que compondrán nuestra aplicación. Por un lado tendremos la extracción de datos donde programaremos a bajo nivel para un hardware específico. Por otro lado estará el front end encargado de mostrar los datos a los usuarios y de realizar acciones exclusivas para cada uno de ellos. Finalmente tendremos un back end encargado de recibir los datos de los sensores y almacenarlos de forma permanente y también de proporcionar a los usuarios acceso al front end.

Debido a que el código desarrollado para los sensores se orienta a un hardware específico no tiene mucho sentido realizar virtualización en este entorno, tampoco lo tiene hacerlo en el front end, principalmente porque en cierto modo el navegador ya realiza esta tarea. No obstante la virtualización tiene un juego un papel muy importante en el back end de la aplicación.

Nuestro back end debe poder escalar con el número de usuarios, adicionalmente no debe depender de un sistema operativo o hardware concreto pues este puede cambiar. Puede que al principio queramos hostear nosotros el back end pero posteriormente portarlo a cloud o incluso que queramos cambiar de host cloud.

Es por ello que utilizaremos Docker para encapsular nuestro back end en contenedores en los que se realice una única tarea de modo que se reduzcan las dependencias entre ellos. Un contenedor almacenará la base de datos, otro contendrá la API que hace de proxy

delante de la base de datos y un tercero almacenará el servidor que nos proporcione acceso a la página web.

De este modo podremos de forma fácil y sencilla replicar los contenedores en caso de necesidad y lanzar el back end sobre cualquier equipo con la única dependencia de que este debe tener docker instalado. Esto también es una ventaja para hacer pruebas ya que cada miembro del equipo puede tener su propia instancia del back end y modificarla localmente sin afectar al resto solo con modificar la IP en la que este se encuentra.

6. Metodología de desarrollo

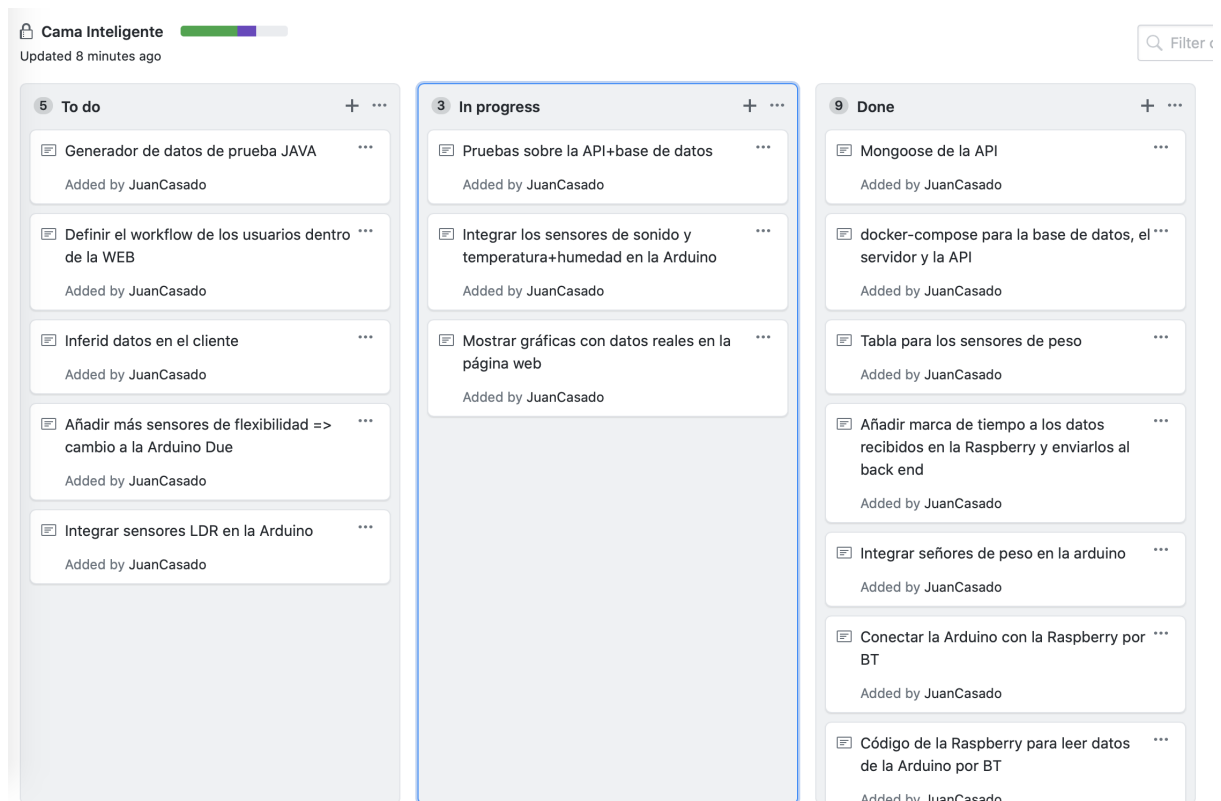
Una metodología de desarrollo adecuada es lo que nos separa entre un proyecto que tuvo buenos resultados y uno que podría haberlos tenido. Es fundamental definir una metodología que se adapte al equipo de trabajo y a los requisitos del proyecto definiendo un ciclo de vida que ayude a desarrollar el producto en vez de entorpecer su desarrollo.

Para definir la metodología que se está llevando a cabo en el proyecto es necesario tener en cuenta los siguientes puntos:

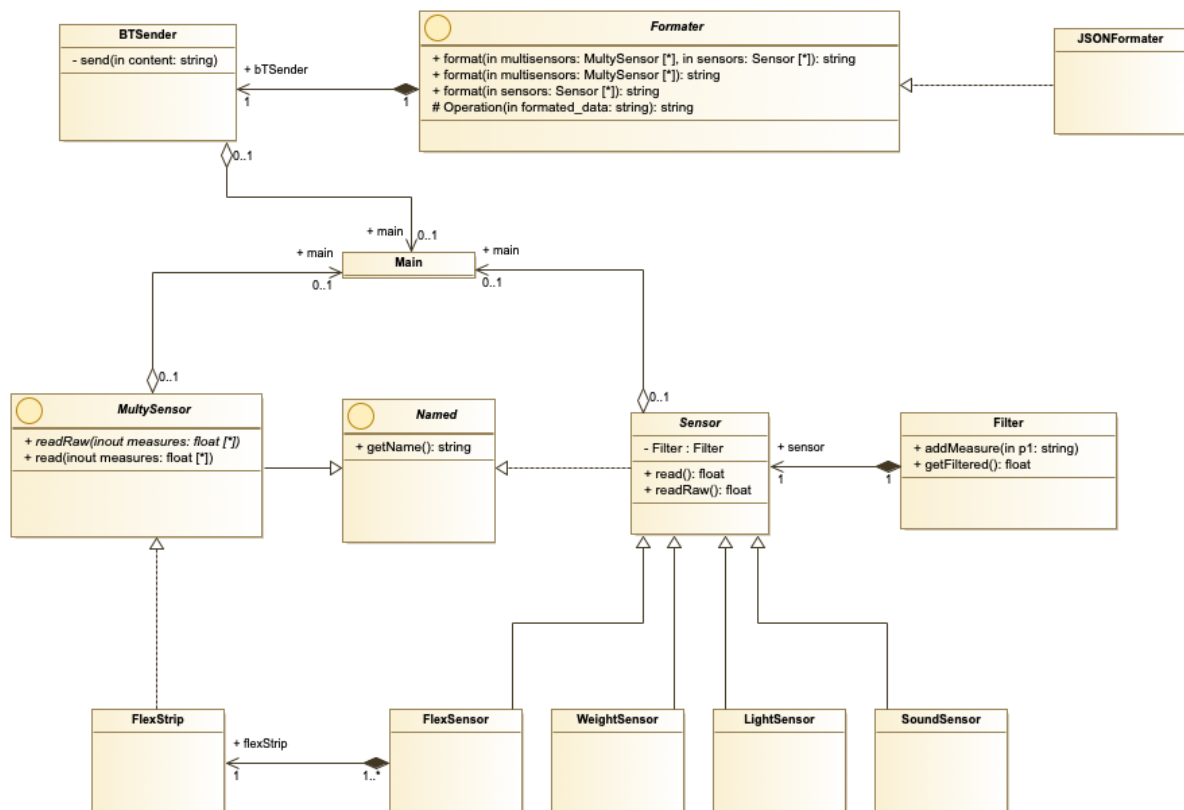
- Aunque el planteamiento del proyecto sea el de diseñar un producto completo y plenamente funcional debemos recordar que lo que debemos construir finalmente es solo un prototipo. Es por ello que aunque el diseño deba ser ambicioso y debemos de ser realistas con lo que finalmente seremos capaces de construir principalmente por las limitaciones temporales.
- Muchas de las tecnologías que vamos a utilizar son nuevas para el equipo de trabajo de modo que necesitaremos tiempos de prueba y aprendizaje.
- Deseamos que el resultado final del proyecto nos aporte un beneficio en el futuro. Para ello deseamos crear una documentación suficientemente buena como para que nos haga de base y tutorial por si en un futuro necesitamos volver a trabajar con algunas de las tecnologías que utilizemos.

6.1 Scrum

6.2 Kanban



6.3 Metodologías pesadas



6.4 Adaptación de la metodología a nuestro caso concreto

6.4.1 Prueba y aprendizaje

6.4.2 Diseño

6.4.3 Ejecución

6.4.4 Pruebas

6.4.5 Reuniones

6.4.6 Sprints

7. Arquitectura de la aplicación

La aplicación basa su arquitectura en múltiples cliente-servidor. Al principio ésta toma los datos desde el Arduino, quién es el responsable de captar datos con los sensores y enviarlos. Estos datos son enviados a un servidor en Python que se encarga de su procesamiento, eliminando valores extremos y agrupándolos correctamente para almacenarlos en la base de datos. El almacenamiento se realiza mediante un servidor API REST, que permite hacer operaciones CRUD con los datos. Este servidor actúa en su mayor parte como un proxy ante la base de datos para proporcionar consultas alto nivel sobre ellas. Por último está el servidor encargado del Back-end y Front-end de la web, que sirve para hacer que la web se muestre y funcione correctamente de cara al usuario. Los usuarios se conectan a este servidor para obtener la página web.

Al ser tan modular la arquitectura se tiene un sistema muy versátil y resistente a fallos. Por ejemplo, si el servidor web cayera se podría seguir almacenando datos, o en caso contrario se podría seguir consultando datos. También en caso de que se quiera cambiar de servidor o tecnología se puede hacer fácilmente cambiando aquellos módulos que nos interesen, como podría ser el servidor web.

Todo el código creado no destinado a correr sobre un hardware concreto (Arduino o Raspberry PI) está virtualizado con Docker. Esto supone una fácil migración, fácil despliegue y alta escalabilidad de la aplicación gracias a la replicación de recursos.

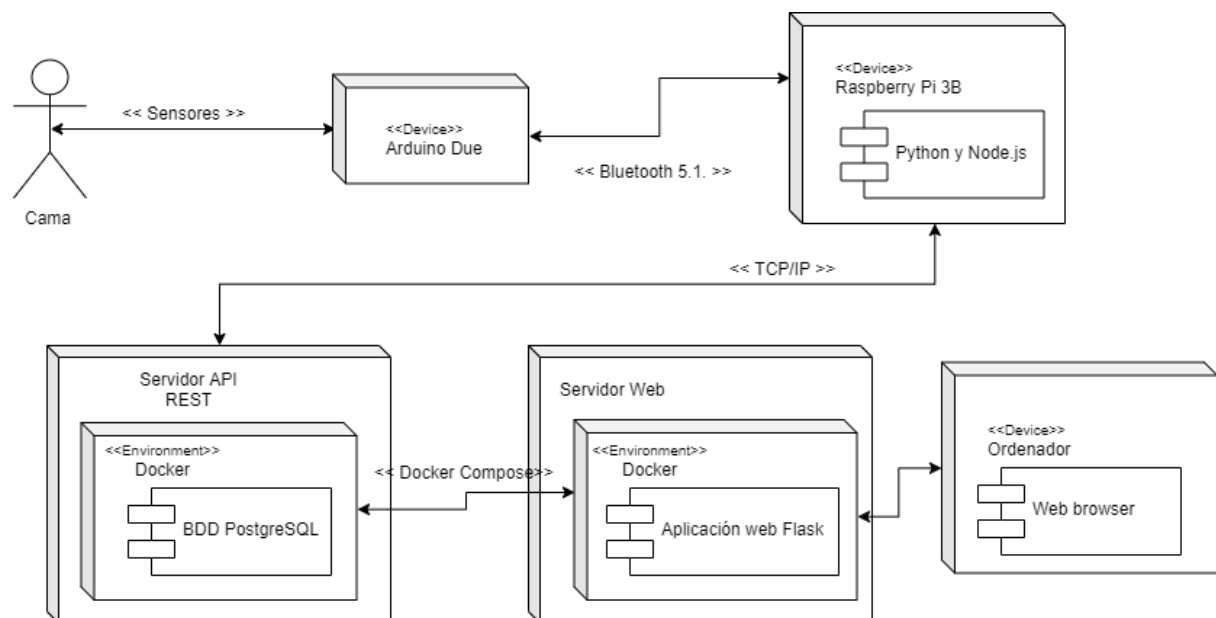


Diagrama de despliegue de la aplicación.

8. Modelo de negocio

En este apartado definiremos el modelo de negocio que seguirá nuestro proyecto de cama inteligente, que será el paso previo al plan de negocio. Este modelo de negocio nos permitirá tener una visión general del proyecto del grupo, viendo cómo se crea, distribuye y retiene el valor, en otras palabras, ganar dinero. Para plasmar el modelo de negocio se usará la plantilla de modelo de negocio Business Model Canvas, que propuso Alexander Osterwalder.

Después para hacer un mayor enfoque en los clientes y su visión del valor se usará el Value Proposition Canvas, también de Alexander Osterwalder, que se centra en el segmento de clientes y la propuesta de valor.

8.1. Business Model Canvas

8.1.1. Propuesta de valor

¿Qué valor entregamos al cliente?

Un sistema que es capaz de seguir toda la actividad del usuario mientras duerme, para que pueda mejorar sus hábitos, guardar registros y automatizar sus rutinas matinales, que se puede instalar en una cama.

¿Qué problema de nuestro cliente solucionamos?

Se solucionan problemas de salud, especialmente de espalda, de los clientes, la falta de sueño por romper ciclos del sueño, un mayor interés en el cuidado personal, no gastar mucho dinero en el producto y el aprovechamiento del tiempo de la mañana.

¿Qué necesidades estamos satisfaciendo?

Las necesidades que se satisfacen son las de un mayor control sobre la salud del cliente, la de poder hacer una cama inteligente sin tener que comprar una y la de la automatización de las rutinas en el hogar.

Estas propuestas se resumen en los siguientes puntos que irán en el lienzo:

- Registro de la salud del cliente.
- Mejora de calidad del sueño.
- Comodidad al cliente
- Evitar la compra de una cama nueva.
- Automatización de rutinas (Nada más levantarse que inicie la casa persianas, luces etc.)

8.1.2 .Segmentos de Cliente

¿Para quién estamos creando valor?

Se crea un valor para las personas que buscan mejorar sus hábitos durante el sueño, optimizar su tiempo y quieren hacerlo con un presupuesto limitado.

¿Quiénes son nuestros clientes más importantes?

Los clientes más importantes son adultos con intereses en la salud y la tecnología con una edad comprendida entre 20 y 55 años.

La segmentación de los clientes se realizará aprovechando el nicho de mercado de la gente que tiene esa edad junto a esos intereses. En el lienzo serán:

- Personas con entre 20 y 55 años con interés en su salud y habilidad tecnológica.
- Empresas dedicadas a mobiliario.

8.1.3. Canales

¿Cómo hacer que nos conozcan?

La principal medida de para que se conozca el producto va a ser la realización de campañas publicitarias en redes sociales, más asociaciones con empresas dedicadas a la producción y venta de camas.

¿Cómo ayudamos al cliente a evaluar la propuesta de valor?

Se ayudará al cliente mediante una sección de guías y videos disponibles en la web para el uso de la cama y mediante la posibilidad de contactar con los expertos de la empresa por correo o teléfono.

¿Cómo permitimos que compren nuestro servicio/producto?

Se permite la compra del producto mediante la web o mediante tiendas asociadas con la empresa.

¿Cómo entregamos la propuesta de valor?

Se entrega al cliente por paquetería o en las tiendas asociadas de forma directa.

¿Cómo proporcionamos servicios de postventa al cliente?

Se proporcionarán los servicios postventa mediante correo electrónico o teléfono.

A partir de las preguntas el espacio de canales quedaría así:

Directos

- Web (Internet)
- Publicidad en redes sociales
- Correo electrónico
- Teléfono
- Paquetería

Indirectos

- Tiendas asociadas

8.1.4. Relaciones con clientes

¿Qué relación queremos establecer y mantener con nuestros clientes?

Se quiere establecer una relación directa entre los clientes y empresa, mediante asistencia personal dedicada y una comunidad relacionada con la cama, para así mantener una relación buena con los clientes para que fidelizarlos.

¿Cómo de costoso es?

Es bastante costoso porque requiere de personal para poder tener una buena atención al cliente y se debe de gestionar la comunidad que se cree.

¿Cómo de integrado está con el resto de nuestro modelo de negocio?

Está moderadamente integrado con el resto del modelo de negocio.

En el lienzo quedaría:

- Asistencia personalizada dedicada sobre la salud.
- Comunidad para los clientes.
- Rápida y eficaz.

8.1.5. Fuentes de ingresos

¿Por qué propuesta de valor están dispuestos a pagar nuestros clientes?

Por un sistema que permita hacer que su cama sea capaz de darles información sobre cómo duermen, que automatice su día a día y mejore su sueño.

¿Por qué están pagando actualmente?

Actualmente está pagando por un sistema que es capaz de informarle de la postura en la que duerma, controlar su peso, saber el ambiente en el que duerme, despertarle e iniciar comandos.

¿Cómo están pagando actualmente?

Las principales formas de pago son online mediante la pasarela de pagos Stripe permitiendo así una gran variedad de formas de hacerlo, incluidas tarjetas y Paypal entre otras. Además de en efectivo en las tiendas asociadas

¿Cómo preferirían pagar?

Por medios electrónicos que fuesen de su comodidad.

¿Cuál es el porcentaje de ingresos de cada línea de ingreso respecto a los ingresos totales?

Hay una única línea de ingresos que es la venta directa de los sistemas para las camas. Aunque en un futuro esto podría verse modificado.

En función a las preguntas el hueco en el Canvas tendría:

- Venta de activos (kits o camas inteligentes).
- Pagos con Stripe.

8.1.6. Recursos Clave

¿Qué recursos clave requiere nuestra propuesta de valor?

La propuesta requiere de un capital inicial para poder poner en marcha la producción, registro de la marca y patentes para la propuesta de valor.

¿Qué recursos clave requieren nuestros canales de distribución?

Los canales de distribución requieren de un almacén, distribución por paquetería, un servidor y dominio web. Además de los recursos para publicidad y de locales de tiendas asociadas.

¿Qué recursos clave requiere la relación con el cliente?

Requiere de empleados con cualificación médica y técnica, la formación de estos empleados, para brindar una atención adecuada a los clientes y oficinas para albergar a los empleados.

¿Qué recursos clave requieren las fuentes de ingreso?

Las fuentes de ingreso requieren de una web donde aplicar la pasarela de pagos y cuentas bancarias para gestionar cobros y ventas.

Por tanto quedará:

Físicos

- Producto, almacén, tiendas asociadas y oficinas.

Intelectuales

- Publicidad, patentes y marcas.

Humanos

- Personal y formaciones.

Económicos

- Capital inicial y cuentas bancarias.

8.1.7. Actividades clave

¿Qué actividades clave intervienen en nuestra propuesta de valor?

Las principales actividades que intervienen son el desarrollo del sistema y la fabricación del sistema por parte del proveedor.

¿Qué actividades clave requieren nuestros canales de distribución?

Se requiere de la distribución de los sistemas mediante las tiendas asociadas, web y paquetería.

¿Qué actividades clave requieren nuestra relación con los clientes?

La recepción y atención a clientes durante la preventa y postventa, el mantener la comunidad de clientes y la formación de los empleados.

¿Qué actividades clave se necesitan para nuestra fuente de ingresos?

La actividad clave es la venta del sistema de cama inteligente a los clientes.

El resultado en el lienzo será:

- Comprar el sistema al proveedor
- Venta y distribución del sistema de cama inteligente (salvo tiendas asociadas)
- Atención a los clientes.
- Formación del personal

8.1.8. Socios Clave

¿Quiénes son nuestros socios clave?

Los socios clave que se tienen son las tiendas asociadas, que permitirán aumentar los clientes posibles, estas serán tiendas de muebles.

¿Quiénes son nuestros proveedores clave?

Los proveedores clave son la empresa especializada en fabricar tecnología de China debido a factores económicos, enviando el sistema ya montado. Estos recibirán las instrucciones del modelo que ensamblarán.

¿Cuáles son los recursos clave que estamos adquiriendo de los socios?

Se obtienen recursos como la posibilidad de vender desde tiendas físicas y el propio sistema de la cama en sí.

¿Cuáles son las actividades clave que hacen los socios?

Los socios pueden proveer del sistema de la cama y realizar ventas del sistema.

Los socios clave serían:

- Proveedores tecnológicos
- Tiendas de muebles

8.1.9. Estructura de costes

¿Cuáles son los costes más importantes de nuestro modelo de negocio?

El modelo de negocio presenta diversos costes destacando los del desarrollo del sistema inteligente, el derivado del desarrollo de la web, la compra al proveedor, los alquileres y el salario al personal. Además de impuestos a pagar, gestión, pedidos mínimos y formaciones.

¿Cuáles son los recursos clave más caros?

Los recursos clave más caros son los sistemas de cama que se comprarán al proveedor, junto a los costes de alquileres, publicidad y salarios de empleados técnicos y médicos.

¿Cuáles son las actividades clave más caras?

Las actividades más caras son la compra de los sistemas, prestar atención al cliente y la formación de los empleados.

Se tendrán en el modelo:

- Desarrollo y mantenimiento web
- Desarrollo del producto
- Formación de empleados
- Compra a los proveedores
- Publicidad
- Gestión
- Impuestos

- Alquileres de oficinas y almacén



9. Plan de marketing

9.1. Análisis de la situación actual

Para hacer un análisis de la situación actual de la empresa emplearemos un DAFO. Este análisis contiene un análisis interno de fortalezas y debilidades, y otro externo de amenazas y oportunidades.

ANÁLISIS DAFO



9.2. Análisis de la competencia

La empresa tiene bastante competencia, esta competencia en su mayoría son empresas estadounidenses, entre las que destacan SleepNumber, RestPerformance y EightSleep.

La empresa **EightSleep** es un startup que ha recibido 70 millones de inversión, los precios de sus productos van desde 895\$ hasta 1195\$ el modelo que permite realizar un seguimiento del sueño, controlar el calor de la cama y poner alarmas personalizadas. Si se quiere el modelo que permite enfriar la cama el precio asciende a los 2095\$ para el más barato. Su modelo de publicidad se basa en el uso de redes sociales y reviews de usuarios. Además ofrecen una gran cantidad de descuentos muchos de ellos enfocados en estudiantes y militares. Otros elementos que tiene son la posibilidad de financiarla y que disponen de una tienda física.

La empresa **Rest Performance** ofrece camas que van desde los 3800\$ a los 6200\$, cuyas funciones son las del monitoreo del sueño y la de ajustarse ergonómicamente al usuario en

tiempo real. Como publicidad emplea las redes sociales y un sistema de embajadores con deportistas de élite. Además ofrece financiación y tiene presencia en más de 30 tiendas físicas.

La empresa **SleepNumber** es la más grande de todas con diferencia, puesto que se trata de una empresa tradicional que ahora desarrolla camas inteligentes. Sus productos van desde 500\$ a 1400\$, que permiten realizar un seguimiento del sueño y ajustar el firme del colchón, aunque también ofrecen camas con ajuste ergonómico a partir de 3600\$. Al ser una empresa tradicional cuenta con presencia en el mercado con más de 600 tiendas. Al igual que las anteriores permite financiación y se publicita por redes.

Empresa	Precio (Rango)	Característica principal	Otras
EightSleep	895\$ - 2095\$	-Control de temperatura.	-Seguimiento del sueño. -Alarmas. -Financiación.
Rest Performance	3800\$ - 6200\$	-Control de ergonómico	-Seguimiento del sueño. -Financiación. -Varias tiendas físicas.
SleepNumber	500\$-3600\$	-Control de firme y ergonómico en las más caras.	-Seguimiento del sueño. -Financiación. -Muchas tiendas físicas.

9.3. Objetivos

Los objetivos que vamos a fijar se basan en el principio SMART, porque van a ser específicos, medibles, alcanzables, relevantes y a tiempo. Así de esta forma seremos capaces de aplicar objetivos realistas a la empresa.

Los objetivos que pondremos serán:

- La incorporación al mercado y ganancia de un 2% de cuota de mercado para el 2020.
- El aumento de ventas de la cama en un 80% en el 2020.
- Aumentar la cantidad de clientes en un 30% mediante las redes sociales en 2020.
- Alcanzar 1 millón de euros de facturación para el segundo semestre de 2020.

9.4. Plan de actuación: estrategias de marketing

Con los objetivos definidos ahora vamos a centrarnos en cómo conseguirlos para ello aplicaremos estrategias enfocadas en el precio, en la distribución y en la promoción de la cama. En la que más nos enfocaremos será en la de promoción puesto que se trata de un producto novedoso, pero que es un desconocido en el mercado. Para cumplir este primer objetivo abriremos cuentas en las principales redes sociales, mejoraremos el SEO de la página web actual y financiaremos una campaña de anuncios en la web con Google Ads. El siguiente paso será centrarnos en el precio de la cama, puesto que nuestro modelo se puede acoplar en las camas supone un ahorro y mayor aprovechamiento de camas, lo que hace que tenga un precio bajo de salida de 300€, frente a la competencia. Además para fomentar las ventas en temporadas especiales como Navidad o Black Friday crearemos promociones de descuentos y regalos. Por último intentaremos mejorar la distribución del producto hablando con nuevos transportistas y tiendas físicas para que puedan actuar como puntos de recogida, e incluso venta.

9.5. Revisión del plan

Para asegurarnos del cumplimiento del plan iremos organizando reuniones periódicas en la que se revisarán los objetivos, el cumplimiento del plan, el funcionamiento de las estrategias y el presupuesto del plan. Una vez se revise esto podremos realizar modificaciones o ampliaciones sobre el plan mismo.

10. Planificación temporal, plan de desarrollo, riesgos, plan de contingencias

11. Resumen y conclusiones

12. Bibliografía

https://content.arduino.cc/assets/UNO-TH_Rev3e_sch.pdf

<https://www.arduino.cc/en/Hacking/PinMappingSAM3X>

<https://github.com/adafruit/DHT-sensor-library>

<https://create.arduino.cc/projecthub/mansurkamsur/upload-sketch-arduino-over-bluetooth-using-android-f1ce55>

https://www.electrodragon.com/w/AT_Commands

<https://es.pinout.xyz>

<https://www.raspberrypi.org/documentation/usage/gpio/>

<https://docs.docker.com>

<https://github.com/chartjs/Chart.js>

<https://expressjs.com/es/4x/api.html>

<https://www.thepolyglotdeveloper.com/2019/01/getting-started-mongodb-docker-container-deployment/>

<https://docs.mongodb.com/ecosystem/drivers/node/>

<https://docs.mongodb.com/ecosystem/drivers/python/>

<https://github.com/Automattic/mongoose/>

<https://babeljs.io/docs/en/>

<https://learn.sparkfun.com/tutorials/getting-started-with-load-cells/all>

<https://www.ardumotive.com/how-to-use-a-flex-sensor-en.html>

<https://www.prometec.net/sensor-sonido-ky038/>

<https://docs.docker.com/compose/>

<https://www.espressif.com/en/products/hardware/esp32/overview>

<http://pages.iu.edu/~rwisman/c490/html/pythonandbluetooth.htm>

<https://people.csail.mit.edu/albert/bluez-intro/c212.html>

https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.Xd17Wi2ZNhE

https://en.wikipedia.org/wiki/Agile_software_development

[https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))

<https://github.com/docker-library/openjdk>

<https://websockets.readthedocs.io/en/stable/>

<https://github.com/theturtle32/WebSocket-Node/tree/master>

<https://github.com/TooTallNate/Java-WebSocket>

<https://es.reactjs.org/docs/getting-started.html>

https://es.wikipedia.org/wiki/Lienzo_de_modelo_de_negocio

https://es.wikipedia.org/wiki/Modelo_de_negocio

<https://advenio.es/tu-propuesta-de-valor-desde-el-cliente-value-proposition-canvas/>

<https://theherocamp.com/blog/value-proposition-canvas/>

<https://www.cleverism.com/business-model-canvas-creating-value-proposition/>

https://es.wikipedia.org/wiki/Análisis_DAFO

https://es.wikipedia.org/wiki/Plan_de_marketing

<https://robertoespinosa.es/2014/03/25/como-elaborar-el-plan-de-marketing>

<https://www.genwords.com/blog/como-hacer-un-plan-de-marketing-ejemplos>

<https://www.inboundcycle.com/blog-de-inbound-marketing/que-es-un-plan-de-marketing-y-como-hacer-uno>

<https://www.cnet.com/news/the-best-smart-bed-could-be-the-one-you-already-own/>

<https://www.restperformance.com/product/rest-bed/>

<https://www.sleepnumber.com/beds-on-sale>

<https://site.eightsleep.com/pages/sale>