



HealthTech: sistema de monitorización especializado en la persona

Pedro Collado Navarro
Víctor Hernández Sanz
Ramón Iñiguez Bascuas
Ruben Ortiz Nieto





Índice

Resumen	3
Introducción	4
Estado del arte	6
Objetivo general y subobjetivos	7
Propuesta técnica	8
Sensores	10
Planificación	16
Programación	18
Optimización	19
Presupuesto	20
Diagrama Entidad Relación	21
Conclusión y Líneas Futuras	22
Referencias bibliográficas utilizadas	23



Resumen

Hoy en día, la población de Europa está envejeciendo; el índice de natalidad está descendiendo mientras que la esperanza de vida aumenta, lo que conlleva a que el número de personas mayores sea superior al de personas jóvenes.

Es un hecho que se producirá un cambio demográfico hacia una población de mayor edad, y esto tiene muchas implicaciones en cómo afecta la forma en la que viviremos nuestras vidas.

Esto se traduce a que se necesita una reforma en el planteamiento de la salud pública, especialmente hacia las personas mayores, y entre esa reforma se encuentran ideas como nuestro producto, que ofrecen una comunicación entre los distintos usuarios y un seguimiento más detallado del paciente, lo cual asegura la prevención de futuros problemas, todo mediante una aplicación intuitiva y con diferentes interfaces para adaptarse a las necesidades de cada usuario.

Nuestro producto intentará garantizar una transición más fluida enfocándose en las personas afectadas: pacientes, médicos, familiares y cuidadores. En esencia, nuestro producto es un sistema de monitorización en el que utilizaremos sensores para cumplir dos objetivos: reducir la brecha de información y progreso del paciente entre las distintas consultas médicas, y reducir el creciente gasto sanitario en el cuidado de estos pacientes que en muchos casos se puede prevenir o al menos reducir.

Introducción

En la figura 1 y figura 2 se aprecia cómo, según un estudio realizado por PopulationPyramid.net [1], la cual obtiene sus datos de fuentes muy fiables [2] como las Naciones Unidas, se prevé que en el 2045, la población de Europa se envejezca de tal manera que casi la mitad de la población sea mayor de 55 años, mientras que para en 1990 este porcentaje no sobrepasaba el 25%.

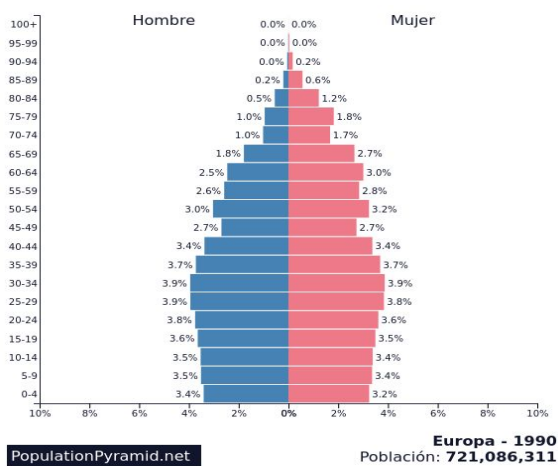


Figura 1
Pirámide de población-1990 [1]

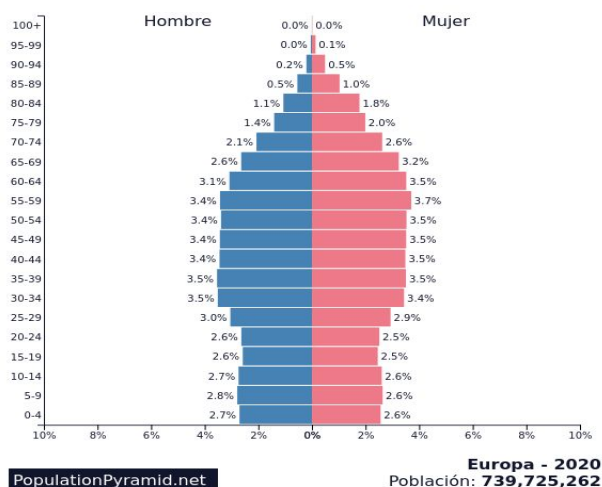


Figura 2
Pirámide de población-2020 [1]

Centrándonos en la población española, esta longevidad en los años de vida se debe sobre todo a dos factores mencionados previamente: la baja natalidad del país y la mayor esperanza de vida de sus ciudadanos. Según los datos de la UN (Naciones Unidas), revisión de 2019 [3], para 2050, una de cada seis personas en el mundo tendrá más de 65 años (16%), frente a una de cada 11 en 2019 (9%).

En 2018, por primera vez en la historia, las personas de 65 años o más superaron en número a los niños menores de cinco años en todo el mundo. Se prevé que el número de personas de 80 años o más se triplique, de 143 millones en 2019 a 426 millones en 2050.

Por otro lado, el Instituto Nacional de Estadística (INE) [4] ha registrado en España que el 17% de los ciudadanos es mayor de 65 años y se prevé que en 2050 este porcentaje casi se duplique situándose en el 33%.

Esto resulta en un incremento en el gasto sanitario, aunque muestra un ligero decrecimiento los últimos años mostrado en la figura 3, y un gran problema para el planteamiento del sistema sanitario, ya que según el informe: “Consecuencias del envejecimiento de la población” [5], realizado por el INE, este destaca el reto que supone para el sistema “... no sólo porque el gasto sanitario aumenta con la edad, sino también porque implica un cambio en el carácter de los cuidados demandados”.

	2010	2011	2012	2013	2014
Servicios de asistencia curativa y de rehabilitación	56.156	56.773	55.627	53.553	55.393
Servicios de atención de larga duración	9.372	9.069	8.949	8.665	8.678
Servicios auxiliares de atención de la salud	5.006	5.025	4.808	4.550	4.700
Productos médicos dispensados a pacientes ambulatorios	21.764	21.227	20.313	21.209	21.074
Servicios de prevención y de salud pública	2.264	2.124	1.983	1.893	1.875
Administración de la salud y los seguros médicos	2.867	3.076	3.089	2.831	2.814
Formación de capital de instituciones proveedoras de atención de la salud	2.470	1.873	1.405	1.156	1.188
Gasto sanitario total	99.899	99.167	96.174	93.856	95.722

Figura 3

Gasto sanitario total según función de atención de la salud. Millones de euros.
España, 2010-2014 [6]

Los mayores costos para la sociedad no son los gastos que se hacen para que los pacientes tengan una vida con capacidades funcionales, sino los beneficios que podrían perderse si no realizamos las adaptaciones apropiadas ahora que estamos a tiempo.

Estado del arte

En este campo se han elaborado varios informes y proyectos con la intención de hacer que las personas mayores y personas con Alzheimer puedan evitar o retrasar el deterioro de la memoria mediante estimulaciones, en un proyecto llamado SED y desarrollado en Murcia en 2010/2011 [7].

El servicio de teleasistencia domiciliaria que se promovió en 2017 en la comunidad de Madrid [8], dotada de las instalaciones de distintos sensores de presión en lugares tan comunes como son los colchones, sillas, sofá... se podrá medir si la persona se levanta de la cama en medio de la noche y no vuelve en un tiempo prudencial, lo que podría llevarnos a pensar que ha sufrido una caída, otros sensores son los de mediciones de combustión, en otras palabras, sensores de humo, para mandar una alerta a los servicios de teleasistencia para una rápida atención, este programa constaba de un presupuesto de 4.8 millones de euros al año, y contó con casi 35.000 usuarios de teleasistencia.

El documento proporcionado por el Ceapat(Centro de Referencia Estatal de Autonomía Personal y Ayudas Técnicas) [9] expande en lo anterior, proporcionando más sensores y más funcionalidades como un terminal GSM/GPS, figura 4, con forma de reloj que proporciona una localización exacta, y permite una configuración de un área de seguridad que si es rebasado manda alertas y se activa un sistema de rastreo.

Junto con un detector de caídas, que manda alertas cuando se detecta una aceleración brusca seguida de un tiempo de inactividad, muy útiles para personas que tienen un alto riesgo de sufrir caídas por mareos o por problemas de movilidad, lo que se complementaría con un botón del pánico que se activa cuando se encuentra en una situación de riesgo o de peligro.



Figura 4
Terminal GSM/GPS con
forma de reloj [9]



Objetivo general y subobjetivos

El objetivo de este proyecto es ayudar a mejorar la vida cotidiana de las personas que sufren despistes, olvidos e incluso tengan síntomas de demencia y reducir posibles gastos sanitarios los cuales se pueden prevenir con un sistema de monitorización como el que ofrecemos.

Con este proyecto vamos a utilizar una serie de sensores en la vivienda del paciente con los que, a través de una aplicación en el ordenador (con vistas a aplicaciones móviles), se puedan observar los datos y alertas recogidos para garantizar la monitorización que nuestro producto ofrece.

Las funcionalidades de los sensores que vamos a utilizar son: medir la temperatura de una habitación, detectar si hay una ventana abierta (junto al sensor de temperatura permitirá saber si el cambio de temperatura se debe por esto), medir la calidad del aire (emitiendo una alerta si se ha dejado el gas abierto en la cocina), detectar y registrar si el paciente se ha pasado tiempo de más en la cama, señalar la ubicación y la ruta a su casa, además de transmitir la propia localización del paciente.

Uno de nuestros subobjetivos más importantes es garantizar un nivel alto de atención al cliente, entre los que incluye un servicio que garantiza una inspección y reparación de algún componente del producto (sensores, cables, placa arduino) en un periodo de 1-3 días laborales.

Para hacer la aplicación lo más intuitiva posible, queremos hacer especial hincapié en en la comunicación entre los diferentes usuarios, ya que nuestra idea es que exista una comunicación dedicada, 1:1 entre dos usuarios. Se basará en una comunicación basada en tickets generados por cualquiera de los usuarios, los cuales permitirán al médico hacer un seguimiento más detallado del paciente e intenta replicar la comunicación real que ocurre en una consulta pero desde casa, y sin añadir una gran carga y ahorrar tiempo tanto al paciente como al médico.

Además, el médico dispondrá de más tiempo para reflexionar sobre la información que está recibiendo del paciente y ayuda a la toma de decisiones adecuadas por su parte.

Propuesta técnica

Vamos a utilizar dos clases de sensores, continuos y discretos. Todos estos se van a poder manejar a través de él arduino MEGA / NodeMCU, ya que se manejan utilizando 3-5V y la placa arduino es capaz de suministrar el voltaje requerido.

Una de las limitaciones que tenemos es la del número de sensores que podemos utilizar, ya que no podemos implementar un número ilimitado de ellos, por lo que junto al arduino y 1-2 de sensores de cada tipo alcanzamos nuestro objetivo con alta eficiencia.

En cuanto a la interfaz de usuario, todos los usuarios tendrán la misma vista la cual se basará principalmente en la navegación de distintas pestañas dentro de la misma ventana, donde dependiendo del rol tendrán acceso a más/menos pestañas. Todos los usuarios (paciente, médico, cuidador, familiar) accederán a una vista intuitiva e interactiva, la cual contiene un login con usuario y contraseña, y después de iniciar sesión tendrá: la opción de buscar un usuario (médico, cuidador) acceder a sus datos y enviarle un mensaje, saldrá una notificación en forma de ventana emergente si se ha disparado un sensor (una ventana abierta o el gas en la cocina) y un calendario personalizado con eventos.

Hacemos visibles las alertas generadas por los sensores al médico y familiar igual que el resto de usuarios, añadiendo las pestañas registros y localización, a las cuales sólo ellos tiene acceso, las cuales contienen un registro (logs) para que pueda estudiar si existe algún patrón que conlleve a otra cosa más seria como mencionado anteriormente. Asimismo incorporamos la posibilidad de compartir eventos en el calendario entre el médico y el paciente, en el cual puede programar una cita la cual aparecerá en ambos calendarios automáticamente.

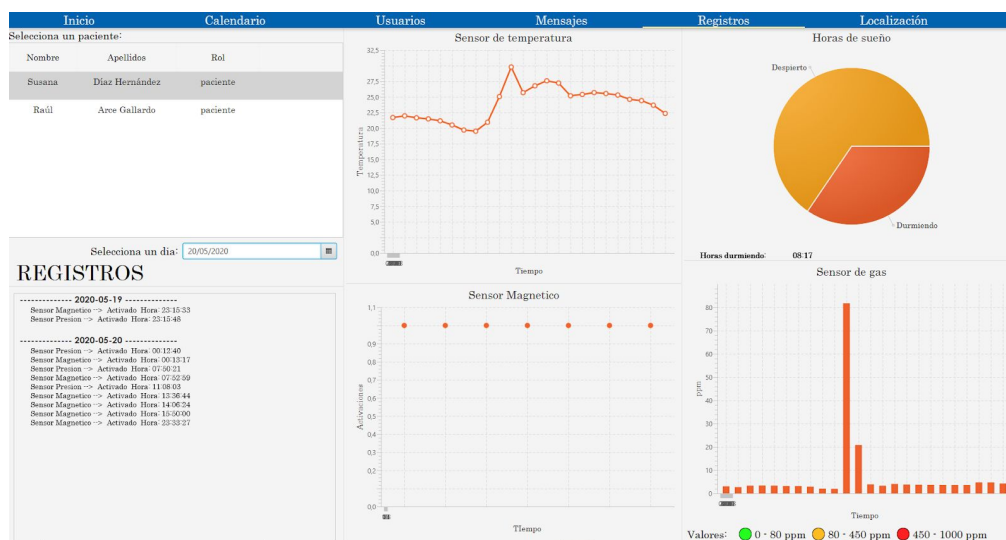


Figura 5
Pestaña de registros

Además todos los usuarios tendrán una pestaña de inicio con recordatorios útiles como: mensajes nuevos, citas programadas en el calendario cercanas, y un FAQ (frequently asked questions), es decir, preguntas y respuestas, el cual se generará a partir del uso del producto el primer año y de información previa dada por los médicos.



Figura 6
Pestaña de inicio

Los cables, los cuales estarán aislados con una capa de plástico y medirán varios metros, comunicaran los sensores con la placa arduino y no estarán al alcance del paciente o del cuidador, ya que estarán cubiertos por una placa de plástico discreta que recorrerá las esquinas de la casa y requerirá uno de nuestros técnicos para poder acceder a ellos. Esto garantiza seguridad a la hora de utilizar el producto y evita que el cuidador o el paciente tenga algún accidente.

Sensores

1. Módulo WIFI:

En un principio empleamos el Arduino MEGA para conectarnos al wifi, utilizando el módulo ESP8266 con el código que se nos proporcionó en los seminarios, pero debido a la carencia de sensores, cambiamos el Arduino MEGA por el NodeMCU, que tiene el módulo WiFi ESP8266 ya incorporado.

Esto nos ha permitido con la librería "ESP8266WiFi.h" conectarnos tan solo pasándole los parámetros de nuestra WiFi:

// Credenciales de WiFi.

```
char ssid[] = "Nombre de la red";           // Nombre de la red WiFi
```

```
char pass[] = "Contraseña de la red";       // Contraseña de la red WiFi
```

Y llamando en el **setup** a la función:

```
WiFi.begin(ssid, pass);                     // Le pasamos al WiFi las credenciales
```

2. Sensores

El **sensor de medición de calidad del aire MQ-135** estará siempre activo en la cocina recogiendo valores para poder realizar un estudio, y cuando este detecte que hay gas o monóxido de carbono en niveles más altos de lo normal, se reflejará una alerta en la aplicación para todos los usuarios. En la figura 6 se muestra su implementación usando una tabla de circuitos con un arduino:

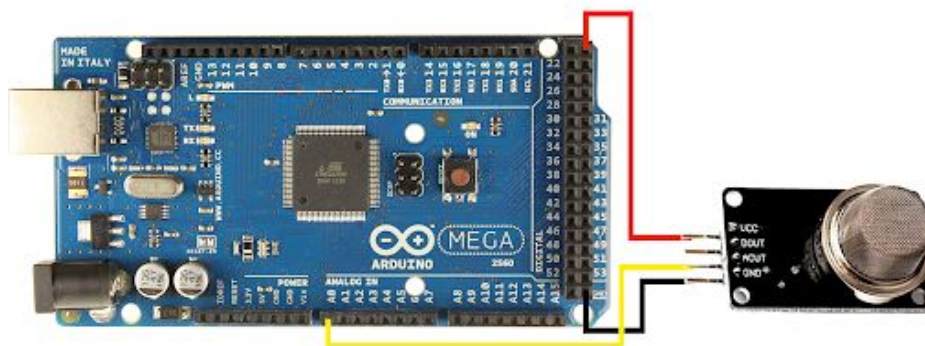


Figura 7

Sensor de gas conectado al arduino MEGA [10]

Podemos acceder al datasheet de este sensor a través del enlace:
<https://datasheetspdf.com/pdf/605076/Hanwei/MQ-135/1>

El **sensor magnético KY-025** tenemos pensado ponerlo en la ventana del salón para saber si esta se encuentra cerrada o abierta.

Este módulo contiene un sensor magnético (reed switch, la ampolla de vidrio con las láminas internas) que se activa cuando se aproxima a un campo magnético. Tiene una salida digital D0 (activa a nivel bajo) que indica si el reed switch está abierto o cerrado. Tiene, también, una salida analógica A0 (regulable mediante un comparador) que indica la intensidad (distancia) del campo magnético [13].

Cuando estén en contacto significará que la ventana está cerrada mientras que en caso contrario la puerta está abierta, reflejándose así en la aplicación con lo que justo al sensor de temperatura se podrá determinar si el cambio de temperatura está provocado porque el paciente se ha podido dejar la ventana abierta, mostrado en la figura 8:

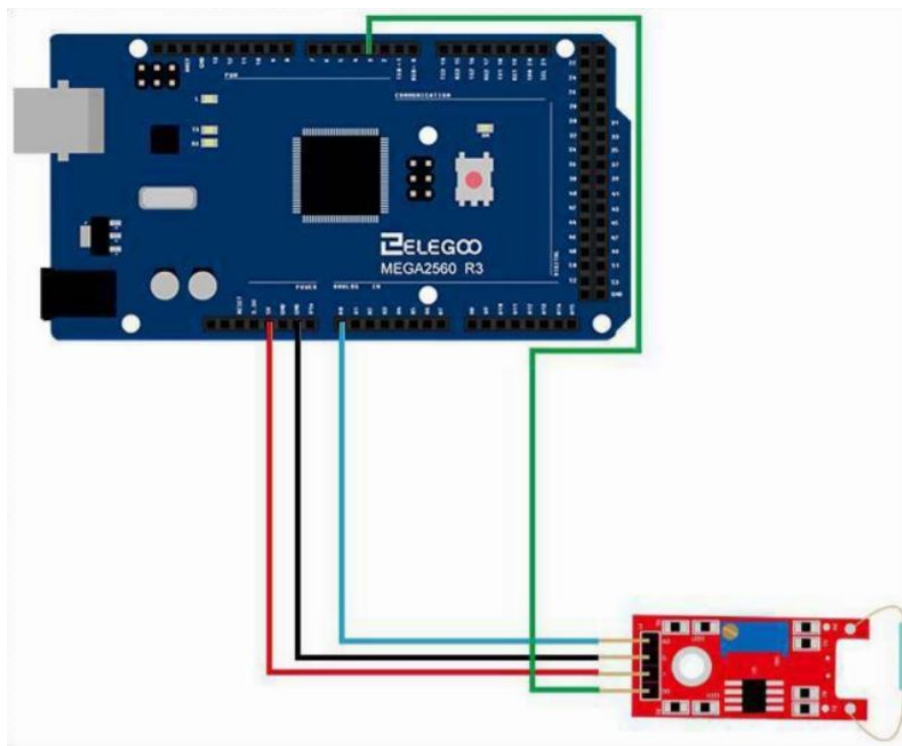


Figura 8
Sensor magnético KY-025 conectado a arduino [11]

Podemos acceder al datasheet de este sensor a través del enlace:
<https://datasheet4u.com/datasheet-pdf/Joy-IT/KY-025/pdf.php?id=1402036>

El **sensor de presión MF01** detecta que hay una persona tumbada en la cama, analizando las horas de sueño y las veces que se ha levantado de la cama y a qué hora. La plataforma es programada para disparar una alerta cuando la ocupación o la falta de ocupación pueden ser debidas a una situación irregular.

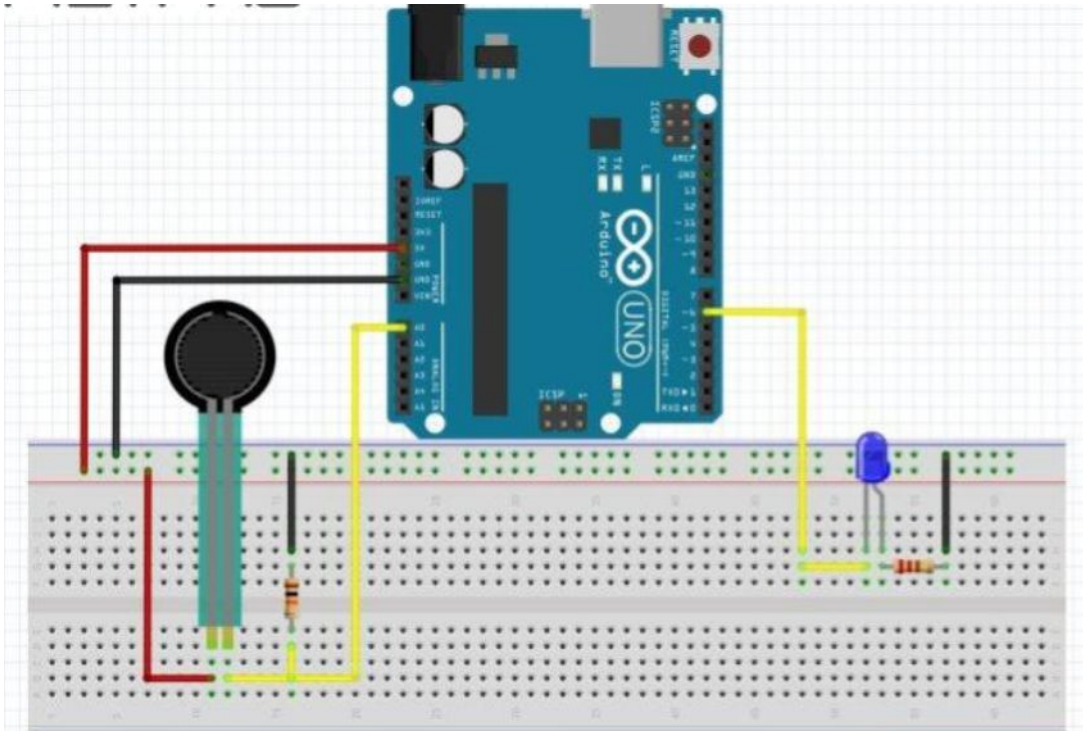


Figura 9
Sensor de presión MF01 conectado a arduino [14]

Podemos acceder al datasheet de este sensor a través del enlace:
<https://www.alldatasheet.com/view.jsp?Searchword=MF01&sField=3>

Con el **GPS NEO 6M** permite aplicar la opción en la que se manda la ubicación del paciente cada minuto y se refleja en un mapa (proporcionado por un API de google), para monitorizar al paciente en caso de que haya decidido salir de casa.

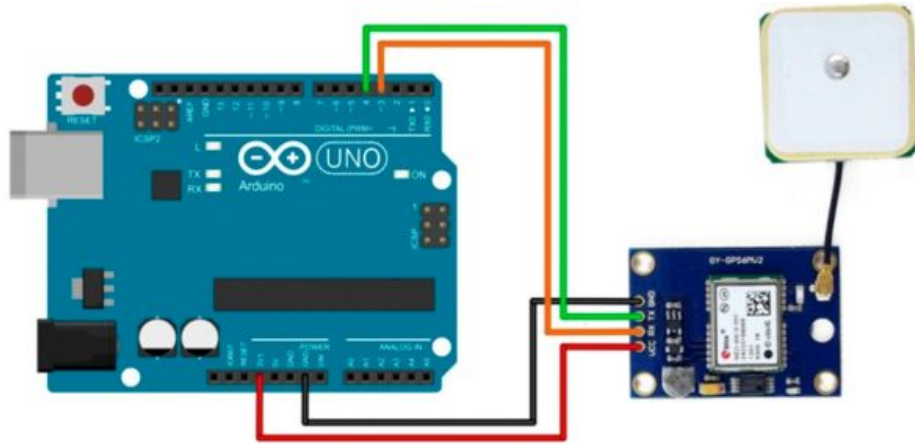


Figura 10

Sensor GPS NEO-6M conectado a arduino [15]

Podemos acceder al datasheet de este sensor a través del enlace: [https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_\(GPS.G6-HW-09005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf)

El **sensor de temperatura DS1820B** recogerá un valor cada diez minutos y se actualizará en los datos mostrados en la aplicación una media de la temperatura por hora, además de saltar una alarma si la temperatura pasa un valor umbral establecido (menos de 18 °C).

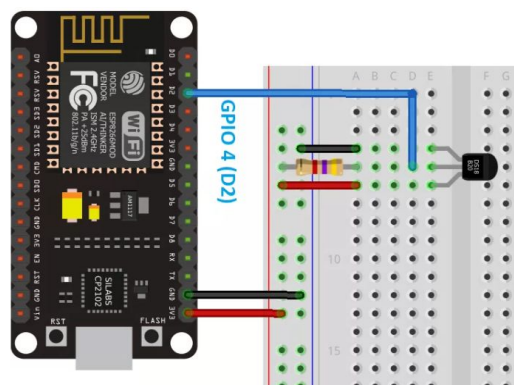


Figura 11

Sensor de temperatura DS1820B conectado a NodeMCU [16]

Podemos acceder al datasheet de este sensor a través del enlace: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

A continuación en la figura 9, se muestra el **esquema** de montaje de nuestra placa con todos los sensores mencionados anteriormente, hemos simulado el sensor magnético y el de presión con un botón debido a los problemas que ha causado el Covid-19, ya que las fechas de entrega de los sensores correspondientes eran demasiado altas.

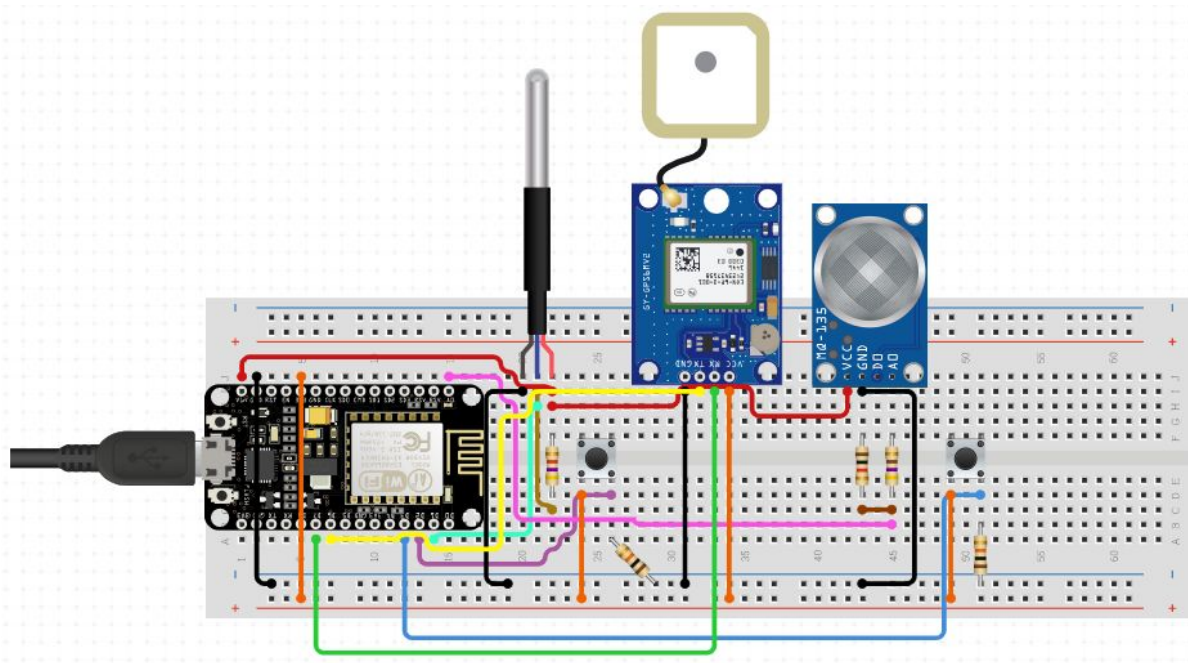


Figura 12

Esquema de los distintos sensores conectados la NodeMCU

3. Planificación de Arduino con la base de datos:

Para las conexiones de nuestra placa con el sistema de Base de Datos, hemos utilizado las librerías: "MySQL_Connection.h" y "MySQL_Cursor.h" las cuales hemos utilizado para conectar directamente a los servidores de base de datos sin necesidad de ningún intermediario.

// Credenciales de la Base de Datos

IPAddress server_addr("IP del server"); // IP del servidor MySQL

char user[] = "Nombre de usuario"; // MySQL user login username

char password[] = "Contraseña"; // MySQL user login password

// Creamos un cliente WiFi y lo asociamos a la conexión con la base de datos

WiFiClient client; // Creamos nuestro cliente WiFi

*MySQL_Connection conn((Client *)&client); // Y lo conectamos a la base de datos*

Y a través de la siguiente función incluida en **loop** mandamos la sentencia SQL a la base de datos, en este caso los datos utilizados del GPS:

// Grabar datos en la base de datos del GPS

void grabarDatoGPSBD (double latitud, double longitud, int idsensor) {

Serial.println("Mandando datos a sensor_gps");

if (conn.connect(server_addr, 3306, user, password)) {

char INSERT_SQL4[] = "UPDATE pr_healthtech.sensor_gps SET Latitude = %f, Longitude = %f, Date_Time_Activation = NOW() WHERE Sensores_ID3 = %d" ;

char query[128];

sprintf(query, INSERT_SQL4, latitud, longitud, idsensor);

*MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);*

cur_mem->execute(query);

delete cur_mem;

Serial.println("Registro grabado ");

conn.close();

}

else

Serial.println("Conexión a la base de datos MySQL fallida.");

}

Planificación

Hemos repartido las tareas a realizar en un tiempo total de 18 semanas, incluyendo el periodo de vacaciones de navidad, como se puede ver en la figura 8. Utilizaremos la metodología ágil SCRUM, en la que la duración de nuestros Sprints será de 2 semanas.

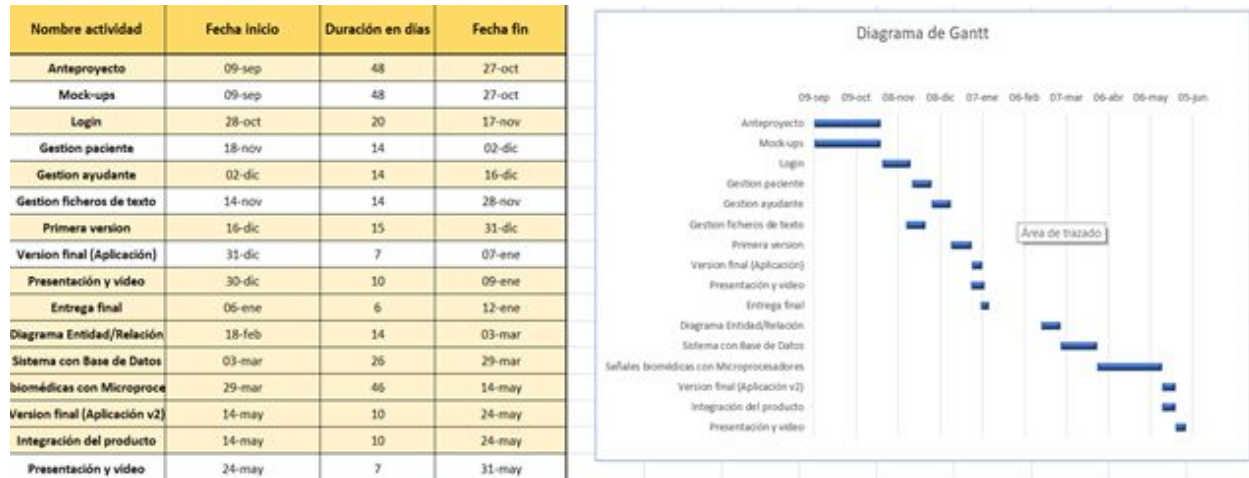


Figura 13

Tabla con las distintas fases del proyecto asociadas a un diagrama Gantt

El proyecto I estará en fase de planificación durante 3 Sprints, es decir, un total de 6 semanas, incluyendo los mock-ups iniciales para dibujar una idea clara de hacia dónde queremos que vaya el producto.

El cuarto sprint abordará la funcionalidad de la vista del Login para el usuario, mientras que el quinto y sexto sprint estarán más enfocados en las vistas de los distintos usuarios e incorporar sus funcionalidades. El séptimo sprint desarrollará la primera versión entregable del producto al cliente, que recorrerá las distintas vistas y contendrá la mayoría las funcionalidades requeridas basándose en el Product Backlog y sus historias de usuario.

Finalmente el octavo sprint pulirá los aspectos de la primera versión mientras que el noveno y último sprint tratará la entrega final del producto.

El proyecto II estará en fase de planificación durante 3 Sprints, es decir, un total de 6 semanas, incluyendo el diagrama Entidad/Relación a partir del cual realizaremos la base de datos.

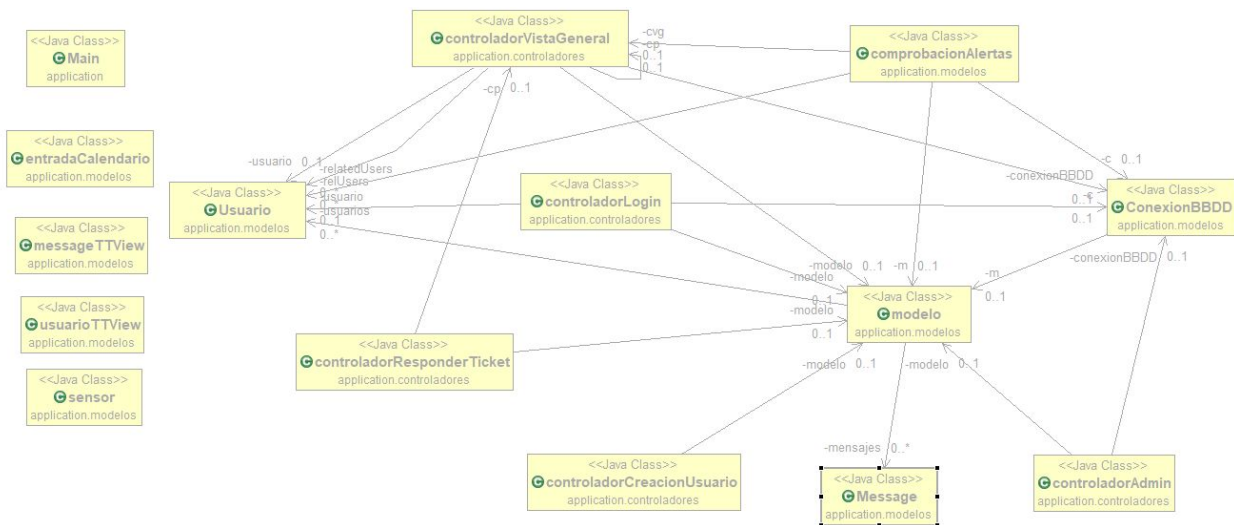
El cuarto sprint abordará la funcionalidad de la base de datos en SQL, mientras que el quinto y sexto sprint estarán enfocados a obtener las señales biomédicas desde procesadores y escribirlas en la base de datos. El séptimo sprint desarrollará la segunda versión del proyecto,



comprobando que todo funcione correctamente y haciéndola totalmente funcional para su presentación.

Finalmente, el octavo sprint integrará el producto final con todas sus funcionalidades, mientras que el noveno sprint tratará la entrega final del producto.

Las clases que no están relacionadas directamente con otra son las que hemos organizado a la izquierda, ya que son clases que usamos para crear objetos propios y nos sirven como recipientes para incorporar nuestros propios métodos y atributos (una entrada del calendario, un usuario en el tree table view al seleccionar un usuario).



El proyecto se ha elaborado siguiendo el patrón Modelo-Vista-Controlador (MVC), es decir, hemos separado las clases en tres apartados, con esto hemos podido separar la representación de la información, y por otro lado la interacción del usuario como se puede ver en ambos diagramas.

Es por ello que nos ha facilitado la reutilización de código y la separación de conceptos, ayudando en el desarrollo de la aplicación. Cabe destacar que también tenemos dos carpetas en las que localizamos todas nuestras biblioteca y recursos utilizados.

Optimización

En cuanto al algoritmo de TPA que hemos usado se trata de un quicksort para reemplazar el order by que teníamos (ordenar las entradas del calendario de un usuario), de forma que a pesar de tener una complejidad de $O(n^2)$ se puede reducir fácilmente para competir con otros algoritmos del tipo $O(n \log n)$ ya que requiere muy poco espacio extra en memoria (al contrario que mergesort) y escogiendo el pivote aleatoriamente (casi siempre evita el caso peor de $O(n^2)$).

```
private int partition(Vector<Integer> IDEntries, int begin, int end) {
    int pivot = IDEntries.get(end);
    int i = (begin-1);

    for (int j = begin; j < end; j++) {
        if (IDEntries.get(j) <= pivot) {
            i++;

            int swapTemp = IDEntries.get(i);
            IDEntries.set(i, IDEntries.get(j));
            IDEntries.set(j, swapTemp);
        }
    }

    int swapTemp = IDEntries.get(i+1);
    IDEntries.set(i+1, IDEntries.get(end));
    IDEntries.set(end, swapTemp);

    return i+1;
}

public void quickSort(Vector<Integer> IDEntries, int begin, int end) {
    if (begin < end) {
        int partitionIndex = partition(IDEntries, begin, end);

        quickSort(IDEntries, begin, partitionIndex-1);
        quickSort(IDEntries, partitionIndex+1, end);
    }
}
```

Figura 14
Algoritmo QuickSort()

Presupuesto

El proyecto se hará en 8 meses, los gastos se resumen a:

1. Horas de trabajo de tres ingenieros informáticos y un ingeniero biomédico:
 - 150 horas por semestre
 - 150×2 semestres = 300 horas por alumno
 - **TOTAL:** $300 \times 4 = 1200$ horas
2. Material comprado:
 - Sensor magnético: 1.40 € (Simulado con un botón)
 - Sensor de presión: 11.89 € (Simulado con un botón)
 - Hazardous Gas Sensor - MQ-135: 0.83 €
 - Ublox NEO-6M Módulo GPS: 2.99 €
 - DS18B20 1-Sensor de temperatura: 3.99 €
 - Ordenadores de desarrollo: 599.99 €
 - Servidores: 19.99 € / mes [12]
 - Node MCU 1.0V: 20.00 €
 - 100 Cables de soldadura PCB: 1.05 € / 100 piezas
 - Pack de resistencias: 5 €
 - BreadBoard: 7€
3. Gastos totales:
 - **TOTAL EN COMPONENTES POR PACIENTE:** $1.40 + 11.89 + 0.83 + 2.99 + 3.99 + 20.00 + 1.05 + 5 + 7 = 54.15$ €
 - **TOTAL EN EQUIPAMIENTO:** $(599.99 \times 4 \text{ ingenieros}) + (19.99 \times 12 \text{ meses}) = 2639.84$ €

Diagrama Entidad Relación

Al nosotros no tener una jerarquía de usuarios en las que tenemos una entidad distinta por el tipo de rol (médico, cuidador, familiar, paciente), los diferenciamos con el atributo ROL, por lo que para implementar bien las relaciones entre usuarios existen 6 relaciones de usuario a usuario del estilo n a n, las cuales al aplicar normalización (paso a tablas) se comprueba que las relaciones son correctas y no hay ningún problema como puede ser el de líneas (rows) duplicadas.

En la relación de usuario con sensores, no es necesario tener otra vez la jerarquía con entidades para cada rol diferente, si no que al normalizar en la tabla de sensores se quedaría como foreign key el ID del usuario, el cual como nosotros siempre introduciremos el valor ese ID iría asociado siempre a un paciente.

En cuanto a la entidad de Sensores está hecha para poder diferenciar bien los distintos tipos de sensores, ya que si quitamos esa entidad y conectásemos directamente cada sensor a la entidad de usuario y añadiésemos muchísimas filas de cada sensor sería imposible diferenciar de qué sensor estamos hablando, ya que con 3 se podría, pero si tuviésemos 500 pues obviamente no. Además la lectura de los datos se envía directamente a esta tabla, la cual luego tiene la relación con usuarios y soluciona el problema mencionado anteriormente.

En cuanto a la relación entre Usuario y Sensores se trata de un 1 a N puesto que 1 Usuario tiene N Sensores, mientras que 1 Sensor está compuesto por N sensores continuos, discretos y GPS. En cuanto a la "self-relation" entre usuarios y usuarios relacionados por mensajes, este link lo explica muy bien:

<https://stackoverflow.com/questions/35310283/how-to-model-messages-exchanged-between-users-er-diagram>

Al normalizar quedaría una tabla con una Primary Key que consiste de una tupla de los dos ID de usuario, con los otros atributos del mensaje (read, text etc.).

Añadiendo a esto la entidad "entrada_calendario" se ocupa de todas las entradas correspondientes al usuario, por ello sigue una relación 1-N del tipo usuario-entrada_calendario permitiéndonos incorporar un historial de entradas si es requerido (la agenda del inicio permite ver todos los eventos pasados y futuros cómodamente sin tener que acceder al calendario principal).

El diagrama Entidad-Relación se encuentra dentro de la carpeta, con el nombre "ER_HealthTech.pdf" en el cual se puede ampliar para verse mejor.

Además, hemos hecho el diagrama de normalización que se encuentra dentro de la carpeta con el nombre "Diagrama Normalización_HealthTech".



Conclusión y Líneas Futuras

Nuestros principales objetivos al comienzo del proyecto fueron, reducir el tiempo entre las comunicaciones del Paciente con su Médico, ya que generalmente solo se comunican durante el horario de consultas, y el tiempo entre éstas puede llegar a ser muy distanciado.

Este objetivo lo solucionamos mediante la implementación del sistema de mensajería basado en “tickets”, permitiendo al paciente u otros usuarios comunicarse con su médico, cuidador y también con sus familiares, en cualquier momento.

También pretendíamos proporcionar seguridad, tanto a familiares como a los pacientes, de que se encuentran conectados en todo momento en caso de emergencia.

Por lo que a través de nuestro sistema de monitorización basado en los sensores, ubicados en el interior de la casa, y un sensor GPS, que nos ayuda a localizarla en situaciones de necesidad, podemos alertar a los demás usuarios si se ha producido una incidencia, como podría ser una subida del nivel de gases nocivos en la cocina, para que se tomen las medidas oportunas.


Al ofrecer un registro de las mediciones de los sensores, el médico dispondrá de más información para analizar y poder reflexionar sobre la decisión más adecuada acerca del tratamiento del paciente.

Pretendemos, en próximas actualizaciones de la aplicación, añadirle la funcionalidad de contactar mediante llamada y videollamada con el médico, para realizar las consultas sin tener que desplazarse al centro de salud o a su hospital, ya sea porque su situación personal no lo permita o por condiciones excepcionales de seguridad, como es el caso actual con la crisis del COVID-19.

Además de esto, nos gustaría poder implementar más sensores para la casa, hasta ser domotizada en su totalidad, y proporcionar una medición de ritmo cardiaco al paciente, o un acelerómetro, para comprobar si se produce alguna caída, todo incluido en una aplicación móvil con las características principales como las alertas, mensajes y eventos del calendario.

Referencias bibliográficas utilizadas

- [1] PopulationPyramid.net, *World 2019* [en línea][fecha de consulta: 12 de octubre de 2019]
Disponible en: <https://www.populationpyramid.net/>
- [2] PopulationPyramid.net, *PopulationPyramid > Sources* [en línea][fecha de consulta: 12 de octubre de 2019] Disponible en: <https://www.populationpyramid.net/sources>
- [3] United Nations, *2019 Revision of World Population Prospects* [en línea][fecha de consulta: 12 de octubre de 2019]
Disponible en: <https://population.un.org/wpp/>
- [4] Instituto Nacional de Estadística [en línea][fecha de consulta: 12 de octubre de 2019]
Disponible en: <https://www.ine.es/welcome.shtml#>
- [5] Instituto Nacional de Estadística, *Consecuencias del envejecimiento de la población: el futuro de las pensiones*[en línea][fecha de consulta: 12 de octubre de 2019] Disponible en:
https://www.ine.es/daco/daco42/sociales/infosoc_envej.pdf
- [6] Ministerio de sanidad, servicios sociales e igualdad, *Gasto Sanitario Informe Anual del Sistema Nacional de Salud 2016* [Imagen digital en línea] Ministerio de sanidad, servicios sociales e igualdad [fecha de consulta: 24 de octubre de 2019] Disponible en:
https://www.msbs.gob.es/estadEstudios/estadisticas/sisInfSanSNS/tablasEstadisticas/InfAnualSNS2016/7Gast_Sanit.pdf
- [7] Gualdo,A. y Chamseddine,M.(2015). Estimulación de la memoria en ancianos con Alzheimer: Experiencia desde el Prácticum de Educación Social. *Revista de Educación Social* [en línea] V.1, no. 21, pp 1-2. [fecha de consulta: 23 de octubre de 2019]
Disponible en:
<http://www.eduso.net/res/21/articulo/estimulacion-de-la-memoria-en-ancianos-con-alzheimer-experiencia-desde-el-practicum-de-educacion-social>
- [8] Belver, M. El servicio de teleasistencia incluirá sensores en las neveras y en los colchones de los usuarios. *El Mundo*. [en línea] 2017, noviembre. [fecha de consulta: 23 de octubre de 2019] Disponible en:
<https://www.elmundo.es/madrid/2017/11/27/5a184adb268e3e912b8b4641.html>
- [9] Gil,S., Rodríguez,C. *Tecnología y personas mayores* [en línea][fecha de consulta: 23 de octubre de 2019] Disponible en:
http://www.ceapat.es/InterPresent1/groups/imsero/documents/binario/reto_8.pdf
- [10] Kookye, *Use arduino to drive mq-5 gas sensor* [Imagen digital en línea] Amber [fecha de consulta: 23 de octubre de 2019] Disponible en:
<http://kookye.com/2016/08/15/use-arduino-to-drive-mq-5-gas-sensor/>



[11] Elegoo, 37 Sensor Kit V2 [en línea] [fecha de consulta: 29 de mayo de 2020] Disponible en: <https://www.elegoo.com/tutorial/Elegoo%2037%20Sensor%20Kit%20Tutorial%20for%20UNO%20R3%20and%20Mega%202560%20V2.0.0.2020.03.10.zip>

[12] Ikoula, *Servidor dedicado* [en línea] [fecha de consulta: 26 de octubre de 2019] Disponible en: <https://www.ikoula.es/es/servidor-dedicado#starter>

[13] e-ika, Detector Magnético para Arduino [en línea] [Fecha de consulta: 2 de febrero de 2020] Disponible en: <https://www.e-ika.com/detector-magnetico-para-arduino-ky-025>

[14] Hetpro-store, *Sensor de Fuerza o Presión MF01* [Imagen digital en línea] Amber [fecha de consulta: 29 de mayo de 2020] Disponible en: <https://hetpro-store.com/TUTORIALES/sensor-de-fuerza-o-presion-mf01/>

[15] Planeta Electronico, *Modulo Arduino GPS NEO-6M* [Imagen digital en línea] Amber [fecha de consulta: 29 de mayo de 2020] Disponible en: <https://www.planetaelectronico.com/modulo-arduino-gps-neo-6m-p-18648.html>

[16] Randomnerdtutorials, *ESP8266 DS18B20 Sensor Web Server Arduino IDE (Single, Multiple)* [Imagen digital en línea] Amber [fecha de consulta: 29 de mayo de 2020] Disponible en: <https://randomnerdtutorials.com/esp8266-ds18b20-temperature-sensor-web-server-with-arduino-ide/>