

Long Life

MEMORIA DEL PROYECTO



GRUPO 7

Sergio Benavente Moreno
Laura Botija Manzano
Fernando Palarea Martín
Aldo Troccoli Montes de Oca



HEALTHY TECH

La tecnología al servicio de nuestra salud

El contenido de este anteproyecto, su propiedad intelectual y todos los derechos están reservados y protegidos en favor de la empresa HEALTHY TECH S.L.

Contenido

Resumen del proyecto	1
Introducción	2
España envejece	2
La Unión Europea también envejece	3
Un problema global.....	4
Estado del arte	5
Long Life frente a la competencia	6
Objetivos generales.....	7
Objetivos y requisitos del proyecto de Informática II.....	7
Mensajería.....	7
Base de datos	7
Sistema de alertas.....	8
Desarrollo Arduino	8
Módulo de monitorización.....	8
Requisitos técnicos hardware	10
Recursos materiales.....	10
Sensor de báscula.....	10
Sensor de temperatura.....	11
Sensor de pulso	12
Baremos estipulados en los sensores	12
Temperatura	12
Horas de sueño	13
Implementación de la base de datos.....	14
Análisis de las entidades	14
Relaciones entre entidades.....	18
Diagrama de entidad-relación.....	19
Plataforma utilizada de la base de datos	20
Entorno de desarrollo de la base de datos.....	21
Configuración del MySQL Workbench.....	21
Tablas de datos en MySQL.....	22
Procedimientos almacenados	26
Desarrollo hardware con Arduino	30

Monitorización de la temperatura	31
Monitorización del tiempo que pasa en la cama.....	31
Conexión Arduino con servidor web	32
Implementación del código en Java	33
Configuración del entorno	33
Integración con la base de datos	33
Concurrencia.....	33
GUI	34
Maquetación y componetización	35
Diagrama de clases	39
Manual de usuario de la interfaz gráfica	40
Funciones extras de Long Life.....	41
Botón de Ayuda.....	41
Planificador de actividades.....	41
Sistema de alertas	41
Listado de usuarios	42
Pantalla de navegación inicial.....	42
Pantallas de navegación personal sanitario.....	43
Tarjetas de Pacientes.....	44
Pantallas de navegación familiar o tutor.....	50
Pantallas de navegación del paciente dependiente.....	53
Repositorio Github	55
• Proyecto JavaFx con Maven	56
• Aplicación Arduino Sensor Temperatura DHT11:.....	56
• Aplicación Arduino Sensor Temperatura Dallas:.....	56
• Aplicación Arduino Sensor Báscula:	56
• Aplicación web php para enviar los datos recogidos por los sensores a la base de datos:.....	57
• Script de creación de la base de datos:	57
• Ejecutable de la aplicación:	57
Organización del trabajo cooperativo y colaborativo.....	58
Posibles vías de desarrollo e investigación en futuras versiones	59
Presupuesto	60

Instalación de la plataforma	61
Dependencias del proyecto definidas en Maven:	63
Metodología.....	64
Referencias bibliográficas.....	65
Imagen corporativa y diferentes alternativas futuras.....	66
Guía de solucionado de errores. FAQs	67
Paciente	67
Tutor.....	67
Especialista sanitario.....	68

Resumen del proyecto

Long Life nace con el propósito de afrontar, sanitariamente, una problemática mundial caracterizada por el envejecimiento de la población en nuestro planeta. Con esta aplicación, se pretende facilitar la vida cotidiana de nuestros mayores, garantizar sus cuidados de manera remota y aportar información relevante al sistema sanitario y los tutores designados de cada paciente, con el objetivo de conseguir un menor grado de dependencia y mayor libertad en las personas dependientes.

Mediante el uso de sensores y sistemas domóticos se podrán monitorizar una serie de parámetros que nos permitirán conocer los hábitos y el estado de salud de los pacientes. Además, Long Life dispone de un servicio de mensajería instantánea con el que los pacientes podrán contactar, en todo momento, con su doctor, familiar y/o tutor. Un sistema de alertas y un botón de ayuda nos advertirá de cualquier incidencia. Asimismo, el calendario incluido en esa aplicación permitirá recordar al paciente sus tareas diarias.

El planeta se está envejeciendo y se requieren de diferentes alternativas para ofrecer la mejor calidad de vida a nuestros mayores. Long Life se posiciona como una opción que cubre muchas de estas necesidades en beneficio del paciente, sus familiares y sistema sanitario.

El desarrollo de Long Life es realizado por Healthy Tech S.L.. Se trata de una empresa especializada en el desarrollo de soluciones informáticas en el ámbito de la salud, permitiendo poner la tecnología en beneficio de los avances sanitarios.

En esta ocasión, Healthy Tech S.L propone el desarrollo de la aplicación “Long Life”, una solución inteligente y sostenible con la que se pretende monitorizar a las personas dependientes con el objetivo de facilitar sus cuidados sanitarios y conseguir un mayor grado de libertad en los pacientes con un control y cuidado prácticamente remoto. La aplicación pretende facilitar la comunicación entre familiar y paciente, así como la comunicación y cuidados desde el personal sanitario con sus pacientes.

Este proyecto se enmarca en los dominios del programa “Active Assisted Living (AAL)” , con el que se pretende mejorar la calidad de vida de las personas mayores mediante el desarrollo de soluciones tecnológicas, debido al alarmante envejecimiento de la población en Europa.

Introducción

Imaginemos un robot moviéndose por el hogar de una anciana. Acompañándola en cada movimiento, reconociendo la actividad que realiza. Recordándole cuando debe tomarse la medicación o advirtiéndole de algún comportamiento nocivo como llevar mucho tiempo en la cama, sentada en el sillón o sin hacer ejercicio. Imaginemos que ese robot nos avisa de si la anciana sale del hogar, deja las ventanas abiertas en días de lluvia y frío o nos manda información del pulso, la respiración, la temperatura u otros valores vitales de esta señora. Sigamos imaginando que ella puede ponerse en contacto, fácilmente, con un familiar que está fuera del hogar o con un personal sanitario si se encuentra mal o tiene alguna urgencia. ¿Sería ideal, verdad?

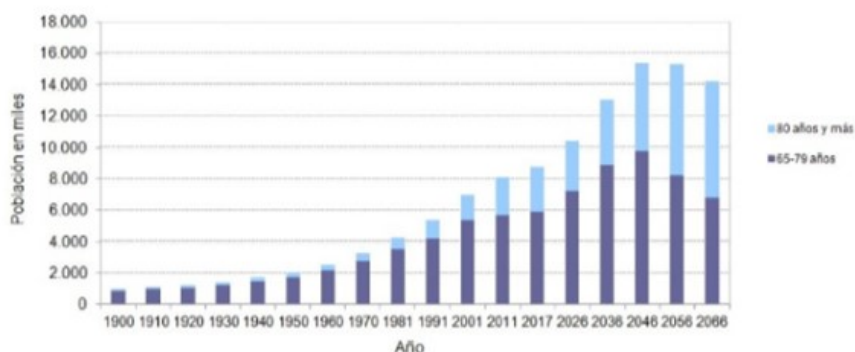
Pues no hace falta imaginar tanto, ya que es una realidad. Las soluciones tecnológicas al servicio y beneficio de la sanidad han permitido el desarrollo de innumerables dispositivos y aplicaciones para el cuidado de las personas mayores, que permiten una mayor independencia a este sector de la población que cada vez es mayor en consecuencia a una población que se ha envejecido globalmente.

España envejece

La edad promedio de la población en España en 2019 es de 43,4 años, frente a los 33.5 años de hace 40 años según los indicadores de estructura de población en España publicados el 25 de junio de 2019 en el Instituto Nacional de Estadística. Unos datos alarmantes que evidencian que España envejece. El incremento de la esperanza de vida y la notable disminución de la natalidad en nuestro país conlleva que, cada año, la edad promedio de nuestra población sea más elevada.

Las personas mayores de 65 años representan casi la quinta parte de la población en España (19'4%), casi diez puntos más que hace 50 años, es decir, hace varias décadas uno de cada diez españoles tenía 65 años, frente a la actualidad en la que se sitúa a uno de cada 5 años.

Figura 1.1.-Evolución de la población mayor. España, 1900-2066

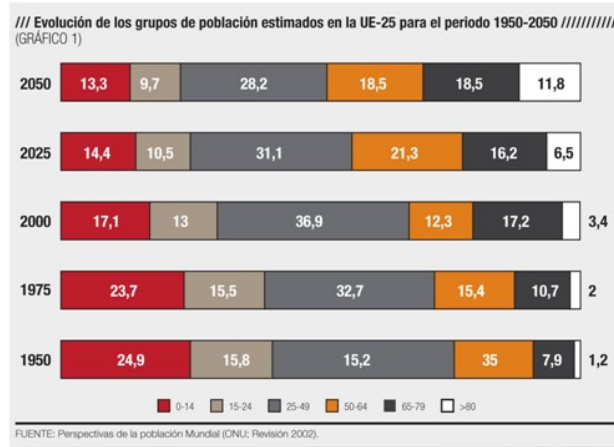


* De 1900 a 2016 los datos son reales; de 2026 a 2066 se trata de proyecciones.
Fuente: INE: INEBASE: 1990-2011: Censos de Población y Vivienda. 2016: Estadística del Padrón Continuo a 1 de enero de 2017. Consulta enero 2018; 2026-2066: Proyecciones de población. Consulta en enero 2018

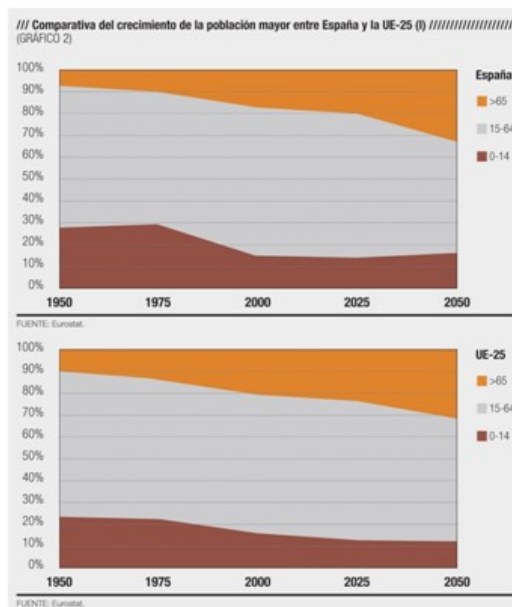
En nuestro país, en el 2050, las personas mayores de 65 años representarán más del 30% de la población y los octogenarios podrían superar la cifra de cuatro millones de personas.

La Unión Europea también envejece

Para enmarcar las cifras del envejecimiento en el contexto de la Unión Europea, se muestra la proyección realizada por el Eurostat en el año 2004 para los 25 países miembros permite enmarcar las cifras de envejecimiento de en la Unión Europea. Divididos por grupos de edad de la población durante un periodo de tiempo de 100 años, desde 1950 hasta 2050. Los grupos de edad más avanzada, de 65 a 79 años y de 80 años en adelante aumentan considerablemente pasando de un 10% en 1950 a un tercio un siglo después. La perspectiva europea, de Par Asghar Zaidi: <http://www.euro.centre.org>).

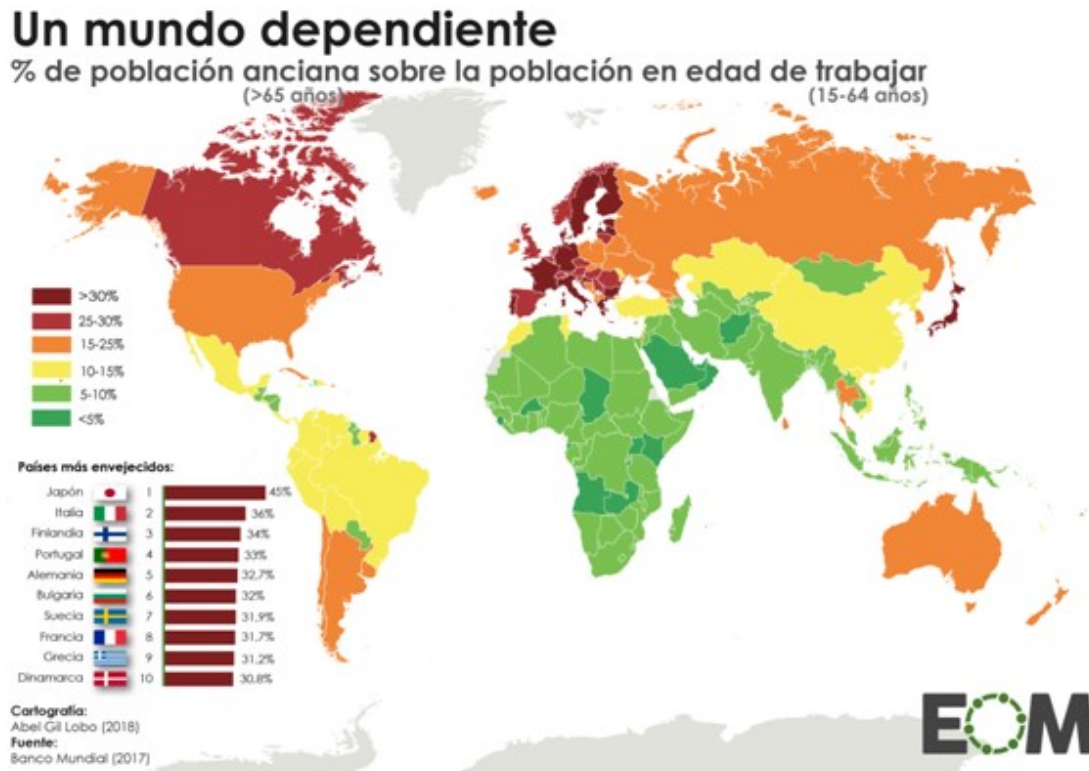


Una comparativa de España con el resto de países de la UE manifiesta que en nuestro país el sector de la tercera edad aumentará más rápidamente que en otros países.



Un problema global

El envejecimiento de la población, con todos los costes y problemas que ello acarrea, es uno de los mayores retos que muchos países, especialmente desarrollados, enfrentan de aquí a unas décadas.



La tecnología en beneficio de este fenómeno conseguirá aportar un granito de arena en beneficio del bienestar y cuidados sanitarios de este sector creciente en nuestra sociedad.

Estado del arte

En 2014, el Parlamento Europeo y el Consejo de la Unión Europea dieron continuidad al “Programa Active Assisted Living Research and Development Programme”, vigente entre 2007 y 2013, dando lugar al programa “Active Assisted Living” (ALL).

Ambos programas contaban con el mismo denominador común, mejorar la calidad de vida de las personas mayores mediante el desarrollo y utilización de soluciones innovadoras sostenibles basadas en las tecnologías de la información y la comunicación (TIC) que respondiesen a la necesidades con un punto de vista multidisciplinar. Este programa, en el que participan una veintena de países de la Unión Europea y también Suiza, se basa en el artículo 185 y 188 del Tratado de Funcionamiento de la Unión Europea y englobado dentro del Horizonte 2020.

Su actividad más importante es la publicación periódica de convocatorias de propuestas de realización de proyectos que cumplan los requisitos y la génesis del programa, mejorando la calidad de vida de las personas mayores, la calidad y eficiencia de los sistemas sanitarios, el bienestar social; y por otro lado, potenciar y fortalecer el desarrollo industrial europeo en el marco sanitario.

Desde entonces, la AAL ha financiado centenares de proyectos de colaboración que trabajan en productos que han aportado resultados notables y varios avances que se han implantado en toda Europa para marcar una diferencia real en la vida de las personas mayores.

Existen, por tanto, varias aplicaciones en el mercado y proyectos en desarrollo que podríamos categorizar según sus principales fundamentos y características en “E-inclusión”, interacción social, monitorización remota, recordatorios, generación de alarmas, análisis de comportamientos, asistencia autónoma...

Algunas de estas aplicaciones y proyectos vigentes y aplicados en Europa son “Mylife”, “HOST”, “MyTherapyAPP”, “GoldUP”, “BePrevent”, “HERA”, “CO-LIVING”, “HOMEdotOLD”, “FOSIBLE”, “GO-MY-LIFE”, “ELISA”; “AMICA”; “ALADDIN”, “PAMP”, “eCAALYX”, “ISACTIVE”, “DOME0”, “HELP”; “ALIAS”, “MedisafeAPP”... Muchos de estos proyectos o aplicaciones han sido financiado gracias a la AAL y pueden conocerse en <http://www.aal-europe.eu/projects-main/>

Aunque se disponen de muchos proyectos y aplicaciones en el mismo ámbito sus soluciones son parciales y no ocupan todos los requisitos y necesidades expuestos por el cliente, siendo simplemente soluciones parciales; no pudiendo disfrutar de ninguna aplicación que nos permita la monitorización de un paciente, su supervisión, oferta de cuidados continuos y la comunicación entre el paciente, su familiar y personal sanitario.

Entendemos, por tanto, que debemos aunar los esfuerzos en potenciar las características diferenciadoras de este proyecto respecto a lo que oferta el mercado, seguir manteniendo un estudio constante de la evolución del sector y desarrollar mejoras,

potenciando fortalezas y minimizando debilidades del producto final que deseamos aportar a nuestros mayores. Esas son las bases de “Long Life”.

Long Life frente a la competencia

	MÉDICO	FAMILIAR	MULTIUSUARIO	ESTADÍSTICA	SOS	CALENDARIO	MENSAJERÍA 2.0
LONG LIFE	✓	✓	✓	✓	✓	✓	✓
MyLife	✗	✗	✗	✓	✗	✓	✗
MyTherapyAPP	✓	✗	✗	✓	✗	✓	✗
Be Prevent	✓	✓	✓	✓	✗	✓	✗
Medisafe APP	✓	✓	✗	✓	✗	✓	✓

Con el objetivo de impulsar diferentes mejoras y funcionalidades en nuestra aplicación, hemos analizado varias aplicaciones semejantes que están en el mercado y pueden suponer una competencia. Mediante el sistema de semáforos hemos advertido las posibilidades que sí ofrece Long Life y que no garantiza la competencia.

Objetivos generales

- Mejorar la independencia de los mayores.
- Ayudar a las personas mayores a participar en actividades sociales, en particular a los que viven solos.
- Incrementar la participación de los sectores objetivo en asistencia basada en ICT.
- Ofrecer información significativa a profesionales de la salud, cuidadores, familiares, psicólogos, diseñadores de sistemas...
- Diseñar una aplicación que permita la conexión de los mayores con sus familiares y personal sanitario para sus cuidados
- Facilitar los recordatorios y agenda referente a cuidados de la salud de los pacientes de la tercera edad
- Aumentar la autoestima y autoeficiencia de los ancianos
- Ofrecer una solución integral que permita detectar posibles incidencias del paciente al que se le hace seguimiento, prevenir y atender.

Objetivos y requisitos del proyecto de Informática II

Entre los objetivos y requisitos planteados para el proyecto de informática II encontramos los siguientes puntos:

Mensajería

- Sistema de mensajería que se describe a lo largo de la memoria, para garantizar la comunicación entre los distintos usuarios. Para este desarrollo, necesitaremos:
- Desarrollado en Java la lógica de funcionamiento del chat, para que cada usuario pueda comunicarse con sus vínculos.
- Conexión con la base de datos, para almacenar los mensajes correspondientes a cada usuario y para que así estos puedan visualizarlos.

Base de datos

- Garantizar la persistencia y accesibilidad a los datos de la aplicación utilizando una base de datos relacional SQL. Como requisitos:

- Diseño de la base de datos, identificando todas las instancias, atributos de cada instancia y las relaciones existentes de ellas.
- Definición de la base de datos a utilizar (Mysql.) y que instalamos en el cloud para que este siempre operativa al 99.99%
- Desarrollo de las clases encargadas de crear la conexión entre la base de datos y el programa Java.

Sistema de alertas

- Desarrollo correspondiente a nuestro sistema de alertas mediante el cual la aplicación advertirá de las anomalías registradas por los distintos sensores conectados a nuestra herramienta:
- Conexión de programa java con la base de datos
- Desarrollo de consultas que nos permitan evaluar los datos
- Definición de la alerta a mostrar en pantalla en función de los valores leídos nuestros sensores

Desarrollo Arduino

- Desarrollo en arduino de toda la lógica de funcionamiento de los sensores y cómo se recolecta y se lleva a cabo la ingesta de datos en nuestra base de datos, para lo cual necesitaremos:
- Todos el hardware mencionado en la propuesta técnica (Placas de arduino & sensores).
- Desarrollo en arduino de la lógica de interacción de los sensores con las personas dependientes, para la correcta recolección de datos.
- Desarrollo de la conexión de las placas de arduino con nuestra base de datos por medio de wifi.

Módulo de monitorización

- Nos conectaremos a nuestra base de datos alojada en la nube de Amazon.
- Se llevan a cabo consultas preparadas para cada una de las distintas gráficas que mostraremos en nuestra aplicación.
- Pintamos las gráficas con los datos obtenidos en la consulta.

- **Manual de Usuario:** Nuestra aplicación contará con un manual de usuario que aglutinará de una forma sencilla y rápida los conceptos clave de usabilidad de nuestra herramienta, para que tanto los asistentes de la aplicación, los familiares, así como nuestros protagonistas, las personas dependientes, puedan comprender rápida y fácilmente el correcto funcionamiento de la herramienta.

- **Vídeo promocional del producto y campaña de difusión.** Desarrollo nuestra campaña publicitaria, definiendo el guión y puesta en escena del video promocional y el plan de difusión para poder captar a los máximos clientes posibles. Para ello haremos una puesta en escena mostrando a nuestros clientes todas las características y beneficios que le brindará nuestra tecnología. Además, estableceremos el plan de acción para hacer nuestra herramienta la más popular del mercado.

Requisitos técnicos hardware

La aplicación para desarrollar debe obtener los datos de los sensores instalados en el hogar para monitorizar ciertos parámetros de las personas dependientes, es decir, poder ser interpretados, exponer alertas si algún valor fuese anómalo y faciliten el control y cuidados sanitarios, a distancia, de los ancianos.

Por tanto, existe una parte basada en el hardware, en los que utilizaremos sensores y la tecnología Arduino y, por otro lado, se realizará una aplicación en Java que será la encargada de recoger los datos, interpretarlos, mostrarlos y permitir una interacción entre personal sanitario, familiar o tutor, y el paciente.

Recursos materiales

En el presente proyecto llevaremos a cabo la instalación de varios sensores que nos ayudarán a poder asistir a la persona dependiente con mayor eficiencia.

A través de estos sensores no sólo podremos recopilar información acerca del comportamiento de la persona dependiente, sino también los cambios que este pueda experimentar.

Además, gracias a la conexión con su contacto de emergencia y su asistente personal, se podrán establecer alertas si los datos nos revelan anomalías sustanciales, pues pueden significar que el usuario puede encontrarse en peligro.

Los módulos que implantamos son los siguientes:

Sensor de báscula

Se trata de una báscula. Este sensor puede llegar a tener varias aplicaciones. Situado en distintos escenarios, bien en el sofá / sillón del salón podremos obtener datos relativos al grado de actividad del sujeto, o bien, si lo situamos en la cama, podremos llevar a cabo la trazabilidad, no sólo de las horas de sueño, sino también de los hábitos de sueño (acostarse / levantarse). Indicadores que nos puede ayudar a estudiar:

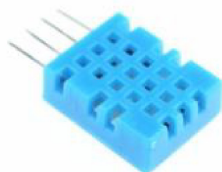


En función de todas las medidas anteriores nos dispondremos a generar las siguientes alertas críticas al contacto de emergencia:

- Si permanece en la cama por un tiempo superior a 14 horas
- Si permanece con inactividad en el salón por un periodo superior a 9 horas.
- Si en la hora normal de acostarse/levantarse sufre una variación sustancial
- Si se levanta numerosas veces durante el descanso nocturno durante un periodo continuado superior a 7 días

Sensor de temperatura

El sensor de temperaturas nos dará los datos referentes a la temperatura de la vivienda del sujeto. Con este sensor podremos conocer si las condiciones climáticas en la vivienda son idóneas para el dependiente.



En cuanto a los datos que podremos recopilar son:

- Temperatura del hogar durante el transcurso del día
- Fuertes variaciones de la temperatura que pueden lastimar la salud del dependiente

En función de todas las medidas anteriores nos dispondremos a generar las siguientes alertas críticas al contacto de emergencia:

- Si la temperatura supera los 35 °
- Si la temperatura es inferior a 15° o Vibraciones anómalas como consecuencias de caídas

Además de los sensores contemplados anteriormente, como futura mejora del proyecto, valoraremos la opción de explorar la viabilidad e instalación del siguiente sensor para detectar las caídas:

Sensor de pulso

Se trata de un sensor óptico, que basa su funcionamiento en el distinto comportamiento que la sangre tiene ante la luz, en función de su grado de saturación de oxígeno.

Incorpora dos LED, uno de espectro rojo y otro de infrarrojo. Su funcionamiento consiste en ponerse sobre la piel, por ejemplo en el dedo o la muñeca del paciente. El sensor detecta la luz reflejada, y determina el grado de saturación. Estos son los datos que recogeremos de cada uno de los pacientes permitiendo realizar un seguimiento del estado de salud diario, semanal y mensual.



Baremos estipulados en los sensores

En el desarrollo de los sensores y después de un estudio minucioso con personal sanitario, se han tomado unos baremos óptimos que se indican a continuación,

Temperatura

NORMAL	entre 17 y 31 de media
WARNING	min 13 y 16 max 32 y 35
DANGER	min < 13 max > 35

Horas de sueño

NORMAL	entre 6 y 9 horas
WARNING	min 4 y 6 - max 9 y 12
DANGER	min < 4 - - max > 12

Pulso

NORMAL	entre 60 y 80
WARNING	min 50 a 60 - max 80 y 100
DANGER	min < 50 max > 100

Toda esta información es recibida tanto por el médico y el tutor. Se han llevado a cabo no solo avisos por escrito sino también visuales que facilitan la rápida reacción sanitaria hacia el paciente. Permitiendo en algunos casos hasta salvar la vida.

Implementación de la base de datos

Nuestra aplicación Long Life consigue la persistencia de su sistema mediante integración con una base de datos. Los registros aportados por los sensores, las fichas de pacientes, doctores y sanitarios, así como otros parámetros son registrados en la base de datos.

Análisis de las entidades

Para conseguir la persistencia de los datos de nuestro sistema, hemos diseñado una base de datos que nos permite el registro ordenado y el acceso eficiente a toda la información.

Teniendo en cuenta todas las funcionalidades definidas en Long Life, y tras realizar un ejercicio de exploración, análisis, y definición de requerimientos, hemos podido identificar las siguientes entidades: *(Dado que vamos a ir introduciendo una breve descripción de cada entidad y sus atributos, hemos considerado apropiado adelantar al lector el tipo de campo que utilizaremos en la base de datos, aunque este punto podrá analizarse en los script de la creación de la base de datos que adjuntamos en el presente ejercicio)*

USUARIOS (id, idRol, nombre, fechaNacimiento, dni, password, email, teléfono,dirección)

Esta instancia recogerá todos los datos personales, de contacto, así como de acceso de todos los usuarios con los que contará la herramienta.

- Id: Correspondiente a la identificación del usuario. Será un valor entero (int) que se auto incrementará conforme más usuarios tengamos. Además, este atributo será la clave principal e identificativa de nuestra instancia Usuarios.
- idRol: Se trata de un atributo que va a definir el papel que ocupará cada usuario en nuestro sistema y que será el que determine cómo debe de comportarse la interacción de nuestra tecnología con el usuario.

Hemos decidido utilizar otra instancia a parte en la que definiremos los distintos roles de usuario, por tanto, el valor de este atributo se relaciona con la instancia de roles. Este campo igualmente será un entero (int)

- Nombre: En el que se almacenarán los nombres y apellidos de los usuarios. Para este campo hemos usado un varchar(100) que sea capaz de almacenar una cadena de caracteres suficiente para registrar cualquier nombre.
- Fecha de nacimiento: Utilizaremos el tipo de formato timestamp
- DNI: que será el que estaremos utilizando en el login de nuestra aplicación. Para este campo hemos utilizado un varchar(45)

- Password: La contraseña del usuario para acceder a la herramienta. Dado que las contraseñas pueden ser largas y estar encriptadas, hemos utilizado un longtext para evitar problemas en el almacenamiento.
- Teléfono: Datos de contacto con nuestros usuarios. Usaremos un varchar(45).
- email: hemos utilizado un varchar(100).
- Dirección: datos de contacto de nuestro usuario, usado un varchar(100).

ROLES (id, rol)

Recordemos que tenemos identificados 3 tipos de usuario, que en función del tipo de usuario, este podrá tener diferentes funcionalidades dentro del sistema:

1. Especialista
2. Paciente
3. Tutor

Esta entidad nos ayudará a definir los distintos tipos de usuario de un modo ordenado, con la posibilidad de poder escalar en un futuro nuestra herramienta y así poder añadir más tipos de usuario.

- Id: Valor autoincrement int
- Rol: Para el que utilizaremos un varchar(45)

EXPEDIENTES (id, idPaciente, idTutor, idEspecialista, estadoAlerta, personaPaciente)

Esta entidad será clave para nuestro programa, pues establecerá las relaciones y asociaciones existentes entre nuestros usuarios (paciente, tutor, y especialista), y el estado de alerta que en función de si el paciente lo tiene activado o no, condicionará a todos los integrantes del expediente.

De este modo, podemos garantizar la correcta relación entre todas las entidades además de asegurar la integridad en nuestros datos.

- Id: Int autoincrement. Partimos del 1000 para diferenciar los id de esta entidad con respecto a los del resto
- idPaciente: utilizaremos un valor int
- idTutor: int
- estadoAlerta: Este valor podrá ser 0 ó 1, Usaremos un int para registrar el estado.
- personaPaciente: Este campo identifica a los diferentes pacientes. Utilizaremos un valor float.
- idEspecialista: Este campo identifica a los diferentes especialista. Utilizaremos un valor int.

CALENDARIOS (id, idExpediente, idActividad, fecha)

En esta entidad se reflejarán los datos relacionados con el calendario de actividades del expediente, en donde el tutor o el mismo paciente podrá añadir nuevos eventos. En cuanto a los atributos localizados podemos encontrar los siguientes:

- Id: Int autoincrement. Cada calendario tendrá su identificador, que será la clave principal de la entidad.
- idExpediente: (int) utilizaremos un valor int, que será el que usaremos para poder relacionarlo con la entidad de expedientes
- idActividad: (int) sera el id de la actividad insertada en nuestro calendario.
- fecha: De acuerdo con el manejo de fechas de nuestra aplicación usaremos un campo de tipo timestamp.

ACTIVIDADES (id, actividad)

Para añadir nuevos eventos deberemos de determinar el tipo de evento que vamos a establecer (como por ejemplo tomar medicina, hacer ejercicio, descansar ..etc). Se trata de actividades predefinidas que nos ayudarán a hacer más ágil la inserción de eventos en nuestro calendario. En esta entidad pretendemos reflejar todos los tipos de actividad que usaremos en nuestro sistema. Gracias a este método podemos hacer escalable el manejo de actividades. Los atributos con los que contarán:

- Id: (Int auto-increment), Será el identificador y clave primaria de la entidad.
- actividad: (varchar (45)) Correspondiente al nombre de la actividad en cuestión (Tomar medicina, Hacer ejercicio, Cita médica...).

SENSORES (id, idExpediente, IdTipo)

Esta entidad consolida todos los sensores utilizados en cada expediente. Aunque si bien estaremos utilizando sólo 3 sensores por cada paciente, este diseño hace más dinámico el control de los sensores, pudiendo escalar nuestra aplicación en caso de tener que añadir nuevos sensores que permitan monitorizar otros aspectos de nuestros dependientes.

Cada sensor tendrá su propio id (que será la clave principal de esta instancia), además de una descripción del tipo de sensor y un expediente asociado. De este modo tendremos control sobre todos los dispositivos utilizados.

- Id: valor int
- idExpediente: int. Utilizaremos un valor int, que será el que usaremos para poder relacionarlo con la entidad de expedientes
- idTipo: valor int que pertenece al id del tipo de sensor.

TIPOSENSORES (id, tipo)

Gracias a esta entidad podremos definir cada uno de los sensores que podremos utilizar en nuestro sistema. Con esta entidad podremos escalar cuanto queramos la adición de nuevos sensores.

- id: utilizaremos un valor int autoincrement que nos servirá de clave principal de la entidad de los tipos de sensores.
- tipo: Será una cadena de caracteres varchar(45), y nos brindará el tipo de sensor(TEMPERATURA, PESO, MOVIMIENTO ETC.)

VALORESENSORES (id, valor, fecha)

Esta instancia será la responsable de reunir toda la información relacionada con los valores proporcionados por nuestro sistema de monitorización de los usuarios dependientes. Desde el Arduino, se irán introduciendo los datos de los sensores automáticamente en función del tiempo de lectura establecido en la lógica de negocio de nuestro sistema de monitorización del paciente.

- Id: Será la identificación de cada sensor del que estamos almacenando el valor obtenido.
- Valor: El valor leído por el sensor. Para que este campo pueda servirnos para almacenar cualquier lectura de cualquier sensor, hemos decidido establecerlo como un float.
- Fecha: Fecha de la lectura del valor proporcionado.

MENSAJES (idEmisor, idReceptor, mensaje, fecha,estadoMensaje)

Esta entidad será la que nos hará posible la comunicación a través de la aplicación entre los miembros de cada expediente.

Hemos apostado por la simplificación en la arquitectura de comunicación de nuestra herramienta, generando una entidad que recoja la actividad de los usuarios.

Para ello los datos que necesitaremos se reducirán al id del emisor del mensaje, el id del receptor, el mensaje enviado, y la fecha del mismo.

- id: int
- idEmisor: int
- idReceptor: int
- Mensaje: Usaremos un long text
- Fecha: timestamp
- estadoMensaje: timestamp
-

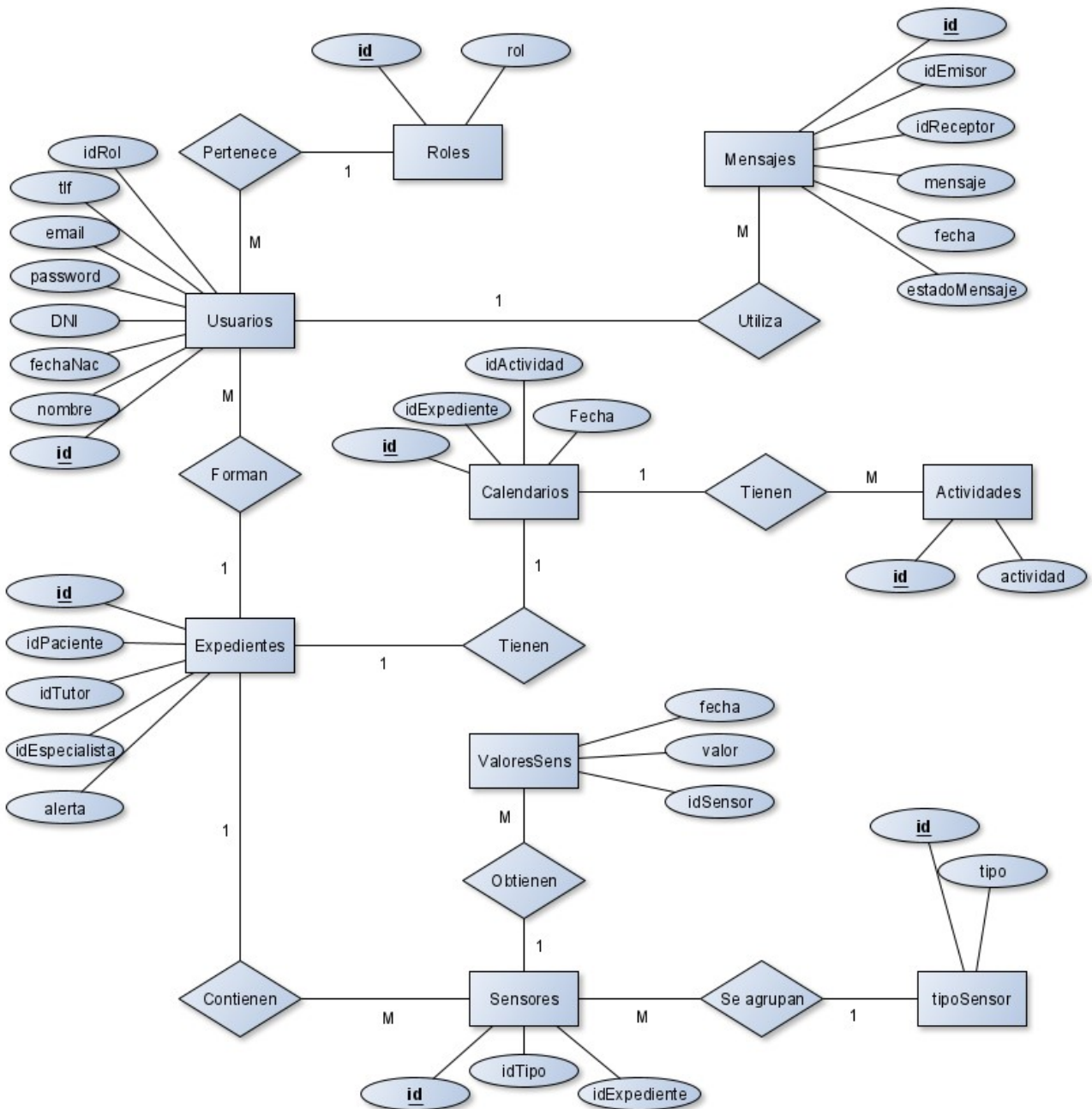
Relaciones entre entidades

Tras analizar todas las entidades, pasamos a identificar algunas de las relaciones existentes entre ellas:

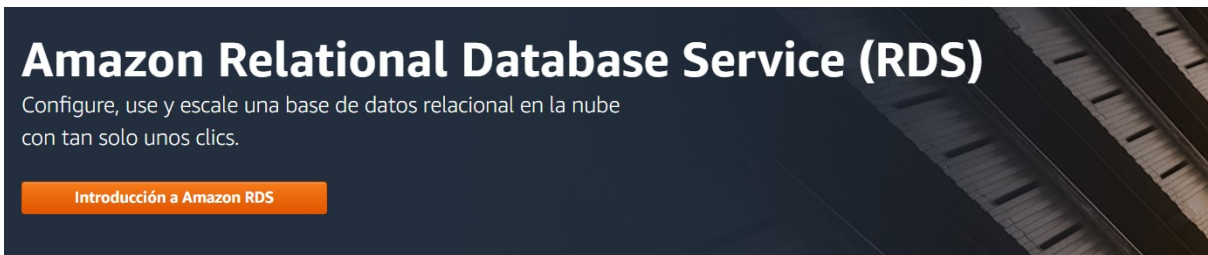
- **Pertenecen(Usuarios,Roles) -> Muchos a uno**
Dado que varios usuarios pueden pertenecer a un mismo rol
- **Envían (Usuarios, mensajes) -> muchos a muchos**
Varios usuarios pueden enviar varios mensajes
- **Forman (Usuarios, expedientes) -> muchos a uno**
Varios usuarios forman un expediente
- **Contienen (Expedientes, Sensores) -> uno a muchos**
Cada expediente tiene varios sensores
- **Obtienen (Sensores, valoresSensores) -> uno a muchos**
Cada sensor puede obtener muchos valores
- **Se agrupan(Sensores, tipoSensor) -> Muchos a uno**
Porque varios sensores se pueden agrupar en un mismo tipo de sensor
- **Tienen (Expedientes, Calendarios) -> Uno a uno**
Pues cada expediente tendrá asociado su propio calendario
- **Tienen (Calendarios, Actividades) -> Uno a muchos**
Cada calendario posee muchas actividades

Diagrama de entidad-relación

Una vez identificadas las instancias que estaremos utilizando en nuestra base de datos , sus atributos, y las relaciones existentes entre ellas, hemos desarrollado el siguiente diagrama de entidad – relación:



Plataforma utilizada de la base de datos



De cara a facilitar el desarrollo de la aplicación, la conectividad entre los distintos miembros del equipo, la escalabilidad y seguridad de la herramienta así como asegurar que nuestros clientes puedan acceder sin problemas, nos disponemos a crear una instancia de base de datos MySQL dentro del entorno de los servicios de bases de datos relacionales (RDS) de AWS, pues se trata de servicios sencillos de configurar, utilizar, y altamente escalables.

Endpoint: long-life-techbd.cb07dfg9wchf.us-east-1.rds.amazonaws.com

Port: 3306

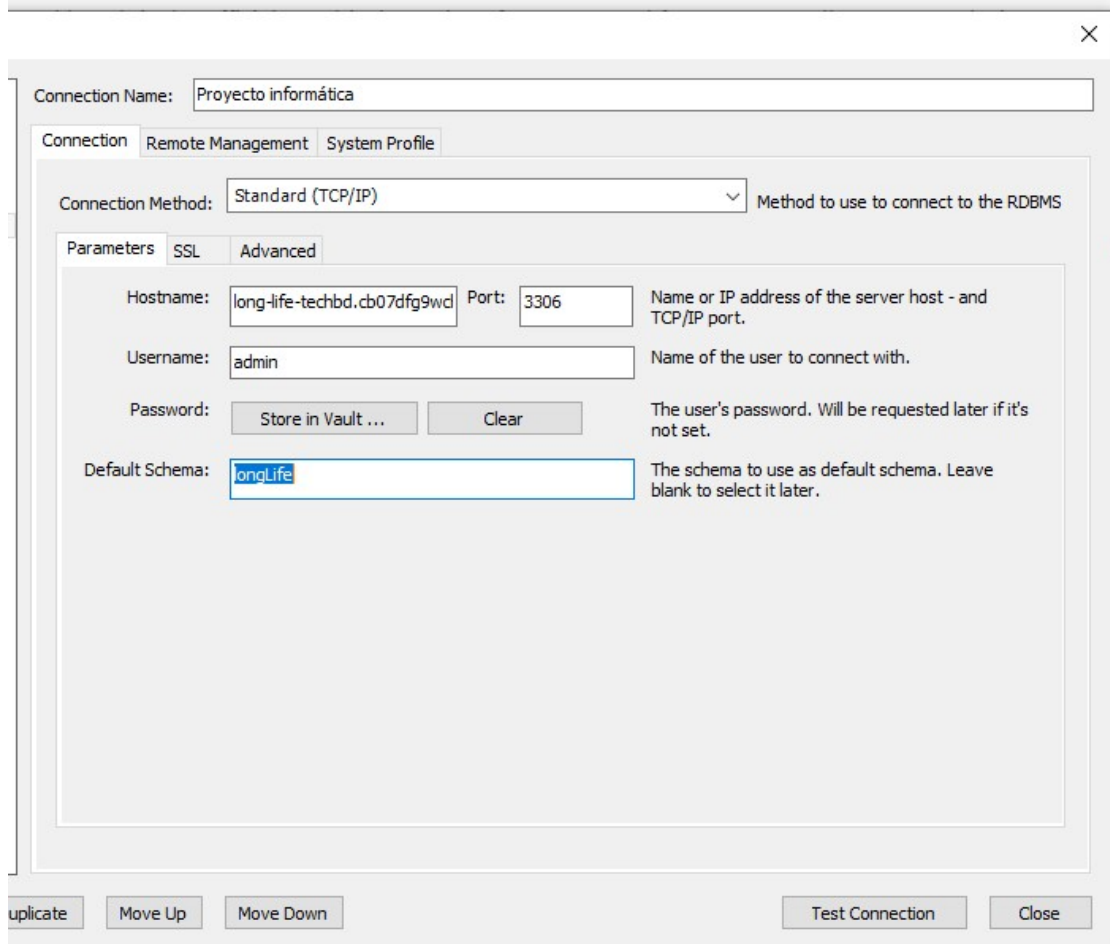
Password: Tech1234.

Entorno de desarrollo de la base de datos

Para la creación de las tablas y llevar a cabo la ingesta de datos que ayudarán a contextualizar la aplicación, utilizaremos **MySQL Workbench**. Gracias a esta herramienta hemos generado el script de creación de la base de datos, así como las consultas que lanzaremos desde nuestra aplicación java, para que, de este modo, podamos implantarlas fácilmente en nuestro desarrollo.

Configuración del MySQL Workbench

elcome to MySQL Workbench



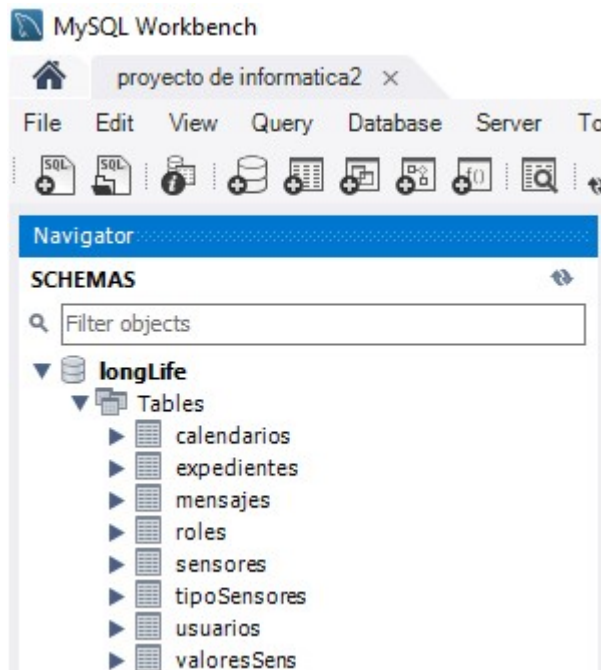
The screenshot shows the MySQL Workbench connection configuration dialog box. The 'Connection Name' field is set to 'Proyecto informática'. The 'Connection Method' is set to 'Standard (TCP/IP)'. The 'Parameters' tab is selected, showing the following fields:

- Hostname: long-life-techbd.cb07dfg9wd
- Port: 3306
- Username: admin
- Password: (with 'Store in Vault ...' and 'Clear' buttons)
- Default Schema: longLife

At the bottom of the dialog, there are buttons for 'Duplicate', 'Move Up', 'Move Down', 'Test Connection', and 'Close'.

Tablas de datos en MySQL

A continuación de recoger cada una de las tablas creadas para desarrollar este proyecto con todos sus campos, el formato de estos y parte de los registros acumulados por parte de los sensores.



Creación de las tablas en MySQL

A continuación, el detalle de los queries generados para crear la base de datos y las distintas tablas.

```
3 #####
4 -- CREAMOS LA BASE DE DATOS:
5 #####
6 • CREATE SCHEMA longLife;
```

```

*****
^
-- Table structure for table `usuarios`
*****

DROP TABLE IF EXISTS `usuarios`;
CREATE TABLE `usuarios` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `idRol` int(11) NOT NULL,
  `nombre` varchar(100) NOT NULL,
  `dni` varchar(15) NOT NULL,
  `password` tinytext NOT NULL,
  `email` varchar(100) NOT NULL,
  `telefono` varchar(45) NOT NULL,
  `direccion` varchar(100) NOT NULL,
  `fechaNacimiento` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

*****
^
-- Table structure for table `calendarios`
*****

) DROP TABLE IF EXISTS `calendarios`;CREATE TABLE `calendarios` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `idExpediente` int(11) NOT NULL,
  `idActividad` longtext NOT NULL,
  `fecha` timestamp NOT NULL,
  PRIMARY KEY (`id`)
- ) ENGINE=InnoDB AUTO_INCREMENT=36 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

*****
^
-- Table structure for table `expedientes`
*****

DROP TABLE IF EXISTS `expedientes`;
) CREATE TABLE `expedientes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `idPaciente` int(11) NOT NULL,
  `idTutor` int(11) NOT NULL,
  `idEspecialista` int(11) NOT NULL,
  `estadoAlerta` int(11) NOT NULL,
  `pesoPaciente` float DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1006 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

```

*****
^
-- Table structure for table `mensajes`
*****

DROP TABLE IF EXISTS `mensajes`;
CREATE TABLE `mensajes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `idEmisor` int(11) NOT NULL,
  `idReceptor` int(11) NOT NULL,
  `mensaje` text NOT NULL,
  `fecha` timestamp NOT NULL,
  `estadoMensaje` bit(1) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=77 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

*****
^
-- Table structure for table `roles`
*****

DROP TABLE IF EXISTS `roles`;
CREATE TABLE `roles` (
  `id` int(11) NOT NULL,
  `rol` varchar(45) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

*****
^
-- Table structure for table `sensores`
*****

DROP TABLE IF EXISTS `sensores`;
CREATE TABLE `sensores` (
  `id` int(11) NOT NULL,
  `idExpediente` int(11) NOT NULL,
  `tipoSensor` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

```

*****
^
-- Table structure for table `tipoSensores`
*****

DROP TABLE IF EXISTS `tipoSensores`;
CREATE TABLE `tipoSensores` (
  `id` int(11) NOT NULL,
  `tipo` varchar(45) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

*****

-- Table structure for table `valoresSens`
*****

DROP TABLE IF EXISTS `valoresSens`;
CREATE TABLE `valoresSens` (
  `id` int(11) NOT NULL,
  `valor` float DEFAULT NULL,
  `fecha` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

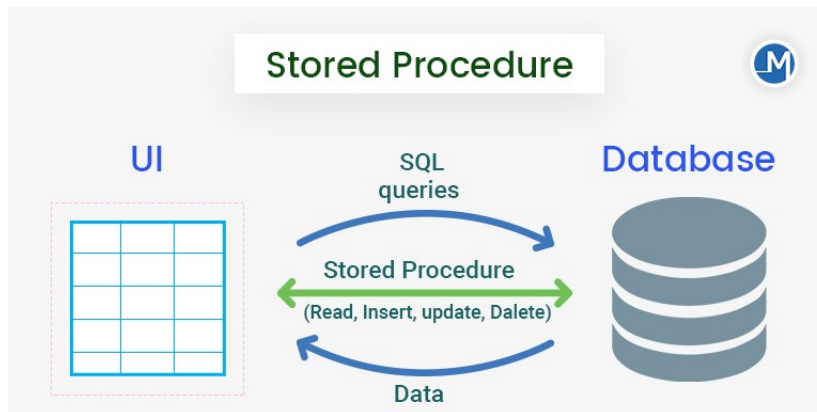
```

Para la inserción de los datos tenemos el detalle en el archivo excel adjunto:

<https://github.com/ProyectoInformatica/proyecto1-long-life-tech/blob/master/bbdd/Proyecto%20Informatica%20-%20Tablas%20base%20de%20datos%20v3.xlsx>

Procedimientos almacenados

Los procedimientos almacenados MySQL son conjuntos de comandos SQL almacenados en el servidor, para que nuestros clientes no tengan que ejecutar los comandos individuales. Esto nos permite trabajar con grandes cantidades de datos delimitando la complejidad a nivel de servidor y ofreciendo a los usuarios directamente lo que están buscando.



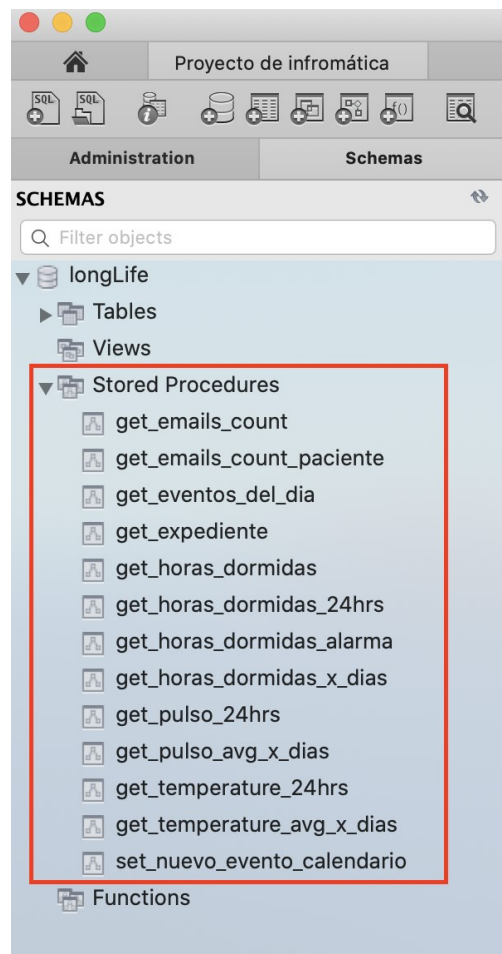
Principales ventajas:

- **Compatibilidad:** los procedimientos almacenados MySQL permiten ejecutar la misma operación sobre la base de datos también en el caso de que las aplicaciones cliente estén implementadas en distintas plataformas y programadas en distintos lenguajes.
- **Integridad:** gracias a los procedimientos almacenados MySQL es posible centralizar el acceso a la información.
- **Seguridad:** cuando necesitamos evitar el acceso directo a la base de datos, los procedimientos almacenados MySQL permiten establecer un entorno seguro, otorgando permisos y privilegios para la ejecución.
- **Rendimiento:** con los procedimientos almacenados MySQL todo el trabajo se ejecuta en el servidor, por lo cual se minimiza de forma importante la cantidad de información intercambiada con las aplicaciones cliente y se reduce el tráfico de acceso a la base de datos además del número de accesos.
- **Centralización:** los procedimientos almacenados MySQL permiten centralizar la lógica comercial ofreciendo a todas las aplicaciones clientes la misma versión actualizada, lo que hace el mantenimiento más sencillo.
- **Reutilización del código:** al escribir un mismo código que se ejecuta por toda las aplicaciones, es posible reducir las inconsistencias.

- **Sencillez:** los procedimientos almacenados permiten la creación de bibliotecas o funciones, lo que facilita mucho el trabajo del programador ya que se trata de características compartidas por los principales lenguajes de programación modernos.

Para el desarrollo de LongLife hemos querido utilizar procedimientos almacenados para nuestras principales consultas debido a las ventajas y valor agregado que estos nos aportan.

A continuación se enumeran y describen cada uno de estos:



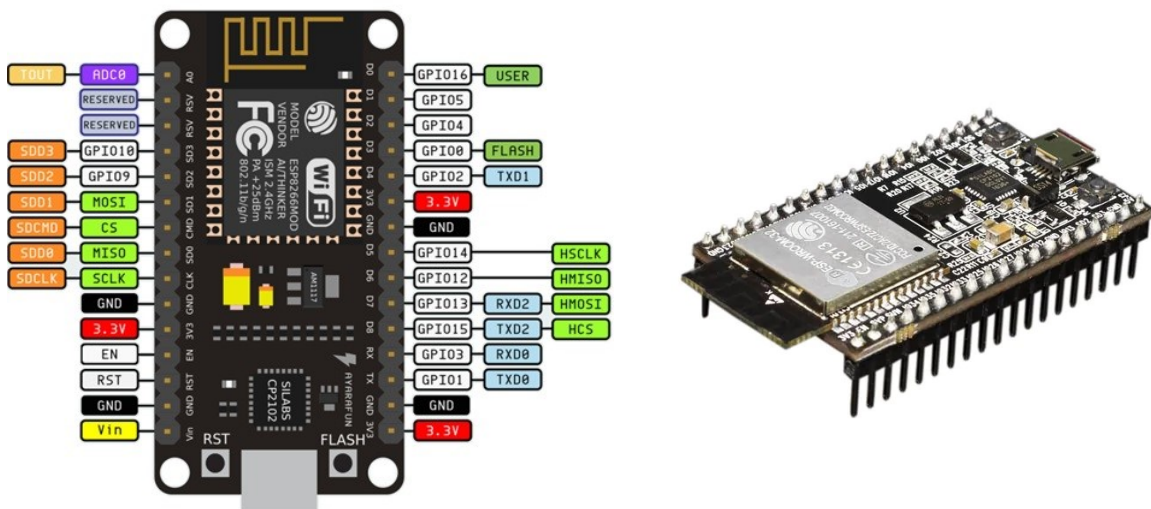
- **get_emails_count:** Devuelve la cantidad de mensajes no leídos para un tutor o especialista. Recibe dos parámetros de entrada: idEmisor e idReceptor.
- **get_emails_count_paciente:** Devuelve la cantidad de mensajes no leídos para un paciente. Recibe solo un parámetro de entrada: idReceptor

- **get_eventos_del_dia:** Dado un id de expediente, nos devuelve las actividades del calendario del día actual para el paciente asociado.
- **get_expediente:** Dado un id de expediente, nos devuelve toda la información asociada a este.
- **get_horas_dormidas:** Devuelve la cantidad de horas que un paciente ha permanecido en su cama en un día dado. Recibe como parámetros de entrada: el peso del paciente, el id de sensor de báscula asociado y el día en que se quiere medir. Como parámetro de salida arroja la cantidad de horas que ha permanecido en la cama el día dado.
- **get_horas_dormidas_24hrs:** Dentro de las últimas 24hrs nos dice en qué horas el paciente ha permanecido en la cama. Recibe como parámetros de entrada el peso del paciente y el id del sensor de báscula asociado.
- **get_horas_dormidas_alarma:** es igual que `get_horas_dormidas` pero sin parámetro de salida.
- **get_horas_dormidas_x_dias:** Este procedimiento construye una tabla que va llenando con un bucle `while` donde se llama al procedimiento
- **get_horas_dormidas** para obtener las horas que ha permanecido el paciente en cama en un rango de días. Es el procedimiento usado para pintar las gráficas semanales y mensuales del paciente. Recibe como parámetros de entrada el rango de días por ejemplo: 7 para una semana, 30 un mes, y el id del sensor de báscula asociado al paciente.
- **get_pulso_24hrs:** Nos da la información del promedio del pulso por hora dentro de las últimas 24 horas. Recibe como parámetro de entrada el id de sensor de pulso asociado al paciente.
- **get_pulso_avg_x_dias:** Devuelve el promedio diario del pulso para un rango de días dado. Es el procedimiento usado para pintar la gráfica de pulso semanal y mensual del paciente. Recibe como parámetros de entrada el rango de días y el id del sensor de pulso asociado al paciente.
- **get_pulso_avg_x_dias:** Devuelve el promedio diario del pulso para un rango de días dado. Es el procedimiento usado para pintar la gráfica de pulso semanal y mensual del paciente. Recibe como parámetros de entrada el rango de días y el id del sensor de pulso asociado al paciente.

- **get_temperature_24hrs:** Nos da la información del promedio de la temperatura de la habitación donde está instalado el sensor por hora dentro de las últimas 24 horas. Recibe como parámetro de entrada el id de sensor de temperatura asociado al paciente.
- **get_temperature_avg_x_dias:** Devuelve el promedio diario de la temperatura de la habitación donde está instalado el sensor para un rango de días dado. Es el procedimiento usado para pintar la gráfica de temperatura semanal y mensual del paciente. Recibe como parámetros de entrada el rango de días y el id del sensor de temperatura asociado al paciente.
- **set_nuevo_evento_calendario:** Es el único procedimiento que tenemos para insertar datos en nuestro modelo. Es usado para crear actividades en calendario del paciente. Recibe como parámetros de entrada el id del expediente, id de la actividad y la fecha.

Desarrollo hardware con Arduino

Tal y comentamos en los requisitos técnicos de nuestra aplicación, estaremos utilizando el arduino para monitorizar los distintos eventos relacionados con el estado de salud de nuestros dependientes. Dado que teníamos que proporcionarle la suficiente escalabilidad y conectividad a nuestro sistema, nos hemos apoyado en el ESP32, dado que se trata de un chip SoC de bajo coste y que consume poca energía y que cuenta tanto con conectividad wifi como bluetooth.



Con este chip, hemos podido conectar cada uno de los sensores que usaremos para monitorizar a nuestros dependientes con la suficiente estabilidad con el fin de evitar errores de hardware en el sistema.

Tal y como hemos estructurado el funcionamiento del sistema, cada paciente tendrá su propio ESP32 conectado a los distintos sensores deseados.

Cada sensor tendrá una identificación única, y estará asociado a un expediente, de modo que sabremos de qué tipo de sensor se trata y a quién pertenece. De este modo aseguramos la integridad de los datos y que todos los miembros de cada expediente cuenta con la información de la monitorización del paciente en cuestión.

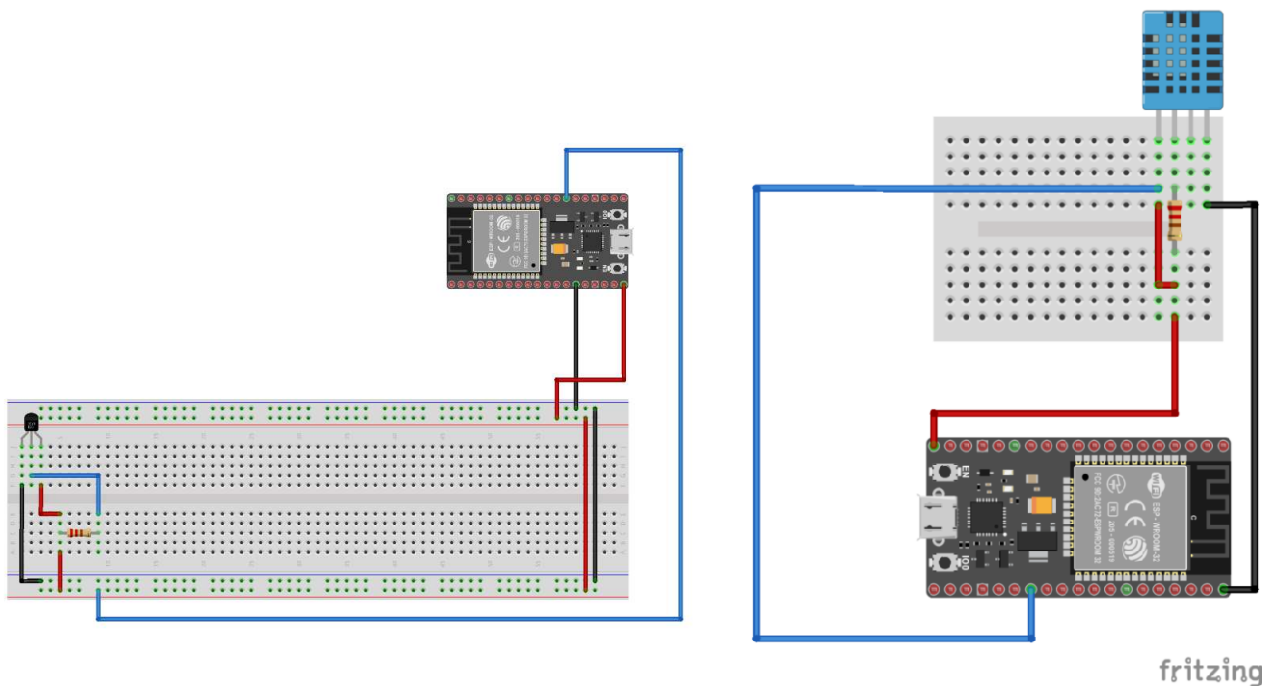
Una vez instalados los ESP32, éste se conectará a una red wifi, y enviará el dato obtenido por los sensores así como la identificación del sensor leído a un servidor web que será el encargado de registrar los datos en la base de datos de nuestro sistema.

Dicho servidor web igualmente lo hemos alojado en la nube, utilizando el servicio de AWS EC2.

Para ello nos hemos apoyado principalmente en **2 sensores, con los que hemos podido configurar los sistemas de monitorización del sueño, la monitorización de la temperatura, y la monitorización

Monitorización de la temperatura

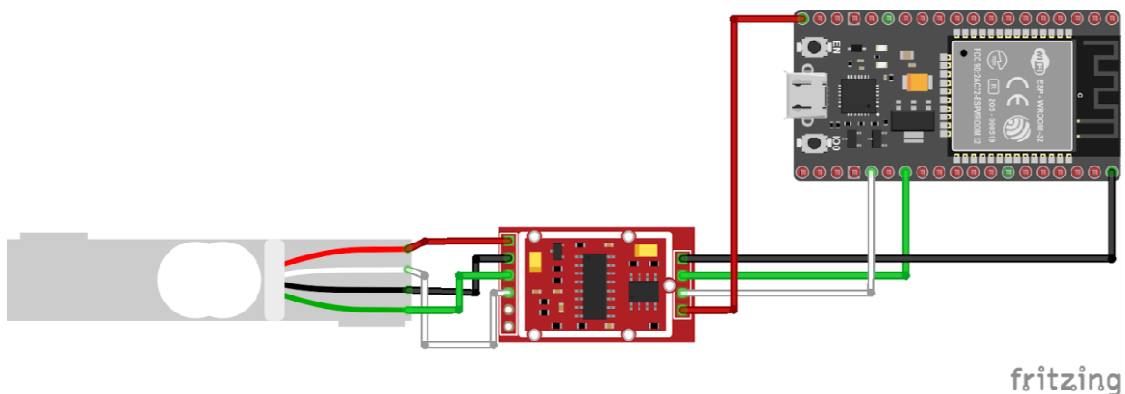
De cara a la lectura de la temperatura, Hemos desarrollado un sistema mediante el cual podemos monitorear la temperatura de varias estancias, y que nos resultará muy útil para obtener información y así confirmar que todo es correcto en las estancias de nuestros pacientes. Para ello hemos hecho uso de dos tipos de sensores distintos DHT11, así como en el DS18B20: (Nos hemos apoyado en la herramienta fritzing para poder hacer el diseño del sistema)



Monitorización del tiempo que pasa en la cama

Para la monitorización del sueño hemos utilizado un sensor que mide el peso, de tal modo que si el paciente se encuentra sobre la cama, este evento será interceptado por nuestro sensor, y viceversa, si el paciente se levanta, no se le escapará el detalle a nuestro sistema.

Concretamente hemos utilizado la celda de carga YZC-1B capaz de sensar hasta 100kg. Este sistema necesita de un transmisor de celda de carga HX711.



(Nota destacar que como consecuencia del COVID19, no hemos podido adquirir a tiempo uno de los sensores que teníamos previsto utilizar en nuestro proyecto.

Conexión Arduino con servidor web

Como comentado anteriormente, hemos utilizado entorno de los servicios de informática en la nube (EC2) de AWS, pues se trata de servicios sencillos de configurar, utilizar, y altamente escalables, que cumple con todas las exigencias de nuestro sistema.

Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
us-east-1d	running	2/2 checks ...	None	ec2-18-215-37-5.comp...	18.215.37.5	-

Dentro de la instancia web comentada, hemos alojado un archivo sencillo que estará escuchando las entradas que le lleguen por POST, se conectará a nuestra base de datos y ejecutará una consulta que insertará los datos correspondientes dentro de la tabla de valores sensores.

Implementación del código en Java

Long Life se ha desarrollado en Java usando la versión 1.8 del JDK. El proyecto es de tipo JavaFx por lo tanto se necesita tener instalado en el IDE el plugin de JavaFX.

Configuración del entorno

Para facilitar las tareas de compilación y empaquetado de una aplicación típica JavaFx hemos usado el plugin **javafx-maven-plugin**, cuya dependencia está definida en el fichero pom.xml de maven.

Para establecer de manera automática la versión del JDK a usar se ha instalado el plugin de Maven **maven-compiler-plugin**, donde se ha establecido la versión 1.8.

Integración con la base de datos

Hemos instalado la dependencia mysql-connector-java que es el driver que nos permite la conexión con una base de datos de tipo MySQL. Esta dependencia está definida en el fichero pom.xml de maven por lo cual será instalada de manera automática en el proyecto.

La clase que se encarga de establecer la conexión con la base de datos es **com.healthytech.core.bbdd.ConnectBd**. Esta proporciona en su interfaz dos métodos **conectar** y **desconectar** que son utilizados en toda la aplicación cada vez que es necesario conectarse a la base de datos.

Concurrencia

Cada operación de conexión a la base de datos se hace desde un thread diferente al main thread donde se ejecuta la interfaz gráfica de LongLife.

Como la información de cada sensor es independiente la aplicación ha sido diseñada para recuperar los datos de manera concurrente. Para ello hemos creado la clase **com.healthytech.core.alerts.AlertThreadWorker** que extiende de Thread y que se

encarga de recuperar la información del tipo de sensor especificado en un hilo de ejecución nuevo.

Para controlar todas las instancias de **AlertThreadWorker** y que puedan realizar el trabajo sin colapsar al ordenador se ha creado la clase **com.healthytech.core.alerts.AlertsManager** que las maneja mediante un **Thread Pool** de tipo **ExecutorService** que se inicializa con la cantidad de cores que tiene la cpu del ordenador donde se está ejecutando la aplicación.

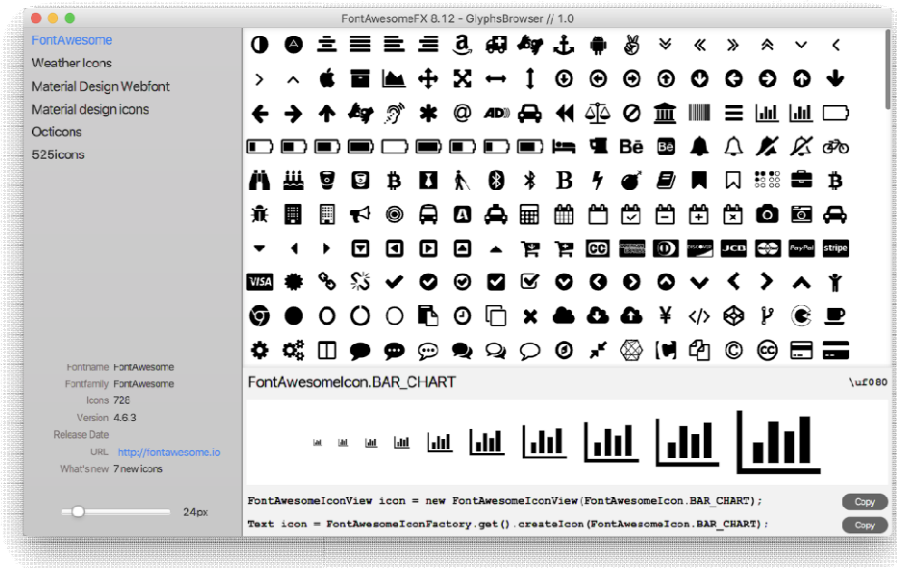
GUI

Para la interfaz gráfica del usuario ya se ha comentado que la aplicación es de tipo JavaFx. Para darle un aspecto más moderno e intuitivo hemos instalado y usado la librería **JFoenix 9** para conseguir el look and feel de **Material Design**, la normativa de diseño creada por Google para sus dispositivos Android y aplicaciones web.

Para más información acerca de JFoenix se deja un enlace: <http://www.jfoenix.com/>



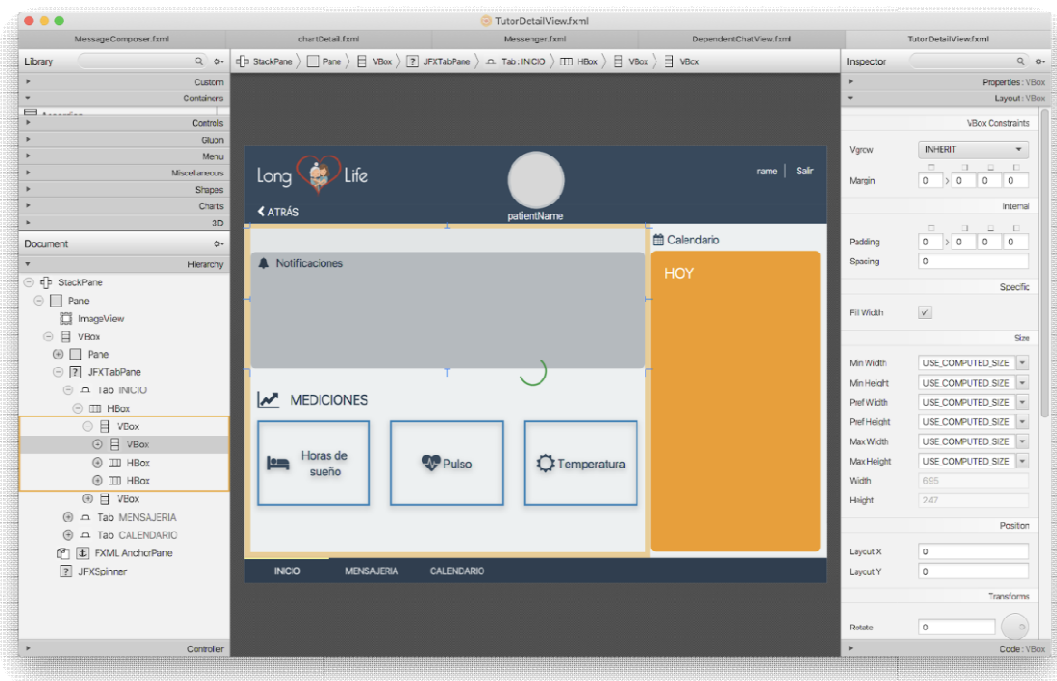
Para crear una interfaz más atractiva se ha instalado la librería **FONTAWESOMEFX 8.2** que ofrece una amplia colección de iconos generados mediante tipografía y que se pueden utilizar y manipular fácilmente en nuestra aplicación sin necesidad de incluir ficheros de imágenes extra que contribuyen a engrosar el peso de la aplicación. A estos iconos se les puede cambiar el tamaño y color fácilmente mediante código.



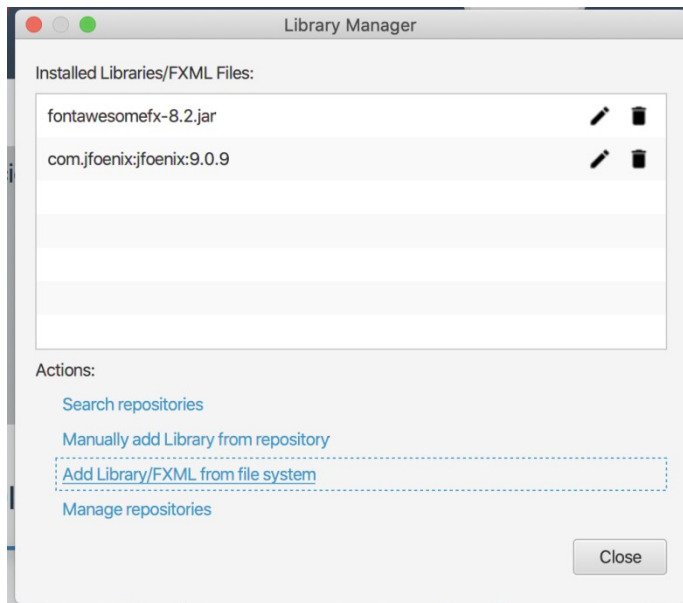
La navegación entre las diferentes vistas es llevada a cabo mediante la clase **com.healthtech.core.navigation.ScreenController**. Su interfaz brinda métodos para añadir nuevas pantallas, eliminar pantallas y activar pantallas.

Maquetación y componetización

Para la maquetación de las vistas y componentes hemos usado en gran parte SceneBuilder 11:



Las librerías JFoenix 9 y FontAwesome también se deben descargar e instalar en SceneBuilder.



A continuación se dejan los enlaces de descargas de estas dos librerías para SceneBuilder:

JFoenix9 para SceneBuilder:

<https://search.maven.org/remotecontent?filepath=com/jfoenix/jfoenix/9.0.8/jfoenix-9.0.8.jar>

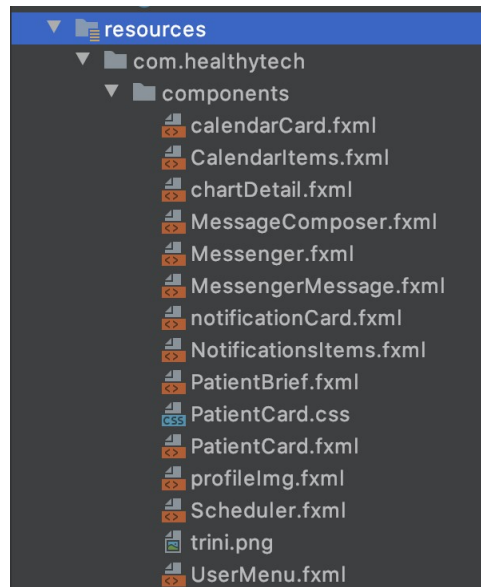
FontawesomeFX 8.2 para SceneBuilder

<https://bitbucket.org/Jerady/fontawesomefx/downloads/fontawesomefx-8.2.jar>

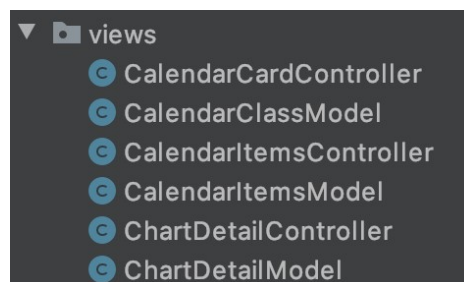
A la hora de maquetar y programar la interfaz de usuario encontramos diferentes partes que pueden ser reutilizadas si son estructuradas y construidas correctamente. A estos es lo que llamamos componentización. Esto nos permite tener nuestro código más organizado y que se pueda mantener con mayor facilidad. Los posibles cambios o ajustes se hacen en un solo lugar evitando tener que malgastar tiempo en este tipo de tareas.

Estos son los ficheros fxml de los distintos componentes que se han definido en LongLife:

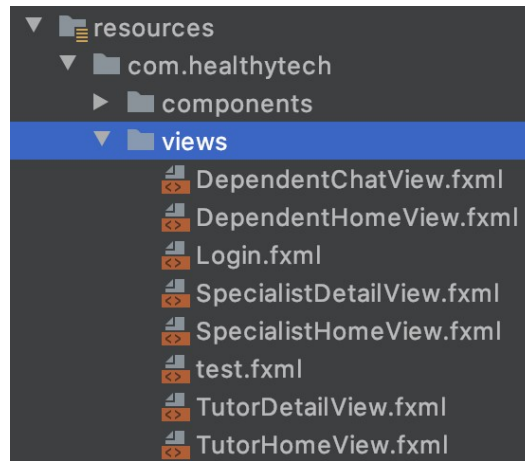
Entre otros hay componentes para el calendario y sus distintas partes, el messenger con sus mensajes, las notificaciones, la tarjeta de usuario, el menú de usuario... entre otros



Nuestros componentes siguen el patrón **MVC**. Por lo que para casi todos los componentes tendremos sus respectivas clases controladora y modelo.



Estos componentes son posteriormente usados en todas las vistas que componen los diferentes flujos de pantallas de los diferentes tipos de usuario. A continuación dejamos la lista de vistas definidas en la aplicación, donde se pueden identificar fácilmente el login, home del paciente, etc..



Por último mencionar que dado que JavaFx permite modificar visualmente los elementos mediante hojas de estilo hemos utilizado la metodología **BEM** (block element modifier) para escribir estilos, ya que nos dota de un patrón para nombrar selectores css:

```
/* User Menu ----- */
.patient-brief {
  -fx-background-color: #ecf0f1;
  -fx-spacing: 10;
  -fx-border-width: 2px;
  -fx-border-color: #7f8c8d;
  -fx-border-radius: 10px;
}
.patient-brief__tutor-info {
  -fx-border-width: 1 0 0 0;
  -fx-border-color: #7f8c8d transparent transparent transparent;
}
.patient-brief__tutor-title {
  -fx-font-size: 22px;
  -fx-text-fill: #7f8c8d;
}
.patient-brief__field {
  -fx-font-size: 15px;
  -fx-text-fill: #34495e;
}
.patient-brief__field--small {
  -fx-font-size: 12px;
  -fx-text-fill: #34495e;
}
```

El primer recuadro comprendería la clase **Bloque**, el segundo clases **Elementos** y el tercero una clase **Modificadora**.

Para saber más de BEM: <http://getbem.com/>

Diagrama de clases

Dado que tenemos más de 50 clases y las relaciones entre ellas es bastante complicada de representar no hemos podido hacer un diagrama de clases como tal. Hemos ido mencionando anteriormente las clases más importantes. A continuación dejamos una lista de las clases más representativas:

Nota: todas las clases se encuentran dentro del paquete **com.healthytech.core**

- AlertThreadWorker
- AlertManager
- PatientAlert
- ConnectBD
- CalendarEvent
- Message
- Conversations
- ScreenController
- Sensor
- Mixins
- Expediente
- PatientData
- User
- Main

Todas las clases modelos y controladores de las diferentes vistas y componentes se encuentran dentro del paquete **com.healthytech.views**

A continuación dejamos una captura con la estadística de los diferentes tipos de ficheros que componen el proyecto, número de líneas, tamaños, etc:

Extension	Count	Size SUM	Size MIN	Size MAX	Size AVG	Lines	Lines MIN	Lines MAX	Lines AVG
java (Java classes)	68x	163kB	0kB	8kB	2kB	5898	4	300	86
fxml (FXML files)	21x	72kB	0kB	10kB	3kB	1367	10	171	65
ds_store (DS_STORE files)	7x	45kB	6kB	8kB	6kB	70	1	45	10
ino (INO files)	4x	9kB	0kB	3kB	2kB	427	37	131	106
css (CSS files)	3x	9kB	0kB	7kB	3kB	463	1	422	154
prefs (PREFS files)	2x	8kB	0kB	8kB	4kB	120	4	116	60
sql (SQL files)	2x	13kB	5kB	8kB	6kB	364	161	203	182
fxbuild (FXBUILD files)	1x	0kB	0kB	0kB	0kB	8	8	8	8
project (PROJECT files)	1x	0kB	0kB	0kB	0kB	29	29	29	29
classpath (CLASSPATH)	1x	1kB	1kB	1kB	1kB	33	33	33	33
xml (XML configuration file)	1x	1kB	1kB	1kB	1kB	71	71	71	71
xlsx (XLSX files)	1x	39kB	39kB	39kB	39kB	268	268	268	268
graphml (GRAPHML files)	1x	102kB	102kB	102kB	102kB	1226	1226	1226	1226

Manual de usuario de la interfaz gráfica

La aplicación para desarrollar debe disponer de la capacidad de una comunicación bilateral entre los diferentes agentes y usuarios. De este modo, el paciente estará en contacto con su familiar o tutor; y con el personal sanitario. De este modo se podrán evitar cualquier incidencia que constaten ambos o que requiera el paciente; o poner en funcionamiento un protocolo de emergencia si la urgencia lo requiriese.

Esta aplicación tiene una serie de requisitos expuestos por el cliente, a continuación, detallamos.

1. Se debe desarrollar una aplicación multiusuario, con diversos roles o perfiles, en el lenguaje de programación JAVA, que sirva para monitorizar los datos obtenidos de los sensores. Debe poseer una interfaz intuitiva y fácil, adaptable a cada perfil o rol.
2. Se requiere que la persona dependiente esté monitorizada mediante la información por los sensores utilizados. Estos sensores pueden estar instalados en el hogar o llevarlos el paciente mediante un dispositivo que se use a modo de pulsera, brazalete o guante, según necesidad. Los sensores que se utilizarán, y el porqué de su uso, se ha detallado en el apartado anterior. Así como futuras mejoras y actualización del desarrollo de esta aplicación con la incorporación de otros sensores alternativos.
3. La aplicación debe disponer de una pantalla de login diferente dependiendo del tipo de rol o perfil con el que iniciamos sesión. A continuación, expondremos las diferencias entre estas pantallas y sus posibilidades

Funciones extras de Long Life

Long Life dispone de las siguientes funciones extra

Botón de Ayuda

Se ha implementado un botón en la interfaz gráfica del paciente mediante el que ante una urgencia, el paciente puede accionarlo y alertará, de manera notoria, al doctor que ese paciente necesita su atención con urgencia.

Una vez accionado, la tarjeta del paciente se mostrará en color rojo en la interfaz gráfica del doctor. Una vez se ha atendido la necesidad del paciente, el doctor podrá presionar el botón “Reset” y devolver al paciente a estado normal.

Planificador de actividades

Se ha incluido un planificador de actividades a modo de calendario en el que el paciente podrá ver las tareas, actividades, medicaciones que deberá llevar a cabo durante el día. Estas indicaciones serán añadidas por el doctor y aparecerán diariamente, en fragmentos horarios, en la ficha del paciente.

Sistema de alertas

Notificar cualquier incidencia basándonos en la monitorización de la información que nos aportan los sensores es una tarea imprescindible en nuestro proyecto. Dependiendo de los parámetros y las condiciones expuestas, se mostrarán las alertas siguiendo el método del semáforo. En verde si todo está correctamente, en amarillo si hay alguna precaución o en rojo si existe alguna gravedad en la situación sanitaria del paciente basándonos en esos parámetros expuestos.

Asimismo, los pacientes estarán categorizados en pacientes en “Estado normal” o “Necesita atención inmediata”. Si los datos monitorizados no advierten, en este sistema de alerta, de ninguna necesidad médica, estarán en el primer grupo.

Por otro lado, se requerirá la presencia del doctor y el paciente será categorizado en el segundo grupo si se da alguna de las siguientes condiciones, en orden de urgencia:

- El paciente ha pulsado el “botón Ayuda” y el sistema de alertas, además de avisar, expondrá en rojo su tarjeta.
- El paciente posee alguna advertencia de peligro, con valores en rojo.
- El paciente tiene 2 o más advertencias (señalización amarilla) por lo que el médico deberá hacerle un control y analizar su situación.

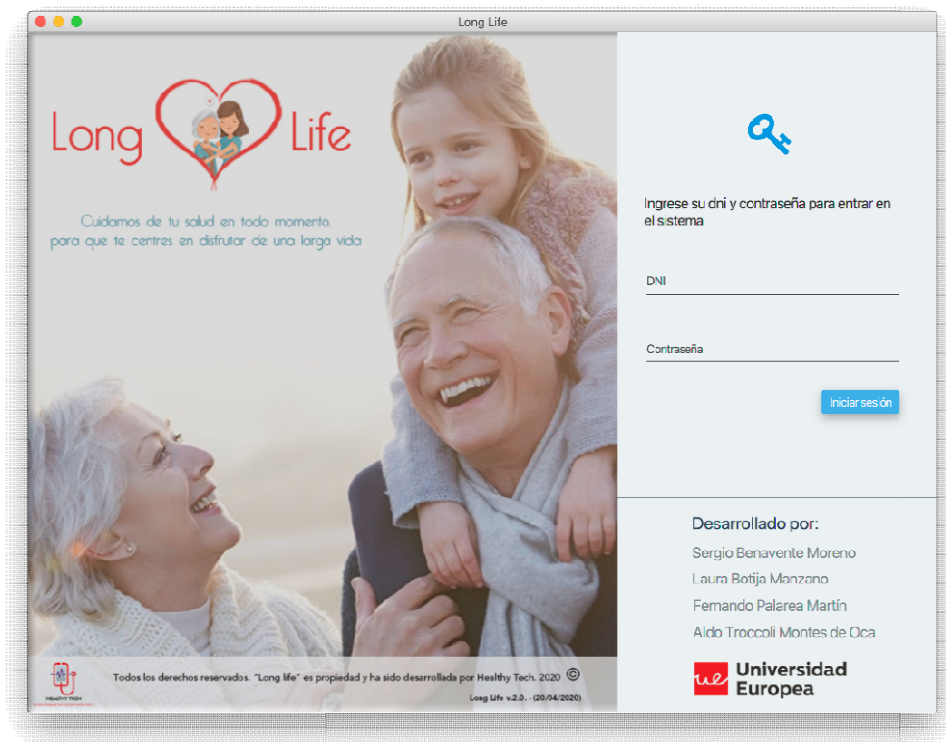
Listado de usuarios

A continuación se encuentra el listado de usuarios con acceso a la aplicación donde se indica el rol, nombre, DNI y password.

Rol	Nombre apellido	DNI	Password
Paciente	María del Carmen Perez	63668659J	1234
Paciente	Pablo Sanchez Balboa	30404092R	1234
Paciente	Trini Diaz Moreno	70596629S	1234
Tutor	Paco Antonio Raimundo Migas	30401232H	1234
Tutor	Fernando Grande	31559486J	1234
Especialista	Pedro Francisco Duque	46132028P	1234
Paciente	Patxi Aramburo Erkoreka	19970890J	1234
Paciente	Amparo Brown Yeah	20675307D	1234

Pantalla de navegación inicial

Donde necesitan un usuario y contraseña. Permite proteger la confidencialidad de estos y sus datos. Es la misma vía de acceso para todos los usuarios de la plataforma.



Pantallas de navegación personal sanitario

El personal sanitario debe obtener la formación de una manera rápida, intuitiva, ordenada y contar con todos los mecanismos necesarios para actuar de inmediato ante cualquier incidencia, notificación y urgencia. Por tanto, el perfil de usuario mostrará en una vista inicial los pacientes que requieren de una urgente atención si ha habido alguna incidencia. Posteriormente, el resto de los pacientes que no requieren de una atención inmediata.

The screenshot displays the 'Long Life' web application interface. At the top, there is a dark blue header with the 'Long Life' logo on the left and the user's name 'Pedro Francisco Duque' and a 'Salir' button on the right. Below the header, the main content area is divided into two sections:

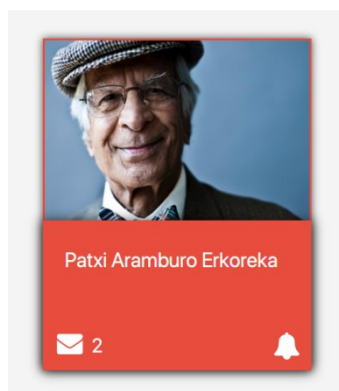
- Pacientes que necesitan atención inmediata:** This section is highlighted with a red border and contains two patient cards. The first card is for 'María del Carmen Perez' and the second is for 'Patxi Aramburo Erkor...'. Both cards feature a red background and a white bell icon, indicating urgent attention.
- Pacientes en estado normal:** This section is highlighted with a green border and contains three patient cards. The first card is for 'Pablo Sanchez Balboa', the second for 'Trini Diaz Moreno', and the third for 'Amparo Brown Yeah'. The third card also features a blue envelope icon and the number '1', indicating a message.

Tarjetas de Pacientes

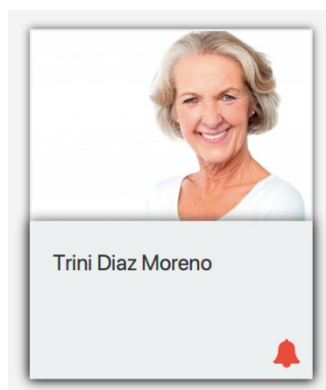
Para representar los expedientes de los pacientes y los diferentes estados en los que estos pueden estar hemos creado un componente llamado *PatientCard*.

Las tarjetas del paciente siempre tendrán la foto del paciente y su nombre.

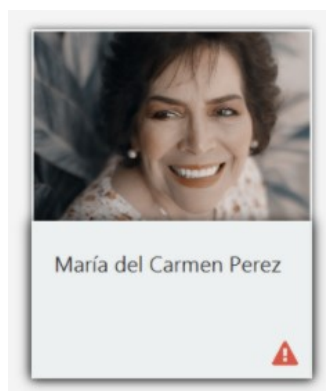
Si el paciente ha pulsado el botón de **AYUDA** entonces la tarjeta se pinta de rojo para llamar la atención del especialista:



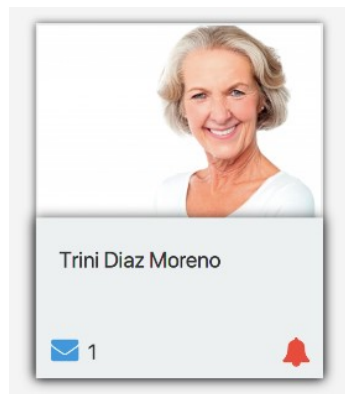
Si al menos uno de los tres sensores registra información dentro del rango de *PELIGRO* que hemos definido en la baremación, entonces en la tarjeta se dibuja el ícono de la *CAMPANA*:



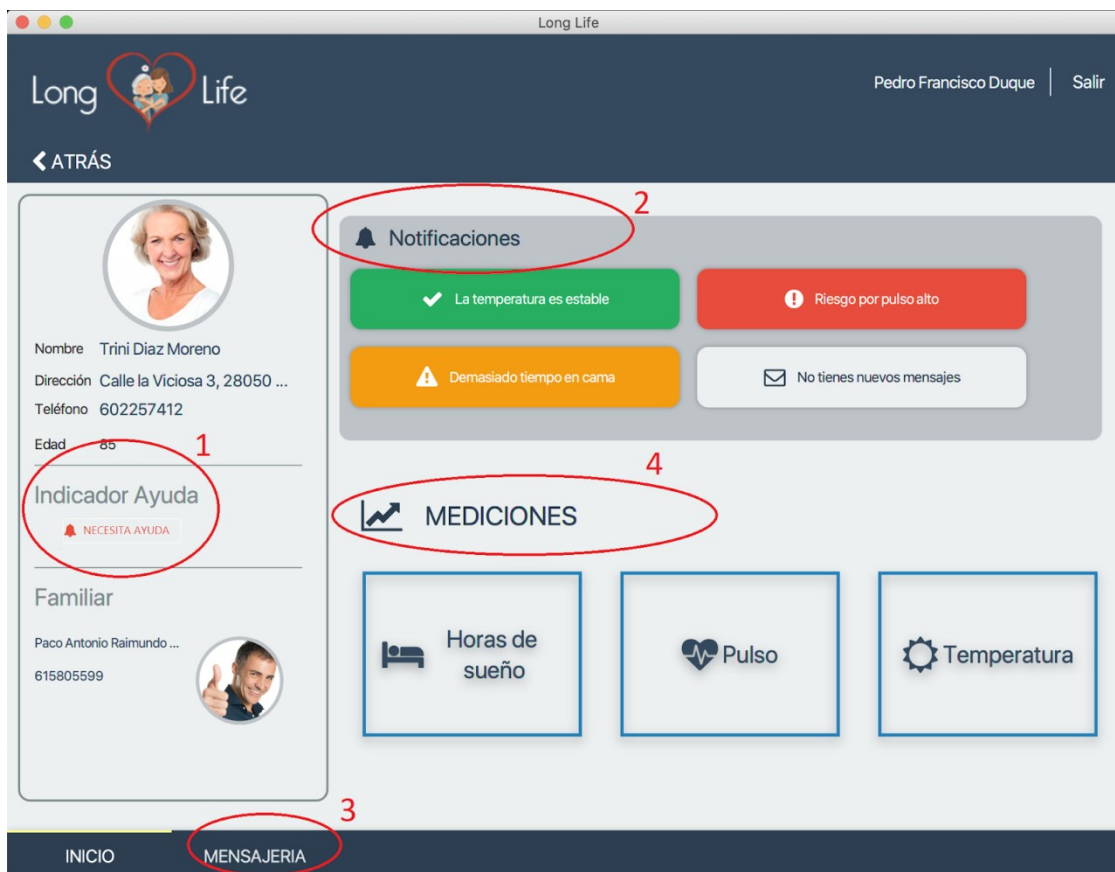
Si no hay ningún sensor que haya registrado información de peligro, pero al menos dos sensores están registrando información clasificada en el rango de *ATENCIÓN*, entonces en la tarjeta se dibuja el ícono *ATENCIÓN*:



Y por último, cuando hay nuevos mensajes no leídos se dibuja el icono de mensajería con el número de mensajes recibidos:



Igualmente, debe tener acceso al historial y a la información detallada del paciente, la persona de contacto (tutor o familiar), medicación, alertas... y los datos que se obtienen por lo sensores que la monitorizan. Por tanto, dispondrá de la visión de una ficha detallada de cada paciente a la que podemos llegar haciendo click en una de las tarjetas de paciente.



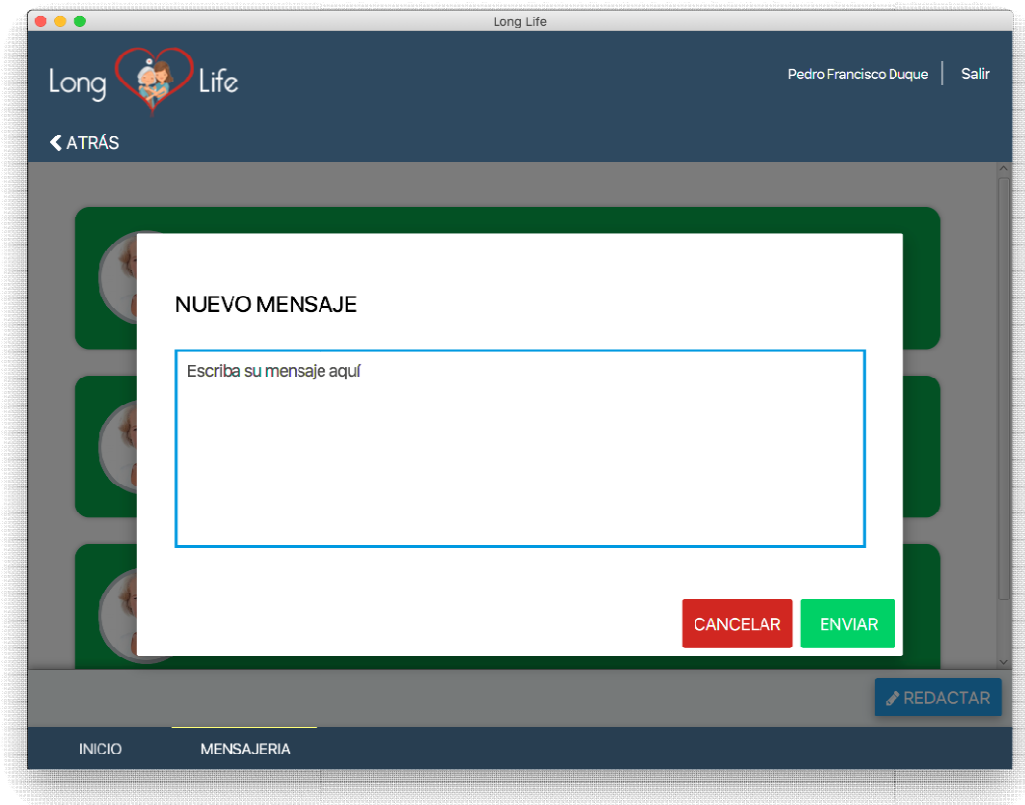
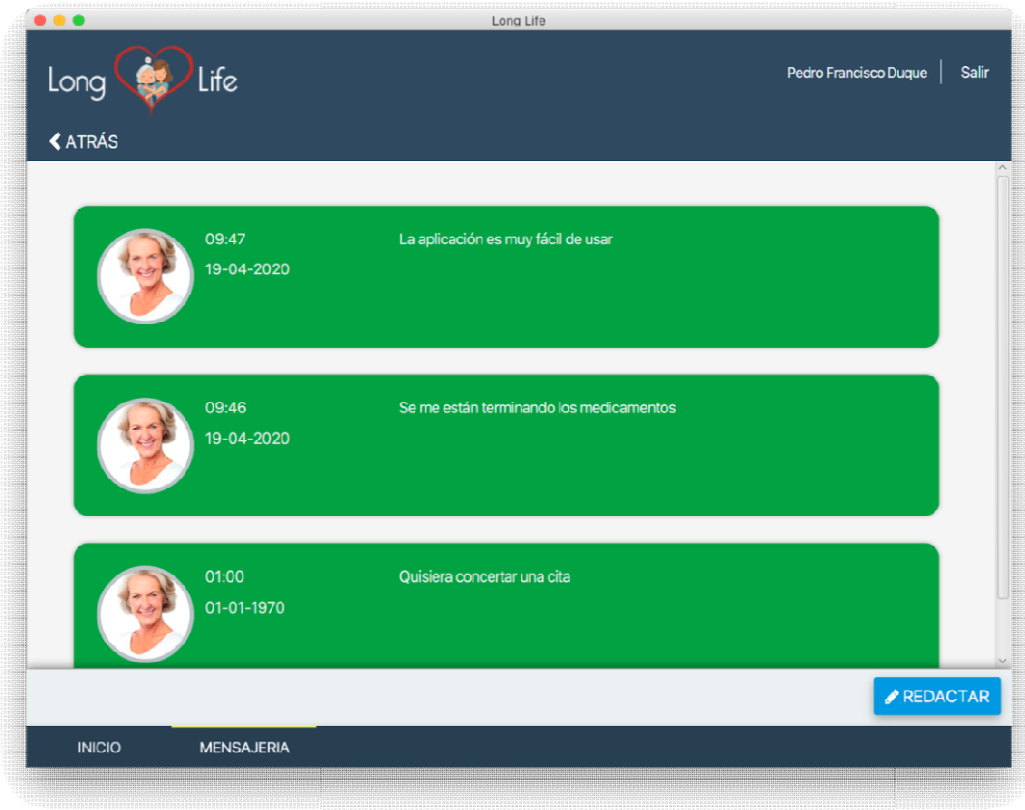
A continuación, se da detalle de la información a la que se puede acceder en cada uno de los puntos señalados en la imagen.

1. En este caso el paciente ha pulsado el botón “AYUDA” y el médico visualiza en su perfil esta solicitud de alarma. Una vez comprobado por el sanitario, este puede quitar la alerta pulsando el botón “RESET”, quedando esta información como se refleja en la siguiente imagen.
2. En la parte “**Notificaciones**” tenemos información visual por colores de los tres sensores implementado “Horas de sueño”, “Pulso” y “Temperatura”. Los registros mostrará diferentes colores (rojo, naranja y verde).



En este caso en particular y como referencia el color verde indica en este paciente que tiene una temperatura estable.

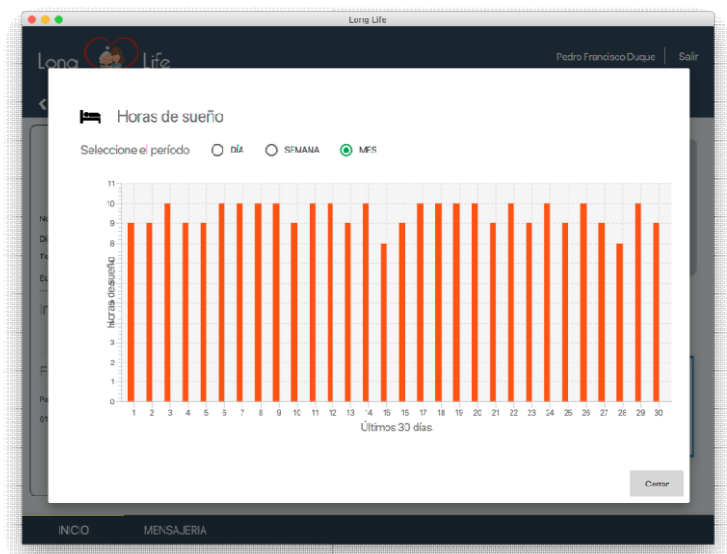
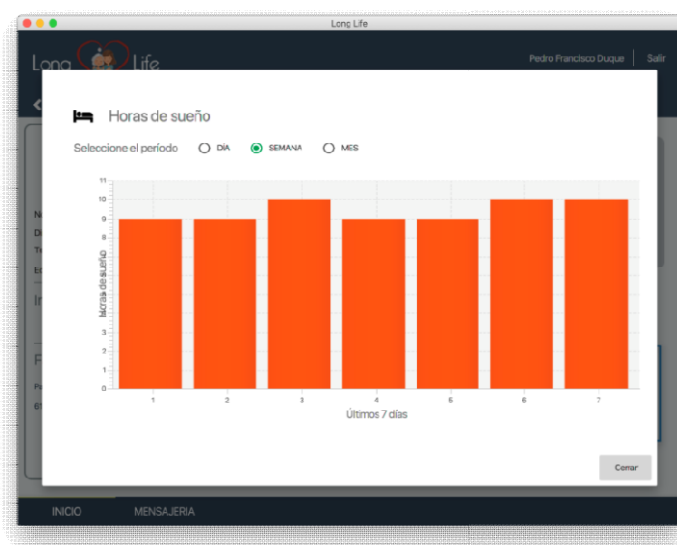
- En naranja tenemos un warning sobre que ha estado más tiempo de lo normal en la cama.
 - En rojo el sensor del pulso detecta que está dentro de unos valores riesgo.
 - En el apartado “**Baremación Sensores**” se puede encontrar información más detallada acerca de las medias de medición que se han implementado.
 - Se dispone una opción de “**notificaciones de mensajería**” que indica si tiene algún mensaje sin leer de este paciente en concreto.
3. En la pestaña “**Mensajería**” es donde se permite al médico enviar mensaje a sus pacientes y estar en permanente comunicación con ellos. Como se muestra en las siguientes imágenes.



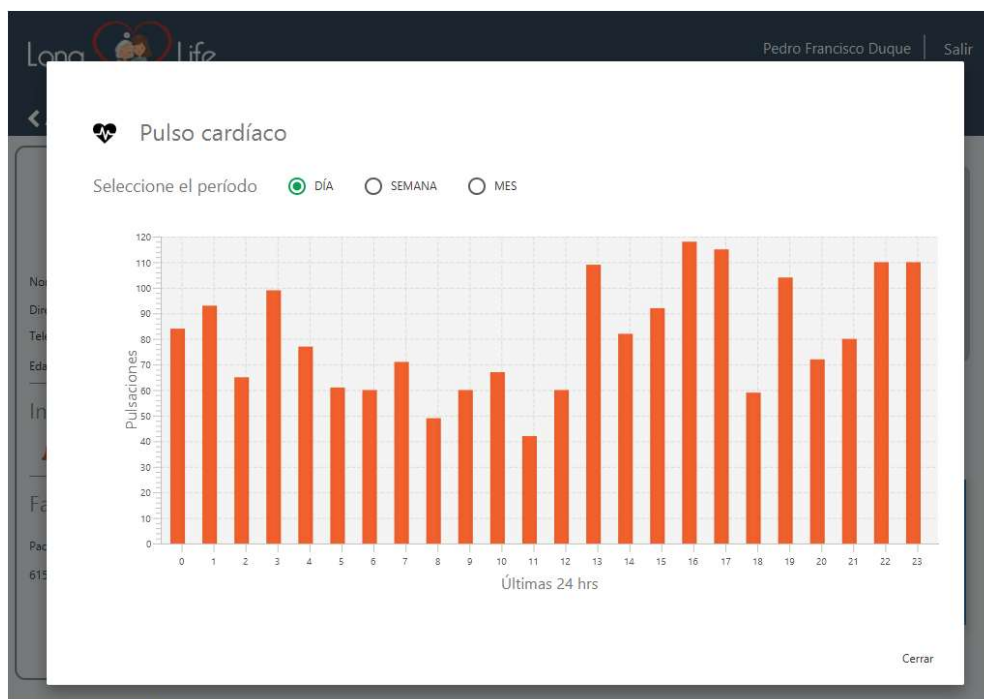
4. En el apartado “**Mediciones**” tenemos una representación gráfica individualizada de los datos obtenido por cada uno de los sensores.

Se nos representará toda la información de la monitorización de nuestro dependiente, ofreciendo un análisis diario, semanal y mensual, para así poder hacer un seguimiento al avance y a las tendencias de nuestros pacientes, con el fin de detectar anomalías que puedan ayudarnos a asistir, e incluso adelantarnos, a posibles problemáticas.

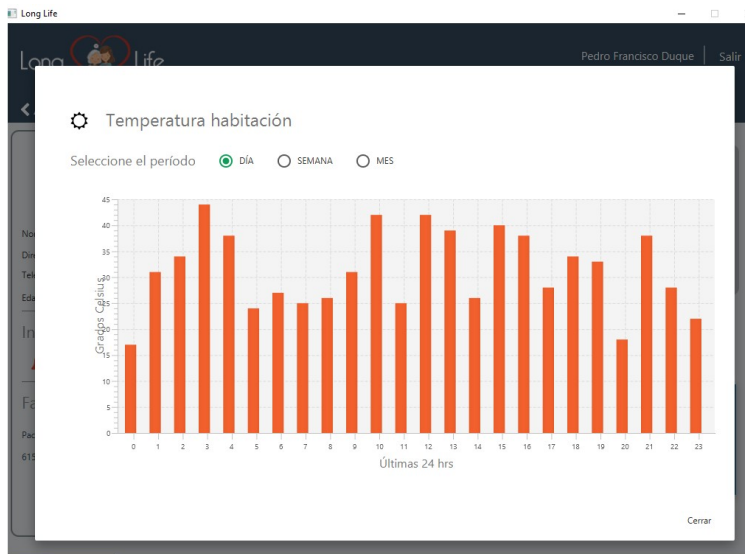
Horas de sueño mensuales recogidas del paciente pero podemos seleccionar “día”, “semana” o “mes”.



Pulso cardíaco del paciente por horas recogidas del paciente pero podemos seleccionar “semana” o “mensual”.



Un seguimiento de la temperatura diaria como se indica en la siguiente gráfica dependiendo de la opción que elijamos pero también tenemos semanal o mensual.



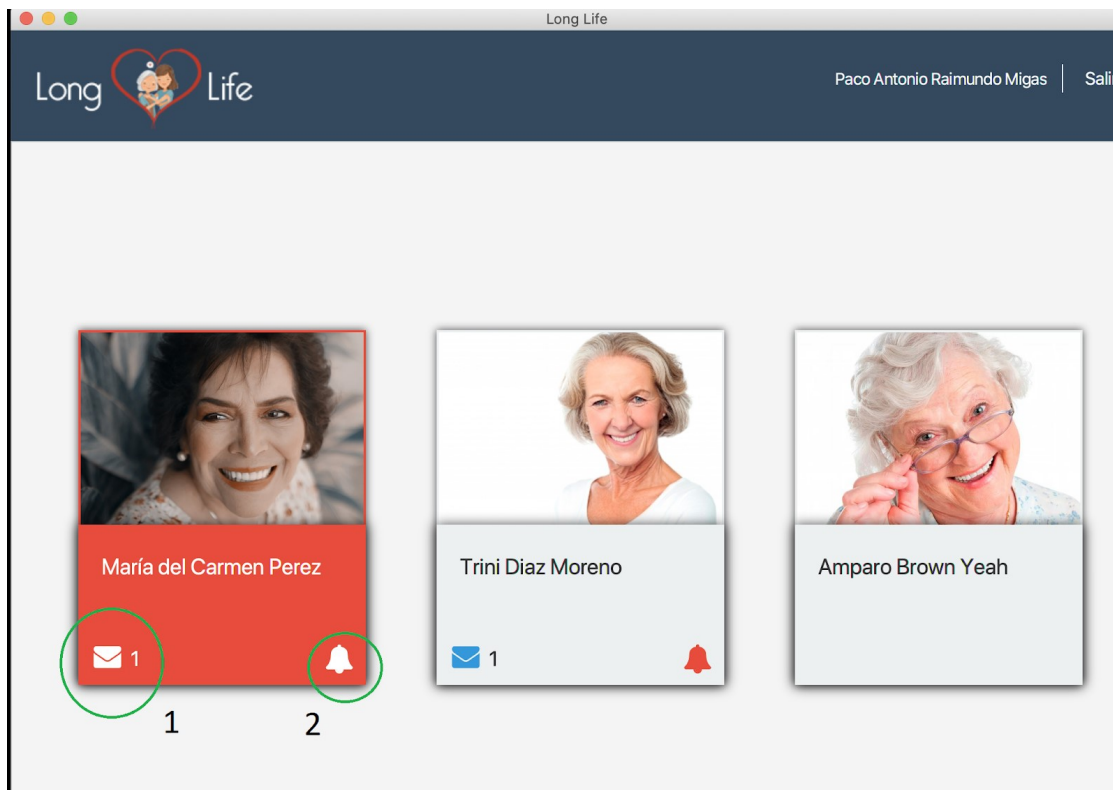
Todos estos datos el médico los obtiene en tiempo real, dando una asistencia de urgencia en caso de que el paciente presentara alguna anomalía en los registros recogidos.



Al mismo tiempo realizar un seguimiento del paciente que puede prevenir situaciones de urgencia si los datos tienen unas variaciones significativas.

Pantallas de navegación familiar o tutor

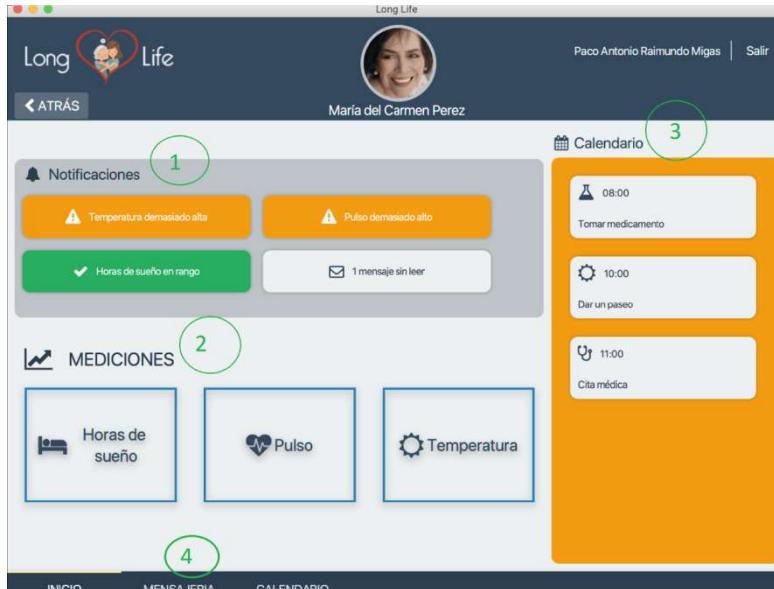
El tutor o familiar después de iniciar sesión dispondrá de la siguiente pantalla que muestra información puntual y básica de los pacientes con los que tiene relación y han autorizado ese vínculo manteniendo la ley de protección del paciente y protección de datos de este.



A continuación, se da detalle de la información a la que se puede acceder en cada uno de los puntos señalados en la imagen.

1. En este caso el tutor tiene un mensaje de Maria del Carmen Perez
2. Se muestra una advertencia con el símbolo de la campana, respecto a los registros de los sensores están fuera de la normalidad.

El tutor al acceder a este paciente tienen la siguiente visualización, en la parte superior se indica el nombre y foto del mismo. A continuación, se da más detalle de toda la información

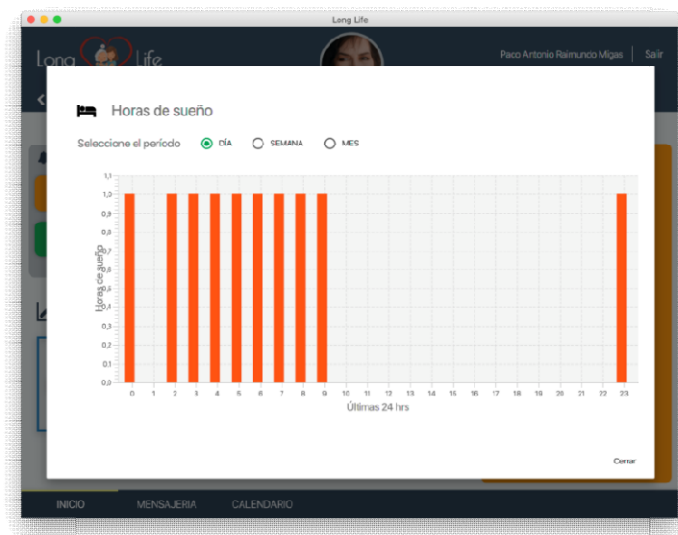


1. En la parte “**Notificaciones**” tenemos información visual por colores de los tres sensores implementado “Horas de sueño”, “Pulso” y “Temperatura”. Los registros mostrará diferentes colores (rojo, naranja y verde). En este caso en particular y como referencia el color verde indica en este paciente que el tiempo de sueño está dentro de la normalidad. En naranja tenemos un warning sobre la temperatura y el pulso.

En el apartado “**Baremación Sensores**” se puede encontrar información más detallada acerca de las medias de medición que se han implementado.

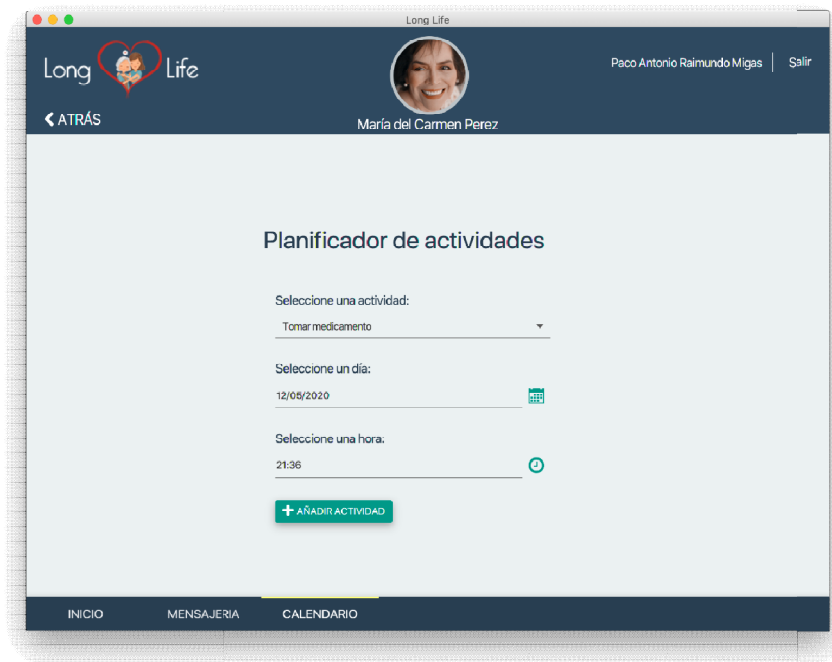
Se dispone una opción de “**notificaciones de mensajería**” que indica si tiene algún mensaje sin leer de este paciente en concreto.

2. El apartado “**Mediciones**”, que tal y como ya hemos comentado anteriormente será donde se visualicen los datos consolidados procedentes de la monitorización de los sensores, y cuyas vistas son iguales a como las ve el asistente.

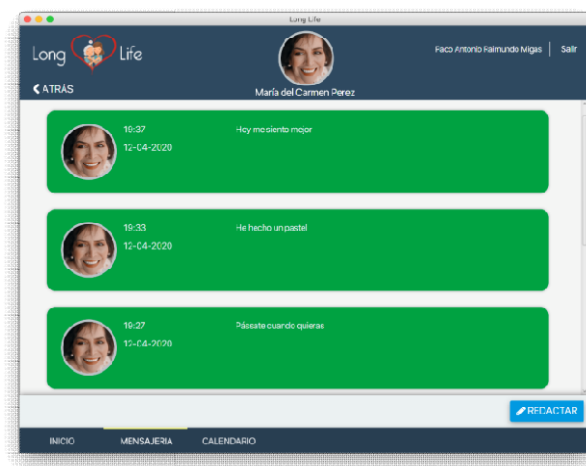


Horas de sueño mensuales recogidas del paciente pero podemos seleccionar “día” o “semana”.

3. En el apartado “**calendario**” tenemos todas las planificaciones activas para este paciente. También se pueden añadir otra planificación de actividades como se indica a continuación,

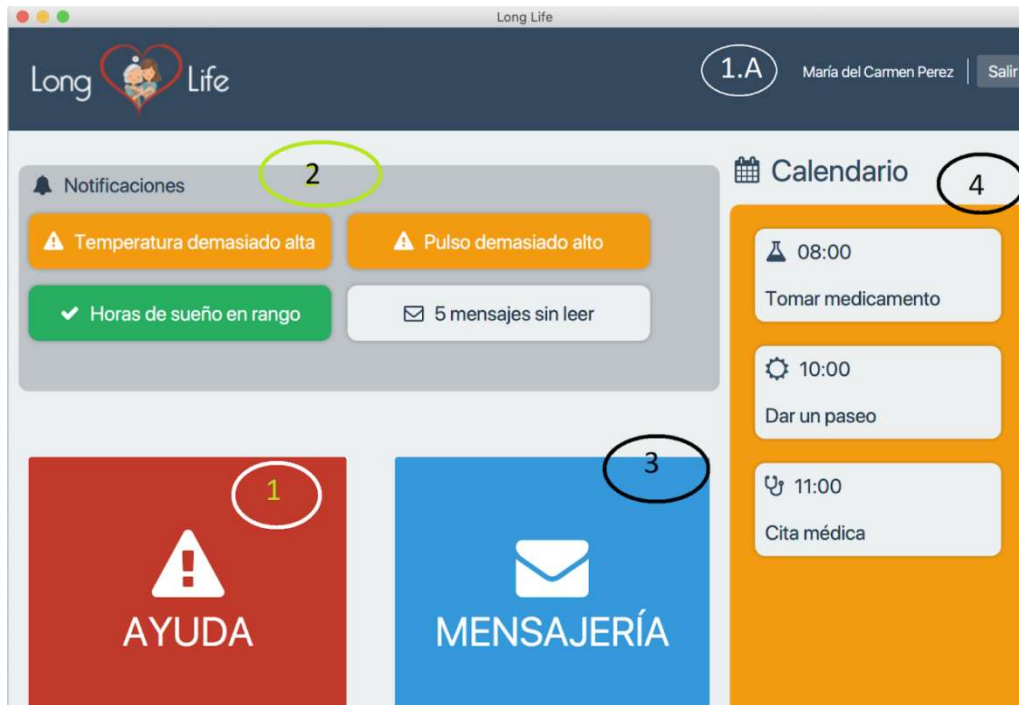


4. En la pestaña “**Mensajería**” es donde se permite al tutor enviar mensaje a sus pacientes y estar en permanente comunicación con ellos. Como se muestra en las siguientes imágenes.



Pantallas de navegación del paciente dependiente

Finalmente, el perfil del paciente, o persona dependiente, debe ser muy intuitivo que no resulte una complicación para estos y sin incluir información alarmante. Simplemente, el calendario con las actividades del día que será meramente informativa y un botón muy importante de ayuda para alertar que no se encuentra bien en cualquier momento.



A continuación, se da un detalle de cada uno de los puntos señalados en la imagen.

“1.A” nos indica el nombre del paciente, con el que se ha iniciado sesión.

1. Los pacientes disponen de un botón “AYUDA” en caso de emergencia lo deben pulsar cambiando inmediatamente al color como se muestra en la anterior imagen. Esta alerta les llega tanto al tutor como al médico. Permitiendo una asistencia de urgencia en el momento que el paciente lo solicite. Esta opción cambia de color como se muestra a continuación,

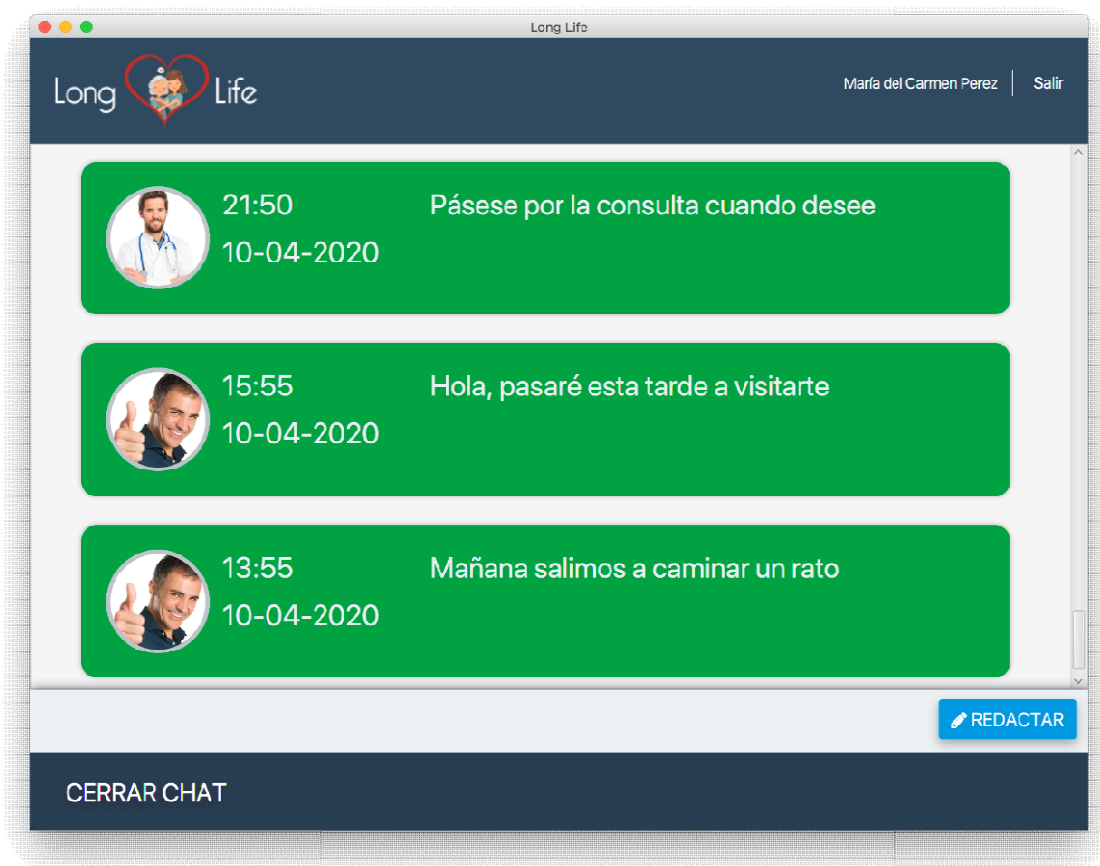


2. En la parte “**Notificaciones**” tenemos información visual por colores de los tres sensores implementado “Horas de sueño”, “Pulso” y “Temperatura”. Los registros mostrará diferentes colores (rojo, amarillo y verde). En este caso son 2 naranjas indican que algo no está correcto y uno verde que es correcto.

En el apartado “**Baremación Sensores**” se puede encontrar información más detallada acerca de las medias de medición que se han implementado.

Se dispone una opción de “**notificaciones de mensajería**” que indica si tiene algún mensaje sin leer de este paciente en concreto.

3. En el apartado “**Mensajería**” el paciente puede comunicarse con su médico y leer todos los emails intercambiados dando lugar a una comunicación activa entre ambos. Favoreciendo al seguimiento médico.

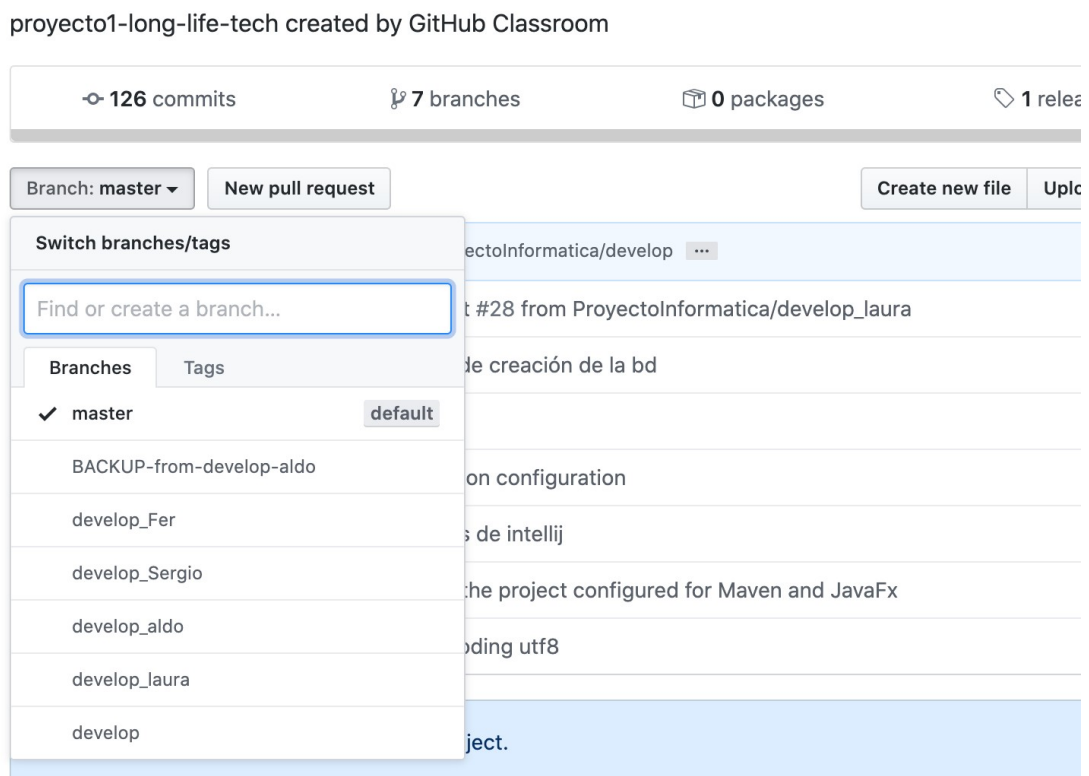


4. En el apartado “**calendario**” tenemos todas las planificaciones activas para este paciente por hora y día.

Repositorio Github

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

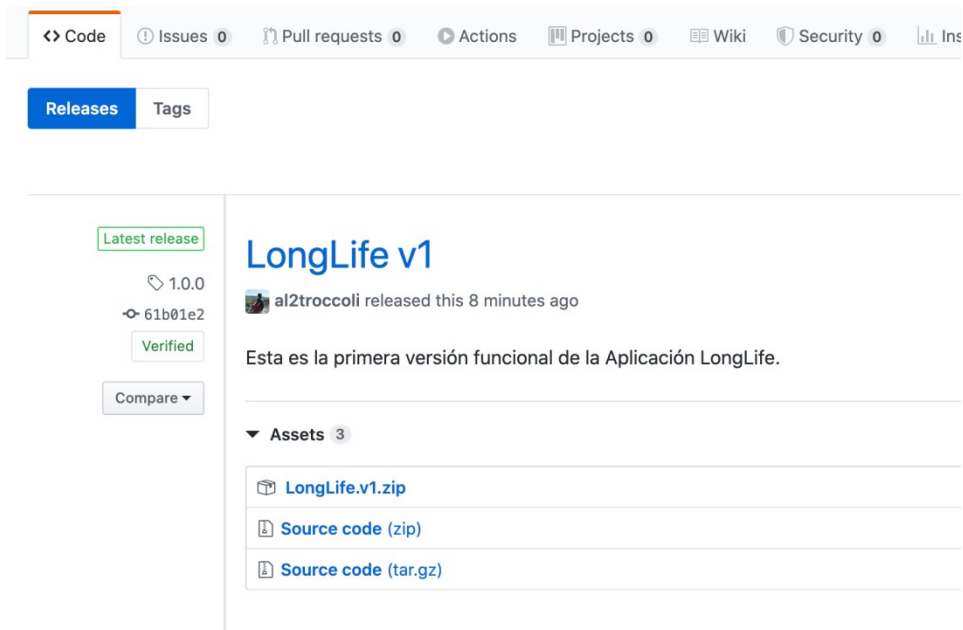
Para facilitar el trabajo colaborativo entre los distintos miembros del equipos hemos



escogido un modelo donde cada desarrollador tiene su propia rama:

En la rama **develop** se van mezclando los nuevos desarrollos progresivos y finalmente **develop** se mezcla en la rama **master** a la hora de sacar una versión.

LongLife ha sacado su primera release que es la versión 1.0.0. Dentro de la esta releasase encuentra etiquetado todo el código fuente correspondiente a la versión y además el ejecutable de la aplicación para poderla ejecutar en el cliente final sin necesidad de compilar, que sería el fichero LongLife.v1.zip



Los recursos y el código del proyecto LongLife se encuentran alojados en el siguiente repositorio de GitHub: <https://github.com/ProyectoInformatica/proyecto1-long-life-tech>

A continuación dejamos los enlaces dentro del repositorio a las distintas partes para facilitar su revisión:

- **Proyecto JavaFx con Maven:**
<https://github.com/ProyectoInformatica/proyecto1-long-life-tech>
- **Aplicación Arduino Sensor Temperatura DHT11:**
https://github.com/ProyectoInformatica/proyecto1-long-life-tech/blob/master/arduino/_longLife_DHT11/_longLife_DHT11.ino
- **Aplicación Arduino Sensor Temperatura Dallas:**
https://github.com/ProyectoInformatica/proyecto1-long-life-tech/blob/master/arduino/_ardTemp_Dallas/_ardTemp_Dallas.ino
- **Aplicación Arduino Sensor Báscula:**
https://github.com/ProyectoInformatica/proyecto1-long-life-tech/blob/master/arduino/_ardBascula/_ardBascula.ino

- **Aplicación web php para enviar los datos recogidos por los sensores a la base de datos:**

https://github.com/ProyectoInformatica/proyecto1-long-life-tech/blob/master/arduino/_servidorPHP/_saveData.php

- **Script de creación de la base de datos:**

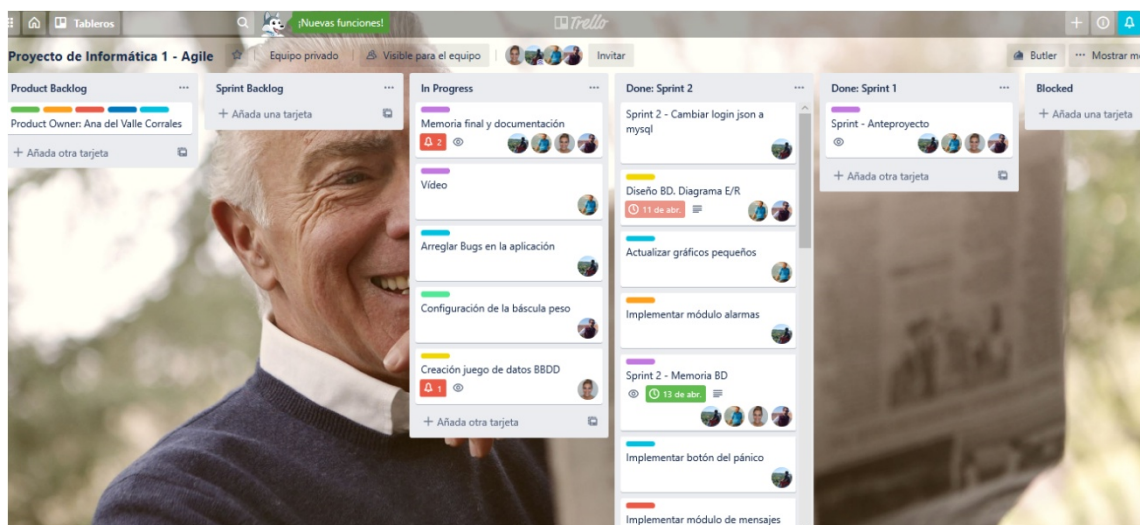
[https://github.com/ProyectoInformatica/proyecto1-long-life-tech/blob/master/bbdd/02.Script SQL BBDD/Long Life - Script de creación de la BBDD.sql](https://github.com/ProyectoInformatica/proyecto1-long-life-tech/blob/master/bbdd/02.Script%20SQL%20BBDD/Long%20Life%20-%20Script%20de%20creaci%C3%B3n%20de%20la%20BBDD.sql)

- **Ejecutable de la aplicación:**

<https://github.com/ProyectoInformatica/proyecto1-long-life-tech/releases/download/1.0.0/LongLife.v1.zip>

Organización del trabajo cooperativo y colaborativo

Hemos utilizado el Trello para la organización de tareas del proyecto. Dividido por diferentes etiquetas, la inicial referente a lo que informamos al cliente que haremos y las siguientes sobre el desarrollo más técnicas. Además, se han fijado miembros, fechas de vencimiento, tiempo previsto de realización...



Posibles vías de desarrollo e investigación en futuras versiones

- Hacer aplicación responsive para que se adapte a la pantalla de cualquier dispositivo.
- Crear la entidad CLINICA para extender el funcionamiento de la aplicación, de esta manera tendríamos una entidad superior donde agrupar los expedientes. Sería una manera de vender la aplicación a diferentes clientes con sus propios expedientes y actores.
- Securizar el envío de datos recogidos por los sensores desde arduino hacia la base de datos.
- Crear un nuevo tipo de usuario con el rol SUPERADMIN que pueda crear expedientes, usuarios, configurar los sensores e identificarlos para los pacientes.
- Se debería poder añadir tantos sensores como se quiera y agruparlos por categorías.
- Crear triggers en la base de datos para optimizar la monitorización de sensores.
- Securizar login de la aplicación
- Creación de reportes en formato pdf

Presupuesto

Para la realización de este proyecto se requiere la contratación de personal especializado, recursos técnicos y se ha presupuestado también una campaña de divulgación, así como de creación de la imagen corporativa, esencial para la óptima venta.

Presupuesto				
Nombre	HEALTHY TECH		Datos cliente	
Dirección	Calle Diputación 15,		Nombre	UEM
Teléfono	657 531 613		Dirección	CAMPUS VILLAVICIOSA
E-mail	info@healthytech.com		Teléfono	91 902 34 34
Fecha presupuesto	30-nov.-19		Validez:	
DESCRIPCIÓN	Recurso	Unidad / horas	Precio	TOTAL
Analista	Contrato	40	35	1.400,00 €
Jefe de proyecto	Contrato	420	30	12.600,00 €
Programador senior	Contrato	420	23	9.450,00 €
Programador junior	Contrato	420	10	4.200,00 €
Ordenador	Material	850	4	3.400,00 €
Sensores	Material	4	6	23,40 €
Arduino	Material	1	33	32,55 €
POSTVENTA				
Director comercial	Contrato	100	25	2.500,00 €
Director de comunicación y marketing	Contrato	200	25	5.000,00 €
Diseño imagen corporativa	Labor	1	1.200	1.200,00 €
Campaña de difusión	Marketing	1	5.000	5.000,00 €
			Total Bruto	44.805,95 €
			I.V.A. %	21%
				9.409,25 €
Total presupuesto				54.215,20 €
Forma de pago	Cheque	Débito	Efectivo	
Firma de la persona que confecciona el presupuesto.			ACEPTO EL PRESUPUESTO. Nombre, apellidos y firma del cliente.	

Instalación de la plataforma

Para ejecutar la presente práctica tendrán que llevarse a cabo las siguientes instrucciones:

1. **Instalar Eclipse**
2. **Instalar JDK 1.8**
3. **Instalar JRE 1.8**
4. **Instalar Maven**
5. **Se deberá de asegurar la asignación de las siguientes variables del entorno del sistema: .**

Panel de control > Propiedades del sistema > Opciones avanzadas > Variables de entorno:

En variables del sistema, asegurarse de tener incluidas las siguientes variables:

Variable	Value
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_191
JDK_HOME	C:\Program Files\Java\jdk1.8.0_191
JRE_HOME	C:\Program Files\Java\jre1.8.0_191
MAVEN_HOME	C:\opt\apache-maven-3.6.3

6. **Instalar plugin e(fx)clipse para eclipse:**
<https://www.eclipse.org/efxclipse/install.html>
7. **Instalar JavaFX Scene Builder 11**
<https://gluonhq.com/products/scene-builder/thanks/?dl=/download/scene-builder-11-windows-x64/>
8. **Instalar JFoenix9 para SceneBuilder**
<https://search.maven.org/remotecontent?filepath=com/jfoenix/jfoenix/9.0.8/jfoenix-9.0.8.jar>
9. **Instalar FontawesomeFX 8.2 para SceneBuilder**
<https://bitbucket.org/Jerady/fontawesomefx/downloads/fontawesomefx-8.2.jar>
10. **Descargar el repositorio de GIT**

11. Sobre el proyecto, clic botón derecho > Run As >

1. Maven update project
2. Maven clean
3. Maven install

12. En la barra de navegación de Eclipse Run > Run Configuration.. Editar las siguientes configuraciones:

Name	Goals
Jfx:run	Jfx:run

13. En la barra de navegación de Eclipse Run > Run Configuration.. Editar las siguientes configuraciones:

Name	Goals
Jfx:jar	Jfx:jar

14. Para ejecutar la aplicación desde Eclipse ejecute la run configuration jfx:run creada en el paso 12

15. Para empaquetar la aplicación y crear el .jar ejecute la run configuration jfx:jar creada en el paso 13

Dependencias del proyecto definidas en Maven:

En el paso 11 de la lista anterior se instalan las siguientes dependencias definidas en el archivo pom.xml de Maven:

- mysql-connector-java
<https://mvnrepository.com/artifact/mysql/mysql-connector-java>
- jfoenix
<https://mvnrepository.com/artifact/com.jfoenix/jfoenix>
- fontawesomefx
<https://mvnrepository.com/artifact/de.jensd/fontawesomefx>
- javafx-maven-plugin
<https://mvnrepository.com/artifact/com.zenjava/javafx-maven-plugin>
- maven-compiler-plugin
<https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-compiler-plugin>

Metodología

Este desarrollo de software se ha decidido utilizar una metodología Scrum.

Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo y obtener el mejor resultado posible de proyectos, caracterizado por:

Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.

Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.

Solapar las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o en cascada.

Por lo que hay que tener un contacto más directo con el cliente y un feedback durante la evolución.

Tener un tablero Kanban digital por diferentes razones algunas de ellas son es accesibles desde cualquier lugar con Internet, más fácil de mantener, proporcionar datos históricos de rendimiento, toda la información se guarda en un lugar central, etc.

Referencias bibliográficas

- AAL Programme <http://www.aal-europe.eu/>
- OnGuardian: <http://www.onguardian.io/>
- Instituto Nacional de Estadística. INE: <https://ine.es/>
- ElderCheck NOW: <https://www.elderchecknow.com/Wheres-Noodle>
- Ambient Assisted Living. Publicado en la revista de Universitat de Barcelona sobre salut i benestar: <http://www.ub.edu/senesciencia/noticia/ambient-assisted-living-aal/>
- Tecnologías Asistivas, de ETSIT, Universidad de Málaga: <https://es.slideshare.net/crisurdiales/ambient-assisted-living-qu-es>
- Características y retos del envejecimiento de la población: La perspectiva europea, de Par Asghar Zaidi: <http://www.euro.centre.org>
- Population Ageing and Development 2009: www.unpopulation.org
- Europa envejece. Publicado en CNN: <https://cnnespanol.cnn.com/2018/05/16/europa-envejece-1-de-cada-5-personas-es-mayor-de-65-y-el-unico-aumento-de-la-poblacion-viene-de-la-inmigracion/>
- El envejecimiento de la población: http://www.fgsic.es/lychnos/es_es/articulos/envejecimiento_poblacion
- Procedimientos almacenados MySQL: qué son, cómo crearlos y ventajas: <https://blog.mdcloud.es/procedimientos-almacenados-mysql-que-son-como-crearlos-y-ventajas/>

Imagen corporativa y diferentes alternativas futuras

Long Life

Long Life



Guía de solucionado de errores. FAQs

Paciente

Tengo una urgencia y no consigo respuesta del doctor ni de mi tutor, ¿qué hago?

Solamente si se trata de una urgencia, deberá pulsar el botón de “Ayuda” y se le notificará al doctor de la necesidad de atención inmediata.

¿Cómo hago para hablar con mi doctor o con mi hij@?

En la parte inferior de la aplicación dispone de un botón llamado “Mensajería” con el que podrá ponerse en contacto con su doctor, con su hij@ y la persona que tenga asignada como tutor/a

Tengo problemas técnicos con Long Life

Ante cualquier incidencia notifíquese al doctor, tutor o familiar para solventarlo.

Tutor

Tengo problemas técnicos con Long Life

Ante cualquier incidencia envíe un correo a info@longlife.com o comuníquese al 91 455 455 45 y le atenderá el equipo técnico.

Mi familiar me escribe demasiado, ¿puedo bloquearle la mensajería?

No. Entendemos que el doctor debe atender a sus pacientes y que estos tengan posibilidad de contactarles para cualquier urgencia. Es tarea sanitaria y familiar concienciar del uso que los pacientes y familiares hacen de la mensajería.

Especialista sanitario

¿Por qué tengo un paciente con la tarjeta en rojo?

Si un paciente tiene una urgencia y requiere de su atención inmediata podrá accionar el botón de “Ayuda”. Su tarjeta se coloreará de rojo y así podrá advertirle de la necesidad inmediata de su atención.

¿Cómo le quito el color rojo a la tarjeta de un paciente?

Tras atender la necesidad de un paciente puede pulsar el botón “Reset” para volverlo a categorizarlo al estado normal.

¿Por qué hay pacientes con una campana roja en su tarjeta?

Si un paciente tiene una campana roja en su tarjeta requiere de su atención. Puede deberse a que algún parámetro monitorizado está en zona peligrosa o porque 2 o más sensores nos advierten de una anomalía en la salud del paciente.

Quiero ver las gráficas monitorizadas, ¿cómo puedo verlas?

Pulsando sobre los sensores expuestos en la ficha de cada paciente podrá ver los valores que los sensores han aportado en las últimas 24 horas, última semana y el último mes.

Un paciente me escribe demasiado, ¿puedo bloquearle la mensajería?

No. Entendemos que el doctor debe atender a sus pacientes y que estos tengan posibilidad de contactarles para cualquier urgencia. Es tarea sanitaria y familiar concienciar del uso que los pacientes y familiares hacen de la mensajería.

¿Cómo puedo ponerle tareas o recordar la medicación a un paciente?

Puede añadir tareas mediante el planificador de tareas a cada paciente. Debe recordar que verá las actividades diarias.

Quiero asignarme diferentes pacientes y asignar sus familiares. ¿Qué debo hacer?

Una vez se crea la ficha del paciente le asignaremos el doctor y familiares que podrán consultar y gestionar la ficha de cada paciente. Ante cualquier incidencia puede ponerse en contacto con el Soporte Técnico de Long Life.

Tengo problemas técnicos con Long Life

Ante cualquier incidencia envíe un correo a info@longlife.com o comuníquese al 91 455 455 45 y le atenderá el equipo técnico.