

UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

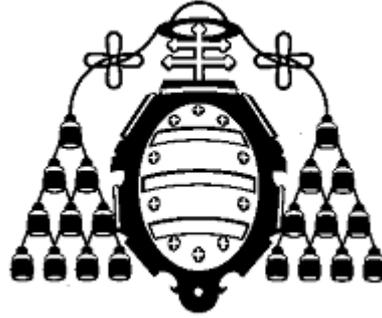
MÁSTER EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

RECONOCIMIENTO DE ACTIVIDAD HUMANA MEDIANTE TÉCNICAS DE
SOFT COMPUTING



PEDRO VICENTE ROMANO GUTIÉRREZ
JULIO 2013



UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

MÁSTER EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

RECONOCIMIENTO DE ACTIVIDAD HUMANA MEDIANTE TÉCNICAS DE
SOFT COMPUTING

DOCUMENTO N° I

MEMORIA



PEDRO V. ROMANO GUTIÉRREZ
JULIO 2013

CIENCIAS DE LA INFORMACIÓN E
INTELIGENCIA ARTIFICIAL

TUTOR: JOSÉ RAMÓN VILLAR FLECHA
COTUTOR: ENRIQUE DE LA CAL MARÍN

ÍNDICE

1.	OBJETIVO	7
2.	INTRODUCCIÓN.....	8
2.1.	DESCRIPCIÓN GENERAL DEL PROYECTO.....	8
2.2.	SUMARIO DEL PROYECTO	8
3.	RECOPIACIÓN DE ANTECEDENTES.....	9
3.1.	INTRODUCCIÓN	9
3.2.	INTELIGENCIA AMBIENTAL Y SU APLICACIÓN EN E – SALUD	9
3.2.1.	<i>Definición.....</i>	9
3.2.2.	<i>Antecedentes históricos.....</i>	9
3.2.3.	<i>Visión actual</i>	10
3.2.4.	<i>Aplicaciones de la Aml en e- salud</i>	10
3.2.5.	<i>Productos disponibles de e- salud</i>	16
3.3.	SISTEMAS EMPOTRADOS Y UBICUOS	17
3.3.1.	<i>Definición.....</i>	17
3.3.2.	<i>Características generales.....</i>	17
3.3.4.	<i>Ámbitos de aplicación</i>	19
4.	DESCRIPCIÓN DEL PROBLEMA A RESOLVER.....	20
5.	DESCRIPCIÓN DE LA SOLUCIÓN	21
5.1.	HARDWARE Y COMPONENTES DISPONIBLES.....	21
5.1.1.	<i>Equipo host.....</i>	21
5.1.2.	<i>Arduino UNO.....</i>	21
5.1.3.	<i>Escudo E-health.....</i>	22
5.1.4.	<i>Sensores.....</i>	25
5.2.	DESCRIPCIÓN DEL SOFTWARE	32
5.2.1.	<i>Arquitectura.....</i>	33
5.2.2.	<i>Programa ejecutado en el microcontrolador de Arduino</i>	35
5.2.3.	<i>Servidor.....</i>	36
5.2.4.	<i>Consola de administración y pruebas.....</i>	38
5.2.5.	<i>Base de datos.....</i>	39
5.3.	ESPECIFICACIÓN DE USO.....	40
9.	PRUEBAS Y RESULTADOS.....	42
9.1.	PRUEBAS CON EL SENSOR DE FLUJO DE AIRE	42
9.2.	PRUEBAS CON EL SENSOR DE TEMPERATURA CORPORAL	42
9.2.1.	<i>Para Frecuencia 1Hz.....</i>	43
9.2.2.	<i>PARA FRECUENCIA 2Hz.....</i>	43
9.3.	SENSOR DE RESPUESTA GALVÁNICA DE LA PIEL	44
9.4.	ELECTROCARDIOGRAMA.....	44
9.4.1.	<i>Prueba a frecuencia 2Hz.....</i>	44
9.4.2.	<i>Prueba a frecuencia de 1Hz</i>	44
9.5.	SENSOR DE POSICIÓN	45

9.6.	SENSOR DE SATURACIÓN DE OXIGENO EN SANGRE	46
9.7.	TENSIÓMETRO.....	47
9.8.	DISCUSIÓN DE RESULTADOS	47
10.	CONCLUSIONES	49
11.	TRABAJOS FUTUROS.....	50
12.	BLIBLIOGRAFÍA.....	51
13.	ANEXO I. ESQUEMA DE LA PLATAFORMA E-HEALTH	53
14.	ANEXO II: MANUAL DE INSTALACIÓN DE RXTX.....	54
15.	ANEXO III. CARGA DE DATOS EN ARDUINO	55
6.	PLANIFICACIÓN Y PRESUPUESTO.....	58
6.1.	PLANIFICACIÓN Y SEGUIMIENTO DEL TRABAJO	58
6.2.	PRESUPUESTO	60
6.2.1.	<i>Recursos humanos</i>	<i>60</i>
6.2.2.	<i>Recursos Técnicos.....</i>	<i>61</i>
6.2.3.	<i>Recursos hardware.....</i>	<i>61</i>
6.2.4.	<i>Recursos software</i>	<i>62</i>
6.2.5.	<i>Presupuesto de ejecución material.....</i>	<i>62</i>
6.2.6.	<i>Presupuesto de ejecución por contrata</i>	<i>62</i>

LISTA DE ILUSTRACIONES:

ILUSTRACIÓN 3.1. AMI APLICADA AL OCIO.....	11
ILUSTRACIÓN 3.2. REPRESENTACIÓN DE LOS AGENTES EN GERAMI.....	15
ILUSTRACIÓN 3.3. E-HEALTH SENSOR PLATFORM.....	17
ILUSTRACIÓN 3.4. ESQUEMA ARQUITECTURA VON-NEUMANN.....	18
ILUSTRACIÓN 3.5. ARQUITECTURA HARVARD.....	19
ILUSTRACIÓN 3.6. EJEMPLO DE APLICACIÓN DE UNO O VARIOS SISTEMAS EMPOTRADOS.....	19
ILUSTRACIÓN 5.1. ARDUINO UNO.....	21
ILUSTRACIÓN 5.2. ESQUEMA DE LA PLATAFORMA E-HEALTH CON TODOS LOS SENSORES.....	23
ILUSTRACIÓN 5.3. CONEXIÓN DE TODOS LOS SENSORES AL PACIENTE.....	24
ILUSTRACIÓN 5.4. VISTA FRONTAL CON LOS CONECTORES DE LOS DIFERENTES SENSORES.....	24
ILUSTRACIÓN 5.5. VISTA TRASERA CON LA CONEXIÓN DEL GLUCÓMETRO.....	25
ILUSTRACIÓN 5.6. CONEXIÓN DE E-HEALTH A ARDUINO UNO.....	25
ILUSTRACIÓN 5.7. ACELERÓMETRO.....	26
ILUSTRACIÓN 5.8. SENSOR OXIGENO EN SANGRE Y PULSÓMETRO.....	27
ILUSTRACIÓN 5.9. CONEXIÓN DE ELECTRODOS A PLATAFORMA E-HEALTH.....	28
ILUSTRACIÓN 5.10. CONEXIÓN DE ELECTRODOS AL PACIENTE.....	28
ILUSTRACIÓN 5.11. SENSOR DE FLUJO DE AIRE.....	29
ILUSTRACIÓN 5.11. CONEXIÓN DEL SENSOR DE TEMPERATURA.....	30
ILUSTRACIÓN 5.12. TABLA CON LOS NIVELES DE PRESIÓN ARTERIAL.....	30
ILUSTRACIÓN 5.13. TENSÍOMETRO CONECTADO A LA PLATAFORMA E-HEALTH.....	31
ILUSTRACIÓN 5.14. COLOCACIÓN DEL TENSÍOMETRO.....	31
ILUSTRACIÓN 5.15. SENSOR GSR Y SU COLOCACIÓN.....	32
ILUSTRACIÓN 5.16. ESTRUCTURA DEL SISTEMA.....	34
ILUSTRACIÓN 5.17. MONTAJE DEL PROTOTIPO.....	35
ILUSTRACIÓN 5.18. ESQUEMA DE COMUNICACIÓN CON SOCKETS EN JAVA.....	38
ILUSTRACIÓN 5.19. CONSOLA DE ADMINISTRACIÓN.....	39
ILUSTRACIÓN 5.20. DIAGRAMA ER DE LA BASE DE DATOS.....	40
ILUSTRACIÓN 5.21. CONEXIÓN DE LOS SENSORES A LA PLATAFORMA E-HEALTH.....	41
ILUSTRACIÓN 9.1. TABLA DE RESULTADOS DE SENSOR DE FLUJO DE AIRE.....	42
ILUSTRACIÓN 9.2. TABLA DE RESULTADOS DEL TERMÓMETRO CON FRECUENCIA 1 Hz.....	43
ILUSTRACIÓN 9.3. TABLA DE RESULTADOS DEL TERMÓMETRO CON FRECUENCIA 1 Hz.....	43
ILUSTRACIÓN 9.4. TABLA DE RESULTADOS DE ECG CON FRECUENCIA 2 Hz.....	44
ILUSTRACIÓN 9.5. TABLA DE RESULTADOS DE ECG CON FRECUENCIA 1 Hz.....	45
ILUSTRACIÓN 9.6. RESULTADOS DE ACELERÓMETRO.....	45
ILUSTRACIÓN 9.7. TABLA CON LOS RESULTADOS DE PULSACIONES POR MINUTO.....	46
ILUSTRACIÓN 9.8. TABLA CON LOS RESULTADOS DE SpO2.....	47
ILUSTRACIÓN 15.1. ARDUINO IDE.....	55
ILUSTRACIÓN 15.2. CARGA DE PROGRAMA EN ARDUINO.....	56
ILUSTRACIÓN 12.1. TABLA DE PLANIFICACIÓN.....	59
ILUSTRACIÓN 12.2. DIAGRAMA GANTT CON LA PLANIFICACIÓN DEL TRABAJO.....	60

PARTE I. MEMORIA

1. OBJETIVO

El reconocimiento de la actividad humana mediante el uso de sensores móviles conjuntamente con las aplicaciones específicas para dispositivos móviles es un tema actualmente en ebullición. Este trabajo tiene como objetivos i) estudiar los aspectos básicos de un sistema e-health, ii) integración de sensores biomédicos en un dispositivo microcontrolado basado en la plataforma Arduino.

Se pretende implementar un sistema que permita monitorizar diferentes constantes vitales de los pacientes mediante sensores biomédicos conectados a una plataforma libre como Arduino, con el fin de obtener datos para su posterior análisis para el diagnóstico preventivo y la rehabilitación ambulatoria de pacientes con problemas motores.

Con todo lo anterior se pretende obtener un prototipo para una plataforma libre de *e-health* capaz de obtener y guardar datos provenientes de sensores distribuidos por todo el cuerpo para su posterior análisis.

2. INTRODUCCIÓN

2.1.Descripción general del proyecto

Actualmente, los sistemas informáticos relacionados con los cuidados sanitarios apoyados en las tecnologías de la información y las comunicaciones (TIC), también denominados e-salud, están en auge para la detección de ciertas anomalías en los estados de salud de los pacientes. Un ejemplo de ello son los escáneres 3D[1] que permiten obtener información sobre una determinada región del cuerpo con características similares a si la visualización de la zona se obtuviese de manera real y además, con este tipo de escáneres, los datos se obtienen en tiempo real.

La e-salud forma parte del término conocido como “Inteligencia Ambiental” (del inglés, *Ambient Intelligence*), en adelante AmI. Desde el punto de vista formal, la AmI es “un área de la Inteligencia Artificial, que intenta dar respuesta a cómo utilizar la información sobre los usuarios de un sistema para mejorar el rendimiento del mismo.”

Por ello, en este proyecto se pretende configurar una plataforma de e-salud basada en una plataforma libre como es Arduino que permita monitorizar y guardar de manera adecuada constantes biomédicas para su posterior análisis.

Las principales motivaciones de la implementación de esta plataforma en Arduino son por una parte, obtener datos de relevancia para su posterior análisis que permitan el diagnóstico preventivo y la rehabilitación ambulatoria de pacientes con problemas motores. Por otra parte, analizar las posibilidades que ofrece en este ámbito una plataforma libre como es Arduino.

De esta manera se puede obtener un prototipo basado en una plataforma de hardware + software libre capaz de hacer mediciones y guardarlas en una base de datos.

2.2.Sumario del proyecto

En la sección 3 se revisará la literatura disponible sobre qué es e-salud y dónde está contenida en el ámbito de la informática junto con qué son y para qué sirven los sistemas empotrados y ubicuos. En la sección 4 se describirá el problema a resolver. En la sección 5 se describirán los componentes que forman parte de la solución así como la solución que se propone para el problema planteado. En la sección 6 se describirán las pruebas hechas sobre los sensores y los diferentes resultados obtenidos. En la sección 7 se discutirán las conclusiones obtenidas tras la realización de este trabajo. En la sección 8 se comentarán los trabajos futuros o trabajos que se pueden realizar como continuación de este. Por último, en la sección 7 se enumeran las diferentes referencias bibliográficas.

En la segunda parte de este documento, se detallará la planificación y el presupuesto de este trabajo.

3. RECOPIACIÓN DE ANTECEDENTES

3.1.Introducción

En esta sección se van a describir en primer lugar qué son y para que se utilizan las técnicas y procesos de e-salud, así como donde están ubicados dentro del campo de la Inteligencia Artificial.

A continuación se describe en primer lugar qué es la inteligencia ambiental (AmI) y que aplicaciones tienen sus técnicas (AAL) [2] en el campo de la medicina, en segundo lugar que son y para que se utilizan las plataformas de e- salud y por último que productos hay disponibles para la implementación de estas plataformas.

Por otra parte también se describen, a grandes rasgos, qué son y para qué se utilizan los sistemas empotrados y ubicuos.

3.2.Inteligencia Ambiental y su aplicación en e - salud

3.2.1. Definición

Como se ha descrito con anterioridad, la Inteligencia Ambiental es “un área de la Inteligencia Artificial, que intenta dar respuesta a cómo utilizar la información sobre los usuarios de un sistema para mejorar el rendimiento del mismo.”

3.2.2. Antecedentes históricos

Se atribuye a Mark Weiser la autoría del concepto AmI cuando trabajaba en 1988 para Xerox en el Parc Lab de California. Weiser y sus colaboradores desarrollaron el primer escenario de cientos de pequeños computadores interconectados.

En junio de 1998, Palo Alto Ventures y Silicon Artists, por encargo de Philips, desarrollan un primer informe sobre AmI como la visión que transformaría la industria electrónica para ofrecer equipos según las ideas originarias de Weiser. [3]

En 1999, Philips se une a la denominada “Oxygen Alliance”, cuyo objetivo estaba determinado por el desarrollo de la tecnología digital del siglo XXI. El consorcio definió una “visión tecnológica” en la que estos ordenadores ubicuos interconectados pudiesen aprender de sus usuarios, de manera que pudiesen mejorar las necesidades de las personas. La idea era que los computadores se adaptasen a las preferencias y/o restricciones de las personas y no al revés.

Posteriormente, sería la Comisión Europea la que incentivaría el área de AmI, convirtiéndola en objetivo prioritario de investigación de la Unión Europea en su VI Programa Marco.

3.2.3. Visión actual

Actualmente, se describe la AmI como un nuevo campo de trabajo consistente en la creación de espacios habitables o entornos inteligentes donde los usuarios interactúen de manera natural e intuitiva con servicios computacionales que les faciliten llevar a cabo una determinada actividad. Estos servicios deben estar caracterizados por lo siguiente.

- Integración con el entorno.
- Capacidad de percibir el contexto en el cual están embebidos.
- Personalización y adaptación al perfil del usuario.
- Capacidad de anticipación a las peticiones del usuario.

La AmI establece una perspectiva de futuro donde las personas se encuentran rodeadas de interfaces inteligentes empotrados en objetos cotidianos, como puede ser un televisor, un PC o una lavadora, generando un entorno capaz de reconocer y dar respuesta a las peticiones de las personas.

De este modo, la persona pasa a ser el centro del desarrollo tecnológico, es decir, todo se mueve alrededor de proporcionar servicios o ergonomía a las necesidades de las personas por lo que los sistemas inteligentes han de poder adaptarse a ellas.

3.2.4. Aplicaciones de la AmI en e- salud

Como se ha comentado con anterioridad, la AmI ha ido evolucionando los dispositivos que rodean a las personas de manera que estos se adapten a las necesidades de las personas.

Existen múltiples aplicaciones de la AmI, desde la mejora de la ergonomía en espacios públicos, oficinas o domótica hasta las mejoras en ciertas ofertas de ocio como pueden ser videojuegos o videoconsolas que interactúan en base a movimientos y/o gestos del cuerpo humano. Se subraya con esto el carácter social de esta tecnología, lo que se denomina AAL. [2]



Ilustración 3.1. AmI aplicada al ocio.

En cuanto a la aplicación de la AmI a entornos de e – salud, hay numerosas aplicaciones, por ejemplo hoy en día ya se pueden ver radiografías o informes sobre determinadas patologías de los pacientes desde un sitio remoto o por ejemplo realizar cirugías a través de robots dirigidos a distancia [4].

También existe una amplia gama de tecnologías de sensores disponibles para medir diversas constantes, por ejemplo, nivel de glucosa en sangre, electrocardiograma (ECG)[5][6], flujo de aire, etc.

Por ejemplo, se ha desarrollado lo que se denomina Monitor de Electrocardiograma Personal (PEM) que forma parte de un proyecto europeo. La motivación principal del PEN es la temprana detección de los diferentes eventos que se producen en el pulso cardiaco para detectar anomalías. El sistema genera diferentes niveles de alarma según la gravedad del análisis de las mediciones. Todos estos datos se envían de forma inalámbrica en tiempo real a las asistencias sanitarias si se produce una situación de gravedad del paciente o se almacena en el propio dispositivo para ser consultado con posterioridad si es un episodio leve.

Todas estas tecnologías son susceptibles de mejora si se tiene en cuenta los diferentes perfiles de los usuarios, tales como historial médico, datos biológicos o hábitos alimenticios.

Este es el nuevo reto que persiguen la informática y la medicina día a día, tratar de apoyarse en las TIC para mejorar los entornos de diagnóstico precoz, tratamiento y prevención de enfermedades, así como para mejorar el estilo de vida de las personas que tienen una determinada discapacidad o tratamiento crónico que no les permite llevar a cabo las tareas cotidianas de manera independiente.

Actualmente, la atención sanitaria mediante TIC está considerado uno de los principales objetivos para mejorar la salud y la independencia de las persona [7]. Por ejemplo, en el Reino Unido, antes de la actual recesión económica, con un presupuesto de 9,2 millones de euros se han asignado para las mejoras de los estos sistemas de información sanitaria durante un periodo

de 10 años. El objetivo es contribuir a la calidad y eficiencia de la atención al paciente, facilitando la labor también a los facultativos, sobremanera con paciente de edad avanzada o que no pueden asistir por sus propios medios a los centros de salud.

Las tecnologías ubicuas posibilitan esta labor gracias a que son dispositivos portátiles, tales como microsensors integrados en prendas textiles, muñequeras, etc. Estas tecnologías están en constante desarrollo para poder cada vez más adaptarse a las necesidades de los usuarios permitiéndoles monitorear y mantener la salud de los pacientes mientras llevan a cabo las tareas habituales de su vida. Con todo esto, además de mejorar la calidad de vida de las personas se produce un moderado ahorro de dinero al reducir la necesidad de la atención constante [7].

Un aspecto importante, es que estas tecnologías de e –salud pueden permitir la detección temprana del deterioro de los patrones de comportamiento y de salud. Algunos patrones, tales como los que afectan a la salud o indican los primeros signos de deterioro físico o cognitivo, no pueden ser fácilmente detectables en un entorno clínico y solo pueden ser detectables en el mediante el monitoreo en el hogar del paciente. Por ejemplo, alguien puede dejar de administrar la medicación correctamente, pero sólo cuando la televisión está encendida o cuando tienen una visita, es decir, se producen situaciones dentro del hogar que son claramente diferentes a las de un centro de salud u hospital.

La Organización Mundial de la Salud (OMS) insta a los estados miembros que cumplan con los siguientes requisitos:

- Desarrollar las infraestructuras necesarias para llegar a las comunidades, grupos vulnerables a los servicios de e- salud adaptándolos a las necesidades de los pacientes.
- Desarrollar tecnologías de la información y las comunicaciones para promover el acceso equitativo, asequible y universal, con el fin de que todo el mundo pueda beneficiarse de estos avances tecnológicos y capaces de reducir los costos en salud.

Sin embargo, el uso de estos sistemas también tiene sus riesgos, sobremanera en temas de seguridad de datos de carácter personal. Con el uso de estos sistemas, la cantidad de datos que se posee de un paciente es mucho mayor que con los sistemas tradicionales por lo que se han de utilizar los mecanismos más sofisticados para la protección de datos.

Se han desarrollado numerosos experimentos y productos en el campo de la e salud, algunos de ellos ya forman parte de la vida cotidiana de las personas. A continuación se van a exponer dos ejemplos desarrollados en la literatura, que evidencian la importancia que tiene la AmI en entornos médicos.

En [9] se hace un estudio de los informes que de manera periódica se llevan a cabo en un hospital. Se han desarrollado dos escenarios para la gestión de información y la interacción con

el entorno. Los diferentes escenarios se han utilizado para la especificación de entornos de AmI en hospitales y para el desarrollo de un *middleware* basado en agentes denominado SALSA.

Las características generales que comparten casi todos los hospitales son la naturaleza distribuida de la información, la necesidad de colaboración y movilidad del personal y la necesidad de poseer información médica precisa sobre el estado de los pacientes.

Uno de los escenarios involucra al médico (García) que está examinando a un paciente en una sala y recibe un mensaje en su PDA que indica que los resultados de los rayos X de un paciente (P) en una determinada habitación (225) están disponibles para ser consultados. El médico cuando va a consultar los datos de ese paciente, la pantalla del sistema de información del hospital identifica al médico y cambia de forma que se adapta a los mensajes, calendario, etc.

En este caso el sistema es consciente del contexto, que es el paciente, los rayos X recién recibidos, proponiendo al médico un tratamiento y/o diagnóstico personalizado para el paciente.

En este escenario, el sistema se adapta a las necesidades tanto del paciente como del médico, ya que sobre este último infiere y recomienda las diferentes fuentes de información acerca de los tratamientos alternativos para el paciente P.

Otro de los escenarios posibles dentro del hospital, implica la colaboración de dos facultativos de diferentes áreas, por ejemplo, si se necesita por parte de la unidad de cardiología la presencia un traumatólogo en un determinado momento en el piso superior. Estratégicamente el cardiólogo envía un mensaje al traumatólogo para solicitar una reunión para discutir un diagnóstico de un paciente determinado. El traumatólogo acepta a través del PDA, se acerca a una pantalla de información centralizada del hospital por medio de la cual los dos especialistas tienen una sesión de colaboración sobre el diagnóstico de un paciente.

En la arquitectura de SALSA, los agentes se utilizan como abstracciones, para actuar en nombre de los usuarios, para representar a los servicios o para proporcionar un interfaz para el usuario de funciones complejas.

Este sistema necesita que sea sensible al contexto, donde el contexto es identificado por la identificación del usuario y su ubicación. Toda la información de los usuarios se almacena en sus PDAs.

En [11] se ha desarrollado un proyecto denominado AmbieSense. Tiene como objetivo el apoyo a los trabajadores de un centro hospitalario para el diagnóstico y tratamiento de los pacientes. Se utiliza un razonamiento basado en casos (CBR) [11]. CBR es una técnica de resolución de problemas que almacena problemas con sus soluciones conocidas. Entonces, dado un nuevo problema, encuentra un caso de los casos conocidos que mejor se adapte o mejor

solución proporcione al nuevo problema. Con todo ello, se genera un contexto basado en varios parámetros, como son la ubicación, la identidad, la hora y el día, y cualquier otro artefacto que esté presente.

Por ejemplo, si el objetivo está asociado a “Evaluar el tratamiento de los pacientes” la arquitectura del sistema es la siguiente:

- La capa de percepción, que es la encargada de recopilar información del entorno con respecto a los parámetros que se usan para clasificar el contexto del paciente.
- La capa de conciencia, donde se aplica el razonamiento basado en casos para que coincida con los datos de la capa de percepción frente a los casos almacenados (ya conocidos) para encontrar la coincidencia más cercana.
- La capa de sensibilidad, que recibe el objetivo reconocido por la capa de conciencia junto con lo que se conoce. Tiene para cada objetivo una secuencia de tareas que le permiten lograr el objetivo deseado.

Un ejemplo donde se aplica esta técnica son las reuniones llevadas a cabo entre los/as médicos/as y los/as enfermeros/as a lo largo de la mañana. El objetivo de estas reuniones es discutir el tratamiento que se les debe suministrar a los pacientes. Con todo ello lo que hacen es dividir las discusiones sobre un determinado paciente en tareas y subtareas.

Otro de los trabajos que son susceptibles de ser reseñados, además por tener aplicación directa en España es GerAmi[11].

Este trabajo está también basado en el uso de CBR. En él se describe un sistema de Inteligencia Ambiental Geriátrica para pacientes de Alzheimer. El centro de atención es la Residencia de la Santísima Trinidad, ubicado en Salamanca, que proporciona atención para un máximo de 60 pacientes de la tercera edad.

Los objetivos del sistema GerAmi son la monitorización de pacientes y gestión del personal médico.

El sistema contiene lectores RFID en las puertas, ascensores, etc. Además se disponen de etiquetas o brazaletes de identificación para los pacientes. De esta manera, el personal sanitario, a través de su PDA puede conocer el estado y ubicación de cada paciente.

La arquitectura del sistema utiliza una estructura multiagente [12] utilizando BDI7 (*belief, desire, intention*).

La aplicación incluye a 30 pacientes, 10 enfermeras y un gerente, cada uno de los cuales se asocia con un agente. El *manager* y los agentes de los pacientes se ejecutan en un ordenador central y los agentes de las enfermeras se ejecutan en los dispositivos móviles.

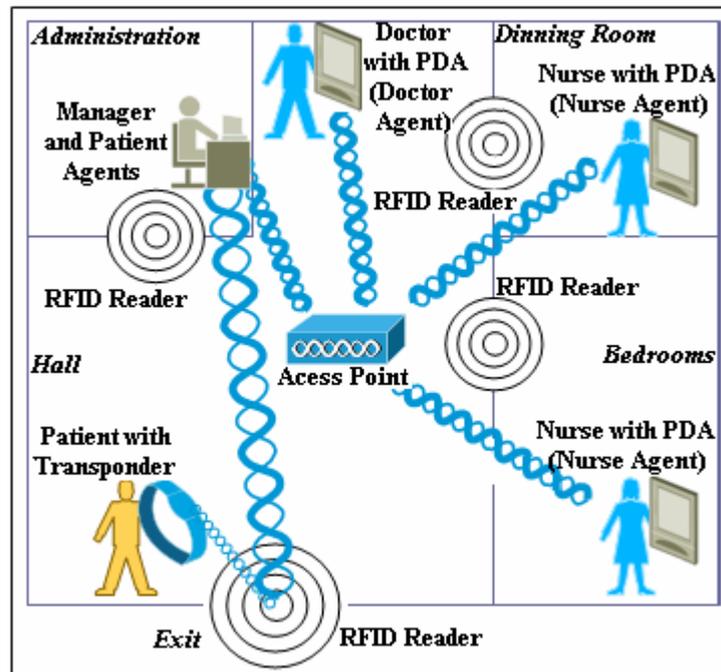


Ilustración 3.2. Representación de los agentes en GerAmi.

El agente del paciente registra la ubicación del paciente cada hora y envía el registro a una base de datos centralizada. El agente que hace de *manager* o de coordinador juega dos roles, por una parte controla las cerraduras y alarmas basadas en ubicaciones de los pacientes, evitando así situaciones de peligro, y por otra parte es el encargado de asignar las tareas que han de llevar a cabo las enfermeras sobre los pacientes. Para este último propósito se utiliza información sobre la disponibilidad de las enfermeras, es decir, si están ocupadas con otro paciente, de vacaciones, etc. También utiliza para la asignación de las tareas los datos sobre los perfiles de las enfermeras en términos de experiencia de prestación de servicios con la restricción de la duración de la jornada laboral.

Para cada tarea se utiliza una estimación del tiempo que ha de emplear la enfermera, por ello cuantos más datos haya en la base de datos, más precisión tendrá el sistema.

Una vez se ha escogido a que enfermera se le va a encargar una determinada tarea, se transmiten estas a través de las PDAs.

La reprogramación de todo a lo anterior puede ser necesaria en casos en los que el paciente no esté en la ubicación correcta o un recurso no esté disponible por alguna emergencia.

Todo ello se almacena en la base de datos con el objetivo de proporcionar conocimiento acerca del estado de los pacientes.

Uno de los beneficios que ha conseguido este sistema, es la disminución del tiempo que las enfermeras dedican a tareas que no están directamente involucradas en la atención de los pacientes como pueden ser la elaboración de informes.

Otro de los beneficios que se han obtenido, es la reducción del número de enfermeras que trabajaban en un turno de guardia de 24 h a pesar que el número de pacientes era el mismo.

Con estos ejemplos, se trata de explicar la motivación sobre la que se ha fundamentado este trabajo.

3.2.5. Productos disponibles de e- salud

Existen numerosos productos comerciales de sistemas de e-salud o al menos que se promocionan como tal, entre ellos cabe destacar los siguientes:

eHealth Systems MyDoctor

Según su web [13] *“monitorea y activa automáticamente a personas con enfermedades crónicas para el diagnóstico oportuno, control automatizado y apoyo al autocuidado de personas con diabetes y otras enfermedades crónicas. Usando llamadas automatizadas y mensajería de texto (SMS), este sistema asegura la continuidad del tratamiento y disminuye la morbilidad y costo del cuidado como ha sido comprobado en diversos países del mundo incluyendo Chile y EE.UU.*

La innovación de este sistema es automatizar el proceso de control y extender el seguimiento del paciente más allá de la organización de salud para:

- *Asegurarse de que tenga las citas necesarias y requeridas por el régimen de garantías explícitas en salud (GES)*
- *Que asista a esas citas o alerte al centro de su inasistencia para optimizar el tiempo profesional*
- *Apoyar su toma de medicamentos y cambio de estilo de vida*
- *Alertar al personal clínico de cualquier factor de riesgo o no adherencia al tratamiento por el paciente para poder apoyarlo en su control”*

Ericsson Mobile Health[14]

Plataforma de software propietario para servicios de asistencia a la salud. Proporciona monitoreo y control automatizado utilizando una plataforma móvil.

Libelium e-Health Sensor Platform

Sin embargo, todos estos sistemas utilizan software propietario y no pueden ser adaptados a unas determinadas situaciones sin el consentimiento de sus propietarios y bajo determinadas condiciones. Por ello existen plataformas libres, como puede ser Arduino o Raspberry PI que poseen escudos para el desarrollo de dispositivos y de monitores de e-salud. Uno de ellos es el desarrollado por Libelium, denominado e- Health Sensor Platform.

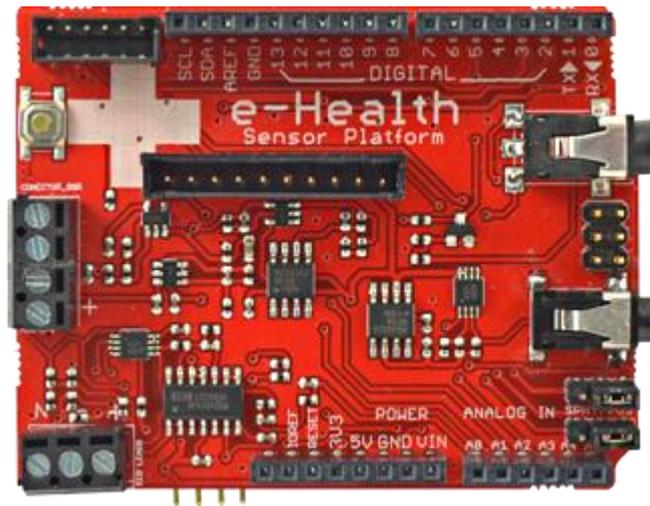


Ilustración 3.3. E-Health Sensor Platform

En este trabajo, por tratarse de un trabajo de investigación se desarrollará un prototipo basado en una plataforma libre concretamente Libelium e-Health. Se detallarán sus características en el siguiente capítulo de este documento.

3.3.Sistemas empotrados y ubicuos

3.3.1. Definición

Un sistema empotrado es un sistema computacional diseñado específicamente para una tarea o un ámbito de aplicación específico. Al contrario de lo que ocurre con los ordenadores personales, que están diseñados para múltiples y diversos tipos de tareas, en un sistema empotrado la mayoría de sus componentes están incluidos en la placa, por ejemplo microprocesador / microcontrolador, memoria, comunicaciones, etc.

Los sistemas empotrados interactúan con su entorno mediante diversos tipos de dispositivos que normalmente no son convencionales, por ejemplo, convertidores A/D y D/A, PWM, entradas y salidas digitales paralelo y serie, interfaces con sensores, actuadores, periféricos especiales, etc.

Un sistema ubicuo es un computador o un componente ubicado que por su construcción o por sus características son invisibles o pasan desapercibidos por los usuarios.

3.3.2. Características generales

Los sistemas empotrados y ubicuos tienen características similares, por lo que a continuación se citan algunas de ellas:

- **Fiabilidad y seguridad:** Son sistemas que pueden controlar mecanismos o actuadores críticos tanto de seguridad como de eficiencia económica por lo que deben ser robustos respecto a la forma de manejar fallos, excepciones, etc.

- **Eficiencia:** si estos sistemas están en un subsistema de control, deben responder con gran rapidez a los cambios en el sistema controlado.
- **Interacción con dispositivos físicos:** estos tipos de sistemas suelen interactuar con el entorno mediante sensores y actuadores que controlan por ejemplo, humedad, temperatura.
- **Robustez:** han de ser sistemas robustos ya que en determinadas circunstancias trabajan a temperaturas y condiciones adversas.
- **Bajo consumo:** Muchos de estos sistemas se alimentan con baterías a muy bajo voltaje y consumo con el fin de que su tiempo de actividad sin interrupción sea alto.
- **Dimensiones y peso.** Suelen ser sistemas con dimensiones reducidas y peso reducido.
- **Dificultad de programación eficiente:** al disponer de poca memoria, la programación debe ser lo suficientemente eficiente para que en un espacio reducido de memoria se pueda programar toda la funcionalidad deseada.

3.3.3. Arquitecturas

Existen básicamente dos arquitecturas en los sistemas empotrados, la arquitectura Von-Neumann y la Harvard. A continuación se detallan sus características principales:

- **Arquitectura Von-Neumann:** En este caso el programa y los datos son almacenados en la misma memoria y se dispone de un único bus de datos. Esto requiere dos búsquedas secuenciales en memoria principal, una para la instrucción y otra para el dato.

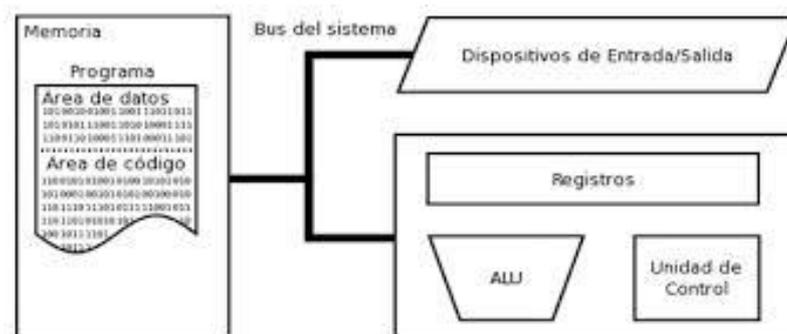


Ilustración 3.4. Esquema arquitectura Von-Neumann

- **Arquitectura Harvard:** En este caso el bus de datos y el de instrucciones están separados lo que produce que se puedan llevar a cabo búsquedas de forma paralela en memoria (Arquitectura segmentada). Se puede estar buscando la siguiente instrucción a la vez que se acceden a los datos de la instrucción en curso.

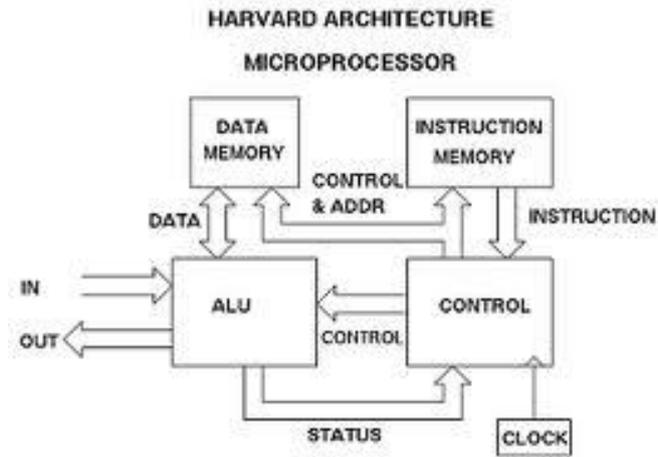


Ilustración 3.5. Arquitectura Harvard

3.3.4. Ámbitos de aplicación

El uso de un sistema empotrado es de lo más heterogéneo, por una parte se pueden encontrar en ámbitos de la vida cotidiana como puede ser lavadoras, coches, televisores o también en ámbitos industriales como puede ser el control de temperatura de altos hornos o en procesos de logística, etc.

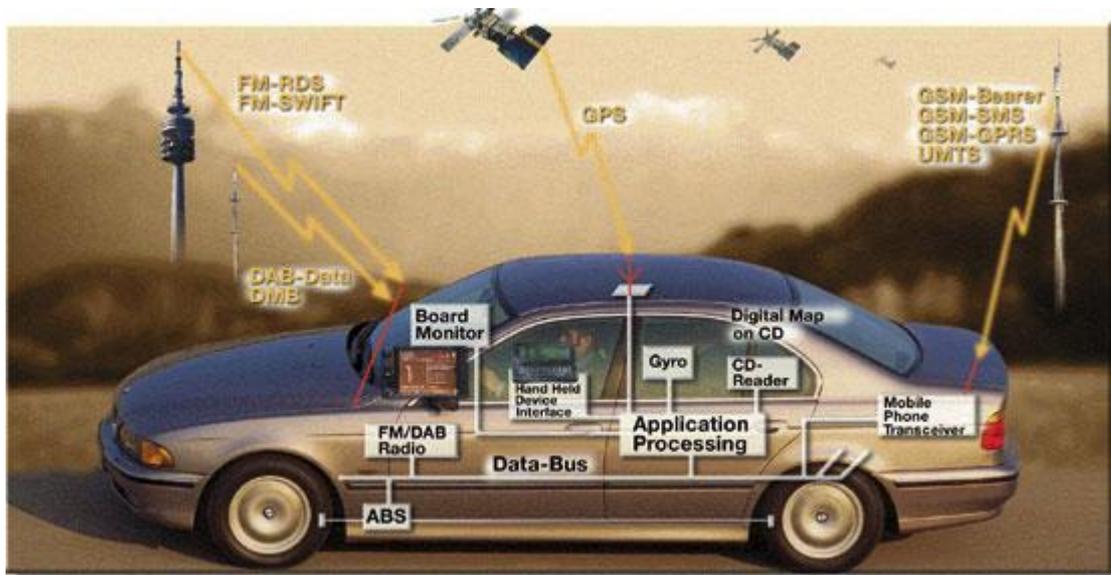


Ilustración 3.6. Ejemplo de aplicación de uno o varios sistemas empotrados.

4. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Una vez analizados algunos de los temas que actualmente están en ebullición sobre las plataformas de e-salud, se pretende construir un sistema de monitorización de las variables biomédicas basado en una plataforma de hardware libre como es Arduino y el conjunto de sensores e-Health.

La principal motivación de este trabajo es conseguir datos de sensores biomédicos con el fin de obtener un histórico de las evoluciones de los pacientes en base a las medidas obtenidas por los sensores.

De esta manera, estos datos pueden ser consultados para su posterior análisis para observar patrones de comportamiento sobre los pacientes que permitan observar anomalías y/o enfermedades como pueden ser la reducción de la movilidad, hipertensión, etc.

Un aspecto importante que caracteriza a este prototipo es que está basado en hardware y software libre por lo que es susceptible de modificación y mejora continua para recoger datos de una forma precisa.

5. DESCRIPCIÓN DE LA SOLUCIÓN

5.1. Hardware y componentes disponibles

En este epígrafe se describe el soporte hardware en el que se basa este trabajo, en primer lugar se describirá que equipo debe contener el prototipo para su correcto funcionamiento, seguidamente se describirá lo que es Arduino, e-Health y cada uno de los sensores.

5.1.1. Equipo host

Para el buen funcionamiento del prototipo se establecerá como hardware mínimo el siguiente ordenador:

- Intel Pentium M,
- 1,86 GHZ,
- 1 GB de memoria RAM,
- Espacio mínimo de almacenamiento de 1 GB

5.1.2. Arduino UNO

Arduino [15] es una plataforma de electrónica abierta para la creación y configuración de prototipos basada en software y hardware flexibles. Permite desarrollar prototipos para proyectos de carácter multidisciplinar.

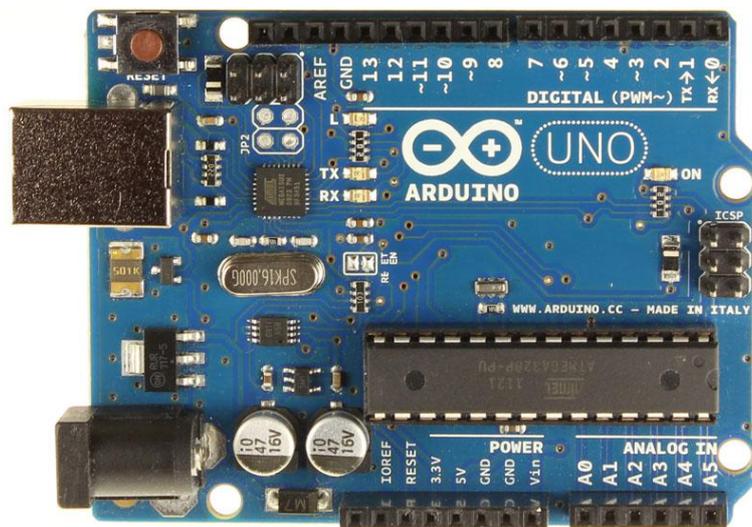


Ilustración 5.1. Arduino UNO

El hardware consiste en una placa con microcontrolador Atmel AVR [16] y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280 y Atmega8 dada su sencillez y bajo coste.

Las especificaciones técnicas del Arduino UNO que se va a utilizar en este trabajo son las siguientes:

- Microcontrolador: ATmega328
- Tensión de funcionamiento (lógica): 5V
- Tensión de alimentación (recomendada): 7-12 V
- Tensión máxima (no recomendada): 20 V
- Pines E/S Digital: 14
- Pines de entrada analógica: 6
- DC I/O Pin: 40 mA
- DC 3.3V Pin: 50 mA
- Memoria Flash: 32KB
- SRAM: 2KB
- EEPROM: 1KB
- Frecuencia de reloj: 16Mhz
- Comunicación serie UART TTL (5v), RX/TX ó USB
- I²C
- SPI
- Bootloader: Optiboot

5.1.3. Escudo E-health

Para la conexión y gestión de todos los sensores se dispone de un escudo denominado E-Health Sensor Platform[17]. Este escudo es compatible con la familia Arduino y también con Raspberry PI.

Este escudo permite realizar aplicaciones biométricas y de vigilancia médica pudiendo obtener datos de hasta nueve sensores diferentes. Por ejemplo, nivel de saturación de oxígeno en sangre (SpO₂), flujo de aire (respiración), termómetro corporal, electrocardiograma (ECG), acelerómetro, etc.

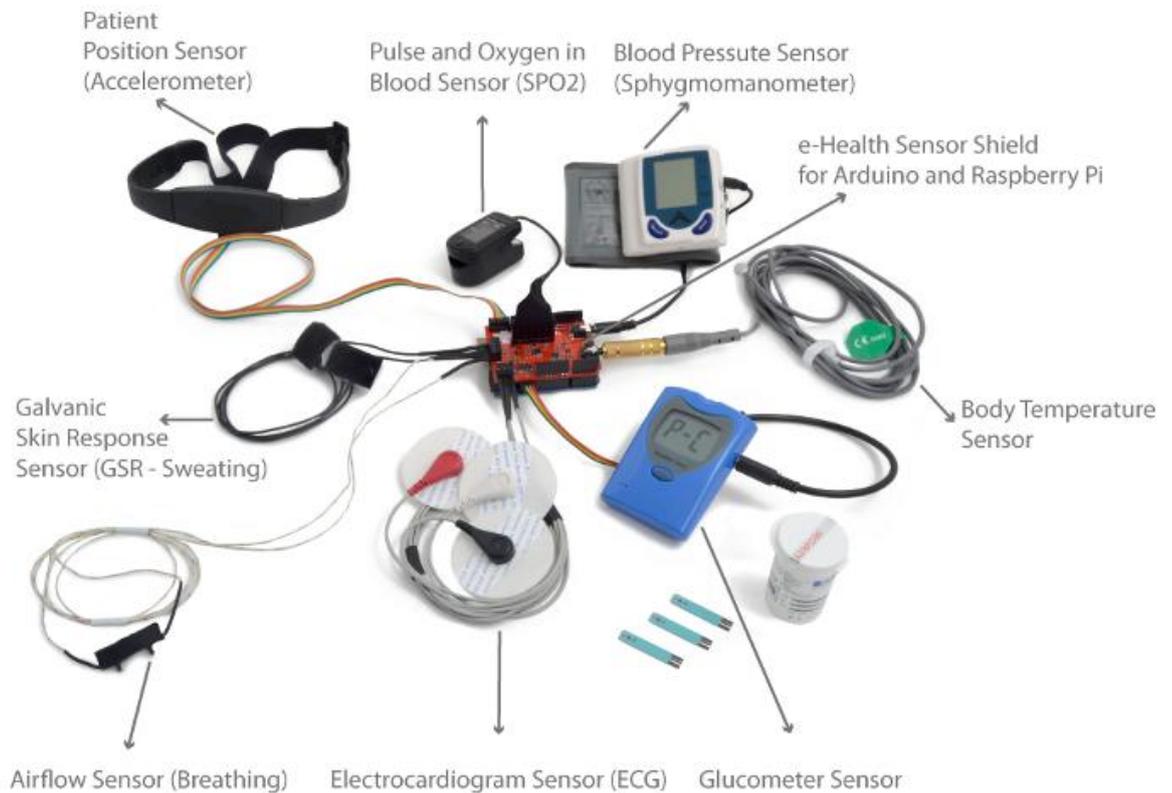


Ilustración 5.2. Esquema de la plataforma E-Health con todos los sensores

A parte de estos sensores, si se necesita un diagnóstico basado en alguna imagen corporal, se puede conectar una cámara al módulo y enviar datos a través de un módulo 3G.

La característica principal que diferencia este dispositivo de e-salud comparado con otros es que tanto el hardware como el software son libres, por lo que, se tiene una plataforma libre para aplicaciones de e-salud.

Uno de los puntos clave de gestión de los pacientes, es la privacidad de los datos personales y médicos, por ello la plataforma implementa los siguientes mecanismos de protección:

- **Capa de enlace de comunicación:** Cifrado AES 128 para 802.14.5/ZigBee y WPA2 para Wifi.
- **Capa de aplicación:** Protocolo HTTPS que utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información privada que el protocolo HTTP. De este modo se consigue que la información privada (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará casi imposible de descifrar.

La siguiente figura muestra el esquema de conexión de todos los componentes:



Ilustración 5.3. Conexión de todos los sensores al paciente.

Al ser una plataforma libre y en constante desarrollo, no hay un manual técnico con las especificaciones técnicas del dispositivo. A continuación se muestran dos imágenes sobre las conexiones de sensores disponibles.

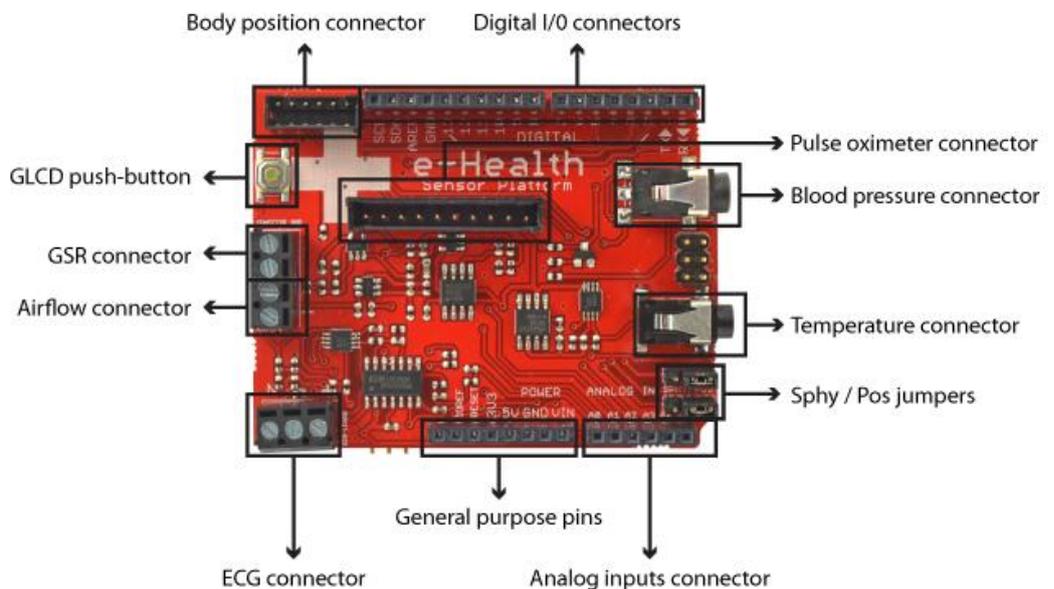


Ilustración 5.4. Vista frontal con los conectores de los diferentes sensores.



Ilustración 5.5. Vista trasera con la conexión del Glucómetro

Para disponer de más información sobre la plataforma, se puede consultar el esquema electrónico del dispositivo en el Anexo I.

A continuación se muestra una imagen de la conexión de Arduino + E-Health Sensor Platform.

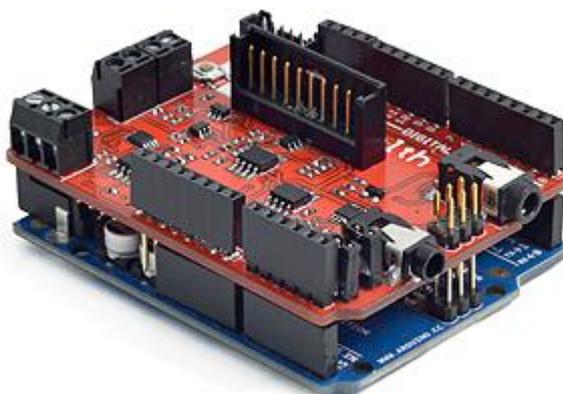


Ilustración 5.6. Conexión de E-Health a Arduino UNO

5.1.4. Sensores

A la plataforma E-Health descrita en el epígrafe anterior se pueden conectar una serie de sensores biomédicos con el objeto de, por ejemplo, monitorizar la actividad pulmonar, esquemas de movimiento, etc.

A continuación se detalla cada uno de los sensores de los que se dispone para el desarrollo de este trabajo.

Acelerómetro

A la plataforma se le puede conectar un acelerómetro similar a que se muestra en la siguiente figura.



Ilustración 5.7. Acelerómetro

El acelerómetro es un sensor que permite medir aceleraciones. En este caso, permite obtener la posición del cuerpo en base a tres acelerómetros, uno para el eje X, otro para el eje Y y otro para el eje Z.

Características:

- Tensión de alimentación: 1.95V – 3.6 V
- Tensión del interfaz: 1.6V – 3.6 V
- Escalas disponibles: +-2g, 4g, +-8g

Concretamente este sensor retorna la posición del cuerpo, pudiendo obtener los datos de aceleración o las siguientes posiciones del paciente en base a los datos de los ejes:

- Posición supina (boca arriba)
- Posición boca abajo
- Posición lateral derecha
- Posición lateral izquierda
- De pie / sentado

Sensor de oxígeno en sangre y medidor de pulso (SpO2)

Este sensor mide la saturación de oxígeno que hay en la sangre y el recuento de pulsaciones por minuto al que late el corazón.

La saturación de oxígeno se define como la medición de la cantidad de oxígeno disuelto en la sangre, sobre la base de la detección de la hemoglobina (Hb) y Desoxihemoglobina (HbO₂). Se utilizan dos longitudes de onda de luz diferentes para medir la diferencia real en los espectros de absorción de Hb y HbO₂. La circulación sanguínea se ve afectada por la concentración de Hb y HbO₂, y sus coeficientes de absorción se miden usando dos longitudes de onda de 660 nm (espectro de luz roja) y 940 nm (espectro de luz infrarroja).



Ilustración 5.8. Sensor oxígeno en sangre y pulsómetro.

La Hemoglobina desoxigenada (Hb) tiene su valor máximo de absorción de luz a 660 nm y la hemoglobina oxigenada (HbO₂) tiene su valor máximo de absorción de luz a 940 nm. Posteriormente un foto-detector percibe la cantidad de luz no absorbida del LED, lo que permite calcular la saturación de oxígeno en sangre.

Este tipo de sensores son útiles cuando se posee un paciente con nivel de oxígeno en sangre inestable, debido a enfermedades crónicas como puede ser la EPOC[18]. Estas enfermedades requieren de un control minucioso de los niveles de oxígeno para controlar si se necesita una fuente de oxígeno suplementario. Los rangos normales aceptables para los pacientes fluctúan entre un 95 a 99 por ciento.

Electrocardiograma (ECG)

El ECG es una de las pruebas médicas más utilizadas hoy en día para el diagnóstico y/o control de enfermedades cardiovasculares.

En este trabajo no se detallarán las interpretaciones médicas de este sensor debido al desconocimiento de las técnicas aplicadas a la medición y observación de resultados.

Este sensor se basa en la medición de impulsos eléctricos en base a tres sondas o electrodos. Se obtiene un valor de tensión medido en voltios.

A continuación se detalla la forma de conexión tanto a la plataforma E-Health como al paciente.

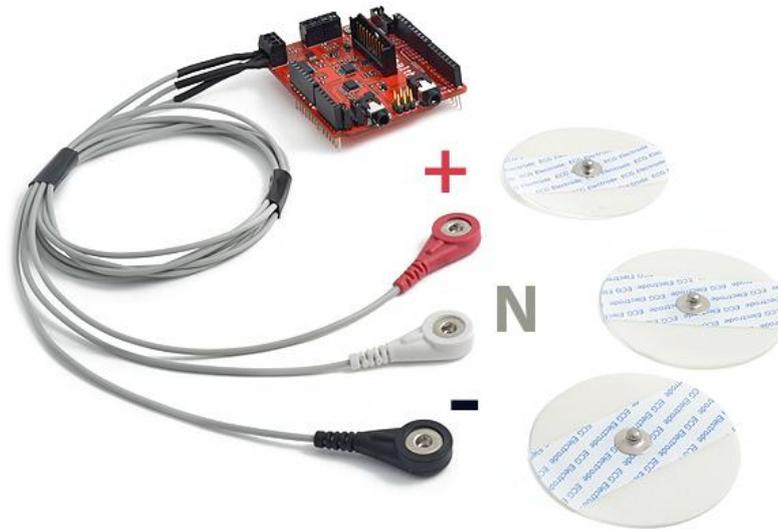


Ilustración 5.9. Conexión de electrodos a plataforma E-Health

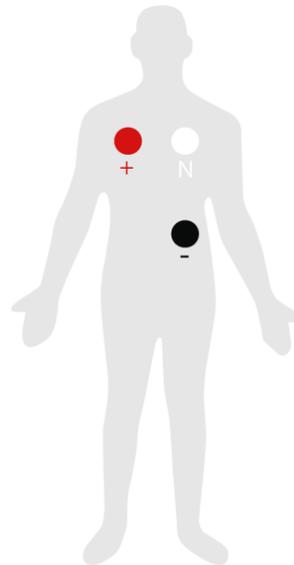


Ilustración 5.10. Conexión de electrodos al paciente

Sonda de flujo de aire

Este sensor mide el flujo de aire permitiendo observar si la curva de respiración del paciente presenta alguna anomalía significativa. Se suele utilizar para la detección de hipoxemia[19] y apnea[21].

Este dispositivo está basado en un sensor termopar que permite detectar con precisión los cambios de flujo de aire en las fosas nasales, así como la temperatura del aire.



Ilustración 5.11. Sensor de flujo de aire

Termómetro o sensor de temperatura corporal

Este es, posiblemente, el sensor más conocido y utilizado, ya que permite conocer la temperatura corporal de un paciente y en base a ello conocer si dicha temperatura está en los niveles normales o por el contrario se encuentra en niveles elevados provocados por algún tipo de afección.

Los rangos son los siguientes:

- **La hipotermia** $<35,0\text{ }^{\circ}\text{C}$ ($95,0\text{ }^{\circ}\text{F}$)
- **Normal** $36,5$ a $37,5\text{ }^{\circ}\text{C}$ ($97,7$ - $99,5\text{ }^{\circ}\text{F}$)
- **Fiebre o hipertermia** $> 37,5$ a $38,3\text{ }^{\circ}\text{C}$ ($99,5$ a $100,9\text{ }^{\circ}\text{F}$)
- **Hiperpirexia** $> 40,0$ a $41,5\text{ }^{\circ}\text{C}$ (104 a $106,7\text{ }^{\circ}\text{F}$)

Cuando se utiliza el sensor de temperatura, en realidad se está midiendo una tensión, por cada tensión se tiene una correspondencia de temperatura, por lo que es muy importante la calibración de este sensor para que mida correctamente el nivel de temperatura corporal.

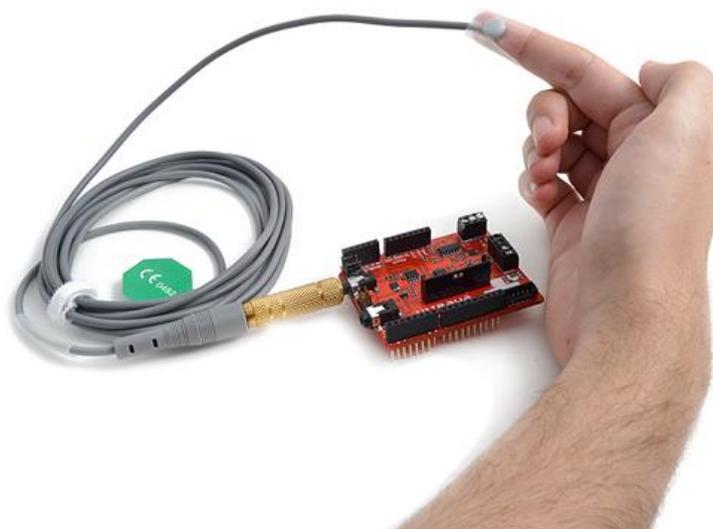


Ilustración 5.11. Conexión del sensor de temperatura

Tensiómetro

La presión arterial es la presión de la sangre en las arterias. Cuando el corazón late, se contrae y empuja la sangre a través de las arterias hacia el resto del cuerpo. Esta fuerza que imprime el corazón sobre la sangre crea la presión en las arterias. La presión arterial se registra como dos valores, la presión sistólica (cuando el corazón late) y la diastólica (cuando el corazón se relaja entre los latidos).

A continuación se muestra una tabla con los valores de la presión arterial para personas mayores de 18 años.

	Sistólica (mmHg)	Diasistólica (mmHg)
Hipotensión	< 90	< 60
Normal	90-119	60-79
Pre-hipertensión	120 – 139	80 – 89
Hipertensión Nivel 1	140-159	90 -99
Hipertensión Nivel 2	160 -179	100 – 109
Crisis Hipertensiva	≥ 180	≥ 110

Ilustración 5.12. Tabla con los niveles de presión arterial

El encargado de medir la presión arterial es el tensiómetro, del que se obtienen las medidas de presión arterial sistólica y presión arterial diastólica.



Ilustración 5.13. Tensiómetro conectado a la plataforma E-Health

Este dispositivo se conecta a través de un cable con conector jack para la comunicación con el escudo E- Health

El tensiómetro tiene que recibir una alimentación mediante baterías internas de 2 x 3A.

Para su correcto funcionamiento se ha de colocar en la muñeca en un plano horizontal con el corazón como se indica en la siguiente figura.

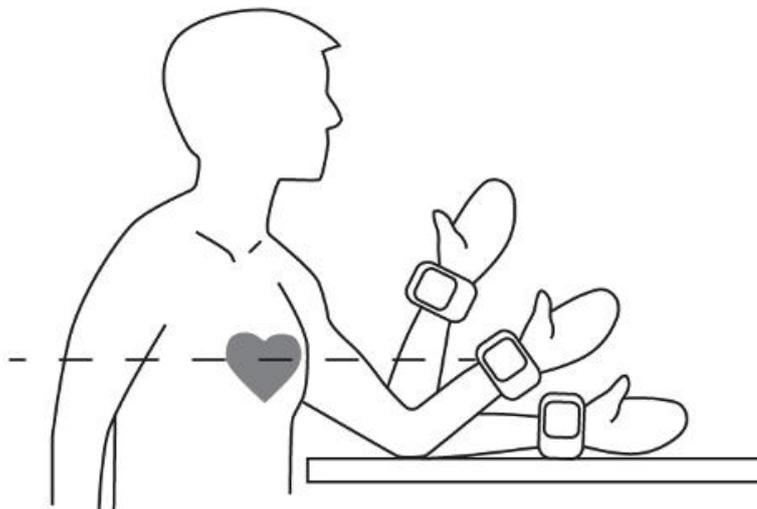


Ilustración 5.14. Colocación del tensiómetro

Sensor de respuesta galvánica de la piel (GSR)

La conductancia de la piel, también conocida como la respuesta galvánica de la piel (GSR) [22] es un método para medir la conductancia eléctrica de la piel, que varía en base a su nivel de

humedad. El interés que tiene este tipo de mediciones es que las glándulas sudoríparas están controladas por el sistema nervioso simpático, por lo que los momentos de una emoción fuerte, cambian la resistencia eléctrica de la piel. Por ello, la conductancia de la piel se utiliza como una indicación de la excitación psicológica o fisiológica.

El sensor de respuesta galvánica de la piel mide la conductancia eléctrica entre dos puntos. La conductividad de la piel se mide en los dedos de la palma de la mano. Lo que se mide es la resistencia eléctrica de la superficie de la piel en base al sudor producido por el cuerpo.

Si se produce un alto nivel de sudoración en la piel, la resistencia eléctrica de la misma disminuye, por el contrario, si se tiene un nivel muy bajo de sudoración superficial en la piel, el nivel de resistencia es mucho mayor.

Estas mediciones se utilizan en muchos ámbitos, tanto médicos, para observar el estado emocional o fisiológico del paciente, o en ámbitos de investigación científica, como pueden ser pruebas de polígrafo.



Ilustración 5.15. Sensor GSR y su colocación.

5.2.Descripción del software

En este apartado se describirá el soporte software sobre el que se desarrolla el prototipo, se presenta tanto los drivers necesarios para controlar la parte de hardware como las herramientas para el diseño del interfaz, así como las principales clases.

Hay que decir que todo el software funciona correctamente si existe todo el hardware antes descrito, es decir, para poder ejecutar la consola de administración se ha de poseer del hardware tanto de Arduino como al menos de la plataforma e-Health.

En primer lugar se citan las diferentes herramientas software necesarias, aunque algunas de ellas se podrían sustituir por otras similares, para el desarrollo de la parte software de este prototipo.

- **Arduino IDE:** Entorno integrado de desarrollo para la programación de la plataforma de Arduino. Proporciona un editor de texto para la programación y también permite seleccionar la forma de comunicación entre el equipo programador al dispositivo programado. También permite la compilación y carga del programa en el microcontrolador de la plataforma Arduino. Además, mediante el puerto serie se puede programar cualquier escudo compatible con la plataforma Arduino. El lenguaje que utiliza está basado en *Wiring*, sintaxis similar a C++ sin implementar herencia múltiple ni *templates*. El usuario debe definir al menos dos funciones `setup()` y `loop()`.
- **NetBeans IDE 7.3.1:** entorno de desarrollo utilizado para el diseño e implementación del interfaz y servidor así como para la programación y compilación de todo el sistema.
- **RXTX:** es una librería Java, que utiliza una implementación nativa (a través de JNI), que proporciona comunicación serie y paralelo para el kit de herramientas de desarrollo Java (JDK). Se basa en la especificación de *Sun Java Communications API*. Sin embargo Java no proporciona soporte para Windows, ese es el motivo de utilizar RXTX. A diferencia de otras librerías para la lectura/escritura del puerto serie, tiene licencia GPL [23]
- **Java Development Kit:** kit de herramientas para el desarrollo de programas en el lenguaje de programación orientada a objetos JAVA, con versión `jdk 1.7.0_25`.

A continuación se describen cada uno de los componentes software que forman parte de la solución.

5.2.1. Arquitectura

Se ha empleado una arquitectura cliente-servidor, de manera que se tiene un programa de bajo nivel en el microcontrolador de Arduino que se encarga de leer datos de los sensores y enviarlos al servidor. Este programa también atiende a peticiones del servidor.

Por otra parte se tiene un programa servidor que se encarga de recibir peticiones de configuración o comandos de selección de sensores y envía a través del puerto serie estos datos al módulo e-Health. El servidor también es el encargado de recibir los datos por parte de los sensores y almacenarlos en una base de datos.

Se tiene también un cliente con una consola de administración, que se comunicará con el servidor para hacer peticiones de lectura de datos de diferentes sensores con determinadas características y parámetros.

Por último se tiene una base de datos donde se almacenan los datos provenientes de los sensores.

A continuación se muestra un esquema con la estructura de sistema propuesta.

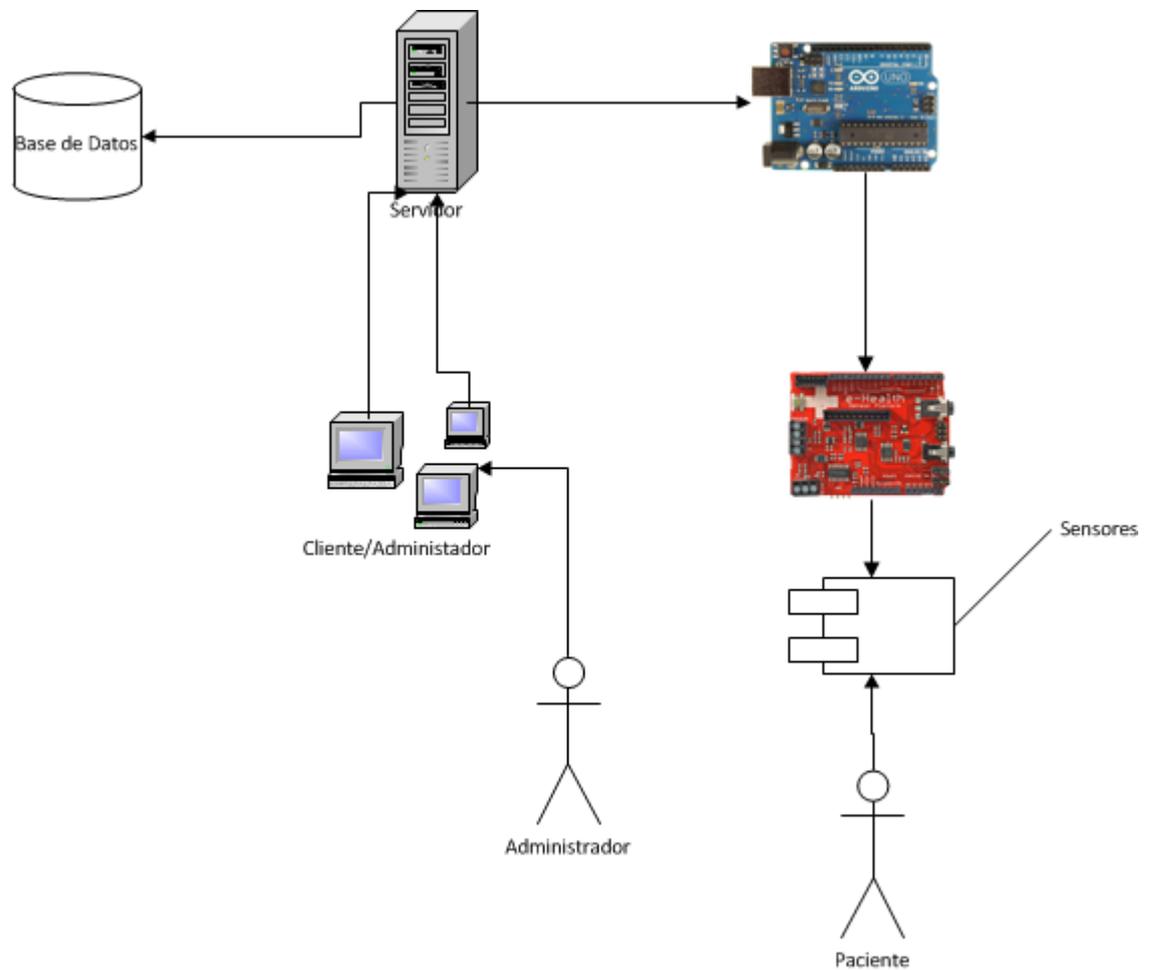


Ilustración 5.16. Estructura del sistema.

A continuación se muestra una imagen real del prototipo construido:



Ilustración 5.17. Montaje del prototipo

5.2.2. Programa ejecutado en el microcontrolador de Arduino

Se ha diseñado un programa que se encarga de la gestión y configuración del escudo E-Health junto con los sensores.

Para ello se dispone de una librería proporcionada por el fabricante de la plataforma E-Health que permite interactuar con los datos de los sensores mediante funciones de alto nivel.

Sin embargo, debido a las necesidades de este trabajo, esta librería está incompleta, por lo que se ha tenido que implementar métodos para por ejemplo obtener los valores numéricos de las posiciones del acelerómetro.

Las funciones que están implementadas en la librería son las siguientes:

Funciones del SpO2:

- `initPulsioximeter()` // Inicialización del pulsómetro.
- `readPulsioximeter()` // Lectura de los datos del pulsómetro.
- `getBPM()` // Devuelve el número de pulsaciones por minuto.
- `getOxygenSaturation()` // Devuelve el nivel saturación de oxígeno en sangre.

Función del Electrocardiograma:

- `getECG()` // Retorna los valores analógicos del electrocardiograma.

Funciones del sensor de flujo de aire:

- `getAirFlow()` // Devuelve el valor del flujo de aire.
- `airFlowWave()` // Dibuja una curva del flujo de aire.

Función del Termómetro:

- `getTemperature()` // Devuelve el valor de temperatura corporal.

Funciones del Tensiómetro:

- `initBloodPressureSensor()` // inicialización del sensor.
- `getSystolicPressure()` // valor de presión arterial sistólica.
- `getDiastolicPressure()` // valor de presión arterial diastólica.

Funciones del Acelerómetro:

- `initPositionSensor()` // inicializa la posición del sensor.
- `getBodyPosition()` // retorna la posición del cuerpo.
- `printPosition()` // imprime la posición del cuerpo (mensaje de estado).

Funciones del sensor de GSR:

- `getSkinConductance()` // devuelve el valor de la conductividad de la piel.
- `getSkinResistance()` // devuelve el valor de la resistencia de la piel.
- `getSkinConductanceVoltage()` devuelve el valor de conductividad de la piel //expresado en voltios

Con todas estas funciones se diseña un programa capaz de enviar los datos de los sensores a través del puerto serie, en base a la parametrización diseñada desde la consola de administración.

5.2.3. Servidor

El servidor consta de tres partes diferenciadas, la primera es la comunicación bidireccional con el puerto serie, la segunda la comunicación bidireccional con la base de datos y por último la comunicación bidireccional que se establece entre el servidor y la consola de administración y clientes de consulta.

Comunicación con el puerto serie

Para la correcta comunicación del servidor con el puerto serie se hace necesario instalar la librería *javax.comm* para Windows, como se expuso con anterioridad no hay soporte de *Java Communications API* directamente para Windows por tanto se ha de instalar el software RXTX,

de adaptación. Para la instalación de este software se debe seguir el manual expuesto en el Anexo II de este documento.

Una vez llevados a cabo los pasos anteriores ya se está en disposición de realizar un programa para la comunicación con los puertos serie de comunicaciones.

Para la comunicación bidireccional con el puerto serie, se dispone de una clase específica basada en la librería de comunicaciones RXTX.

En esta clase se implementan varios métodos que permiten establecer la comunicación y enviar parámetros de configuración como pueden ser la frecuencia de muestreo o el sensor o sensores de los cuales se han de obtener datos.

El nombre de esta clase es “ComunicacionPuertoSerie” y sus métodos más importantes, los que permiten la lectura y escritura en un puerto son: SerialReader() y SerialWriter().

Esta clase permite la parametrización por ejemplo del nombre del puerto serie al que se conecta o el ancho de banda de la comunicación, ya sea en Windows, Linux o Unix.

Comunicación con la base de datos

Para la comunicación con la base de datos, se utiliza una conexión basada en el driver JDBC. Esta conexión permite ser configurada mediante la consola de administración para establecer un enlace entre el servidor y cualquier base de datos.

Para ello, se dispone de una clase denominada “ConexionBBDD” que posee métodos para conectar, leer y escribir en la base de datos.

Tanto el nombre de la conexión como el usuario y la contraseña de la base de datos se almacenan en un archivo de configuración con el objeto que sean recordados para la próxima conexión en el caso que se disponga del mismo PC.

Comunicación con los clientes de consulta y administración

Para llevar a cabo la comunicación entre la consola y el servidor, se ha optado por la comunicación por sockets de comunicación.

El concepto de socket está basado en la comunicación de dos o más programas, pudiendo estar ubicados en computadores diferentes, que intercambian datos entre sí.

Los sockets han de ser capaces de utilizar el protocolo de flujos TCP (Transfer Control Protocol) y el de datagramas UDP (User Datagram Protocol).

Utilizan una serie de primitivas para establecer el punto de comunicación, para conectarse a una máquina remota en un determinado puerto que esté disponible, para escuchar en él, para leer o escribir y publicar información en él, y finalmente para desconectarse.

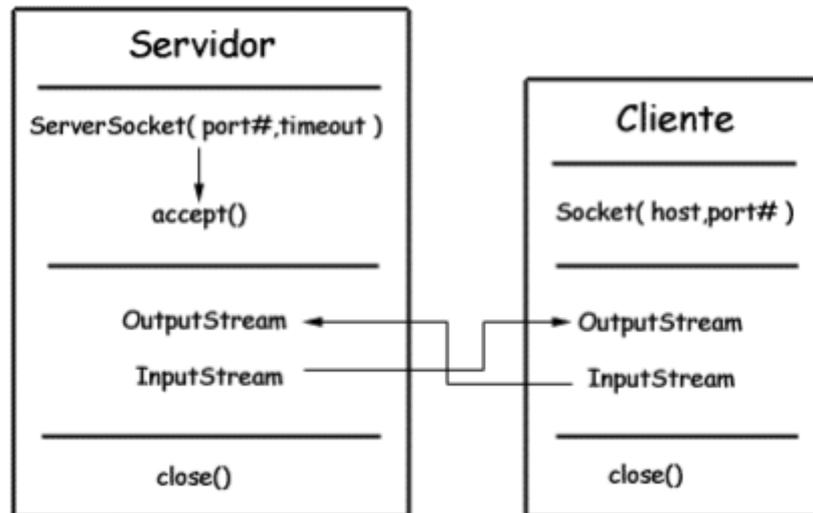


Ilustración 5.18. Esquema de comunicación con sockets en Java

En este caso, la parte servidora lo que hace es abrir un socket y esperar a que un cliente se conecte a él mediante su dirección IP o nombre y un puerto definido en el servidor, una vez se ha conectado un cliente, se crea un nuevo hilo que permite atender las peticiones del cliente y gestionarlas de manera adecuada.

Para ello se disponen de dos clases, la primera de ellas es la denominada “Server” que lo único que hace es esperar a que un cliente se conecte. La segunda de ellas es la denominada “HiloDeCliente” que permite gestionar la conexión para el cliente k que se haya conectado al servidor.

5.2.4. Consola de administración y pruebas

Para llevar a cabo las diferentes pruebas, se dispone de una consola de administración y pruebas que permite, entre otras cosas elegir los parámetros de los diferentes sensores disponibles y empezar a consumir los datos.

Los datos, a la vez que se muestran en la consola, son enviados a la base de datos para su posible análisis.

El cliente se basa en la apertura de un socket de comunicaciones con el servidor, para ello dispone de métodos capaces de conectarse mediante un socket al servidor, introduciendo su ip y puerto.

A continuación se muestra una imagen de la consola de administración:

```
-----> Menu Principal EHEALTH <-----  
  
A : Sensor de flujo de aire  
O : Sensor de saturacion de oxigeno en sangre  
S : Sensor de respuesta galvanica de la piel  
E : Electrocardiograma  
P : Pulsometro  
B : Sensor de posicion  
Q : Sensor de posicion valores ejes  
T : Termometro  
N : Tensiometro  
G : Glucometro  
v
```

Ilustración 5.19. Consola de administración

5.2.5. Base de datos

Se ha de disponer de una base de datos para guardar los datos leídos de los sensores, para ello hay que optimizarla de manera que la lectura y escritura de datos sea lo más eficiente posible dentro de las capacidades del servidor de la BBDD

Se ha optado por un SGBD de código abierto como es MySQL Community Edition bajo licencia GPL[24].

El servidor de la base de datos puede estar alojado en el mismo equipo que hace de servidor de la aplicación o puede estar distribuido.

La conexión que se utiliza para comunicar el servidor de e-salud con la base de datos está implementada bajo JDBC.

A continuación se muestra un diagrama ER con las tablas que hay en la base de datos para este experimento.

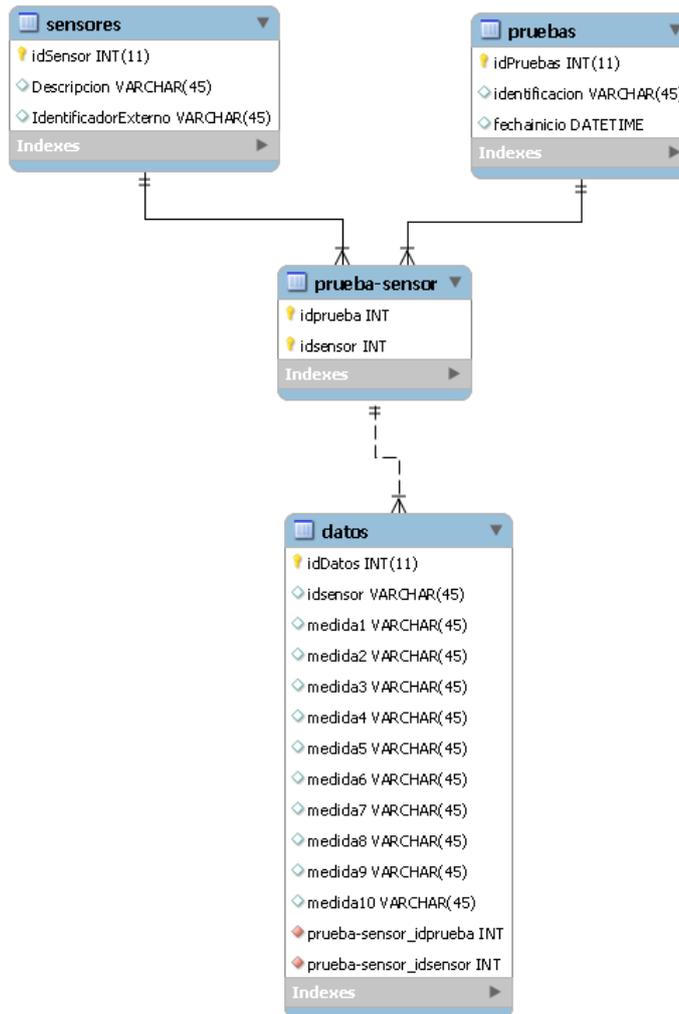


Ilustración 5.20. Diagrama ER de la base de datos

5.3. Especificación de uso

En primer lugar hay que hacer constar, que sin el equipo hardware descrito, por lo menos, el Arduino más el escudo e-Health las funcionalidades del sistema carecen de sentido más allá de la consulta de los datos almacenados en la base de datos.

A continuación se describen los pasos básicos para poner en funcionamiento el sistema.

Para el correcto uso de todo el prototipo se seguirán unos pasos determinados y en el orden siguiente:

1. Conectar la plataforma e-Health al Arduino tal y como se indica en la siguiente imagen.

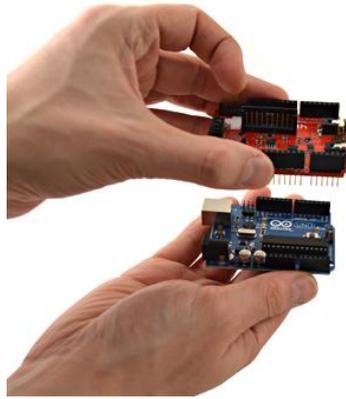


Ilustración 5.21. Montaje escudo e-Health sobre Arduino

2. Conectar todos los sensores a la plataforma tal y como se indica en la siguiente imagen

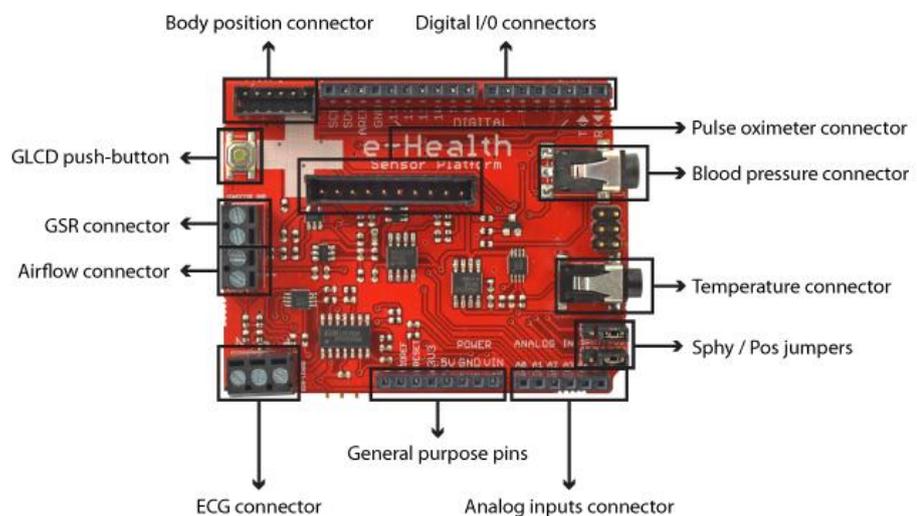


Ilustración 5.21. Conexión de los sensores a la plataforma E-health

3. Conectar el Arduino mediante un cable UBS a un puerto disponible del equipo que haga de host. Una vez el Arduino esté alimentado, ya se ejecutará el programa cargado. Para conocer como cargar este programa consultar el Anexo III.
4. Arrancar el servidor de la BBDD.
5. Arrancar el servidor del host. Para ello hacer doble clic sobre el archivo Server.jar.
6. Arrancar la consola de administración. Para ello hacer doble clic sobre Consola.jar.
7. Por último, basta con seguir el menú de la consola para obtener los datos de los sensores que se deseen.
8. A la vez que se visualizan los datos, se inserta una copia de los mismos en la base de datos.

9. PRUEBAS Y RESULTADOS

En este apartado se detallan las diferentes pruebas llevadas a cabo sobre los diferentes sensores disponibles. Se ha llevado a cabo pruebas para verificar tanto la correcta comunicación con los dispositivos como la recepción de datos válidos.

Las pruebas se llevan a cabo sobre todos los sensores disponibles, por lo que al final, se expondrán las valoraciones conjuntas acerca del funcionamiento del sistema.

9.1. Pruebas con el sensor de flujo de aire

Se han llevado a cabo pruebas con diferentes frecuencias de muestreo. A continuación se detallan los resultados para cada frecuencia.

En el eje de abscisas se representa el tiempo en segundos, en el eje de ordenadas se representa el flujo de aire.

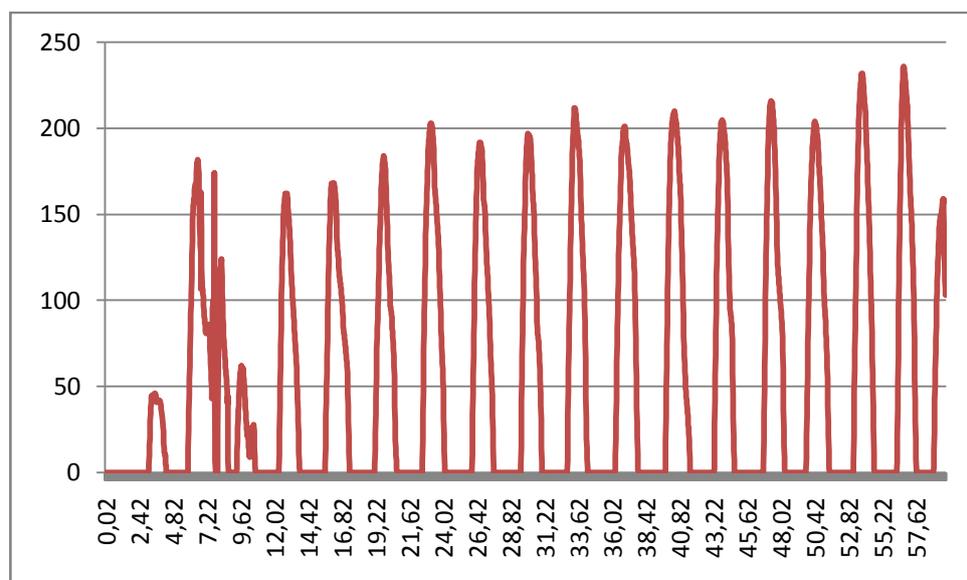


Ilustración 9.1. Tabla de resultados de sensor de flujo de aire

Estos datos, para saber si poseen alguna deficiencia médica deberían ser analizados por un especialista. El sensor aparentemente funciona como se espera. En este caso la frecuencia de muestreo no se puede variar ya que el hardware no lo permite.

A la vista de los resultados, los picos en la gráfica representan cuando una persona esta respirando y los valles cuando se está inspirando.

9.2. Pruebas con el sensor de temperatura corporal

En este caso se van a tomar datos del sensor de temperatura corporal. Se van a tomar datos a diferentes frecuencias de muestreo. En el eje de abscisas se muestra el tiempo en segundos y en el de ordenadas la temperatura en grados centígrados.

9.2.1. Para Frecuencia 1Hz

Muestreando los datos del sensor cada segundo se obtuvieron los siguientes datos:

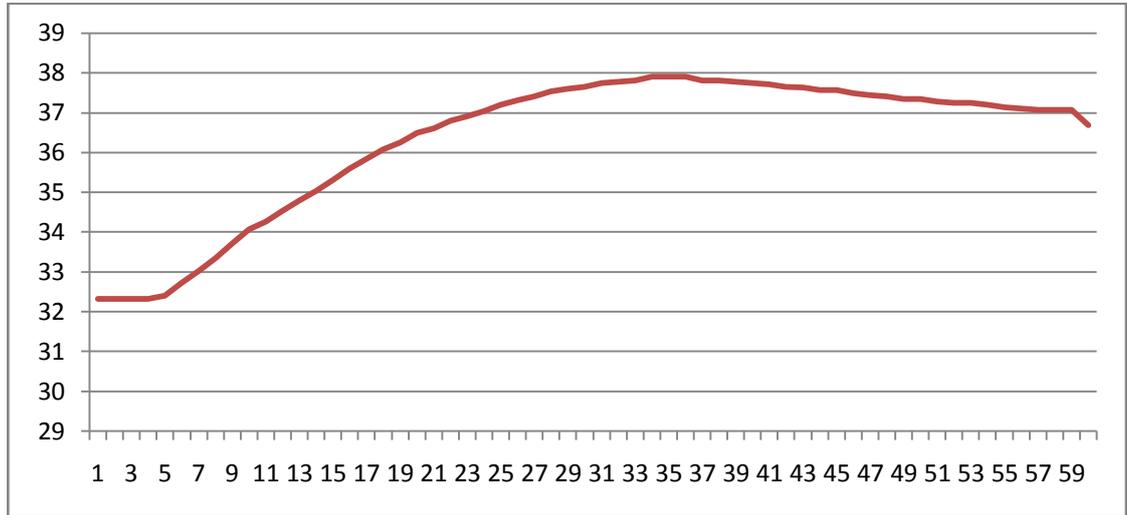


Ilustración 9.2. Tabla de resultados del termómetro con frecuencia 1 Hz

Como se puede observar es una curva típica de temperatura, ya que no se producen cambios bruscos en la misma.

9.2.2. Para frecuencia 2Hz

Muestreando los datos del sensor cada segundo se obtuvieron los siguientes datos:

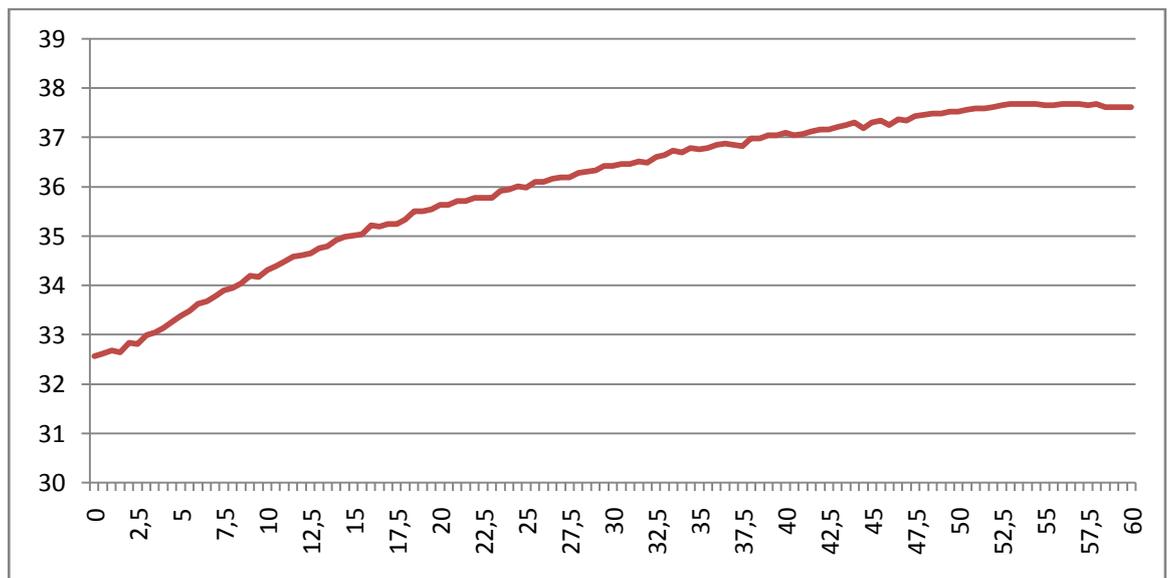


Ilustración 9.3. Tabla de resultados del termómetro con frecuencia 1 Hz

9.3.Sensor de respuesta galvánica de la piel

En este caso, se han hecho pruebas con este sensor y devuelve tres datos, el valor de resistencia de la piel con un valor de seis cifras, el valor de conductividad de la piel que está en un intervalo de 1 a 4 y por último el valor de conductividad expresado en voltios.

No se han representado los datos porque no se tiene una base de representación para compararlos, es decir, no se conoce si los datos son buenos o malos, dado que son datos biomédicos que han de ser comparados por un experto.

9.4.Electrocardiograma

En este apartado se hacen dos pruebas para obtener datos del ECG. En el eje de abscisas se muestra el tiempo en segundos y en el de ordenadas el valor del ECG en voltios

9.4.1. Prueba a frecuencia 2Hz

Los datos obtenidos a una frecuencia de 2 Hz (configuración por defecto) son los siguientes:

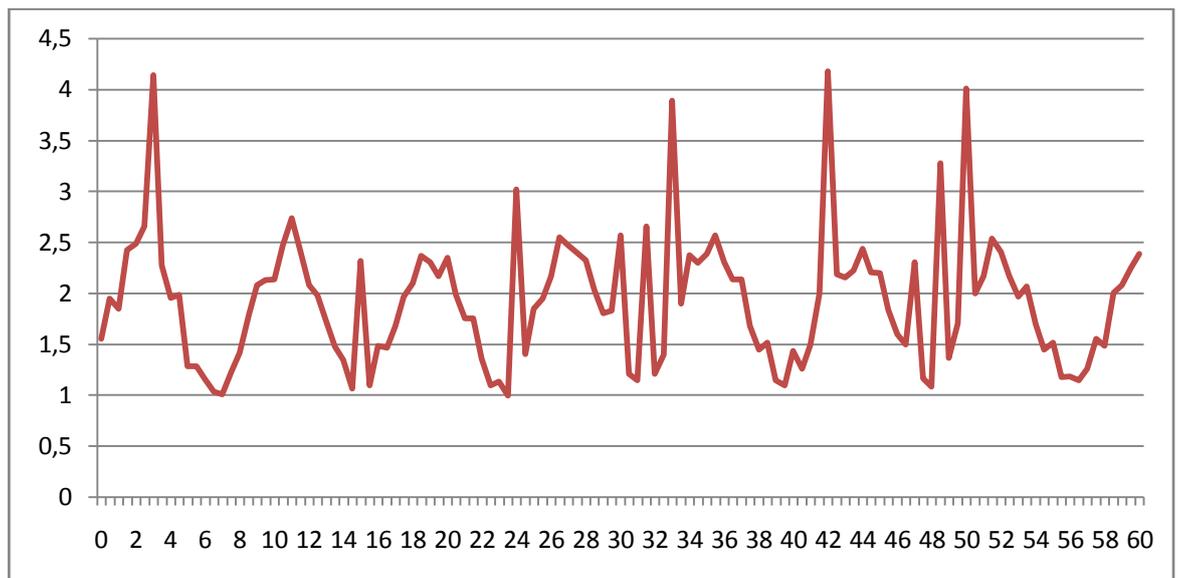


Ilustración 9.4. Tabla de resultados de ECG con frecuencia 2 Hz

9.4.2. Prueba a frecuencia de 1Hz

La siguiente gráfica muestra los datos de la prueba realizada con una frecuencia de muestreo de 1 Hz:

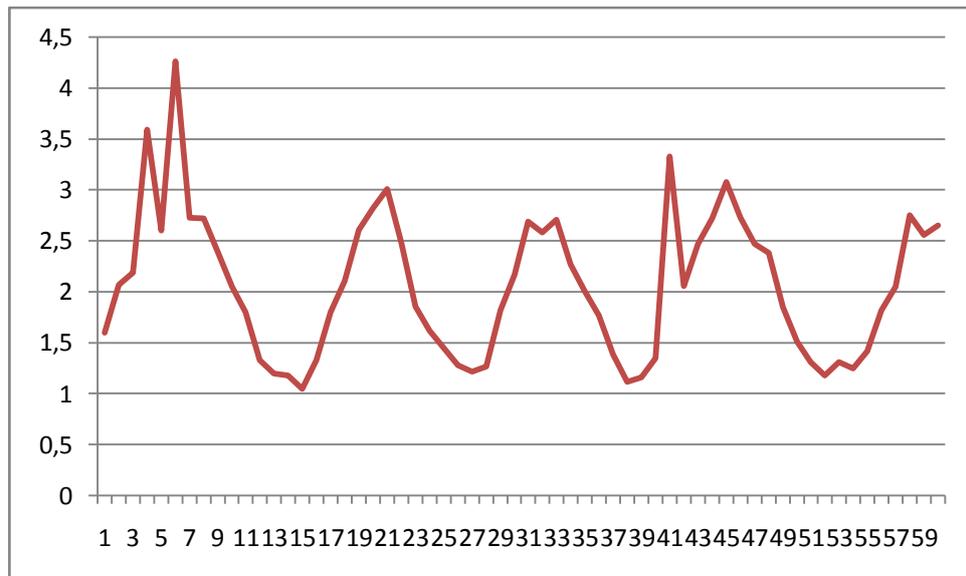


Ilustración 9.5. Tabla de resultados de ECG con frecuencia 1 Hz

9.5.Sensor de posición

Se han hecho pruebas para este sensor, y funciona razonablemente bien, es decir el sensor detecta las posiciones correctamente. La frecuencia de muestreo se ha mantenido ya que el tiempo de respuesta del sensor es similar. A continuación se muestra un ejemplo:

```

Posicion Actual : Supine position

Valores ejes:
X: 0.53
Y: 0.22
Z: 0.93

Posicion Actual : Supine position

Valores ejes:
X: -0.37
Y: 0.30
Z: 0.72

Posicion Actual : Stand or sit position

Valores ejes:
X: -0.83
Y: 0.51
Z: 0.63

Posicion Actual : Rigth lateral decubitus

Valores ejes:
X: -1.28
Y: 0.51
Z: 0.58

Posicion Actual : Rigth lateral decubitus

Valores ejes:
X: -0.60
Y: 0.72
Z: 0.25

```

Ilustración 9.6. Resultados de acelerómetro

9.6.Sensor de saturación de oxígeno en sangre

Este sensor mide el nivel de saturación de oxígeno en sangre y el recuento de pulsaciones por minuto.

A continuación se muestra una gráfica con las pulsaciones por minuto, con muestras con una frecuencia de 1 Hz.

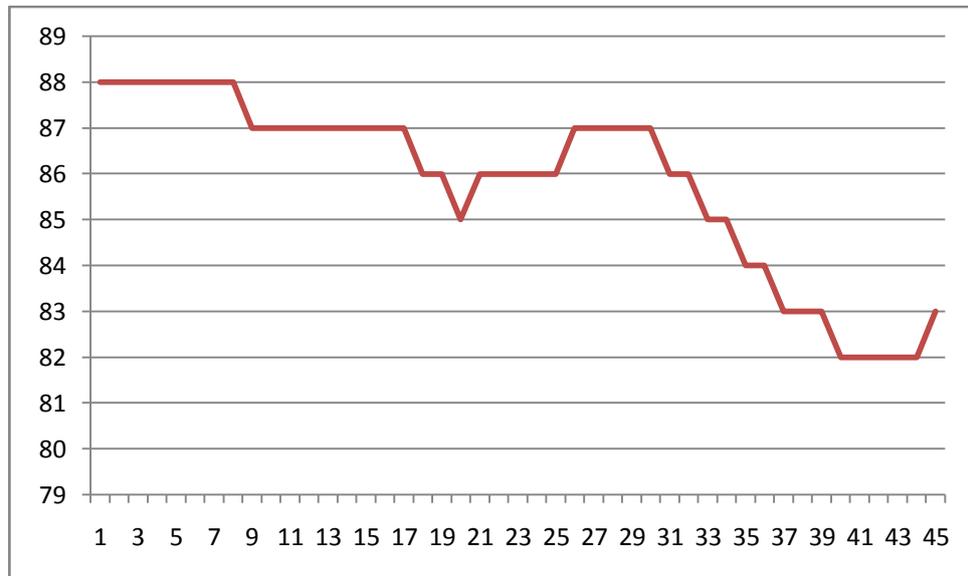


Ilustración 9.7. Tabla con los resultados de pulsaciones por minuto

En este caso, el incremento o decremento de la frecuencia solo va a permitir disponer de más o menos muestras, ya que por ejemplo el valor de oxígeno en sangre suele tender a ser constante.

La siguiente gráfica representa en el eje de abscisas el tiempo expresado en segundos y en el eje de ordenadas la saturación de oxígeno en sangre expresada en % SpO₂.

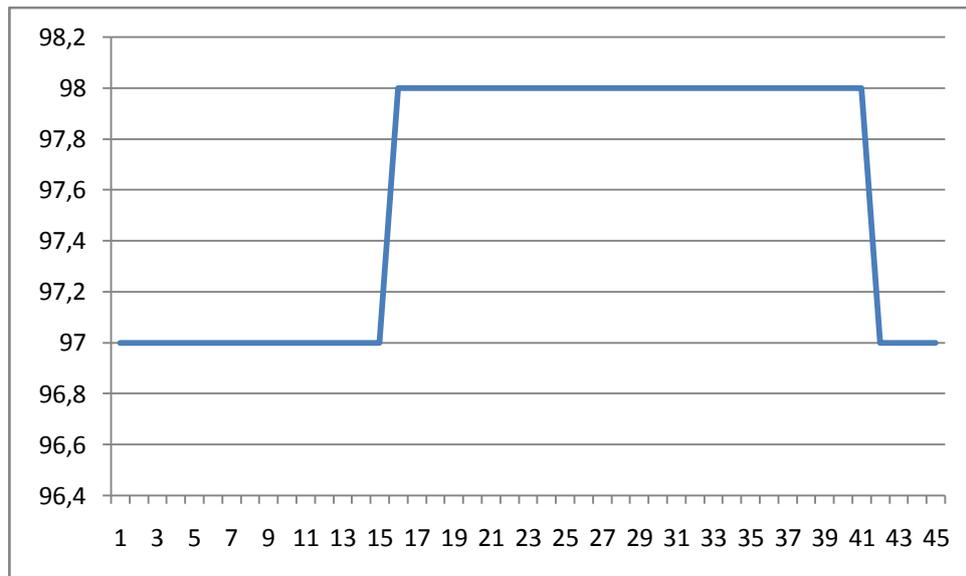


Ilustración 9.8. Tabla con los resultados de SpO2

9.7. Tensiómetro

En este caso, se toma una medición cada vez que se pulsa el botón *on* del tensiómetro. Evidentemente, esta medida se hace como mucho dos veces al día en casos especiales, ya que se supone que la variación de tensión no debe ser muy acusada en un intervalo corto de tiempo.

El tensiómetro del que se dispone, mide correctamente los datos de presión arterial, sin embargo, al enviar estos datos a la plataforma e- Health los datos no se corresponden con los observados en el *display* del Tensiómetro.

Consultando el manual, los parámetros están aparentemente correctos por lo que se supone que esta parte no funciona correctamente en la plataforma o el tensiómetro del que se dispone no envía los datos de forma correcta.

9.8. Discusión de resultados

Los resultados obtenidos a través de los diferentes experimentos parecen ajustarse a la realidad en casi todas las situaciones.

Por ejemplo, los valores obtenidos de los sensores de saturación de oxígeno en sangre, flujo de aire y respuesta galvánica de la piel parecen correctos si se comparan con los que se obtienen en el manual o en el propio *display* del sensor de saturación de oxígeno.

Para algunas pruebas, la frecuencia de muestreo no parece que intervenga en los resultados, por ejemplo es el caso del sensor de temperatura, del cual a diferentes frecuencias la diferencia es la obtención de más o menos datos, pero el comportamiento es el mismo.

Sin embargo, en otros sensores, como puede ser el ECG, aparentemente se obtienen mejores resultados con una frecuencia de 2Hz, esto puede estar motivado porque a mayor o menor

frecuencia de muestreo se tenga una precisión que no se corresponda con la optima, bien sea por exceso o por defecto.

Tras la obtención de los resultados, se ha notado que algunos sensores no están suficientemente calibrados o no están tomando datos correctamente para obtener los resultados correctos. Un ejemplo es el sensor de temperatura corporal, que en vez de estabilizarse con el tiempo en una temperatura, hace fluctuaciones tanto de exceso de temperatura como de defecto. Dicho esto, el sensor está calibrado tal y como se muestra en el manual, por lo que las razones de estos cambios de temperatura se desconocen.

10. CONCLUSIONES

Tras la realización de este trabajo se han comprendido la importancia que tiene la informática en la vida de las personas, por ello, tras estudiar qué es la Inteligencia Ambiental y a qué se aplica, se ha comprendido la necesidad de desarrollar sistemas que se adapten al usuario y no sea el usuario el que se tenga que adaptar a el funcionamiento de los sistemas.

Por otra parte también se ha tomado consideración de la importancia que tienen los sistemas empotrados y ubicuos en el desarrollo de la vida cotidiana, ya que, hoy en día se pueden encontrar diversos objetos, sistemas, herramientas de la vida cotidiana que están gobernados por este tipo de sistemas.

Esto abre un abanico de posibilidades enormes para diseñar prototipos y productos que sean capaces de mejorar el bienestar de las personas. Por ello, tras implementar un prototipo de medición de constantes biomédicas basado en una plataforma libre, se ha conocido de primera mano qué características poseen estos tipos de sistemas y qué posibilidades ofrecen.

También cabe destacar, que al implementar un prototipo sobre una plataforma libre como es Arduino, también surgen diversos problemas y dificultades tanto en el diseño como en la adaptación de la solución.

Respecto a los resultados obtenidos del estudio, se han considerado, de manera subjetiva, que han sido satisfactorios, sin incluir algunas excepciones que no se han conseguido solucionar como la obtención de datos correctos de algunos sensores o la comunicación de la plataforma a través de medios inalámbricos.

El sistema diseñado permite integrar varios sensores en una sola plataforma, los beneficios de esto es que se puede monitorizar a un paciente desde su domicilio con un sistema de bajo coste, si se compara con los dispositivos comerciales. Este prototipo, por tanto, si bien se destinará a fines docentes, se podría seguir desarrollándolo de manera que sería útil, con unas pequeñas ampliaciones, en un sistema hospitalario.

11. TRABAJOS FUTUROS

Tras recoger datos de los diferentes sensores conectados a la plataforma y almacenar estos en una base de datos, un trabajo inminentemente posterior sería el análisis de los datos.

Para el análisis de los datos obtenidos se podría implementar diversas técnicas, entre ellas el reconocimiento de patrones mediante series temporales como puede ser SAX u otras técnicas de *soft computing*. Incluso se podrían aplicar técnicas de fusión de información [25][26] para obtener un sistema de decisión capaz de emitir valoraciones o diagnósticos en base al estudio de los datos de los sensores.

También, se podría desarrollar la plataforma o evolucionarla de manera que fuese totalmente inalámbrica. En el transcurso de este trabajo se ha intentado esta opción, pero resultaron algunas incompatibilidades con el hardware del que se disponía de tal manera que no fue posible su implementación.

La idea sería tener poseer un dispositivo portátil que se conectara al equipo servidor a través de un canal de comunicación sin cable. También, para la alimentación del sistema se podría insertar alimentación autónoma.

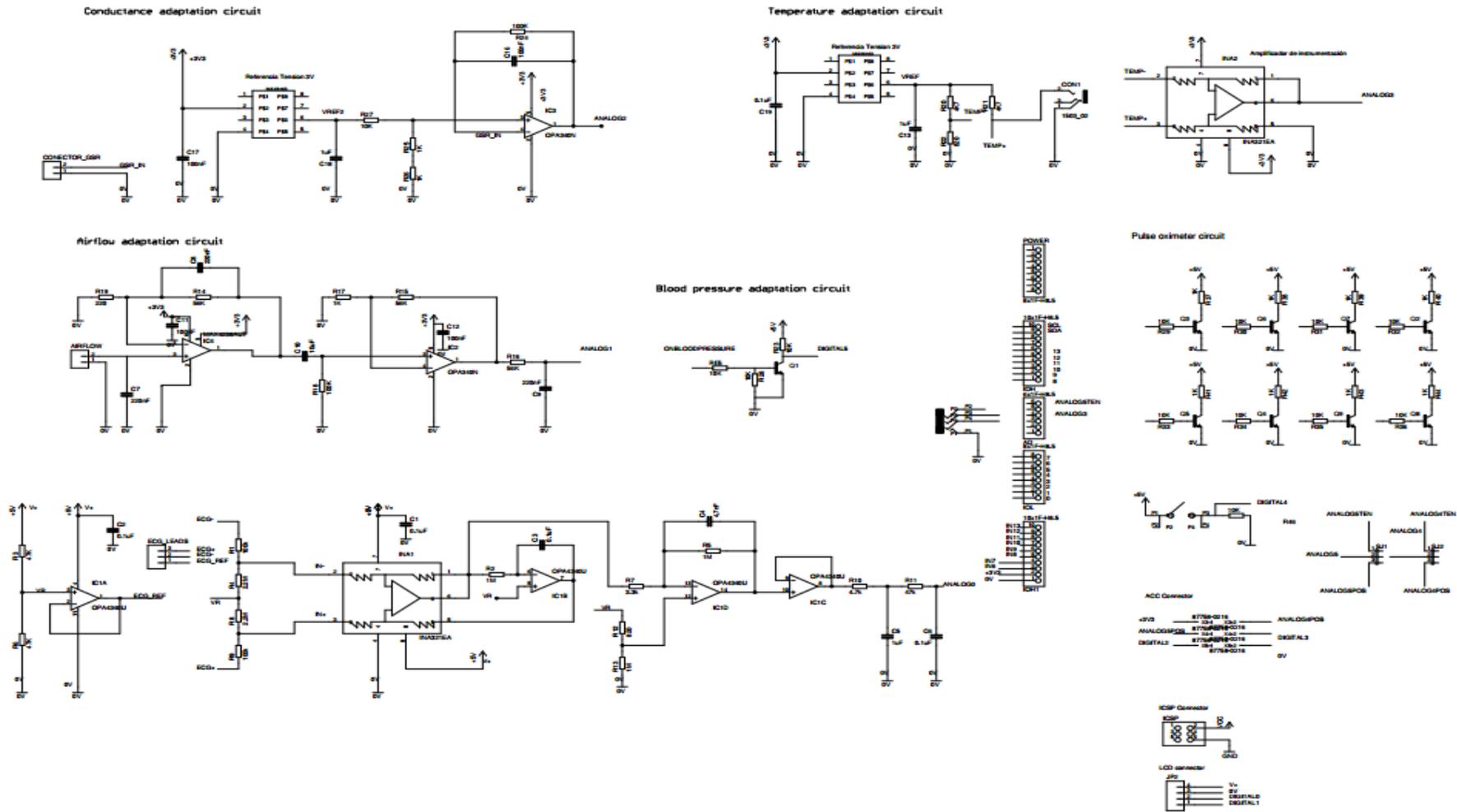
Otro de los posibles trabajos que se pueden proponer a partir del estudio que se ha hecho en este, es sustituir el equipo que hace de host, en este caso un PC, por un equipo como puede ser un Raspberry PI, al que se le puede instalar un sistema operativo Linux y puede hacer las veces de servidor.

12. BIBLIOGRAFÍA

- [1] Digitalización 3D aplicada a la medicina:
<http://www.cinpi.es/blog/2012/03/la-digitalizacion-3d-aplicada-a-la-medicina/>
- [2] Web de Ambient Assisted Living: <http://www.aal-europe.eu>
- [3] Definición AmI: http://en.wikipedia.org/wiki/Ambient_intelligence
- [4] RIVA, G. 2003. Ambient intelligence in health care. *CyberPsych. Behav.* 6, 3, Mary Ann Liebert, Inc. Publishers, 295–300.
- [5] Electrocardiograma, qué es y para que se utiliza:
<http://www.nlm.nih.gov/medlineplus/spanish/ency/article/003868.htm>
- [6] GOUAUX, F., SIMON-CHAUTEMPS, L., FAYN, J., ADAMI, S., ARZI, M., ASSANELLI, D., FORLINI, M.C., MALOSSI, C., MARTINEZ, A., PLACIDE, J., ZILIANI, G.L., AND RUBEL, P. 2002. Ambient intelligence and pervasive systems for monitoring of citizens at cardiac risk: New solutions from the EPI-MEDICS project. *Comput. Cardiol.* 29, 289–292.
- [7] HAUX, R. 2006. Individualization, globalisation and health—about sustainable information technologies and the aim of medical informatics. *Int. J. Medical Informatics* 75, Elsevier, 795–808.
- [8] Web sobre e-salud:
<http://www.gencat.cat/salut/ticsalut/flashticsalut/html/es/articulos/doc36506.html>
- [9] MUNOZ, M. A., RODROGUEZ, M., FAVELA, J., MARTINEZ-GARCIA, A. I., AND GONZALEZ, V. M. 2003. Context aware mobile communication in hospitals. *IEEE Comput.* 36, 38–46.
- [10] KOFOD-PETERSON, A., AND AAMODT, A. 2006. Contextualised ambient intelligence through case-based reasoning. In *Proceedings of the 8th European Conference on Case-Based Reasoning (ECCBR)*. T. R. Roth-Berghofer et al. Eds., Lecture Notes in Artificial Intelligence, vol. 4106, Springer-Verlag, Berlin, 211–225.
- [11] CORCHADO, J. M., BAJO, J., AND ABRAHAM, A. 2008. GerAmi: improving healthcare delivery in geriatric residences. *J. IEEE Intell. Syst. (Special Issue on Ambient Intelligence)*, 3, 2, 19–25.
- [12] Sistemas multiagente: http://jmas.us.es/doc/1.Delgado_JMASweb.pdf
- [13] Soluciones comerciales para e-health: <http://ehs.cl/soluciones/telemedicina/>
- [14] Soluciones comerciales par e-health:
http://www.ericsson.com/hr/ict_solutions/e-health/emh/
- [15] Web Arduino: <http://www.arduino.cc>
- [16] Web Atmel: <http://www.atmel.com/products/microcontrollers/avr/default.aspx>
- [17] Web e-Health Platform Sensor:
<http://www.cooking-hacks.com/index.php/documentation/tutorials/ehealth-bio-metric-sensor-platform-arduino-raspberry-pi-medical>
- [18] Definición y características de la EPOC:
http://www.texasheartinstitute.org/HIC/Topics_Esp/Cond/copd_span.cfm
- [19] Definición de hipoxia:
<http://escuela.med.puc.cl/publ/Aparatorespiratorio/11Hipoxia.html>
- [20] Definición de apnea:
- [21] <http://www.nlm.nih.gov/medlineplus/spanish/ency/article/003069.htm>
- [22] Definición y ejemplos de GSR: <http://bsamig.uef.fi/research/gsr.shtml>

- [23] Librería Java de comunicación con el puerto serie: <http://rxtx.qbang.org/>
- [24] Web de MySQL: <http://www.mysql.com/>
- [25] XIONG N., SVENSSON P.; Multi-sensor management for information fusion: issues and approaches; Information Fusion 3 (2002) 163–186.
- [26] DIA, H., THOMAS, K., Development and evaluation of arterial incident detection models using fusion of simulated probe vehicle and loop detector data; Information Fusion 12 (2011) 20–27. doi:10.1016/j.inffus.2010.01.001

13. ANEXO I. ESQUEMA DE LA PLATAFORMA E-HEALTH



14. ANEXO II: MANUAL DE INSTALACIÓN DE RXTX

Para la lectura de datos desde el puerto serie, se ha empleado una librería de Java denominada Javax.comm, para instalarla en Windows hay que seguir una serie determinada de pasos que se describen a continuación:

- Descarga [\[1\]](#)
- Copiar rxtxSerial.dll en la carpeta C: \ Archivos de programa \ Java \ jre-version \ bindir
- Copiar RXTXcomm.jar en la carpeta C: \ Archivos de P... \ Java \ jre-version \ lib \ extdir
- Cambiar todas las referencias de "javax.comm 'a' gnu.io "
- Recompilar

DESCARGA:

1. En primer lugar hay que descargar el paquete [rxtx-2.1-7r2.zip](#) desde <http://rxtx.qbang.org/>
2. Descomprimir los archivos en una carpeta o directorio de trabajo.

INSTALACIÓN:

1. Hay que instalar mingw32 <http://www.mingw.org>
2. instalar un JDK <http://java.sun.com/j2se> asegurandose de que el ming32\bin y jdk \ bin se encuentran en el path.
3. En el directorio principal de rxtx hay que hacer lo siguiente:

```
mkdir build
copy Makefile.mingw32 build\Makefile
cd build
```

4. Editar el archivo Makefile, comprobando que todos los directorio y/o rutas son correctos
5. Ejecutar el siguiente comando: `make make install`

Con todo esto, hacer un programa que lea desde el puerto serie, se pueden seguir los ejemplos que se muestran en <http://rxtx.qbang.org/wiki/index.php/Examples>. Una vez instalado todo lo anterior ya se tendrá *gnu.io* lista para su uso.

15. ANEXO III. CARGA DE DATOS EN ARDUINO

A continuación se detallan los pasos para cargar un programa para Arduino desde el Arduino IDE.

1. Abrir el Arduino IDE.

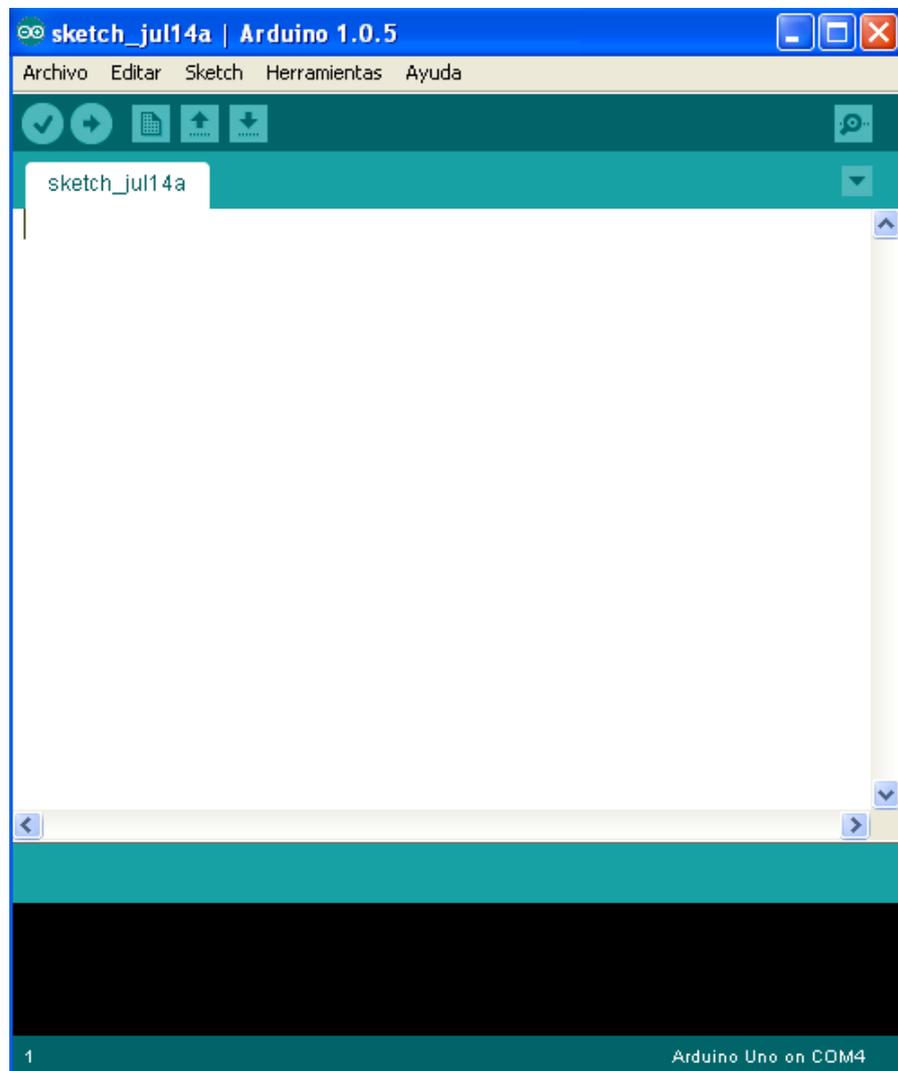


Ilustración 15.1. Arduino IDE

2. Ir a Herramientas → Puerto serial y seleccionar el que se corresponda con el Arduino.
3. Ir a Archivo → Abrir y escoger el fichero que contiene el programa que se desea cargar
4. Pulsar sobre el siguiente botón. Una vez que se muestre el mensaje de “Carga Terminada” ya está cargado el programa en el microcontrolador de Arduino.



Ilustración 15.2. Carga de programa en Arduino

5. Una vez cargado ya se puede abrir el monitor serial o cualquier programa que lea del puerto serie.

PARTE II. PLANIFICACIÓN Y PRESUPUESTO

6. PLANIFICACIÓN Y PRESUPUESTO

Para llevar a cabo la planificación del trabajo y hacer una aproximación del tiempo para llevarlo a cabo, se dividirá este en un número determinado de tareas, que se exponen a continuación:

- **Análisis del estado del arte e investigación:** Se busca y lee parte de la literatura correspondiente a los diferentes trabajos implementados de entornos de AmI para plataformas médicas, así como los diferentes sistemas empotrados disponibles en el mercado para el desarrollo de una plataforma de e-salud, seleccionando aquella literatura que más se adapta a las necesidades de este trabajo.
- **Estudios iniciales previos:** Se revisa toda la información Arduino, módulo e-health, combinación de Arduino + e-Health + módulo WIFI (no implementado) instalación y configuración, así como manuales del lenguaje Java y librerías para la comunicación con el puerto USB.
- **Estudio de características:** se hará un estudio de las diferentes soluciones al problema, estableciendo cual será implementada en base a un conjunto de ventajas e inconvenientes.
- **Implementación:** Propuesta de implementación de un prototipo para simular el monitoreo de datos biomédicos.
- **Pruebas:** Se realizarán pruebas, para observar los resultados y comprobar que la solución propuesta tiene validez o no.
- **Documentación:** Todas las tareas anteriormente enumeradas han de ser documentadas, la realización de dicha documentación es la labor de esta tarea.

6.1. Planificación y seguimiento del trabajo

En este apartado se realiza la planificación de las tareas vistas anteriormente. El trabajo fin de máster se va a planificar en semanas de trabajo. Cada semana de trabajo implicará, aproximadamente, cuarenta horas de trabajo en el proyecto, estas horas son estimadas, puede que una semana se dedique más tiempo y otra menos.

En la siguiente tabla se muestra de manera aproximada las tareas anteriormente mencionadas y la duración prevista de las mismas, partiendo de que la persona que realiza todas estas tareas, no es una persona experimentada en la materia, por tanto el tiempo se eleva considerablemente.

Orden	Nombre de tarea	Días Laborables	Comienzo	Fin	Predecesoras	Nombres de los recursos	Horas
1	Trabajo Fin de Máster	44 días	mié 15/05/13	lun 15/07/13			352
2	Análisis	29 días	mié 15/05/13	lun 24/06/13			116
3	Análisis e investigación	28 días	mié 15/05/13	vie 21/06/13		Investigador	80
4	Estudios previos	21 días	lun 27/05/13	lun 24/06/13		Investigador	36
5	Diseño	14 días	mar 25/06/13	vie 12/07/13			83
6	Estudio de características	4 días	mar 25/06/13	vie 28/06/13	4;3	Analista de sistemas	13
7	Implementación	10 días	lun 01/07/13	vie 12/07/13	6	Programador	70
8	Pruebas	2 días	jue 11/07/13	dom 14/07/13			18
9	Ejecución	3 días	jue 11/07/13	dom 14/07/13		Tester	9
10	Documentación pruebas	3 días	jue 11/07/13	dom 14/07/13		Documentador;Tester	9
11	Documentación TFM	41 días	vie 17/05/13	dom 14/07/13			132
12	Memoria	42 días	vie 17/05/13	dom 14/07/13		Documentador	117
13	Planificación y presupuesto	5 días	mar 09/07/13	dom 14/07/13		Manager	15
14	Entrega	1 día	lun 15/07/13	lun 15/07/13			3
15	Entrega de documentación	1 día	lun 15/07/13	lun 15/07/13			3

Ilustración 12.1. Tabla de Planificación

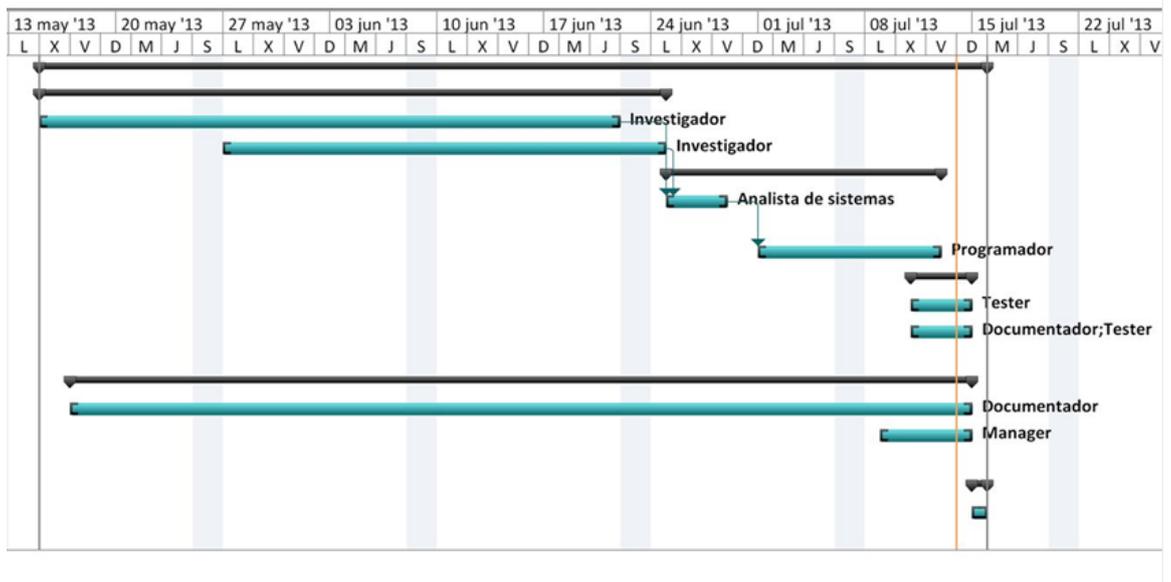


Ilustración 12.2. Diagrama Gantt con la planificación del trabajo.

6.2.Presupuesto

En primer lugar, hay que comentar, que el objetivo del trabajo no es generar un producto comercial para ser vendido sino más bien un prototipo mediante el cual se puedan realizar pruebas sobre sensores biométricos con el fin de estudiar diferentes comportamientos de la salud de los pacientes.

Por ello, las cifras de presupuesto aquí descritas son orientativas, calculadas por aproximación al trabajo y recursos empleados. Los precios de los recursos hardware sí que son realistas, obtenidos de la web del proveedor.

A continuación se detalla el coste de cada recurso por separado, por último se hace un resumen con el coste total.

6.2.1. Recursos humanos

A continuación se muestra una tabla relacionando el coste por hora de los diferentes perfiles de trabajador, estos costes son de valor aproximado, atendiendo a criterios que pueden no ajustarse a la realidad, ya que cada persona o puesto tendría una remuneración mayor o menor en función de otras consideraciones que en este trabajo no es posible estimar.

ID	Descripción	Unidades (h)	Precio Unidad	Precio final
1.1	Investigador	116	90	10.440
1.2	Analista de sistemas	13	50	650
1.3	Programador	70	20	1.400

1.4	Tester	18	50	900
1.5	Documentador	126	10	1.260
1.6	Manager	3	80	240

6.2.2. Recursos Técnicos

A continuación se detallarán los costos de los recursos técnicos para poder abordar este proyecto:

ID	Descripción	Unidades	Precio Unidad	Precio final
2.1	Bibliografía	18	0,00	0,00
2.2	Manuales y tutoriales de Arduino	10	0,00	0,00
2.3	Manuales y tutoriales e-Health	1	0,00	0,00
2.4	Manuales de BBDD	1	0,00	0,00
2.6	Manuales Java y RXTX	1	0,00	0,00

6.2.3. Recursos hardware

En este apartado se van a incluir solamente los materiales específicos para la ejecución de este trabajo, ya que el ordenador del proyectante, puesto que se amortiza cada 5 años y su precio fue de 600€, la parte proporcional a la ejecución de este proyecto le corresponderá 50 €, a causa de que no ha sido adquirido específicamente para el estudio de este proyecto. Los materiales necesarios se citan a continuación junto a su coste.

ID	Descripción	Unidades	Precio Unidad	Precio final
3.1	Ordenador Portátil	1	50,00	50,00
3.2	Arduino UNO	1	24,20	24,20
3.3	Escudo e-Health	1	90,75	90,75
3.4	Cable USB 2.0 Tipo A Macho/Hembra	1	2,88	2,88
3.5	Acelerómetro	1	48,40	48,40
3.6	Tensiómetro	1	90,75	90,75
3.7	Sensor SpO2	1	66,55	66,55
3.8	Sensor de flujo de aire	1	30,25	30,25
3.9	Termómetro	1	24,20	24,20
3.10	Sensor ECG	1	42,35	42,35
3.11	Sensor GSR	1	36,30	36,30

6.2.4. Recursos software

A continuación se detallarán los costes producidos por la obtención de los recursos software, al igual que en el epígrafe anterior no se considerarán aquellos que se han adquirido con anterioridad o no específicamente para este proyecto (*).

ID	Descripción	Unidades	Precio Unidad	Precio final
4.1	NetBeans IDE 6.9.1 (GPL)	1	0,00	0,00
4.2	Mozilla Firefox 4.0 (GPL)	1	0,00	0,00
4.3	Microsoft Office Excel 2007*	1	0,00	0,00
4.4	Microsoft Office Word 2007*	1	0,00	0,00
4.6	Microsoft Office Project 2010*	1	0,00	0,00
4.5	Microsoft Office Visio 2010*	1	0,00	0,00
4.5	Librería e-Health	1	0,00	0,00
4.6	Librería RXTX	1	0,00	0,00

6.2.5. Presupuesto de ejecución material

Se detallan las sumas totales de los costos de los recursos anteriores:

Descripción	Importe (€)
1. Recursos Humanos	14.890,00
2. Recursos Técnicos	0,00
3. Recursos Hardware	506,63
4. Recursos Software	0,00
TOTAL	15.396,63

6.2.6. Presupuesto de ejecución por contrata

Descripción	Importe (€)
Ejecución Material (12.2.5)	15.396,63
Gastos Generales (13%)	2.001,56
SUBTOTAL	17397,19
I.V.A. (21%)	3.643,41
TOTAL	21.050,60