



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Máster

Aplicación para registros fisiológicos a través de Arduino

Alumno: Cristina Lechuga Chica

Tutores:

Prof. Dña. Lidia Ortega Alvarado

Prof. Dña. Silvia Moreno Domínguez

Dpto.: Departamento de informática

OCTUBRE, 2019



Universidad de Jaén

Escuela Politécnica Superior de Jaén

Departamento de Informática

Doña Lidia Ortega Alvarado y Doña Silvia Moreno Domínguez, tutoras del Trabajo Fin de Máster titulado: **Aplicación para registros fisiológicos a través de Arduino**, que presenta Cristina Lechuga Chica, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, octubre de 2019

El alumno:

Los tutores:

ORTEGA
ALVARADO LIDIA
- 24241033E

Firmado digitalmente por
ORTEGA ALVARADO LIDIA -
24241033E
Fecha: 2019.10.25 19:56:17
+02'00'

Cristina Lechuga Chica

Doña Lidia Ortega Alvarado y

Doña Silvia Moreno Domínguez

AGRADECIMIENTOS

En primer lugar, todo se lo debo agradecer mis padres y mi hermano, por su apoyo, su confianza y sus ánimos cuando me venía abajo con el proyecto, sin ellos, no hubiese sido posible.

Gracias especialmente a mi equipo developers, por ayudarme y aconsejarme en todo lo que les he pedido, por enseñarme tantas cosas y por aguantar que les contara diariamente mis avances en el proyecto. Andrés, Joseda, Raúl, Fran, Adrián, Javi. Estáis contratados. ¡MIL GRACIAS!

Aunque me ayudó en los comienzos, no me olvido de Adrián, por que sin su ayuda no hubiese sido posible la impresión de una carcasa tan perfecta.

A mi tutora Lidia, por brindarme todo su apoyo y ayuda siempre que lo he necesitado. A Silvia y Gustavo, gracias por vuestra colaboración y entusiasmo, espero que este sea el comienzo de un gran proyecto.

Y finalmente, gracias a Edu, por que ha sido el que más me ha animado, el que ha estado a mi lado durante todo el proyecto ayudándome en todo lo que ha podido y cargándome de energía para continuar cada día.

ÍNDICE

1. INTRODUCCIÓN	8
1.1. Motivación	8
1.3. Objetivos	9
1.2. Estado del arte	10
2. PLANIFICACIÓN	14
2.1 Metodología	14
2.2. Tareas	15
2.3. Estimación de tiempo	19
2.4. Estimación de costes	21
2.5. Tecnología utilizada	25
2.5.1. Apache Tomcat	25
2.5.2. Apache 2	26
2.5.3. MySql	26
2.5.4. Spring	26
2.5.5. Maven	27
2.5.6. Yeoman	27
2.5.7. Grunt	27
2.6. Software utilizado	28
2.6.1. Trello	28
2.6.2. TeamGantt	29
2.6.3. SourceTree	29
2.6.4. Putty	30
2.6.5. CyberDuck	31
2.6.6. Postman	31
2.6.7. HeidiSQL	32
2.6.8. MarvelAPP	33
2.6.9. IntelliJ IDEA	33
2.6.10. Arduino IDE	33
2.6.11. Android Studio	34
3. ANÁLISIS DEL SISTEMA	35
3.1. Historias de usuario	35
3.2. Requisitos funcionales y no funcionales	36
3.2.1. Requisitos funcionales	37
3.2.2. Requisitos no funcionales	38
3.3. Arquitectura del sistema	38
3.4. Hardware	40
3.4.1. MySignals	40
Escuela Politécnica Superior de Jaén	3

3.4.1. HealthyPi v3	44
4. DESARROLLO	48
4.1. Primera iteración	48
4.1.1. Configuración del servidor	48
4.1.2. Diseño y configuración de base de datos	49
4.1.3. Desarrollo del sistema de recogida de datos fisiológicos	52
4.2. Segunda iteración	61
4.2.1. Diseño de la aplicación Android	61
4.2.1. Desarrollo aplicación Android	66
4.3. Tercera iteración	70
4.3.1. Diseño de la aplicación web	70
4.3.2. Desarrollo de la aplicación web	74
4.3. Cuarta iteración	76
5. CONCLUSIONES Y TRABAJO FUTURO	79
4.3. Conclusiones	79
4.3. Trabajos futuros	81
BIBLIOGRAFÍA	83
ANEXO I. Manual de uso del sistema de recogida de datos	87
ANEXO II. Manual de uso de la aplicación móvil	93
ANEXO III. Manual de uso de la aplicación web	102
ANEXO IV. Documentación API REST	111

ÍNDICE DE ILUSTRACIONES

Ilustración 1.1. Espejo para la terapia	12
Ilustración 2.1. Diagrama de Gantt - Investigación e iteración 1	20
Ilustración 2.2 Diagrama de Gantt - Iteración 2	20
Ilustración 2.3 Diagrama de Gantt - Iteraciones 3 y 4	21
Ilustración 2.4. Trello	28
Ilustración 2.5. TeamGantt	29
Ilustración 2.6. SourceTree	30
Ilustración 2.7. Putty	30
Ilustración 2.8. CyberDuck	31
Ilustración 2.9. Postman	32
Ilustración 2.10. HeidiSQL	32
Ilustración 2.11. MarvelAPP	33
Ilustración 3.1 Historias de usuario	36
Ilustración 3.2. Arquitectura del sistema	40
Ilustración 3.3. Flujo MySignals	41
Ilustración 3.4. Placa MySignals parte delantera	42
Ilustración 3.5. Placa MySignals parte trasera	42
Ilustración 3.6. Placa MySignals con los sensores	44
Ilustración 3.7. Placa HealthyPi parte delantera	45
Ilustración 3.8. Placa HealthyPi parte trasera	45
Ilustración 4.1. Modelo de datos	50
Ilustración 4.2. Sensor airflow	53
Ilustración 4.3. Sensor de conductividad de la piel	53
Ilustración 4.4. Electrocardiograma	54
Ilustración 4.5. Electromiograma	54
Ilustración 4.6. Diagrama de flujo implementado	55
Ilustración 4.7. Placa MySignals obtenida	56
Ilustración 4.8. Placa MySignals junto con Arduino	57
Ilustración 4.9. Pruebas de carcasa	58
Ilustración 4.10. Visualización de la impresión	59
Ilustración 4.11. Impresión de la carcasa	59
Ilustración 4.12. Sistema de recogida de datos	60
Ilustración 4.13. Diseño aplicación - login	61
Ilustración 4.14. Diseño aplicación - bienvenida	62
Ilustración 4.15. Diseño aplicación - recogida de datos subjetivos	63
Ilustración 4.16. Diseño aplicación - comienzo en reposo	63
Ilustración 4.17. Diseño aplicación - recogida de datos en reposo	64
Ilustración 4.18. Diseño aplicación - comienzo frente al espejo	65

Ilustración 4.19. Diseño aplicación - recogida de datos frente al espejo	66
Ilustración 4.20. Diagrama de secuencia	69
Ilustración 4.21. Diseño web - login	71
Ilustración 4.22. Diseño web sesiones	71
Ilustración 4.23. Diseño web - detalle sesión	72
Ilustración 4.24. Diseño web - usuarios	73
Ilustración 4.25. Gráfica con las medias de los datos recogidos	77

ÍNDICE DE TABLAS

Tabla 2.1. Comparación de metodologías	14
Tabla 2.2. Resumen coste hardware	22
Tabla 2.3. Resumen coste equipo desarrollo	23
Tabla 2.4. Resumen coste software	23
Tabla 2.5. Resumen coste personal	24
Tabla 2.6. Resumen coste total	25
Tabla 3.1. Ventajas e inconvenientes de MySignals	44
Tabla 3.2. Ventajas e inconvenientes placa healthyPI	47

1. INTRODUCCIÓN

En este apartado, se detalla la motivación que ha llevado a la realización de este Trabajo Final de Máster, los objetivos planteados y un estudio relacionado con la temática de este: la insatisfacción corporal y la terapia de exposición frente al espejo.

1.1. Motivación

Actualmente, existen unos estándares de belleza basados en cuerpos imposibles y siluetas perfectas. Las mujeres tenemos que mantenernos delgadas, con cuerpos bien definidos, pechos firmes y rostros perfectos sin arrugas, con labios carnosos y nariz respingona. Los hombres altos, con cuerpos musculados y con una sonrisa propia de un anuncio de dentífricos.

La lucha por alcanzar el canon de belleza impuesto por la sociedad ha contribuido a la aparición de trastornos de la imagen corporal, modificando la autopercepción de la propia imagen. Esto desemboca en el rechazo del propio cuerpo, en una insatisfacción que puede llegar a afectar en la vida de una persona.

He conocido a bastantes personas que no se sentían a gusto con su propio cuerpo, estaban continuamente siguiendo dietas de adelgazamiento o entrenando en el gimnasio, algunas incluso odiaban ir a la piscina y tener que mostrar su cuerpo.

La insatisfacción corporal, provoca que una persona no sea feliz, y en caso extremo, puede derivar en trastornos de la conducta alimentaria y problemas de salud. Por ello, considero que este Trabajo Final de Máster mediante la tecnología móvil y sistemas de bajo coste, puede ayudar a la investigación y facilitar la realización y el seguimiento de terapias dedicadas a atajar este problema y conseguir que una persona se sienta a gusto consigo misma y acepte su cuerpo tal y como es.

1.3. Objetivos

El objetivo general de este Trabajo Fin de Máster es diseñar y desarrollar un sistema de bajo coste para su utilización en el tratamiento de personas que padecen trastorno de conducta, específicamente, en las técnicas de exposición frente al espejo. Con la construcción de este sistema ubicuo, se permite abrir una nueva línea de investigación para determinar la efectividad de este tipo de terapia en un entorno familiar para el paciente.

Para concretar del objetivo general, se dividirá en los siguientes objetivos específicos:

- Estudiar el proceso de las técnicas de exposición frente al espejo y recoger documentación relacionada.
- Realizar un estudio para la determinación de la tecnología a utilizar.
- Montar y desarrollar un sistema encargado de la recogida de distintas variables fisiológicas utilizando un sistema de bajo coste como arduino.
- Diseñar y desarrollar una aplicación móvil para facilitar y guiar el uso del sistema y que permita la recogida de medidas subjetivas.
- Diseñar y desarrollar una aplicación web para el estudio de los datos obtenidos, ayudando así a la toma de decisiones por parte de los profesionales.
- Estudio y desarrollo de la comunicación entre los distintos aplicativos.
- Estudiar la usabilidad de las aplicaciones y realizar pruebas de uso con pacientes y terapeutas reales.

1.2. Estado del arte

La insatisfacción corporal se refiere al disgusto con el propio cuerpo, esto supone la presencia de juicios sobre el cuerpo, que usualmente no coinciden con las características reales de la persona.

Las alteraciones de la imagen corporal incluye dos aspectos relacionados: por una parte, incluye la distorsión perceptiva de la talla, que se considera una sobreestimación del tamaño o aspecto de las partes del propio cuerpo. Por otro lado, la alteración cognitivo-afectiva asociada a la insatisfacción y preocupación por la figura, provocando un rechazo al propio cuerpo. Este último tipo de alteración será el que mejor define el concepto de insatisfacción corporal ^[1].

Hay numerosos estudios que demuestran que una gran parte de la población, especialmente mujeres, se encuentran insatisfechas con su físico:

- En la Universidad de Hidalgo (México), el 82% de las mujeres están insatisfechas con su imagen corporal, mientras que en los hombres el porcentaje de insatisfacción fue menor (70%) ^[2].
- En una encuesta realizada en Estados Unidos, Cash y Henry encontraron que un 48% de mujeres adultas evaluaban negativamente su apariencia, un 63% estaban insatisfechas con su peso y un 49% estaban preocupadas con el sobrepeso. ^[3]

La insatisfacción con el propio peso y la forma corporal de una persona, puede desencadenar una serie de comportamientos inadecuados relacionados con la alimentación y el peso, que suponen un riesgo para el desarrollo de algún tipo de trastorno alimentario ^[4].

La alteración de la percepción de la forma o tamaño corporal es uno de los elementos centrales de los trastornos de la conducta alimentaria y tiene un papel fundamental en la intervención con estas personas^[5]. Todo tratamiento de los trastornos de la conducta alimentaria debería contar con una intervención específica sobre la imagen corporal^[6].

Un estudio determina que más de la mitad de la población investigada (59.1%) muestran insatisfacción corporal. Asimismo se detecta que el 26.89% de la muestra señalan no haber practicado ninguna conducta alimentaria de riesgo en los últimos 6 meses, mientras que el 31.06% señalan que han practicado una de estas conductas, el 35.23% dos conductas, y el 6.82% confirman haber realizado tres o cuatro conductas alimentarias de riesgo. Entre esas conductas destacan, por ejemplo, los atracones, el seguimiento de dietas crónicas y restrictivas, el abuso de laxantes, diuréticos o anfetaminas, el vómito autoinducido y la práctica de ayunos^[7].

Para el tratamiento de los trastornos de conducta alimentaria, se considera la terapia cognitivo-conductual como una de las más efectivas. Una de las técnicas conductuales más utilizadas en la actualidad son basadas en la exposición del cuerpo frente al espejo.

El procedimiento de las técnicas de exposición frente al espejo, consiste en exponer a la persona a la visión de su propio cuerpo en un espejo, asumiendo que la exposición provocará un incremento en el nivel de malestar y/o ansiedad. La exposición prolongada a la imagen en el espejo permite la reducción del malestar y de las conductas de evitación^[8].

Uno de los distintos mecanismos que se estudian en estas técnicas, es la respuesta fisiológica que provoca la exposición frente al espejo, pero hasta el

momento, no queda claro si la aparición de emociones negativas se acompaña de una activación fisiológica ^[8].

Actualmente, el grupo de investigación de Psicología clínica de la Universidad de Jaén, está llevando a cabo terapias de exposición frente al espejo a personas con distintos grados de insatisfacción corporal.

La terapia se realiza en uno de los laboratorios de la Universidad. El laboratorio está equipado con un gran espejo, el cual se puede observar en la ilustración 1.1, y un sofisticado equipo encargado de recoger todas las respuestas fisiológicas de la persona que está realizando la sesión.



Ilustración 1.1. Espejo para la terapia

Una vez finalizada la terapia, se extraen los valores fisiológicos recogidos por los sensores en formato CSV y se exportan a otra herramienta encargada de la realización del análisis de estos datos, pudiendo así investigar la evolución de la respuesta fisiológica en las distintas sesiones y poder determinar si la aparición de emociones negativas se acompañan de una activación fisiológica.

La terapia de exposición al espejo realizada en el laboratorio, presenta una serie de limitaciones que se pretenden subsanar con el sistema propuesto en este Trabajo Final de Máster. Por un lado, la terapia únicamente se puede realizar en el laboratorio habilitado para ello, ya que es el que dispone del equipamiento necesario y de la máquina con los sensores que registra las variables fisiológicas, además este lugar propicia un ambiente más serio, rígido e imponente. Se propone un sistema de recogida de datos ubicuo, de bajo coste, donde el paciente puede realizar la terapia en un entorno más familiar, donde puede encontrarse más cómodo y relajado para la terapia. Por otro lado, se proporciona a los terapeutas una herramienta web donde tener centralizado toda la gestión de los usuarios, el control de las sesiones y todos los datos recogidos para su posterior análisis.

2. PLANIFICACIÓN

En este apartado, se estudiarán las cuestiones generales relativas a la planificación y gestión del proyecto.

2.1 Metodología

Definir la metodología a que se va a seguir en un proyecto es algo crucial para la línea de vida de este, ya que determinará la forma de desarrollo y la planificación del proyecto.

En primer lugar es necesario decidir entre escoger una metodología tradicional o una metodología ágil. En la siguiente tabla se puede observar las diferencias más significativas entre ambas metodologías.

Tradicional	Ágil
Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
El cliente delega su responsabilidad	El cliente es parte del equipo
Se prioriza en base a la visión propia	Se prioriza en base a las necesidades del cliente
Modelo de desarrollo en cascada	Entregas incrementales e integración continua
Control centralizado	Auto-organización
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios
Grupos grandes y posiblemente distribuidos	Grupos pequeños y trabajando en el mismo sitio

Tabla 2.1. Comparación de metodologías

Basándonos en las características del proyecto a realizar, se escoge seguir una metodología ágil, ya que se necesitará una comunicación constante con el cliente, y debido a las características del proyecto no se definen unos requisitos claros, por lo que debe estar preparado para constantes cambios y nuevos requerimientos que puedan surgir durante el desarrollo.

La metodología se basará en iteraciones con las cuales el proyecto irá evolucionando. Las iteraciones serán aproximadamente de un mes, en las que tras cada una de ellas se obtendrá una versión entregable del proyecto.

Un aspecto fundamental para guiar el desarrollo iterativo e incremental es la priorización de las tareas. Para ello utilizaremos la herramienta Trello (detallada en el apartado 2.6.1.), en la cual se irán definiendo y priorizando las tareas a realizar.

2.2. Tareas

A continuación, se especifican las tareas a realizar durante la ejecución del proyecto con el fin de facilitar su desarrollo y la consecución de los objetivos fijados.

Tarea 1 - Búsqueda de bibliografía y estudio de la terapia de exposición frente al espejo.

Descripción: se realizará una búsqueda de bibliografía y su posterior estudio para comprender y conocer el proceso que se realiza en las terapias de exposición frente al espejo, tanto por parte de los pacientes como de los terapeutas.

Participantes: Cristina Lechuga y Silvia Moreno.

Tarea 2 - Búsqueda de las diversas opciones de hardware disponible para la recogida de datos.

Descripción: se realizará una investigación y análisis de soluciones hardware disponibles en el mercado, que se encarguen de manera sencilla de recoger medidas fisiológicas a través de sensores, concretamente:

electrocardiograma, electromiograma, sensor de conductividad de la piel y airflow.

Participantes: Cristina Lechuga

Tarea 3 - Estudio y configuración del servidor

Descripción: Decidir qué tipo de servidor utilizar y realizar su configuración

Participantes: Cristina Lechuga

Tarea 4 - Programación del hardware para la recogida de los datos.

Descripción: consiste en la programación de la placa Arduino para que se encargue de la recogida de los datos de los distintos sensores.

Participantes: Cristina Lechuga

Tarea 5 - Diseño e implementación de la base de datos

Descripción: esta tarea se centra en el diseño y creación de la estructura de la base de datos utilizada por el sistema.

Participantes: Cristina Lechuga

Tarea 6 - Programación de la comunicación entre el hardware del sistema y el servidor

Descripción: se programa e implementa el envío de los datos desde el dispositivo que se encarga de su recogida hasta el servidor.

Participantes: Cristina Lechuga

Tarea 7 - Diseño de la aplicación móvil

Descripción: diseño de la interfaz de la aplicación móvil que se encargará de asistir y guiar al paciente durante la sesión. Además recogerá datos subjetivos del paciente.

Participantes: Cristina Lechuga, Silvia Moreno, Gustavo Reyes

Tarea 8 - Desarrollo de la aplicación móvil

Descripción: desarrollo en Android de la aplicación diseñada junto con los especialistas.

Participantes: Cristina Lechuga

Tarea 8.A - Desarrollo de la aplicación móvil - Login

Descripción: implementación del inicio de sesión para los pacientes activos.

Participantes: Cristina Lechuga

Tarea 8.B - Desarrollo de la aplicación móvil - Recogida de datos subjetivos

Descripción: se recogerán datos subjetivos mediante preguntas acerca del estado en el que se encuentra el paciente.

Participantes: Cristina Lechuga

Tarea 8.C - Desarrollo de la aplicación móvil - Gestión del estado de la sesión

Descripción: gestionar los distintos estados de la sesión para el correcto almacenamiento de los datos.

Participantes: Cristina Lechuga

Tarea 9 - Diseño de la aplicación web

Descripción: Diseño de la interfaz gráfica de la aplicación web que utilizarán los especialistas para el análisis de los datos y seguimiento de los distintos pacientes.

Participantes: Cristina Lechuga, Silvia Moreno, Gustavo Reyes

Tarea 10 - Desarrollo de la aplicación web

Descripción: Desarrollo de la aplicación web que utilizarán los especialistas para el análisis de los datos y seguimiento de los distintos pacientes.

Participantes: Cristina Lechuga

Tarea 10.A - Desarrollo de la aplicación web - Login

Descripción: Implementación de un login para el inicio de sesión de los administradores

Participantes: Cristina Lechuga

Tarea 10.B - Desarrollo de la aplicación web - Gestión de usuarios

Descripción: Desarrollo de la gestión de los usuarios del sistema: creación de nuevos usuarios, modificación de datos y activación o desactivación de usuarios.

Participantes: Cristina Lechuga

Tarea 10.C - Desarrollo de la aplicación web - Visualización de las sesiones

Descripción: Desarrollo de la visualización de las sesiones de usuarios y los datos asociados a cada una de ellas.

Participantes: Cristina Lechuga

Tarea 10.D - Desarrollo de la aplicación web - Descarga en formato CSV de los datos recogidos por los sensores

Descripción: Desarrollo de una funcionalidad que permita descargar en formato CSV los datos recogidos durante la sesión de un paciente.

Participantes: Cristina Lechuga

Tarea 10.E - Desarrollo de la aplicación web - Realización de un análisis de los datos recogidos durante la sesión

Descripción: Se determinará un análisis básico de los datos recogidos en una sesión, para poder comparar el estado del paciente antes y durante la realización de la terapia.

Participantes: Cristina Lechuga

Tarea 11 - Testeo y pruebas

Descripción: Realizar pruebas para comprobar el correcto funcionamiento de las aplicaciones y su posterior corrección.

Participantes: Cristina Lechuga

Tarea 12 - Desarrollo y documentación de la memoria del proyecto

Descripción: se desarrollará una memoria del Trabajo Fin de Máster con toda la documentación acerca de la planificación, análisis y desarrollo del proyecto.

Participantes: Cristina Lechuga y Lidia Ortega

2.3. Estimación de tiempo

En la realización del Trabajo Final de Máster es imprescindible determinar una planificación temporal adecuada a las tareas especificadas en el apartado anterior. Para establecer las horas dedicación al proyecto, nos ajustaremos a las fijadas en la Guía Académica de este, que especifica 300 horas de trabajo por parte del estudiante.

Estas 300 horas se dividirán en una jornada laboral de cuatro horas (media jornada) en días laborables, lo que supone un cómputo semanal de veinte horas, por lo que, cumpliendo los horarios establecidos, el proyecto se finalizará aproximadamente en cuatro meses.

En el siguiente resumen, se establece una correspondencia entre las iteraciones estimadas para el desarrollo del proyecto y las distintas tareas que se han definido en el apartado anterior:

Previo a comenzar el desarrollo - Investigación: Tareas 1, 2 y 12.

Iteración 1. Tareas 3, 4, 5, 6 y 12.

Iteración 2. Tareas 7, 8 y 12.

Iteración 3. Tareas 9, 10 y 12.

Iteración 4. Tarea 11 y 12.

A continuación se muestra un diagrama de Gantt, una herramienta muy útil para la visualización de la planificación temporal de las tareas establecidas, donde se han descompuesto en tareas más específicas.

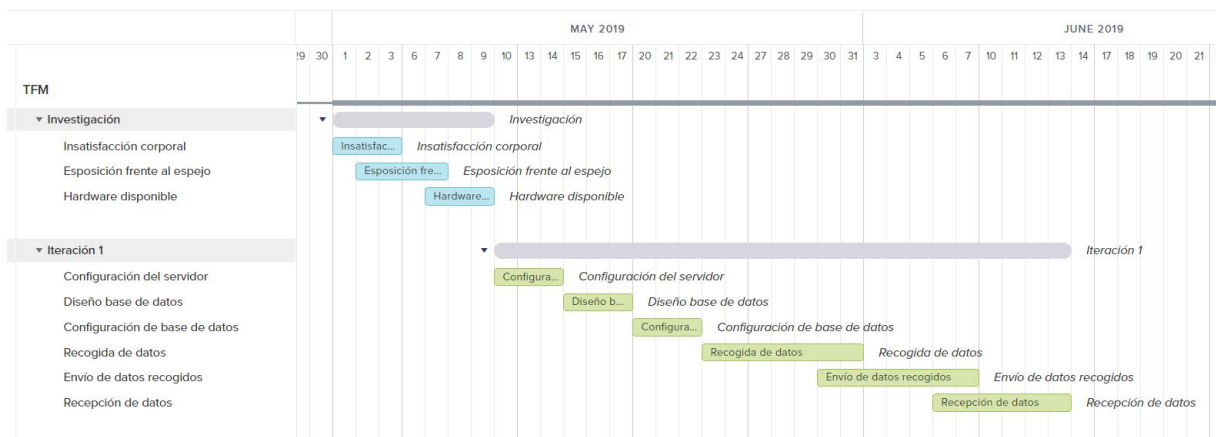


Ilustración 2.1. Diagrama de Gantt - Investigación e iteración 1

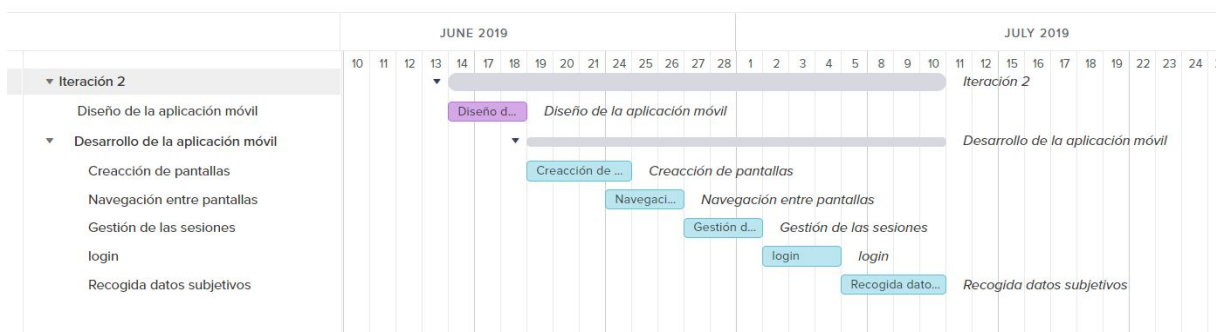


Ilustración 2.2 Diagrama de Gantt - Iteración 2

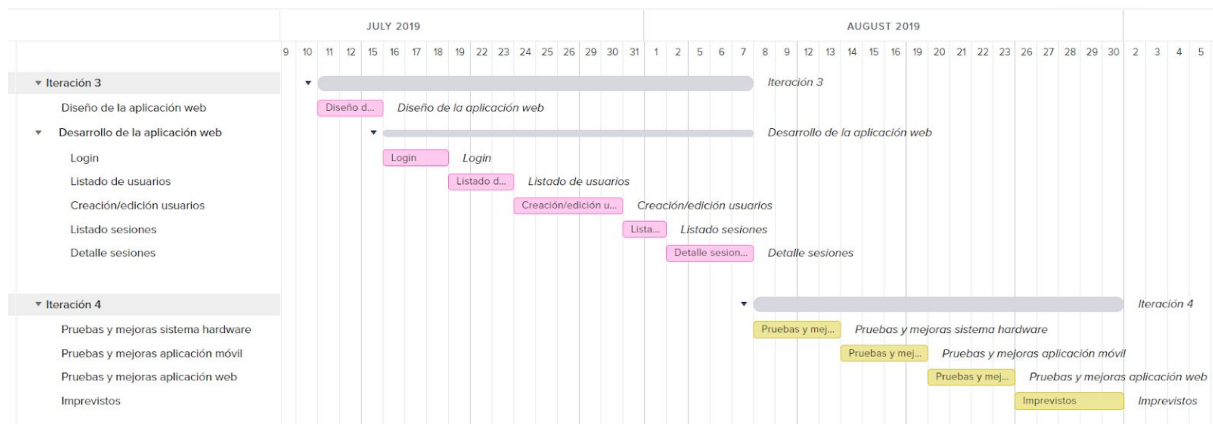


Ilustración 2.3 Diagrama de Gantt - Iteraciones 3 y 4

2.4. Estimación de costes

Un aspecto fundamental en la gestión y planificación de un proyecto, es la estimación del coste que supondrá su desarrollo, ya que esto determina la viabilidad de este. Hay que garantizar que se podrá contar con el software, equipo y personal necesario para el completo desarrollo del proyecto.

A continuación, se realizará una estimación del coste que supondría el desarrollo de este Trabajo Final de Máster, suponiendo que se solicita a una empresa externa encargada de otros proyectos.

Comenzamos evaluando el coste del equipo hardware necesario para la recogida de las medidas fisiológicas. Tras el estudio realizado (ver 3.4. Hardware), se decide comprar la placa MySignals la cual facilita la recogida de estos datos junto con la placa Arduino UNO necesaria para su funcionamiento, así como los siguientes sensores: Electrocardiograma, electromiograma, airflow y conductividad de la piel (GSR). Este equipamiento se podrá utilizar para otro proyectos similares, por lo que se calculará la amortización por uso durante los cuatro meses de duración del proyecto, estimando su vida útil en cuatro años. A continuación se detalla el coste de cada uno de estos elementos:

Hardware	Coste sin amortizar	Coste amortizado
MySignals HW	379.00 €	31.58 €
Arduino UNO	19.95 €	1.66 €
Electrocardiograma	55.00 €	4.58 €
Electromiograma	57.00 €	4.75 €
Airflow	55.00 €	4.58 €
GSR	74.00 €	6.16 €
Total	639.95 €	53.31 €

Tabla 2.2. Resumen coste hardware

Además del hardware necesario para la toma de datos, también hay que disponer del equipo para el desarrollo. En este caso, contaremos con un ordenador de clase media valorado en 500€, y un dispositivo móvil Android para realizar las pruebas, valorado aproximadamente en 200€. Se calculará su valor teniendo en cuenta la amortización a cuatro meses de ambos equipos, se considera que vida útil de estos equipos es de cuatro años.

En este apartado, también sumaremos el coste del alojamiento web. A pesar de que realmente se ha utilizado un servidor del departamento de informática de la Universidad de Jaén gratuito, la empresa ficticia debería asumir el coste de este alojamiento, donde se desplegará el proyecto y se almacenarán los datos. Se contratará un servidor privado virtual de la empresa OVH, el cual se utilizará durante los cuatro meses del proyecto, y su coste es de 2.99€ mensuales. Para este concepto no es necesario calcular la amortización, ya que se contrata exclusivamente para este proyecto ^[9].

Equipo	Coste sin amortizar	Coste amortizado
Ordenador	500 €	41,66 €
Dispositivo móvil	200 €	16,66 €
Alojamiento web	11.96 €	11.96 €
Total		70.28 €

Tabla 2.3. Resumen coste equipo desarrollo

A continuación, evaluaremos el coste que supone utilizar las licencias de software necesarias (ver apartado 2.5) para el desarrollo del proyecto. La mayoría de las herramientas que se utilizará durante el proceso son gratuitas, pero, la empresa debe contar con la licencia para la utilización de IntelliJ IDEA para el desarrollo de los servicios web, la cual se puede obtener por meses.

Software	Coste por mes	Coste total
Licencia IntelliJ IDEA Ultimate	49.90 €	199.60€

Tabla 2.4. Resumen coste software

El proyecto ha sido enteramente asumido por el alumno, el cual abarcará todos los roles necesarios. A pesar de esto se realizará el cálculo del coste que asumiría la empresa con respecto al personal necesario para el desarrollo del proyecto.

Se requerirán los siguientes perfiles profesionales para poder completar la entrega del proyecto:

- Jefe de proyecto. Es la figura clave del proyecto, ya que será el encargado de supervisar todo el proceso de desarrollo, desde el inicio hasta la entrega final al cliente. Se encarga de la toma de decisiones a nivel operativo, de garantizar la coordinación del equipo y la consecución de los objetivos. Se contratará con la calificación profesional correspondiente al grupo A-I.

- Analista/Diseñador. Será el encargado del análisis y diseño del sistema, teniendo en cuenta los requisitos establecidos y las características del proyecto. Se contratará con la calificación profesional correspondiente al grupo B-I.
- Programador. Será el encargado de todo el desarrollo software y la implementación del sistema. Deberá tener un perfil de desarrollador full-stack debido a la necesidad de conocimientos en distintos lenguajes de programación y en el desarrollo de Front-End y Back-End. Se contratará con la calificación profesional correspondiente al grupo A-I.

Para calcular el coste del personal requerido, se tendrá en cuenta el tiempo que dedicarán a este proyecto, teniendo en cuenta que únicamente el jefe del proyecto trabajará en él durante todo el periodo de ejecución del proyecto, ya que la empresa se dedica a otros proyectos además de a este.

En la siguiente tabla, se analizará el porcentaje dedicado al proyecto por cada uno de los distintos perfiles así como el salario que recibirán por las horas dedicadas al proyecto, se calculará teniendo en cuenta el salario base más plus salariales, el cual estará determinado por XVII Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública ^[10]. Tampoco podemos olvidar añadir en la tabla la cantidad asociada a las cotizaciones a cargo de la empresa de cada uno de los trabajadores, lo que supone un 33% de su salario base.

Perfil	Meses	Dedicación	Horas totales	Salario base total	Plus salarial	Cotización empresa	Total
Jefe de proyecto	4	10%	30 h.	417,24 €	29,24 €	137,68 €	584,16 €
Analista/Diseñador	1	20%	60 h.	809,91 €	56,30 €	267,27 €	1133,48 €
Programador	3.5	70%	120 h.	2834,70 €	197.06 €	935,45 €	3967,20 €
Coste total							5.684,84€

Tabla 2.5. Resumen coste personal

Para finalizar, hay que tener en cuenta los costes indirectos que supone la realización del proyecto, como el coste de la electricidad, la conexión a internet, el agua, la oficina... así como gastos imprevistos que puedan surgir durante el desarrollo. Esto se estimará estableciendo un incremento del 25% al coste total del proyecto.

Para resumir y obtener el costo total que asumiría la empresa, se analizarán todos los conceptos anteriores en la siguiente tabla:

Concepto	Coste
Hardware	53.31 €
Equipamiento	70.28 €
Software	199.60 €
Personal	5.684,84 €
Costes indirectos	1502,00 €
TOTAL	7.510,03 €

Tabla 2.6. Resumen coste total

El presupuesto total para la realización del proyecto asciende a 7.510,03 €.

2.5. Tecnología utilizada

A grandes rasgos, se necesitará un servidor web para la aplicación del servidor, otro para la aplicación web y tecnología para la creación de la base de datos y el compilado y ejecución de las distintas aplicaciones.

2.5.1. Apache Tomcat

Apache Tomcat, es el servidor web usado tradicionalmente para proyectos Java por su implementación de servlets o páginas JSP. Es desarrollado, en un entorno abierto y participativo y publicado bajo la licencia Apache versión 2, por

miembros de la Apache Software Foundation y voluntarios independientes. Es gratuito, sencillo de configurar y dispone de abundante documentación ^[11].

2.5.2. Apache 2

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP. El servidor Apache es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la Apache Software Foundation dentro del proyecto HTTP Server (httpd). Es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web ^[12].

2.5.3. MySql

MySQL es un sistema de gestión de base de datos relacional de código abierto, basado en lenguaje de consulta estructurado (SQL).

Es la base de datos de código abierto más popular del mercado. Gracias a su rendimiento probado, a su fiabilidad y a su facilidad de uso, MySQL se ha convertido en la base de datos líder elegida para las aplicaciones basadas en web y utilizada por propiedades web de perfil alto ^[13].

2.5.4. Spring

Spring es un framework de código abierto para la creación de aplicaciones Java. Tiene una estructura modular y una gran flexibilidad para implementar diferentes tipos de arquitectura según las necesidades de la aplicación. Spring nos permite desarrollar aplicaciones de manera más rápida, eficaz y corta, saltándonos tareas repetitivas y ahorrándonos líneas de código ^[14].

Se utilizará el módulo de SpringBoot, el cual proporciona un conjunto de herramientas para facilitar la creación y configuración de aplicaciones basadas en Spring.

2.5.5. Maven

Maven es una herramienta de software para la gestión y construcción de proyectos Java. Nos permite una gestión correcta de las librerías necesarias, proyectos y dependencias ^[15].

Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos, componentes externos, y el orden de construcción de los elementos.

2.5.6. Yeoman

Yeoman es un creador de proyectos, de tal forma que evita a los desarrolladores los problemas de crear un proyecto de inicio. Se ejecuta dentro de NodeJs por el cual podemos crear proyectos de los principales proyectos Web, como puede ser una aplicación vNext, una SPA basada en el framework JavaScript que decidamos utilizar en nuestro desarrollo (ReactJS, AngularJS, Backbone, etc..). De la misma forma, estas plantillas pueden utilizar otras herramientas de FrontEnd como Grunt, Bower o npm ^[16].

Dispone de multitud de plantillas y se utilizará para la creación del proyecto de la aplicación web.

2.5.7. Grunt

Grunt ejecutor de tareas de JavaScript, una herramienta que nos permite automatizar las tareas de arranque del servidor y preprocesamiento de archivos y así ahorrarnos tiempo en nuestro desarrollo y despliegue de aplicaciones webs. Utiliza NodeJS para su ejecución lo que lo hace multiplataforma ^[17].

Con un simple fichero JS que llamaremos Gruntfile, indicamos las tareas que queremos automatizar con un simple comando y las escribimos en él en formato JSON.

Se utilizará Grunt para realizar de manera automática el despliegue de la aplicación web para depurar en local.

2.6. Software utilizado

Se necesitarán distintas herramientas para facilitar la planificación y gestión del proyecto, así como para facilitar el acceso al servidor y a la base de datos. También, se incluye los entornos de desarrollo utilizados para los distintos aplicativos.

2.6.1. Trello

Es una herramienta de gestión de proyectos que proporciona una amplia perspectiva de estos. Se decide el uso de Trello para facilitar la planificación del proyecto ya que es una herramienta muy visual y permite organizar y priorizar tareas de forma rápida y sencilla ^[18]. Además ayudará para contabilizar las horas y comprobar la variación de la planificación inicial.

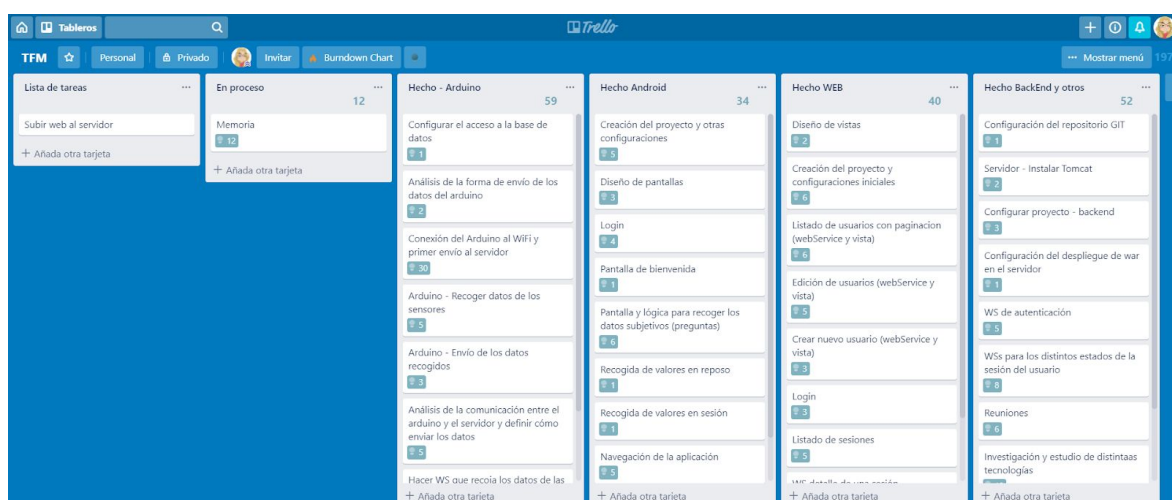


Ilustración 2.4. Trello

2.6.2. TeamGantt

TeamGantt es una herramienta online gratuita que permite la creación de diagramas de Gantt. Se utilizará para crear la planificación inicial de tareas ^[19]. Es muy sencillo y cómodo de usar y dispone de muchas características personalizables.

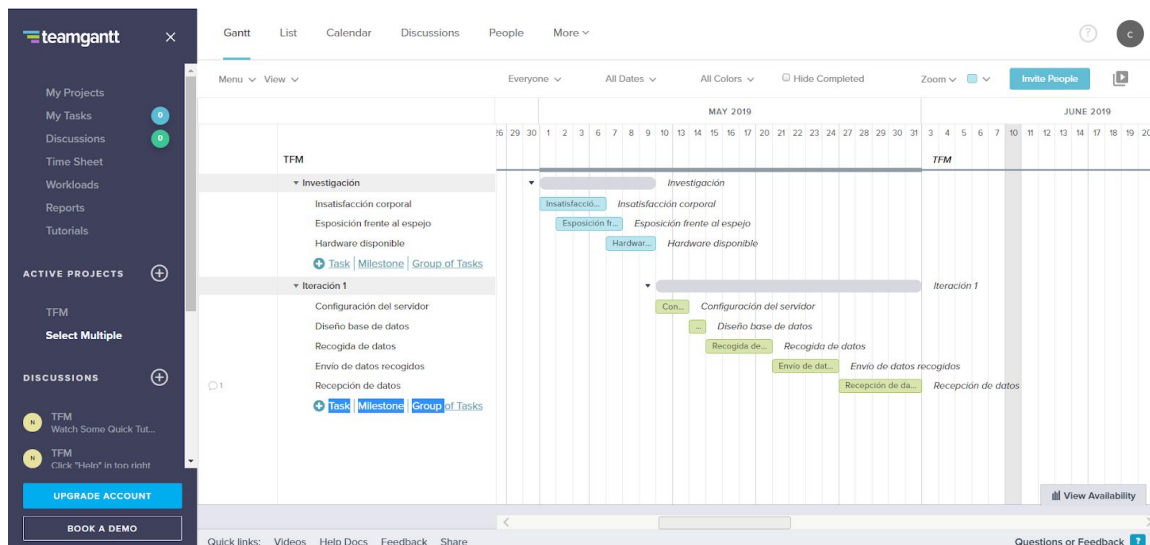


Ilustración 2.5. TeamGantt

2.6.3. SourceTree

Durante el desarrollo de un proyecto, es primordial la gestión de los cambios que se realizan sobre los distintos archivos a lo largo de todo el desarrollo, es por ello que se utilizará un control de versiones ^[20]. Debido a la experiencia con la herramienta, se utiliza Git, un sistema de control de versiones distribuido de código abierto.

SourceTree es un cliente de Git gratuito, el cual simplifica la forma de interactuar con los repositorios de Git. Permite visualizar y administrar los repositorios mediante una sencilla interfaz de usuario.

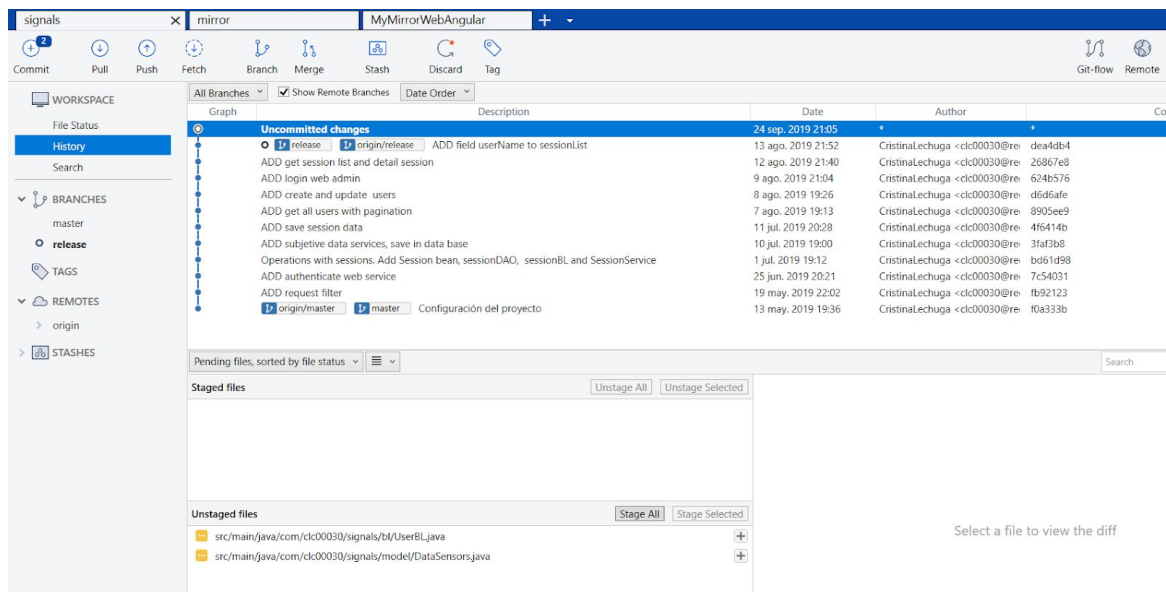


Ilustración 2.6. SourceTree

2.6.4. Putty

Putty es un cliente SSH y Telnet con el que podemos conectarnos a servidores remotos iniciando una sesión en ellos que nos permite ejecutar comandos [21]. Se utilizará esta herramienta para la comunicación con el servidor remoto, instalar y configurar lo necesario así como el acceso a logs del servidor para pruebas y depuraciones.

```

root@usuario-KVM: /home/usuario
login as: usuario
usuario@kefren.ujaen.es's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-161-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Pueden actualizarse 138 paquetes.
5 actualizaciones son de seguridad.

*** Es necesario reiniciar el sistema ***
Last login: Mon Sep 16 20:19:32 2019 from 139.47.102.8
usuario@usuario-KVM:~$ sudo su
[sudo] password for usuario:
root@usuario-KVM: /home/usuario# tail -f /opt/tomcat/logs/catalina.out
Hibernate: SELECT COUNT(id) FROM Users
Hibernate: SELECT ID, userName, firstName, lastName, email, phoneNumber, DNI, password, status FROM Users LIMIT 0,5
23-Sep-2019 21:14:15.614 INFORMACIÓN [http-nio-80-exec-11] com.clc00030.signals.ws.rest.filter.RequestFilter.doFilter REST Request -
[HTTP METHOD:GET]
[PATH INFO:/signals-1/user/get-all]
[REQUEST PARAMETERS:{pageNumber=0, resultsPerPage=5}]
[REQUEST BODY:]
[REMOTE ADDRESS:139.47.100.86]
[META:200]
[RESPONSE:{meta:{code:200,totalQueryCount:8},response:[{"userName":"clc00030","firstName":"Cristina","lastName":"Lechuga","email":"clc00030@red.ujaen.es","phoneNumber":"686151244","password":"123456","status":1,"dni":"773602056","id":1}, {"userName":"clc","firstName":"Cris","lastName":"Test","email":"clc00030@red.ujaen.es","phoneNumber":null,"password":"123","status":1,"dni":null,"id":2}, {"userName":"jdca00022","firstName":"Jose David","lastName":"Calada","email":"jjj","phoneNumber":null,"password":"123","status":1,"dni":"11122119","id":3}, {"userName":"jana111","firstName":"Ana","lastName":"Montes","email":"jjj@gmail.com","phoneNumber":null,"password":"123","status":1,"dni":null,"id":4}, {"userName":"srs111","firstName":"Carmen","lastName":"Rodriguez","email":"jjj","phoneNumber":null,"password":"123","status":0,"dni":null,"id":5}]]}]}

```

Ilustración 2.7. Putty

2.6.5. CyberDuck

Se trata de un cliente FTP de código abierto muy sencillo de utilizar y de configurar el cual utilizaremos para transferir archivos y los ejecutables del proyecto al servidor remoto mediante protocolo SFTP [22].

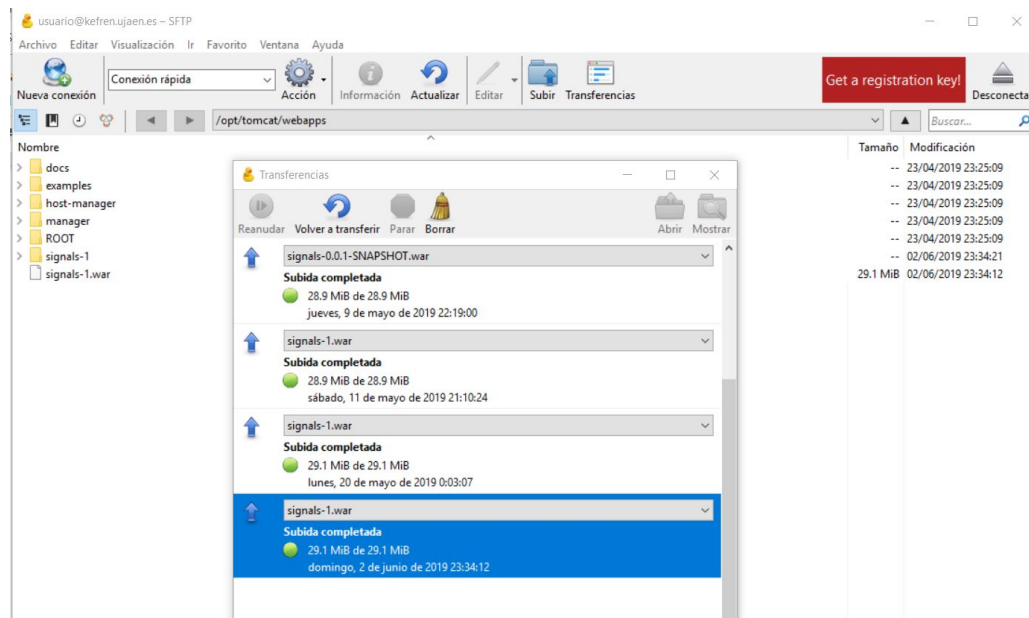


Ilustración 2.8. CyberDuck

2.6.6. Postman

Postman es un cliente HTTP que nos permite gestionar las peticiones a nuestras API's. Actualmente es una de las herramientas más populares utilizadas en las pruebas ya que tiene muchas funcionalidades para gestionar todo el ciclo de vida de nuestra API [23].

Será utilizado para realizar las pruebas necesarias de la API.

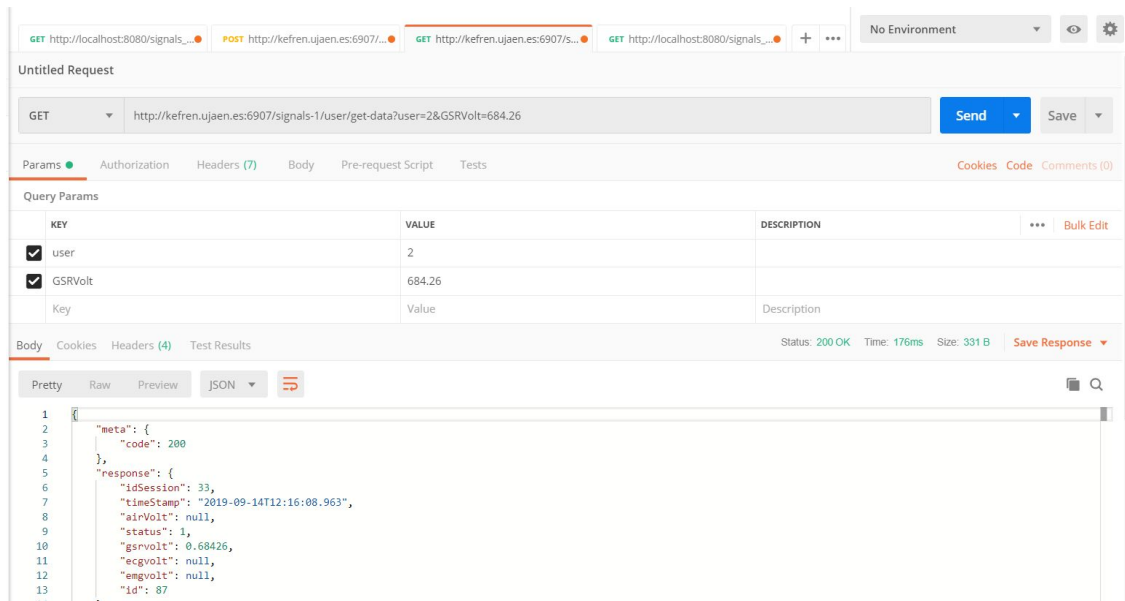


Ilustración 2.9. Postman

2.6.7. HeidiSQL

HeidiSQL un software de código abierto que permite, de manera rápida y sencilla, la administración, desarrollo y gestión de bases de datos [24]. El gestor de base de datos se utilizará para conectarnos de manera remota a la base de datos alojada en el servidor.

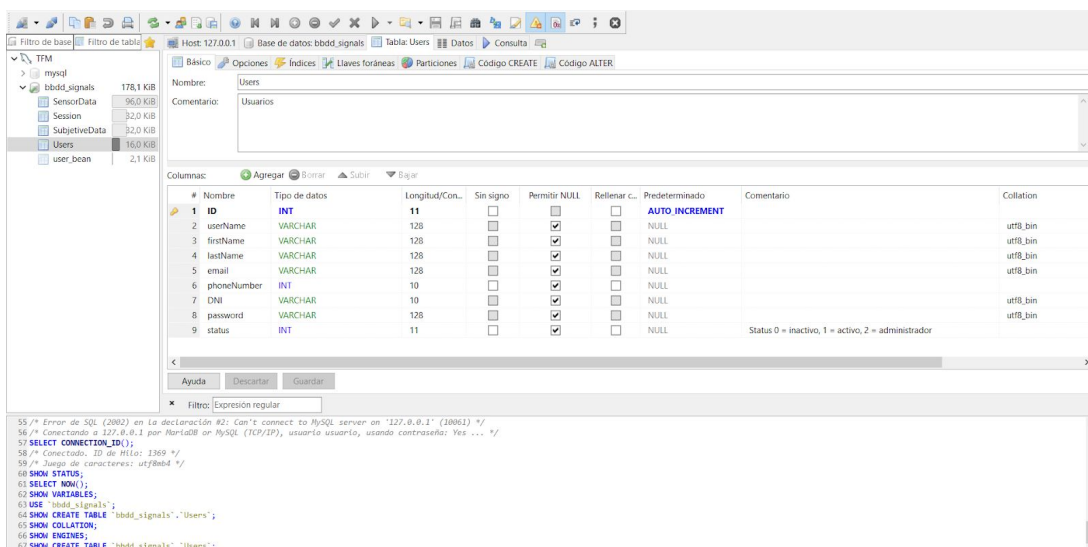


Ilustración 2.10. HeidiSQL

2.6.8. MarvelAPP

Marvel es una herramienta online que nos permite crear prototipos de aplicaciones móviles y proyectos web [25]. Es muy fácil de usar y tiene la posibilidad de añadir bocetos desde la aplicación móvil de Marvel. Además permite agregar interactividad entre las pantallas y múltiples tipos de transiciones y gestos.

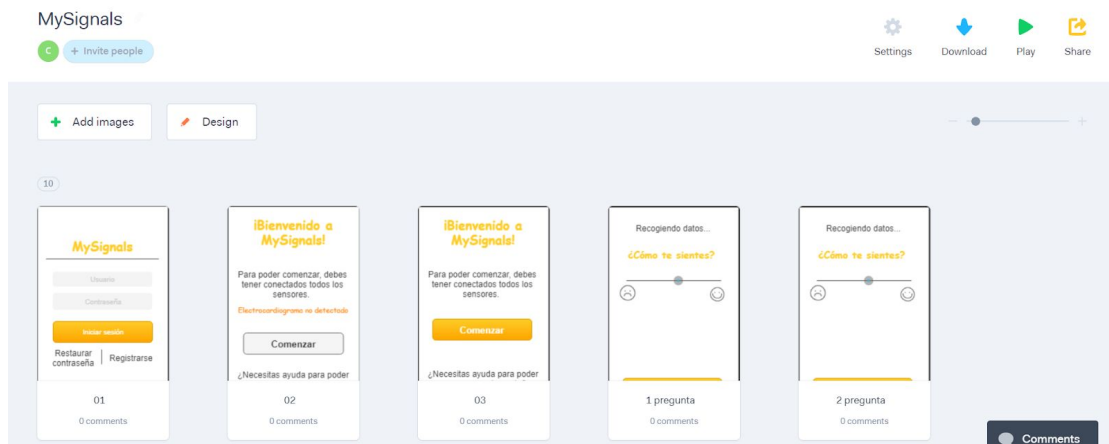


Ilustración 2.11. MarvelAPP

2.6.9. IntelliJ IDEA

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) para el desarrollo de programas informáticos. Es un entorno muy potente diseñado específicamente para maximizar la productividad del desarrollador. Es desarrollado por JetBrains. Para el desarrollo del proyecto se utilizará IntelliJ IDEA debido a la experiencia y familiaridad con este IDE, con él se desarrollará tanto el back como la aplicación web [26].

Para su uso, se obtendrá una licencia gratuita de estudiante.

2.6.10. Arduino IDE

El entorno de desarrollo integrado de Arduino es una aplicación multiplataforma que está escrita en el lenguaje de programación Java. Se utiliza para escribir y cargar programas en tableros compatibles con Arduino [27].

Será el utilizado para el desarrollo del hardware encargado de la recogida de datos de los sensores y su envío al servidor.

2.6.11. *Android Studio*

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android, lanzado por Google. Está diseñado para acelerar su desarrollo y ayudar a construir las aplicaciones de alta calidad ^[28].

Ofrece un entorno de desarrollo claro y robusto que facilita el entorno de pruebas en los distintos tipos de dispositivos.

3. ANÁLISIS DEL SISTEMA

En el siguiente apartado se realiza un análisis del sistema requerido, detallando las historias de usuario, definiendo los requisitos del sistema y finalmente planteando la arquitectura del mismo.

3.1. Historias de usuario

Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la facilitar la especificación de requisitos, se escriben en lenguaje coloquial, son cortas y esquemáticas, y resumen la necesidad concreta de un usuario al utilizar un producto o servicio.

El estilo de redactar una historia de usuario es libre, pero debe responder a tres preguntas: ¿Quién se beneficia?, ¿qué se quiere? y ¿cuál es el beneficio? Por ello, algunos autores recomiendan redactar las historias de usuario según el formato: Como (rol) quiero (algo) para poder (beneficio) [29].

A continuación, se muestran las historias de usuario descritas conforme a este formato.

Como paciente, quiero una aplicación móvil para que me guíe durante la sesión

H.U.1

Como paciente, quiero tener una guía que me explique cómo tengo que conectar los sensores para utilizarla cuando tenga dudas

H.U.1.2

Como paciente, quiero iniciar sesión en la aplicación móvil para garantizar que sólo puedo entrar yo

H.U.1.3

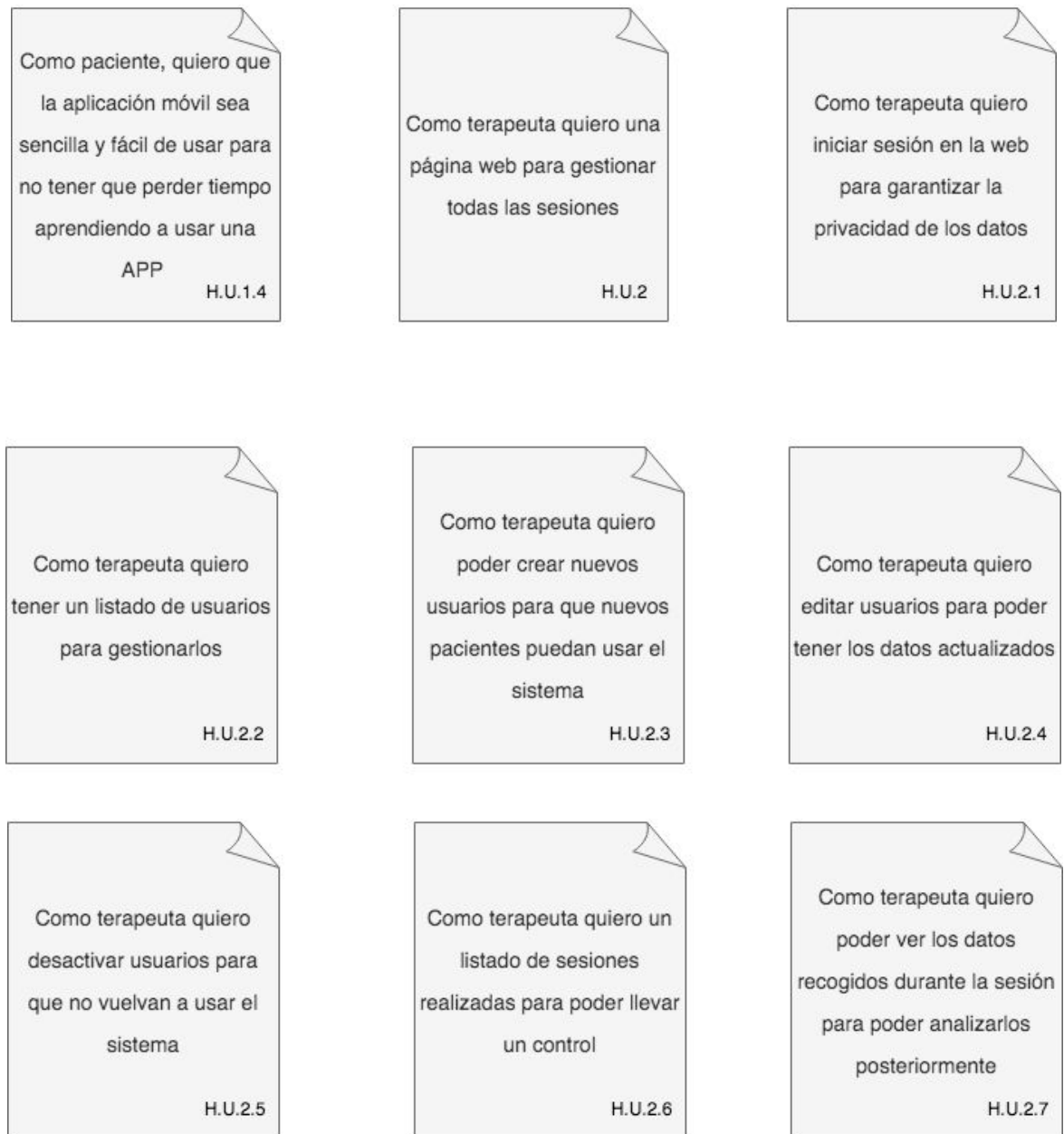


Ilustración 3.1 Historias de usuario

3.2. Requisitos funcionales y no funcionales

Tras el estudio de las historias de usuario, los requisitos se especificarán junto al cliente en la etapa inicial del proyecto. A pesar de esto, estos requisitos podrán ir evolucionando a lo largo del desarrollo del proyecto.

Los requisitos funcionales, son aquellas declaraciones que se refieren a funciones específicas que entrega el sistema, es decir, denotan la funcionalidad del sistema.

Los requisitos no funcionales se refieren a las propiedades o características requeridas del sistema, es decir, todas las exigencias de cualidades que se imponen al proyecto.

A continuación se especifican los requisitos funcionales y no funcionales de los distintos elementos:

3.2.1. Requisitos funcionales

- Dispositivo Hardware
 - El dispositivo debe recoger los datos de los sensores.
 - El dispositivo debe enviar los datos recogidos al servidor.
- Aplicación Android
 - La aplicación deberá guiar al usuario con respecto a las acciones a realizar para la correcta recogida de los datos.
 - Mediante distintas preguntas, la aplicación deberá recoger datos subjetivos del paciente y los subirá al servidor en tiempo real
- Aplicación Web
 - La aplicación web permitirá la gestión de usuarios: creación, modificación y activación/desactivación.
 - La aplicación web deberá mostrar al profesional los datos obtenidos de los pacientes.
 - La aplicación web permitirá la descarga en formato CSV de los datos recogidos.
 - La aplicación web contendrá gráficas para la ayuda del análisis de los datos recogidos.

3.2.2. Requisitos no funcionales

- Dispositivo Hardware
 - Sistema de bajo coste
 - El sistema hardware deberá ser cómodo y fácil de usar.
 - El muestreo de los sensores serán como mínimo de 100 veces por segundo para electrocardiograma y electromiograma, mientras que para los demás sensores será de una muestra por segundo.
 - Los datos serán recogidos y entregados al cliente en voltios
- Aplicación móvil
 - La aplicación se desarrollará para dispositivos Android.
 - Aplicación intuitiva y usable.
- Aplicación web
 - Carga de datos rápida y eficiente.
 - Interfaz intuitiva y usable.
 - La aplicación web debe garantizar la seguridad de los datos con respecto a lecturas no autorizadas o modificaciones de datos, así como la disponibilidad de estos.

3.3. Arquitectura del sistema

En este apartado, se definirá la arquitectura del sistema, es decir, la representación de los distintos elementos que compondrán el sistema y la relación entre ellos.

Se implementará una arquitectura API REST, caracterizada por la separación del cliente y servidor. La Interfaz uniforme definirá una interfaz genérica para administrar cada interacción que se produzca entre el cliente y el servidor de manera uniforme, lo cual simplifica y separa la arquitectura.

También se caracteriza por presentar un conjunto de operaciones bien definidas (GET, POST, PUT y DELETE) empleadas en todos los recursos. Además, utiliza URIs únicas siguiendo una sintaxis universal y emplea hipermedios para representar la información, que suelen ser HTML, XML o JSON.

En el conjunto de todo el sistema, podemos distinguir los cuatro elementos principales que lo compondrán:

- **Sistema de recogida de datos (cliente):** Es el encargado de recoger los datos fisiológicos de los pacientes. Se compone de una placa MySignals conectada a una placa arduino junto con los distintos sensores.
La placa Arduino será programada para recoger los datos de los sensores y enviarlos a nuestro servidor a través del módulo WiFi del que dispone la placa MySignals.
- **Aplicación móvil (cliente):** Aplicación desarrollada en Android para guiar al paciente durante la sesión, se comunicará mediante API con el backend para realizar el ciclo de vida de las sesiones así como el almacenamiento de los datos subjetivos recogidos.
- **Aplicación web (cliente):** La página web se alojará en el servidor y será accesible para los profesionales, desde esta, podrán realizar la gestión de los pacientes y el análisis de los datos recogidos en las distintas sesiones.
- **Servidor:** es el núcleo de todo el sistema y el encargado de realizar toda la lógica necesaria para el correcto funcionamiento del proyecto. Mediante Hibernate, se realiza la conexión con la

base de datos para facilitar la persistencia y consulta de los datos.

En el siguiente esquema, podemos ver la representación de los distintos elementos y la comunicación entre ellos.

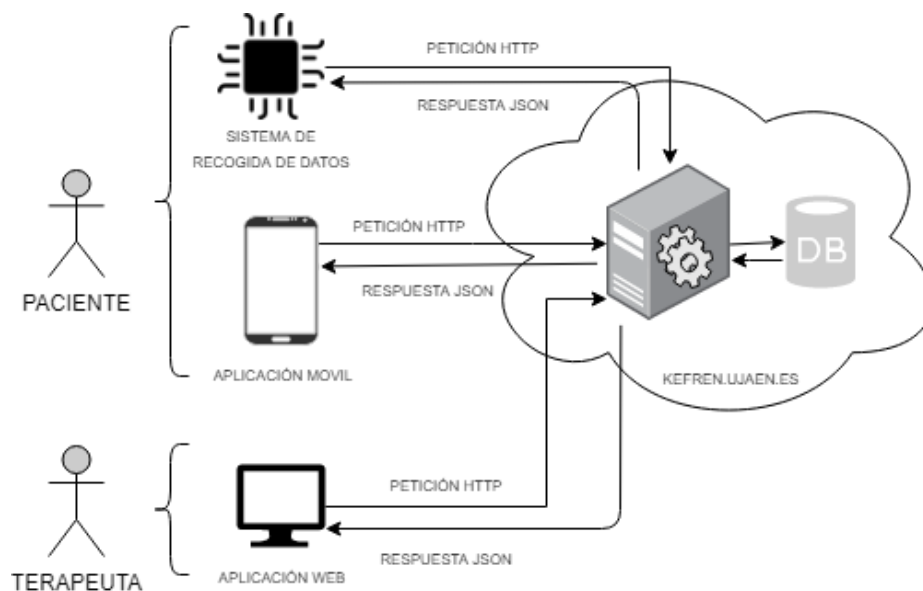


Ilustración 3.2. Arquitectura del sistema

3.4. Hardware

Una de los primeros pasos de este proyecto, es decidir qué equipo hardware utilizar para la recogida de las respuestas fisiológicas del paciente. Se realizará un estudio con las distintas alternativas que existen en el mercado teniendo en cuenta que debe ser de bajo coste y garantizar unas mediciones fiables. A continuación se mostrarán las dos opciones que más se ajustaban a los requisitos.

3.4.1. MySignals

MySignals HW Development Platform para Arduino soporta hasta 17 sensores que pueden utilizarse para monitorizar más de 20 parámetros biométricos.

Todos los datos recopilados por MySignals pueden cifrarse y enviarse a la cuenta privada del usuario en Libelium Cloud a través de WiFi o Bluetooth, también es posible configurarlo para enviar a un servidor de terceros (Uso directo de WiFi (HTTP, TCP / IP, etc.)). Para para hacer uso de la placa MySignals es necesario un Arduino Uno ^[30].

En el siguiente esquema, podemos observar las distintas alternativas y flujo de los datos con este hardware.

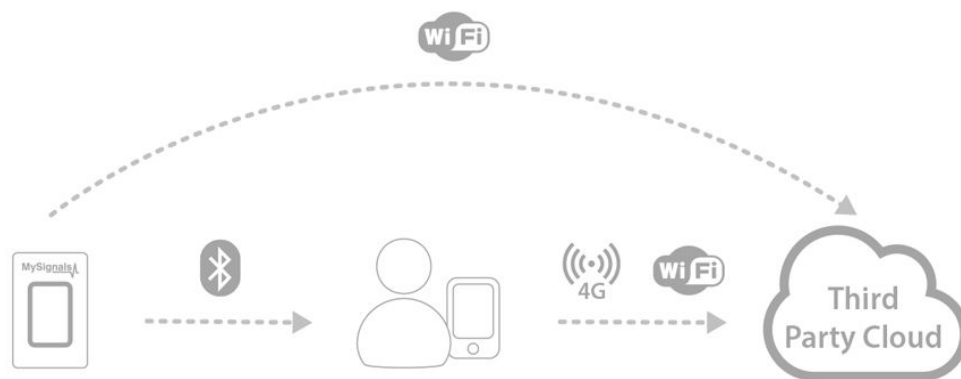


Ilustración 3.3. Flujo MySignals

En cuanto a características técnicas, destacaremos las siguientes:

- Memoria RAM: 2K
- Microprocesador: ATmega 328 (Arduino Uno)
- Sockets UART: 1 (multiplexado)
- Pantalla: TFT (gráficos básicos)
- Radios a bordo: BLE, WiFi

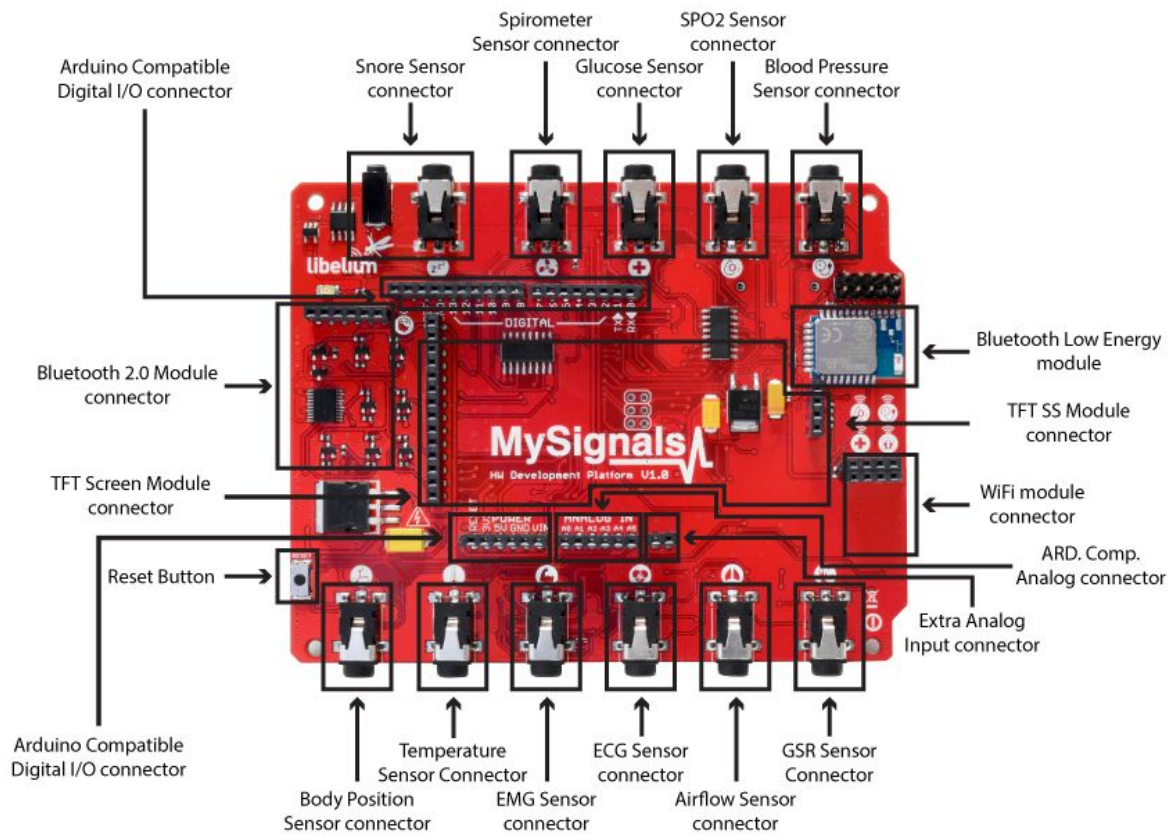


Ilustración 3.4. Placa MySignals parte delantera

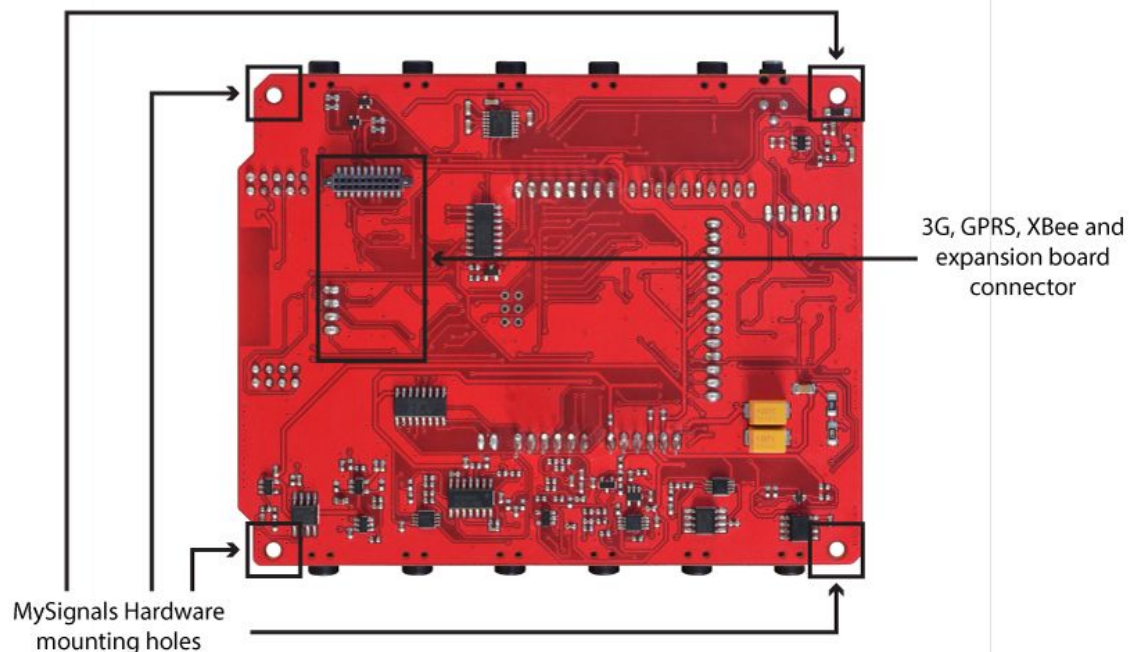


Ilustración 3.5. Placa MySignals parte trasera

El precio de esta placa asciende a 379€, a lo que hay que sumar el coste del Arduino necesario para su utilización y de los sensores que se quieran utilizar en este primer prototipo. A continuación se muestra el listado de sensores que se pueden adquirir para esta placa.

- SPO2 Pulse Oxygen in Blood
- ECG Electrocardiogram
- Airflow (Breathing)
- Blood Pressure
- Glucometer
- Body Temperature
- EMG Electromyography
- Spirometer Air Capacity
- GSR Galvanic Skin Response
- Body Position
- Snore Sensor
- Body Scale BLE
- SPO2 Pulse Oxygen in Blood BLE
- Blood Pressure BLE
- Glucometer BLE
- Mindwave EEG Electroencephalogram
- Body Temperature BLE

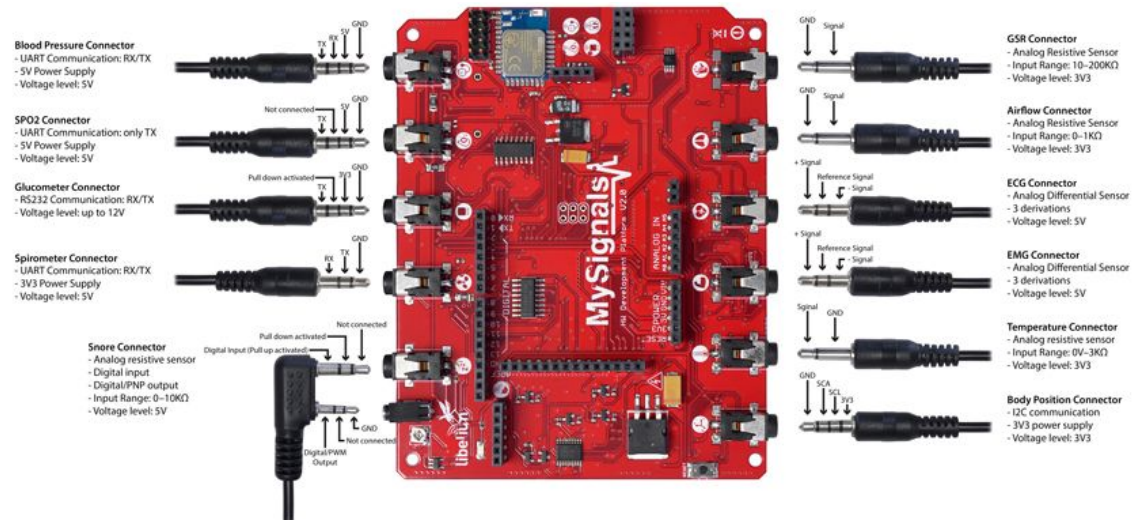


Ilustración 3.6. Placa MySignals con los sensores

Finalmente, para facilitar la toma de decisión posterior, se crea la siguiente tabla resumiendo las ventajas y los inconvenientes encontrados a esta alternativa.

Ventajas	Inconvenientes
- Gran variedad de sensores - Certificaciones (FDA / CE / IEC)	- Arduino 1 - No se pueden utilizar todos los sensores al mismo tiempo. Realizar combinación

Tabla 3.1. Ventajas e inconvenientes de MySignals

3.4.1. HealthyPi v3

HealthyPi es un monitor de signos vitales con todas las funciones, completamente de código abierto. Usando el Raspberry Pi como su plataforma de computación y visualización, el HAT complementario HealthyPi convierte el Raspberry Pi en un sistema de monitoreo de signos vitales ^[31].

En cuanto a características técnicas, destacaremos las siguientes:

- Microcontrolador principal: Atmel ATSAM21 ARM Cortex M0 - compatible con Arduino Zero
- Programabilidad: Arduino Zero Bootloader precargado
- Interfaces: cabezal Raspberry Pi de 40 pines (pines UART conectados), dispositivo USB CDC, conector UART adicional para conectar un módulo de presión arterial externo
- Dimensiones (solo placa): 65 mm x 56.5 mm x 6 mm
- Peso: 100 g

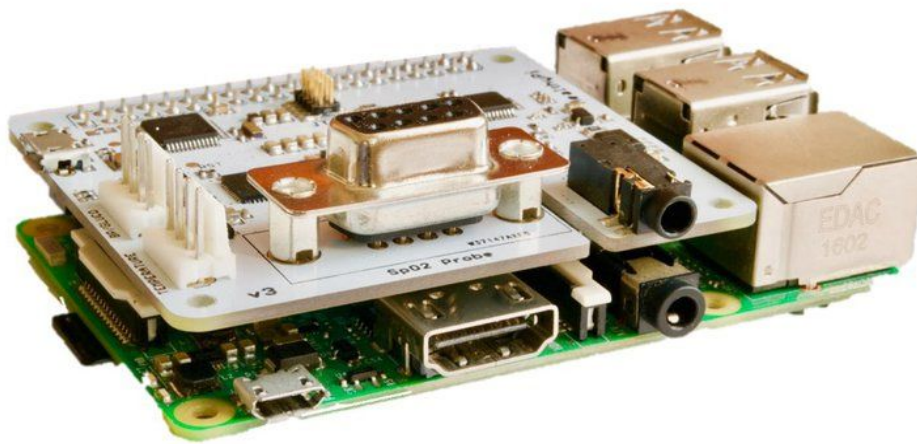


Ilustración 3.7. Placa HealthyPi parte delantera

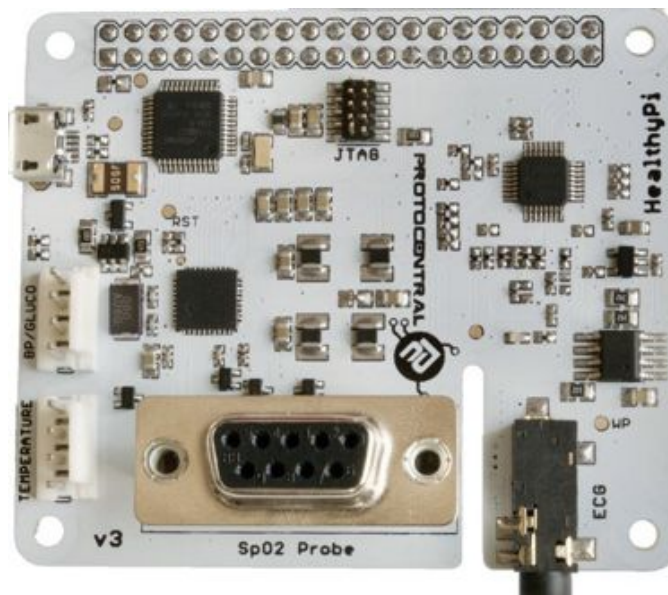


Ilustración 3.8. Placa HealthyPi parte trasera

Esta placa, está limitada a los siguientes sensores:

- ECG y respiración: utiliza la interfaz analógica TI ADS1292R de 24 bits con SNR de 107 dB
- Pulsioximetría : utiliza el front end TF-Oximetry TI AFE4400 con controlador LED integrado y ADC de 22 bits
- Temperatura: utiliza el sensor de temperatura corporal digital Maxim MAX30205 para la detección de la temperatura de la piel

El precio de todo el kit Healthy Pi3 HAT asciende a 249 €. Este kit incluye:

- PCB HealthyPi v3
- Cable de tres electrodos con conectores "de botón" en un extremo y conector estéreo en el otro extremo
- Oxímetro: Sonda Spo2 de clip para el dedo con conector DB9 "compatible con Nellcor"
- Sensor de temperatura de la piel digital montado en el cable con contactos de acero inoxidable.
- Una hoja de 20 Electrodo de ECG desechables de un solo uso (Covidien Medi-Trace)
- Kit de montaje HAT

Por lo que sería necesario adquirir también la raspberry Pi ya que no se encuentra incluida.

Al igual que en la alternativa anterior, se crea la siguiente tabla resumiendo las ventajas y los inconvenientes encontrados en esta solución.

Ventajas	Inconvenientes
• Raspberry • Código abierto • Barato	• Menos variedad de sensores • Sin certificaciones (FDA / CE / IEC) • Proyecto en proceso

Tabla 3.2. Ventajas e inconvenientes placa healthyPI

Analizadas las dos soluciones, se opta por utilizar la placa MySignals para la recogida de las variables fisiológicas. Se toma esta decisión debido a que proporciona una mayor fiabilidad con respecto a healthyPI, ya que dispone de certificaciones, y esta última se encuentra aún en proyecto y en estado de crowdfunding. Además, se prefiere MySignals por la gran variedad de sensores que ofrece los cuales se podrán ir añadiendo en versiones posteriores del proyecto.

4. DESARROLLO

En este apartado, se detallarán las tareas que se han realizado en las distintas iteraciones del proyecto.

4.1. Primera iteración

Tras el análisis inicial del proyecto, definición de las tareas, requisitos y todo lo visto anteriormente, se comienza el desarrollo del proyecto. En esta primera iteración, se definirá el diseño base de datos junto con su posterior configuración, se configurará el servidor y se realizará la programación del sistema de recogida de datos.

El producto resultante de esta iteración será el sistema de recogida de datos programado para realizar la obtención de datos del paciente y su almacenamiento en el servidor.

4.1.1. *Configuración del servidor*

La primera tarea que se realizará, será la preparación del servidor donde se alojará toda la lógica y los datos del proyecto.

El departamento de informática de la Universidad de Jaén dispone de un servidor de virtualización (kefren.ujaen.es) con máquinas virtuales para los estudiantes que lo necesiten, por ello, se solicita acceso a una máquina virtual donde poder configurar el servidor y todo lo necesario para el correcto funcionamiento del sistema.

También se ha estudiado la posibilidad de albergar el servidor en las instancias ofrecidas por Amazon Web Services, debido a la gran variedad de servicios de los que dispone, a su alta fiabilidad y flexibilidad. Finalmente se opta por la opción del servidor de la universidad debido a que nos ofrece los suficientes

recursos para garantizar el correcto funcionamiento del sistema, además de la reducción de costes que esto supone.

Las características de la máquina obtenida son las siguientes:

- Sistema operativo Ubuntu 16.04 LTS 64bits
- 2 núcleos
- 2Gb RAM
- 100Gb de disco duro

El acceso a la máquina se realizará con PuTTY mediante el protocolo SSH, para garantizar que todas las comunicaciones hacia y desde el servidor remoto sucedan de manera encriptada.

Se instalará un servidor Tomcat escuchando en uno de los puertos para el servicio de la aplicación del servidor y un Apache 2 en otro puerto para el acceso la página web. Además, se instalará MySQL para gestión de la base de datos.

4.1.2. Diseño y configuración de base de datos

Una vez conocidas las historias de usuario y las distintas interacciones con el sistema, se diseña la estructura lógica de la base de datos para su posterior implementación.

Se intentará simplificar lo máximo posible la base de datos, por que únicamente serán necesarias cuatro tablas: una para almacenar los usuarios, otra para las sesiones, otra para el registro de los datos recogidos por los sensores y la última para el almacenamiento de los datos subjetivos.

A continuación se observa el modelado de datos realizado y una pequeña descripción de cada uno de los atributos.

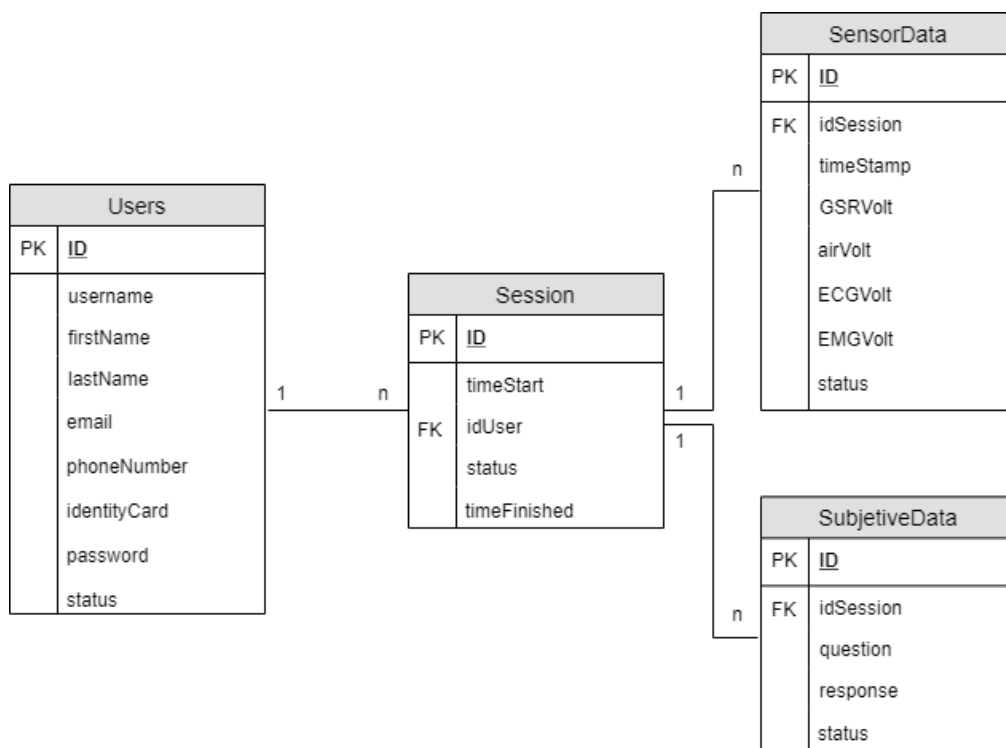


Ilustración 4.1. Modelo de datos

Users (usuarios del sistema)

- ID: identificador único de usuario
- username: nombre de usuario, utilizado para hacer login en la aplicación
- firstName: nombre del usuario
- lastName: apellido del usuario
- email: email del usuario
- phoneNumber: número de teléfono del usuario
- identityCard: número de identidad del usuario
- password: contraseña para acceder a la aplicación
- status: estado del usuario (0 - inactivo, 1 - activo, 2 - usuario administrador)

Session (sesiones realizadas)

- ID: identificador único de sesión
- timeStart: fecha y hora de comienzo de la sesión
- idUser: id de usuario que realiza la sesión
- status: estado de la sesión (0 - pendiente de comenzar, 1 - recogiendo datos en reposo, 2 - recogiendo datos frente al espejo, 3 - sesión finalizada)
- timeFinished: fecha y hora de finalización de la sesión

SensorData (datos recogidos por los sensores)

- ID: identificador único del registro de los datos
- idSession: id de la sesión
- timeStamp: instante en el que se han recogido los datos
- GSRVolt: valor recogido del sensor de respuesta galvánica medido en voltios
- airVolt: valor recogido del sensor de airflow medido en voltios
- ECGVolt: valor recogido del electrocardiograma medido en voltios
- EMGVolt: valor recogido del electromiograma medido en voltios
- status: estado de la sesión (1 - recogiendo datos en reposo, 2 - recogiendo datos frente al espejo)

SubjectiveData (datos subjetivos recogidos por la aplicación móvil)

- ID: identificador único del registro de datos
- idSession: id de la sesión
- question: pregunta que se le ha hecho al usuario
- response: respuesta del usuario (valor entre 0 - 100)
- status: estado de la sesión (1 - recogiendo datos en reposo, 2 - recogiendo datos frente al espejo)

Realizado el modelo de datos y toda la estructura lógica, se da paso a su creación en el servidor que hemos configurado. Se creará una base de datos relacional MySQL y se establecerá un túnel para poder acceder a ella desde el equipo de desarrollo mediante HeidiSQL. Con el uso de este software, se facilitará la creación de las distintas tablas, así como el acceso a todos los datos almacenados.

Desde la aplicación del servidor también es necesario realizar la conexión a la base de datos. Para esto, se configurará Hibernate en el proyecto, un framework de mapeo objeto-relacional para la plataforma Java que, además, facilita el mapeo de atributos entre nuestra base de datos relacional y el modelo de objetos de la aplicación. Para ello, basta con realizar anotaciones en los beans de las entidades y cumplimentar el fichero de configuración.

4.1.3. Desarrollo del sistema de recogida de datos fisiológicos

En este apartado, se presentará el sistema de recogida de datos y veremos como se ha llevado a cabo su desarrollo.

Como hemos visto en otros capítulos, los datos fisiológicos del paciente se recogerán a través de distintos sensores, a continuación se se presentarán los utilizados en este proyecto.

Airflow: El sensor de flujo de aire nasal/oral es un dispositivo utilizado para medir la frecuencia respiratoria de una persona, es decir, el número de respiraciones por minuto. Este dispositivo consta de un hilo flexible que se ajusta detrás de las orejas y un conjunto de dos puntas que se colocan en las fosas nasales.

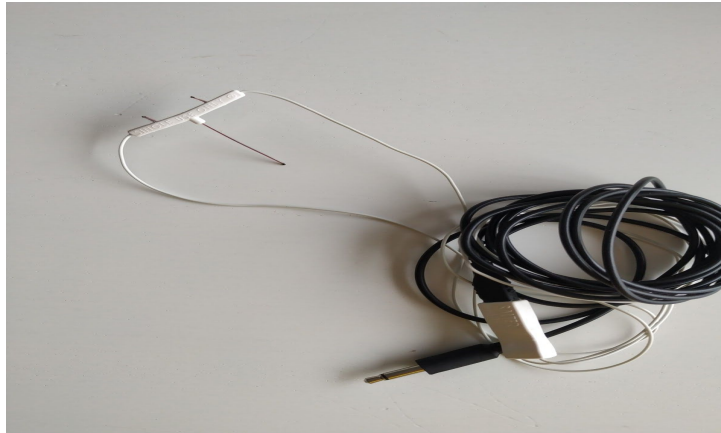


Ilustración 4.2. Sensor airflow

Conductividad de la piel: Este sensor mide la conductancia eléctrica de la piel, que varía con su nivel de humedad.

Esta medida, podría ser interesante ya que las glándulas sudoríparas están controladas por el sistema nervioso simpático, por lo que los momentos de emoción fuerte cambian la resistencia eléctrica de la piel.



Ilustración 4.3. Sensor de conductividad de la piel

Electrocardiograma: El electrocardiograma (ECG) es una herramienta de diagnóstico que se usa habitualmente para evaluar las funciones eléctricas y musculares del corazón. El sensor utiliza el "electrocardiograma de telemetría continua" para una monitorización prolongada que incluye el uso de tres electrodos.



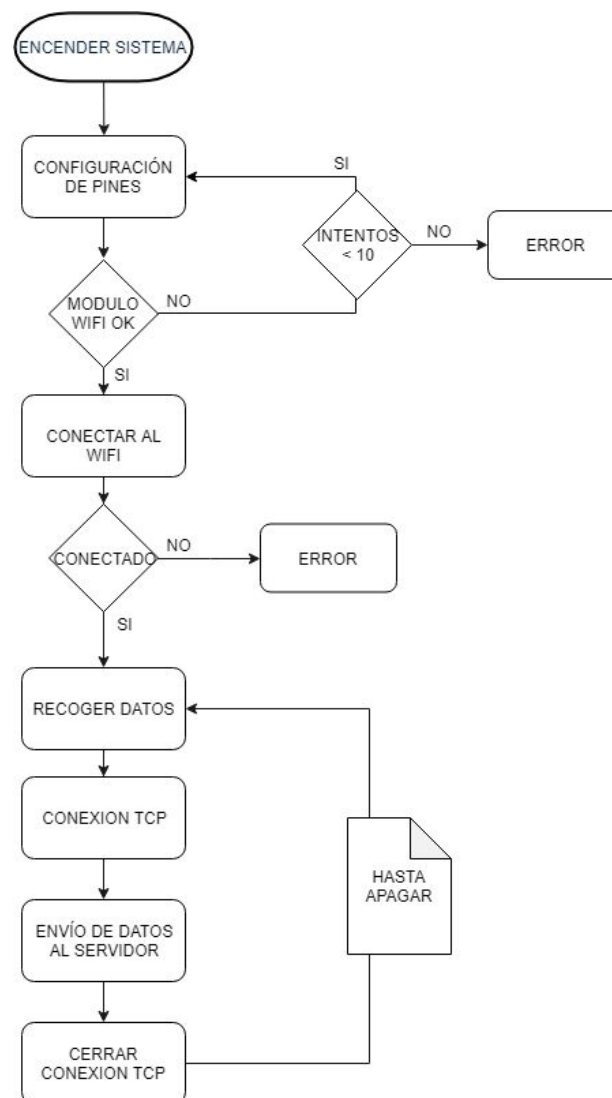
Ilustración 4.4. Electrocardiograma

Electromiograma: Un electromiógrafo detecta el potencial eléctrico generado por las células musculares cuando éstas se activan eléctrica o neurológicamente. Las señales se pueden analizar para detectar anomalías médicas, nivel de activación, orden de reclutamiento o para analizar la biomecánica del movimiento humano o animal.



Ilustración 4.5. Electromiograma

Estos sensores se conectarán a la placa MySignals, que controlada mediante un Arduino, recogerá los datos del paciente y se enviarán vía WiFi al servidor. En la ilustración 4.6. se puede observar un diagrama de flujo en el que se representa la lógica que se ha desarrollado en la placa.

**Ilustración 4.6. Diagrama de flujo implementado**

Como podemos ver, en primer lugar se realiza la activación de la placa junto con la configuración de los pines para poder activar el módulo WiFi integrado. Después se conecta a la red WiFi, para ello hay que indicar el SSID y la contraseña en código, siendo conscientes del problema que supone esto, se anotará como mejora para siguientes prototipos.

Posteriormente se realiza la recolección de datos en ese mismo instante de los distintos sensores, se utilizarán las librerías disponibles de MySignals que facilitan enormemente este trabajo.

Finalmente, una vez tenemos los datos, se envían al servidor mediante la API desarrollada. Es necesario establecer una conexión TCP con el servidor, indicar la cantidad de bytes que se va a enviar, realizar la petición HTTP y cerrar la conexión TCP. Ésta ha sido la parte más compleja de todo el desarrollo, ya que la comunicación con el módulo WiFi únicamente se permite hacerla mediante comandos AT, un lenguaje a muy bajo nivel. Además, tener que realizar este ciclo para el envío de datos cada vez que se recogen, provoca una limitación en la tasa de muestreo de los datos que se establece a dos muestras por segundo.

En las siguientes imágenes, se presenta la placa MySignals.

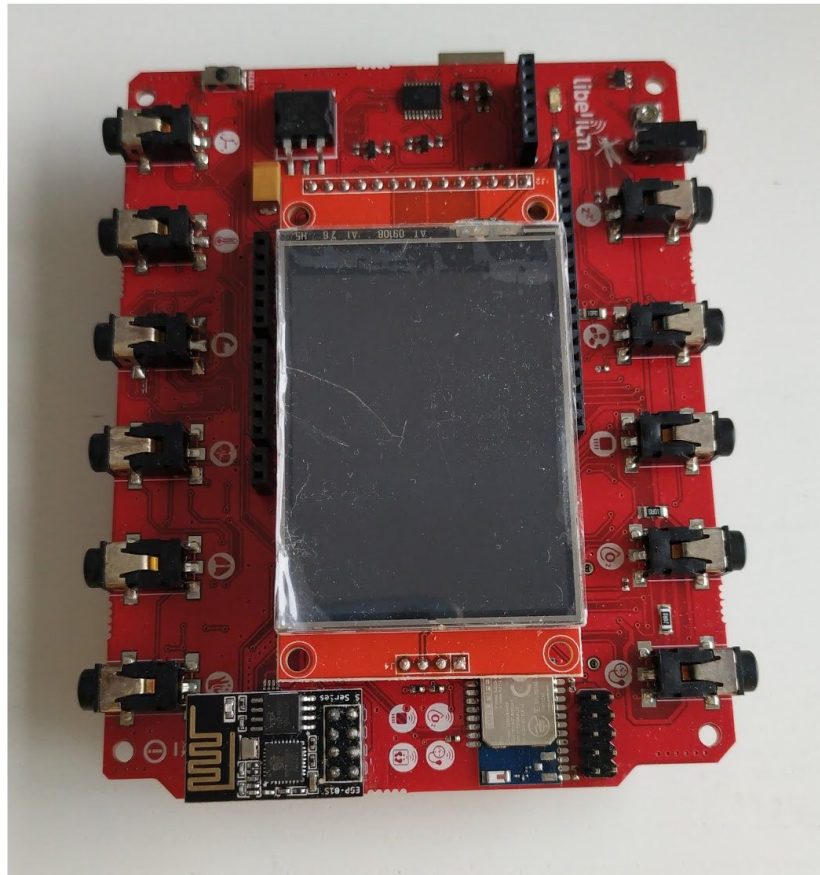


Ilustración 4.7. Placa MySignals obtenida

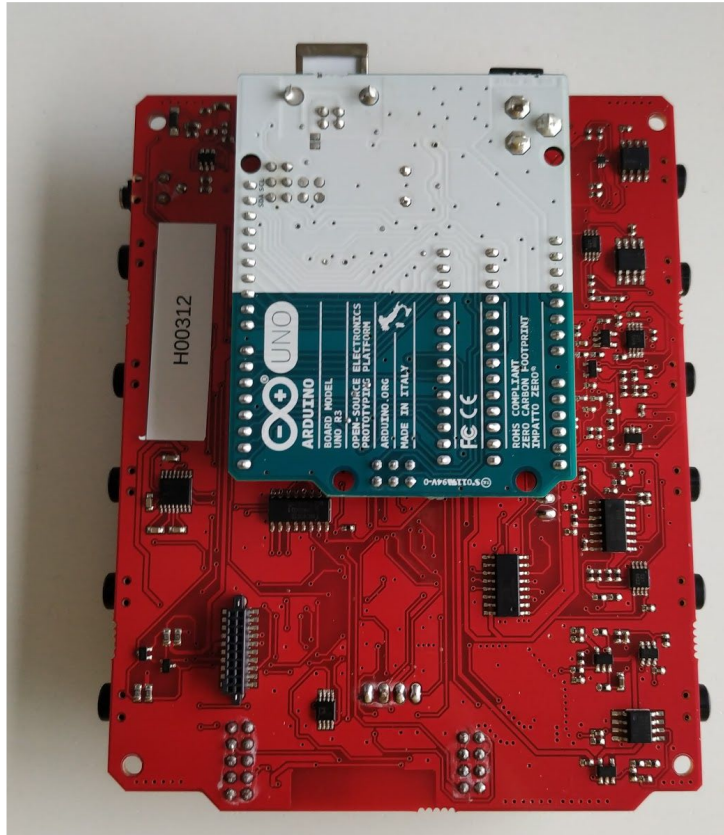


Ilustración 4.8. Placa MySignals junto con Arduino

Como se puede apreciar en las imágenes anteriores, la utilización de la placa tal cual está, puede suponer un problema para los usuarios, ya que es poco manejable y el aspecto puede resultar intimidante. Para solventar esto, se decide imprimirle una carcasa utilizando una impresora 3D.

La impresión de la carcasa se realizará totalmente a medida, por ello, en primer lugar se toman medidas de la placa y se modela usando 3DBuilder. Como podemos ver en la imagen 4.9. se realizan distintas impresiones del modelo para validarlo y ajustar las mejoras necesarias hasta que se da con el diseño exacto.

Se utilizará la impresora 3D de la que dispone el grupo de investigación de Informática Gráfica y Geomática de la Universidad de Jaén. Considerada una de las mejores impresoras del mercado, se trata de una impresora Prusa I3 MK3S. Posee un espacio de construcción de 250 x 210 x 200 mm, cama de calor magnética

MK52 con lámina de impresión reemplazable de acero y PEI y recuperación de energía. El extrusor ha sido mejorado con respecto a la versión anterior, una gran cantidad de sensores y otras mejoras han sido añadidos.

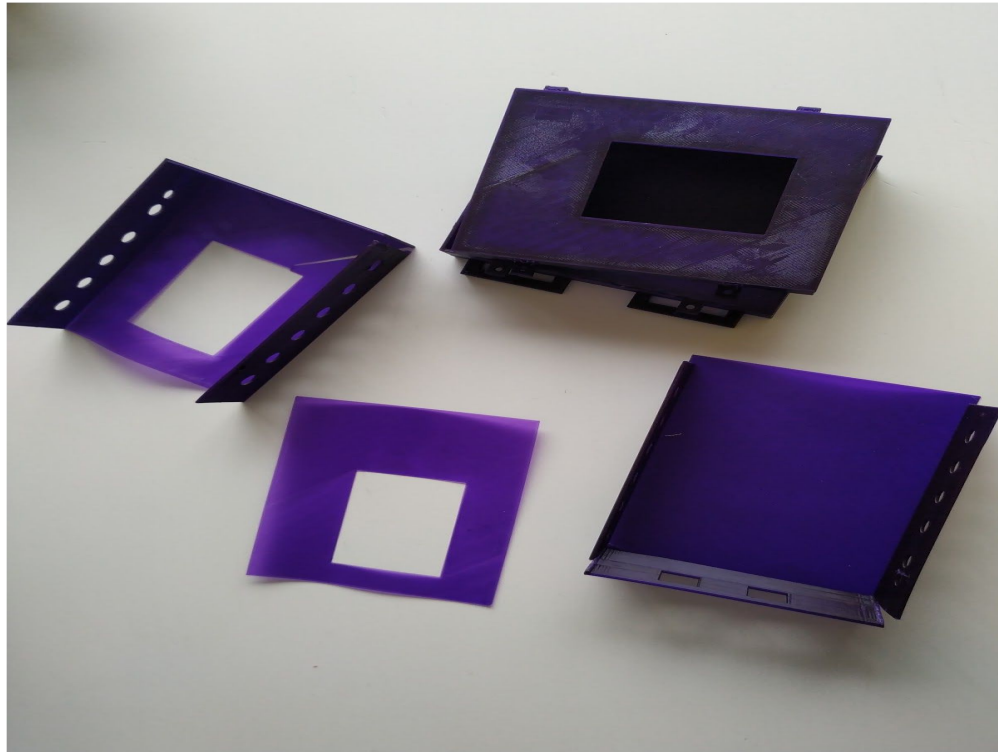


Ilustración 4.9. Pruebas de carcasa

A continuación podemos ver desde distintas perspectivas la impresión de la carcasa. En la Ilustración 4.10. se puede ver la impresión en directo a través de una página web junto con información relativa al proceso de impresión, gracias a la cámara que se ha incorporado en la impresora, mientras que en la ilustración 4.11. se puede ver la impresora durante el proceso de impresión..

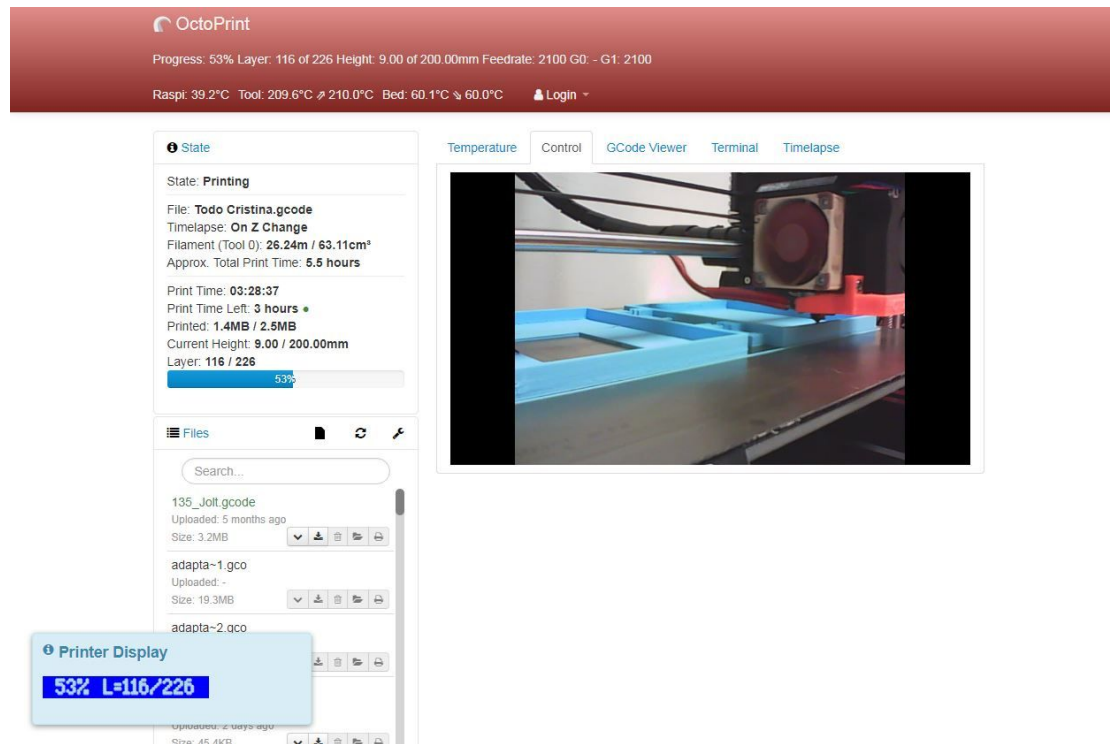


Ilustración 4.10. Visualización de la impresión

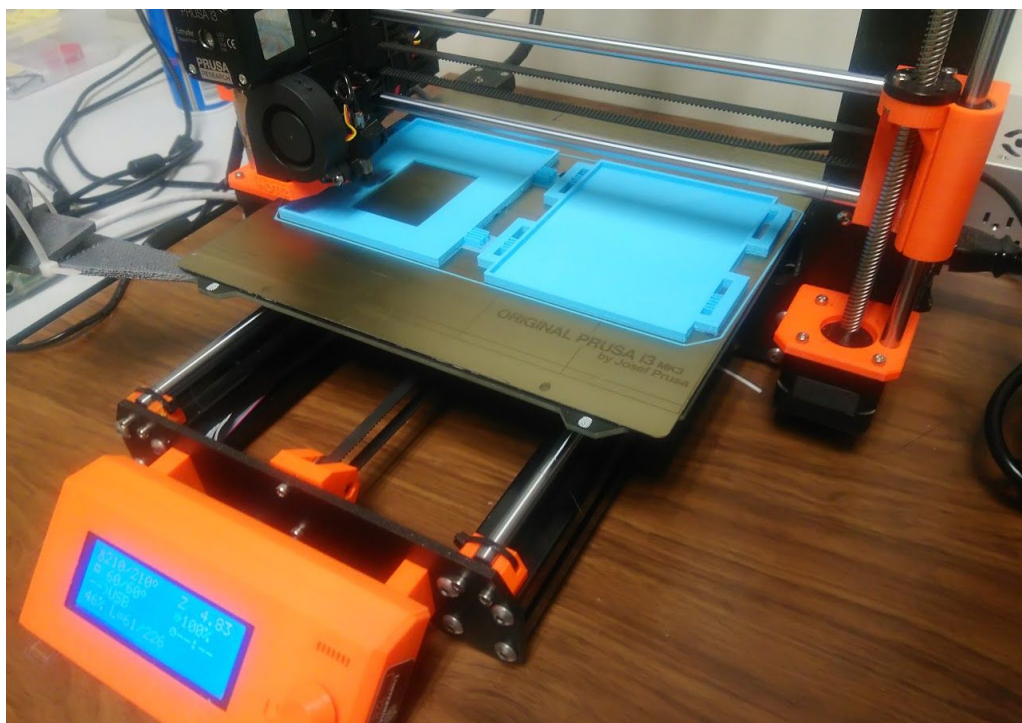


Ilustración 4.11. Impresión de la carcasa

El resultado final del sistema encargado de la recogida de datos fisiológicos de los pacientes se puede observar en la siguiente imagen.



Ilustración 4.12. Sistema de recogida de datos

4.2. Segunda iteración

En la segunda iteración, nos centraremos en el la aplicación Android, lo que supone tareas de diseño y desarrollo.

Como resultado de esta iteración se espera una aplicación móvil funcional conforme al diseño previsto.

4.2.1. Diseño de la aplicación Android

El diseño de la aplicación móvil se realiza junto a los terapeutas. Tras las reuniones iniciales donde se detallan los requisitos, se lleva a cabo una propuesta de diseño que será expuesta a los especialistas para realizar los cambios que consideren oportunos y den su validación antes de comenzar el desarrollo.

Se crean las pantallas de la aplicación y para emular la navegación entre ellas se utilizará MarvelApp.

La primera pantalla será el login de la aplicación. El paciente debe rellenar el nombre de usuario y la contraseña que le le facilitarán los terapeutas antes de comenzar con la sesión.



Ilustración 4.13. Diseño aplicación - login

Una vez el paciente ha iniciado sesión en la aplicación, se mostrará la pantalla de bienvenida, desde donde se podrá comenzar una sesión o en caso necesario ver el manual de ayuda donde podrá encontrar las instrucciones para el uso del sistema de recogida de datos y colocación de los sensores, así como los pasos necesarios para realizar la sesión de manera correcta.



Ilustración 4.14. Diseño aplicación - bienvenida

Al comenzar la sesión, el primer paso es recoger medidas subjetivas para registrar el estado del paciente antes de la exposición. Estos datos se recogerán mediante preguntas establecidas por los especialistas y mediante un seekBar el usuario podrá seleccionar su respuesta.



Ilustración 4.15. Diseño aplicación - recogida de datos subjetivos

Tras contestar a las preguntas establecidas, se presenta la pantalla previa a la recogida de datos en reposo, donde se le indica a la paciente que comenzará la recogida y qué debe de hacer.



Ilustración 4.16. Diseño aplicación - comienzo en reposo

Comienza la recogida de datos previa a la exposición, durante el periodo determinado por los terapeutas, en este caso, cinco minutos.



Ilustración 4.17. Diseño aplicación - recogida de datos en reposo

Una vez recogidos los datos durante el tiempo determinado, aparecerá la pantalla que guía al paciente para comenzar la recogida de datos frente al espejo



Ilustración 4.18. Diseño aplicación - comienzo frente al espejo

A continuación, se muestra una pantalla informativa de que se están recogiendo los datos de los sensores. Tras pasar un tiempo determinado por los terapeutas, aparecerá un botón para finalizar la sesión cuando el usuario considere necesario.



Ilustración 4.19. Diseño aplicación - recogida de datos frente al espejo

Al pulsar sobre finalizar la sesión, se volverá a abrir la pantalla donde se muestran las preguntas (ilustración 4.15.) para volver a recoger los datos subjetivos tras la realización de la terapia. Una vez contestadas todas las preguntas, se mostrará un mensaje informando de la finalización de la sesión.

El resultado final del diseño de la aplicación móvil con las interacciones, se puede observar en el siguiente enlace: [Diseño aplicación](#)

4.2.1. Desarrollo aplicación Android

La aplicación móvil se desarrollará para móviles Android y se utilizará el IDE Android Studio.

El principal componente en el desarrollo de una aplicación Android son las actividades. Una aplicación generalmente consiste en múltiples actividades vinculadas de forma flexible entre sí. Están conformadas por dos partes: la parte gráfica y la parte lógica.

La parte gráfica, se centra en la creación de las pantallas de la aplicación, donde se seguirá el diseño acordado en el apartado anterior. En el proyecto, se han llevado a cabo dos maneras de definir los elementos en la interfaz:

- Declaración de elementos de la interfaz de usuario en XML. Android proporciona un vocabulario XML simple que coincide con las clases y subclases de vistas, como las que se usan para widgets y diseños. También se ha utilizado la función Layout Editor, para crear algunos de los diseños XML ya que proporciona una interfaz de arrastrar y soltar, lo cual facilita mucho el desarrollo.
- Creación de una instancia de elementos de diseño durante el tiempo de ejecución. La aplicación creará algunos de los objetos y realizará la modificación de propiedades de forma programática.

La parte lógica es un archivo java que es la clase que se crea para poder manipular, interactuar y colocar el código de esa actividad. En esta clase, se crean todos los métodos necesarios para la realización de la lógica de la aplicación, la implementación de las llamadas necesarias a la API y la navegación entre las distintas pantallas.

De manera paralela al desarrollo de la aplicación móvil, también se va desarrollando la API y creando los web services necesarios en la aplicación del servidor para la ejecución de la lógica relacionada.

En la aplicación del servidor, se creará toda la gestión y ciclo de vida del estado de sesiones de usuario, esto es, creación de la sesión, estado “recogiendo valores en reposo”, estado “recogiendo valores frente al espejo” y la finalización de la sesión. Otro punto destacable que se añade en esta iteración, son los web services relacionados con la obtención y almacenamiento de los datos subjetivos,

estos son las preguntas que se hacen al usuario y el almacenamiento de sus respuestas.

La comunicación con la API se realizará haciendo uso de la librería Retrofit, es un cliente REST muy simple y fácil de aprender. Permite hacer peticiones GET, POST, PUT, DELETE y HEAD; gestionar diferentes tipos de parámetros y parsear automáticamente la respuesta a un POJO (Plain Old Java Object) ^[33].

La documentación de la API REST desarrollada se encuentra en el Anexo IV. Documentación API REST.

A continuación, podemos observar el diagrama de secuencia donde se detalla en qué momentos la aplicación se comunica con la aplicación servidor y lo que se desencadena. El manual de usuario de la aplicación móvil se puede revisar en el Anexo II. Manual de uso de la aplicación móvil.

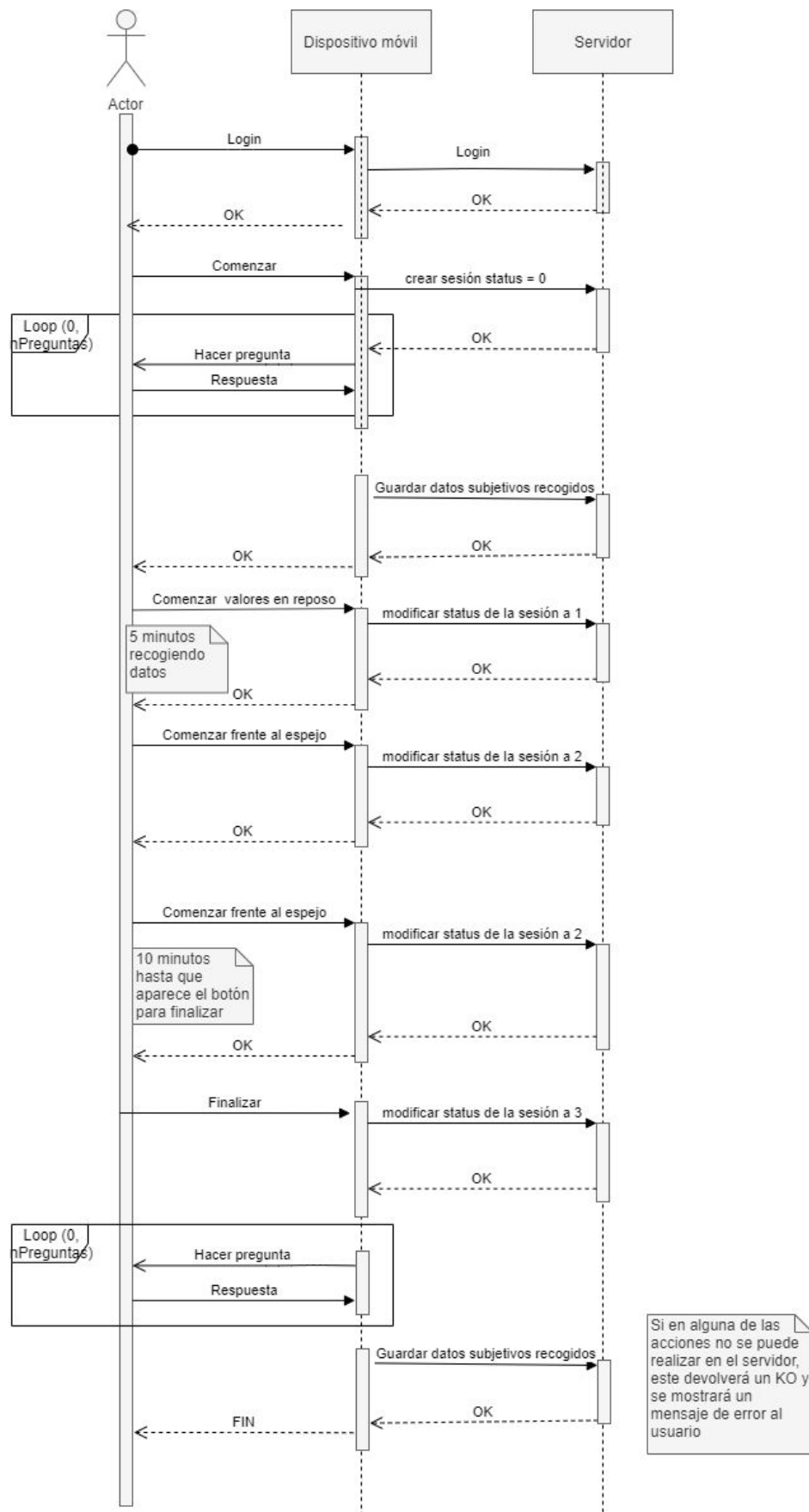


Ilustración 4.20. Diagrama de secuencia

4.3. Tercera iteración

Durante el desarrollo de esta iteración, se realizan las tareas relacionadas con el diseño y desarrollo de la aplicación web utilizada por los terapeutas.

Finalizada esta iteración, se obtendrá la página web accesible, funcional y lista para ser utilizada por los terapeutas.

4.3.1. Diseño de la aplicación web

El diseño de la aplicación web se realizará mediante bocetos en papel. Esta es una técnica muy sencilla y muy eficaz. Se puede preparar un diseño muy básico de las páginas y la estructura general de manera rápida y sencilla, sobre el cual se pueden realizar todas las mejoras que se consideren oportunas.

A continuación se muestran los esbozos realizados en papel junto con una explicación de cada una de las pantallas.

En primer lugar, es necesario establecer una vista para realizar el login en la página web. El terapeuta debe rellenar su email y contraseña y pulsar sobre LOGIN.

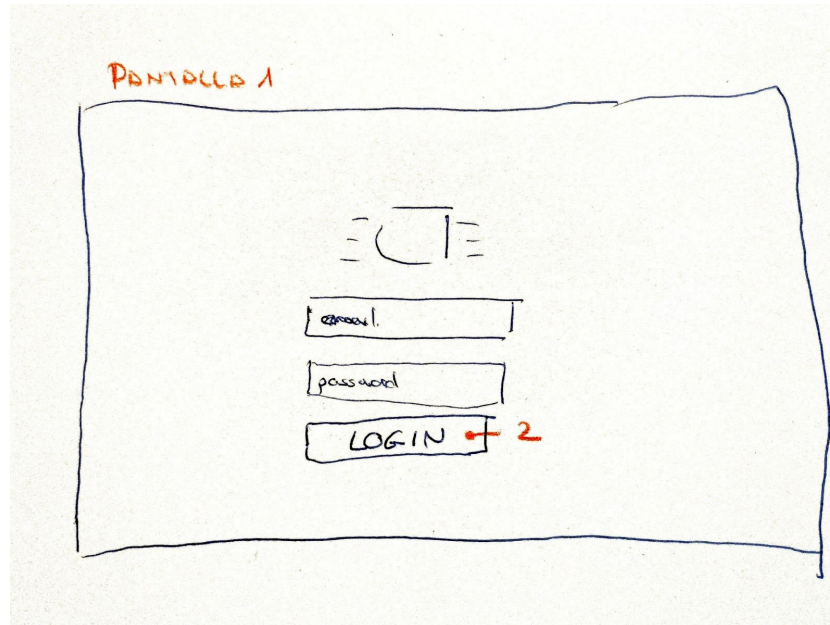


Ilustración 4.21. Diseño web - login

Al iniciar sesión, entra en el listado donde se muestran las sesiones que se han realizado, indicando quien la ha hecho, cuando comenzó, cuando finalizó y el estado en el que se encuentra la sesión. En la parte superior de la pantalla, se encontrará el menú, que en este caso tendrá las opciones “sesiones” y “usuarios”.

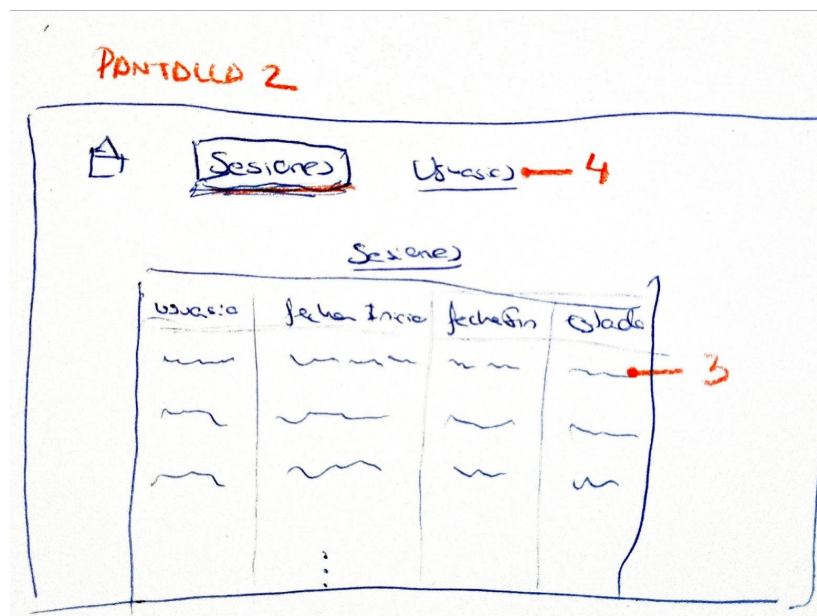


Ilustración 4.22. Diseño web sesiones

Para ver el detalle y todos los datos que se han recogido de una sesión, se pulsará sobre ella y se mostrará la siguiente pantalla. En ella se podrán diferenciar dos apartados: los datos recogidos antes de comenzar la sesión y los datos recogidos durante la sesión. En cada uno de estos apartados, se detallarán las preguntas que se le han hecho al paciente junto con el valor que ha contestado y una tabla con los datos recogidos por cada uno de los sensores y el instante de tiempo en el que se han recogido.

PANTALLA 3

Sesiones / Usuario — 4

Detalle

EN REPOSO

¿Cómo te sientes? * 50 **

~ ~ ~ ~

Tiempo	Sensor 1	Sensor 2	Sen. 3
13/05/18 13:40	5'01	0'05	2'13
~	~	~	~
~	~	~	~

EN SESIÓN

¿Cómo te sientes? * 50 **

~ ~ ~ ~

Tiempo	Sensor 1	Sensor 2	Sen. 3
13/05/18 15:15	2'01	3'44	0'02
~	~	~	~
~	~	~	~

ATRAS — 2

* Pregunta Datos subjetivos
** Respuesta

Ilustración 4.23. Diseño web - detalle sesión

Al listado de usuarios se puede llegar desde el menú superior. En esta pantalla se puede observar el listado de los usuarios que existen en el sistema indicando su nombre, apellidos, email y el estado en el que se encuentra.

Nombre	Apellido	email	Estado
Cristina	Lechuga	de@..	Activo
~	~	~	~
~	~	~	~
~	~	~	~

Ilustración 4.24. Diseño web - usuarios

El diseño de la pantalla para crear o modificar un usuario será el mismo. Se diferenciarán en que para un nuevo usuario todos los campos se encontrarán vacíos mientras que para la modificación de uno ya existente estos campos vendrán prerellenos con la información almacenada del usuario en cuestión. Se mostrará el nombre del campo junto con una caja de texto para rellenar.

PANTALLA 5

2 -> Sesiones | Usuarios

Nuevo usuario // modificar usuario

Nombre: [] Apellido: []

DNI: [] + []

mes: []

Guardar

Ilustración 4.25 . Diseño web - detalle usuario

Los diseños en papel vistos anteriormente se corresponden con las vistas de la aplicación web. Tras la validación con el cliente y teniendo en cuenta las modificaciones y mejoras propuestas, se procede a su desarrollo.

4.3.2. Desarrollo de la aplicación web

El proceso de creación del proyecto se agilizará utilizando el generador de proyectos Yeoman. La creación se realiza mediante una plantilla que ya incluye la librería Bootstrap, Angular JS, y un servidor Grunt para el arranque del proyecto en local.

Se seguirá el patrón Modelo-Vista-Controller, es un patrón de diseño de software para el desarrollo de aplicaciones web, el cual se compone de las siguientes tres partes:

- Modelo: es el nivel más bajo del patrón, es responsable de mantener y administrar los datos de la aplicación. Responde a la solicitud desde la vista y a las instrucciones del controlador para actualizarse.
- Vista: es responsable de mostrar todos o una parte de los datos al usuario, desencadenada por la decisión del controlador de presentar los datos. Las vistas de la aplicación se realizarán en HTML y CSS.
- Controlador: es el encargado de controlar las interacciones entre el modelo y la vista. El controlador responde a la entrada del usuario y realiza interacciones en los objetos del modelo de datos. Recibe información, la valida y luego realiza las operaciones necesarias que pueden modificar el estado del modelo de datos.

El controlador, se encargará de la conexión con la API del servidor, por lo que, de manera paralela al desarrollo web, también es necesario desarrollar los web services de los que hará uso la aplicación. Estos web services son los relativos a

obtener los listados tanto de usuarios como el de sesiones, a la creación y modificación de usuarios y a la obtención del detalle de una sesión concreta.

La documentación de la API REST desarrollada se encuentra en el Anexo IV.
Documentación API REST.

El manual de usuario de la aplicación web, puede consultarse en el Anexo III.
Manual de uso de la aplicación web

4.3. Cuarta iteración

En esta última iteración, se completará el desarrollo de la página web con funcionalidades más específicas que no se habían añadido en anteriores iteraciones. Además, se realizarán pruebas en todas las aplicaciones, y pequeñas mejoras de cara a la entrega final del proyecto.

El resultado de esta iteración será el sistema completo preparado para su entrega al cliente.

A continuación, se explican las funcionalidades incorporadas a la aplicación web:

- **Descarga CSV:** En la pantalla de la página web donde el terapeuta puede ver los datos recogidos por los sensores, se habilitará un botón para poder realizar una descarga de un fichero CSV con todos los datos recogidos por los sensores. De esta manera, los terapeutas podrán exportar los datos a la plataforma que utilizan actualmente para realizar análisis de las variables fisiológicas registradas durante la terapia.
- **Análisis de una sesión:** Se añade en la vista donde se encuentran los datos recogidos en la sesión, un análisis básico de los valores recogidos. Se incorpora una gráfica de barras donde se puede observar de manera clara una la comparativa de los valores medios registrados cuando el paciente se encontraba en reposo y durante la exposición frente al espejo, por cada uno de los distintos sensores. En la siguiente imagen, se puede observar la gráfica que se obtiene en una sesión.



Ilustración 4.25 . Gráfica con las medias de los datos recogidos

En cuanto a las pruebas, primero se realizarán a cada uno de los aplicativos. En primer lugar, se comprueba que el sistema de recogida de datos funciona correctamente y recoge los datos de los cuatro sensores una vez están conectados al cuerpo, además se revisa que los datos se envían al servidor y se almacenan en la base de datos.

A continuación se realiza un testeo de la aplicación móvil, este consiste en completar una terapia completa y comprobar que todo funciona de manera robusta. Comprobado que en el caso correcto funciona bien, también se realizan pruebas en las que se esperan mensajes de error y comportamientos específicos.

También se realizarán pruebas a la aplicación web, donde se comprobarán todas las funcionalidades que esta ofrece. Se comprueba que se visualizan correctamente los listados, que se pueden modificar y añadir nuevos usuarios, así como que se muestra el detalle de una sesión de manera correcta, además de todos los mensajes de error y distintas casuísticas que se pueden dar.

Para completar las pruebas, se realiza mediante la herramienta Postman (apartado 2.6.6) peticiones contra la API del servidor, para revisar que todos los web services funcionan conforme a lo esperado.

Para finalizar las pruebas, se realizan flujos completos, probando a conectar todos los sensores e iniciar la terapia con la aplicación móvil. Una vez finalizada la terapia, se accede a la página web y se comprueba que se han registrado correctamente los datos de la sesión, se pueden descargar y se visualiza la gráfica de manera correcta, garantizando el buen funcionamiento de todo el sistema.

5. CONCLUSIONES Y TRABAJO FUTURO

Una vez finalizado este Trabajo Final de Máster, es momento de realizar una retrospectiva del trabajo realizado. En este apartado, se presentan las conclusiones a las que se ha llegado y propuestas para la continuación de este proyecto.

4.3. Conclusiones

Durante la realización del proyecto, me he enfrentado a trabajar de manera autónoma en un proyecto compuesto por varios aplicativos, suponiendo la programación en distintos lenguajes y paradigmas, además de haber avanzado y profundizado en el aprendizaje de distintas metodologías y aspectos clave en la ejecución de un proyecto, así como la adquisición de experiencia a nivel tanto profesional como personal.

A continuación, se reflexiona respecto a los objetivos planteados al inicio del proyecto, afirmando que la mayoría de ellos han sido alcanzados satisfactoriamente.

En cuanto al estudio de las técnicas de exposición frente al espejo he de decir que no abunda el uso de nuevas tecnologías en dichas técnicas. Me resulta realmente interesante que, la incorporación del sistema propuesto en este trabajo pueda suponer una nueva línea de investigación para evaluar la implicación de las respuestas fisiológicas durante la evolución de la terapia, además de permitir la realización de estas terapias en un entorno menos hostil para el paciente.

Para la implementación del sistema completo, ha sido necesario el estudio de tecnologías como Arduino, Android y tecnologías de desarrollo web. A pesar de tener ciertos conocimientos en desarrollo android y web, se ha profundizado en conceptos y se ha adquirido experiencia en los lenguajes utilizados. Por otro lado, Arduino era prácticamente desconocido para mi y ha supuesto uno de los retos y desarrollos más complejos de este proyecto.

Referente al objetivo de montar y desarrollar un sistema encargado de la

recogida de distintas variables fisiológicas, como he indicado anteriormente, ha sido uno de los desarrollos más complejos. Finalmente el sistema recoge las variables fisiológicas y las envía al servidor tal cual se había marcado en los objetivos, pero realmente sería necesario un estudio en profundidad para la validación de los valores que recogen los sensores y analizar la fiabilidad del sistema.

El diseño y desarrollo de la aplicación móvil se ha completado satisfactoriamente, contando con la validación por parte del cliente. Se contaba con los conocimientos básicos en el desarrollo de aplicaciones Android antes del comienzo de este trabajo, y gracias a la existencia de una extensa documentación sobre este lenguaje se han adquirido nuevos conocimientos y experiencia en este ámbito. En cuanto a la aplicación desarrollada, cumple los requisitos que se marcaron al inicio del proyecto. Se encarga de guiar al paciente durante la sesión y recoge medidas subjetivas mediante preguntas dirigidas al usuario, con una interfaz simple y sencilla.

En cuanto al diseño y desarrollo de la aplicación web para el estudio de los datos obtenidos, se ha conseguido este objetivo obteniendo como resultado una aplicación web que facilita la gestión de usuarios y la visualización de los datos recogidos durante las sesiones. En cuanto al análisis de los datos, es necesario determinar qué análisis y estadísticas pueden ser realmente de utilidad para los terapeutas, a pesar de esto, se incluye un pequeño análisis de los datos obtenidos.

Y, por último, a pesar de realizar testeos completos del sistema, aún no se han llevado a cabo pruebas con usuarios reales, por lo que, el objetivo referente al estudio de la usabilidad de las aplicaciones y la realización de pruebas de uso con pacientes no podrá ser constatado hasta su ejecución por parte de las personas a las que se destina.

En cuanto a experiencia personal conseguida, resaltar el trato con los investigadores encargados de la realización de las terapias frente al espejo, cuya

ayuda para este proyecto ha sido imprescindible, aportando conocimientos esenciales en mi etapa de investigación.

Para finalizar, la realización del Trabajo Fin de Máster ha sido más que satisfactoria, contando con la validación por parte del cliente y cumpliendo las expectativas marcadas. Espero que el sistema propuesto, a pesar de ser un prototipo, sirva como comienzo para su puesta en marcha y llegue a ser utilizado realmente en las terapias de exposición frente al espejo que se emplean en el tratamiento de los trastornos de conducta alimentaria.

4.3. Trabajos futuros

Una vez finalizado el desarrollo del primer prototipo del sistema y su entrega al cliente, a continuación, reflexionamos sobre las mejoras que se le deberían añadir para completar su desarrollo y obtener un sistema robusto.

En primer lugar, es necesario validar con los terapeutas los valores fisiológicos que toman los distintos sensores y comprobar que tienen la misma fiabilidad que los equipos de los que disponen actualmente en el laboratorio. Además, es necesario ajustar el muestreo de datos al indicado en los requisitos del sistema, y realizar un estudio en profundidad para comprobar si la placa obtenida presenta limitaciones en cuanto a la frecuencia de envío.

También sería necesario realizar pruebas de uso con usuarios reales, tanto con pacientes que estén realizando la terapia como con los terapeutas para comprobar la usabilidad de las aplicaciones y realizar los cambios oportunos.

Finalmente, como se ha indicado en las conclusiones, se debe realizar un estudio en profundidad acerca de los distintos tipos de análisis que se pueden realizar con los datos recogidos e implementarlos en la aplicación web para facilitar aún más el trabajo de los terapeutas.

Como ampliación del sistema propuesto, es muy interesante el hecho de poder utilizar la arquitectura y las aplicaciones para su uso en otro tipo de terapias cognitivo-conductuales en las que también es necesario la medida de los datos fisiológicos de los pacientes, por ejemplo, en terapias destinadas a tratar las fobias o tratamientos de vergüenza y autocrítica, en los que se también se puede realizar una exposición.

Para esto, y para poder convertir el sistema más flexible, se pueden parametrizar las medidas subjetivas y los tiempos de exposición, haciéndolos dinámicos y permitiendo a los terapeutas que los configuren para las distintas terapias o pacientes.

De esta manera, se podría conseguir un sistema de adquisición de respuestas fisiológicas para su uso en una gran variedad de tratamientos en los que sea de utilidad la medida de los valores fisiológicos del paciente. Se podrían añadir los sensores necesarios y automatizar los análisis de los datos para obtenerlos de la manera más cómoda y sencilla por parte de los terapeutas.

BIBLIOGRAFÍA

- [1]. Garner, David & Garfinkel, Paul. (1981). Body Image in Anorexia Nervosa: Measurement, Theory and Clinical Implications. *International journal of psychiatry in medicine*. 11. 263-284.
- [2]. Zaragoza Cortes Jessica , Saucedo-Molina Teresita de Jesús, Fernández Cortés Trinidad Lorena (2011). Asociación de impacto entre factores socioculturales, insatisfacción corporal, e índice de masa corporal en estudiantes universitarios de Hidalgo, México. *Archivos Latinoamericanos de Nutrición Vol. 61 N° 1*.
- [3]. Thomas E Cash, Patricia E. Henry (1995) Women's Body Images: The Results of a National Survey in the U.S.A. *Sex Roles*, Vol 33, Nos. 1/2 .
- [4]. Stice E, Marti CN, Durant S. (2011). Risk factors for onset of eating disorders: evidence of multiple risk pathways from an 8-year prospective study. *Behav. Res. Ther.* 49 622–627.
- [5]. Espina, Alberto., Ortego, María Asunción., Ochoa de Alda, Íñigo., Yenes, Félix., Alemán, Amagoia (2001). La imagen corporal en los trastornos alimentarios. *Psicothema*. 13(4), 532-538.
- [6]. Ana Rosa Sepúlveda, Juan Botella y José Antonio León (2001). La alteración de la imagen corporal en los trastornos de la alimentación: un meta-análisis. *Psicothema*. Vol. 13, nº 1,. 7-16.
- [7]. Berengüí, Rosendo, Castejón, M. Ángeles, & Torregrosa, M. Soledad. (2016). Insatisfacción corporal, conductas de riesgo para trastornos de la conducta alimentaria en universitarios. *Revista mexicana de trastornos alimentarios*, 7(1), 1-8.
- [8]. Silvia Moreno Domínguez y Sonia Rodríguez Ruiz. (2012) La insatisfacción corporal: Bases teóricas y mecanismos psicológicos implicados en el tratamiento basado en técnicas conductuales de exposición al espejo. *Mágina: Revista Universitaria*, N°. 16, págs. 47-60

- [9]. OVH Hispano, S.L.U, «OVH - VPS SSD» [En línea]. Available: <https://www.ovh.es/vps/vps-ssd.xml>. [Último acceso: 25 junio 2018].
- [10]. Boletín Oficial del Estado. «XVII Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública». [En línea]. Available: https://www.boe.es/diario_boe/txt.php?id=BOE-A-2018-3156 [Último acceso: 25 Julio 2019].
- [11]. The Apache Software Foundation. «Apache Tomcat». [En línea]. Available: <http://tomcat.apache.org/> [Último acceso: 6 junio 2019].
- [12]. The Apache Software Foundation. «The Apache HTTP Server Project». [En línea]. Available: <http://httpd.apache.org/> [Último acceso: 9 junio 2019].
- [13]. Oracle Corporation. «Oracle MySQL». [En línea]. Available: <https://www.oracle.com/es/mysql/> [Último acceso: 4 junio 2019].
- [14]. Pivotal Software, Inc. «Spring Boot». [En línea]. Available: <https://spring.io/projects/spring-boot> [Último acceso: 25 junio 2019].
- [15]. The Apache Software Foundation. «Apache Maven Project». [En línea]. Available: <https://maven.apache.org/> [Último acceso: 25 junio 2019].
- [16]. Yeoman «Using Yeoman». [En línea]. Available: <https://yeoman.io/learning/> [Último acceso: 13 Mayo 2019].
- [17]. Grunt JS «Grunt» . [En línea]. Available: <https://gruntjs.com/> [Último acceso: 17 Mayo 2019].
- [18]. Atlassian «Trello» . [En línea]. Available: <https://trello.com/> [Último acceso: 1 julio 2019].
- [19]. TeamGantt «TeamGantt». [En línea]. Available: <https://www.teamgantt.com/> [Último acceso: 2 julio 2019].

[20]. Atlassian «SourceTree» . [En línea]. Available: <https://www.sourcetreeapp.com/> [Último acceso: 2 julio 2019].

[21]. Simon Tatham «PuTTY» . [En línea]. Available: <https://www.putty.org/> [Último acceso: 10 Mayo 2019].

[22]. Atlassian «CyberDuck» . [En línea]. Available: <https://cyberduck.io/> [Último acceso: 3 Mayo 2019].

[23]. Postman, Inc «Postman» . [En línea]. Available: <https://www.getpostman.com/> [Último acceso: 4 Mayo 2019].

[24]. HeidiSQL «HeidiSQL » . [En línea]. Available :<https://www.heidisql.com/> [Último acceso: 16 Julio 2019].

[25]. Marvel «Marvel App» . [En línea]. Available: <https://marvelapp.com/> [Último acceso: 17 julio 2019].

[26]. JetBrains «IntelliJ IDEA» . [En línea]. Available: <https://www.jetbrains.com/idea/> [Último acceso: 25 junio 2019].

[27]. Arduino «Arduino IDE». [En línea]. Available: <https://www.arduino.cc/en/main/software> [Último acceso: 3 junio 2019].

[28]. Google «Android Studio». [En línea]. Available: <https://developer.android.com/studio> [Último acceso: 3 junio 2019].

[29]. Mike Cohn (2004). User Stories Applied. *Addison Wesley*

[30]. Cooking Hacks «MySignals HW». [En línea]. Available: <https://www.cooking-hacks.com/mysignals-hw-ehealth-medical-biometric-iot-platform-arduino-tutorial/> [Último acceso: 28 junio 2019].

[31]. Protocentral HealthyPI «HealthyPi v3». [En línea]. Available: <http://healthyapi.protocentral.com/> [Último acceso: 2 junio 2019].

[32]. Prusa Research s.r.o. «PRUSA I3 MK3S 3D PRINTER». [En línea]. Available: <https://www.prusa3d.es/original-prusa-i3-mk3-spa/> [Último acceso: 11 julio 2019].

[33]. Square «Retrofit». [En línea]. Available: <https://square.github.io/retrofit/> [Último acceso: 12 julio 2019].

ANEXO I. Manual de uso del sistema de recogida de datos

En este manual, se indicarán las instrucciones para el correcto funcionamiento del sistema de recogida de datos. El sistema de recogida de datos, actualmente cuenta con cuatro sensores los cuales iremos viendo a continuación. En primer lugar veremos cómo se conectan los distintos sensores al sistema y una vez los tengamos todos conectados, se mostrará cómo se deben de colocar en el cuerpo para que recojan las medidas correctamente.

El primer sensor en conectar, será el sensor de conductividad de la piel. Este se conectará en el primer conector del lateral derecho del sistema, como se puede ver en la siguiente imagen.

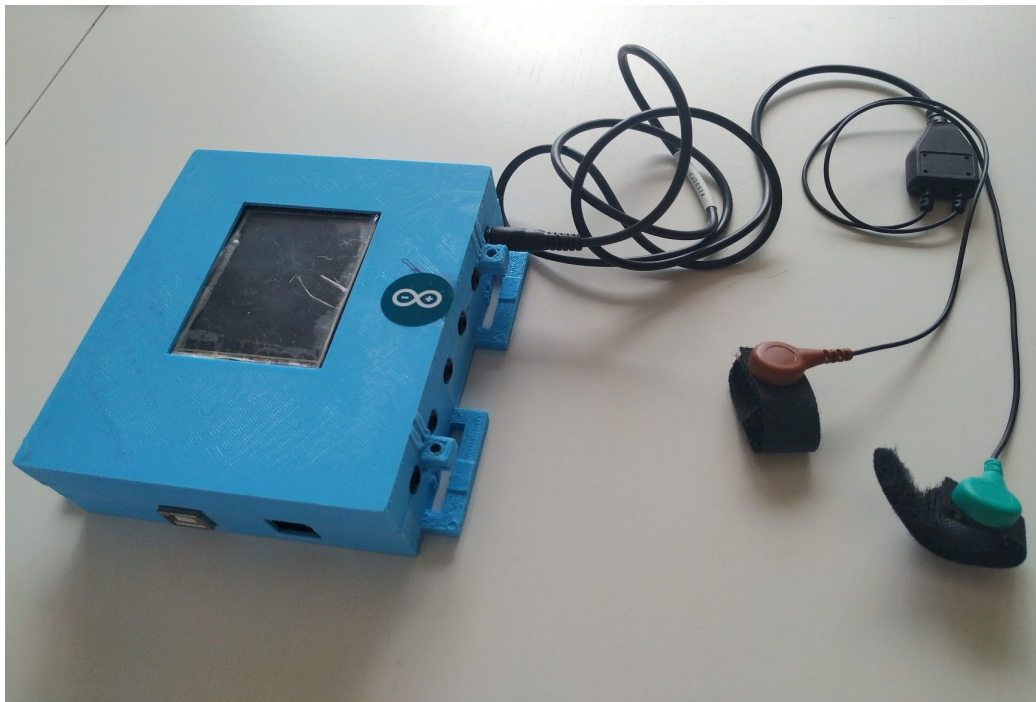


Ilustración 1. Conexión sensor conductividad

aire.



Ilustración 2. Conexión airflow

El electrocardiograma, se conectará en el tercer conector.



Ilustración 3. Conexión electrocardiograma

El último sensor en conectar será el electromiograma, que se conectará en el cuarto conector del lado derecho del sistema.



Ilustración 4. Conexión electromiograma

Conectados al sistema de manera correcta todos los sensores, pasaremos a colocarlos en el propio cuerpo. El orden de colocación de los sensores en el cuerpo no es relevante, pero por comodidad del paciente, se recomienda el siguiente orden.

En primer lugar, se colocará el electrocardiograma. Para ello se conectará el cable a los electrodos como se puede ver en la ilustración 5 y posteriormente, quitándoles la pegatina, se colocarán en el cuerpo siguiendo el esquema de la ilustración 6.

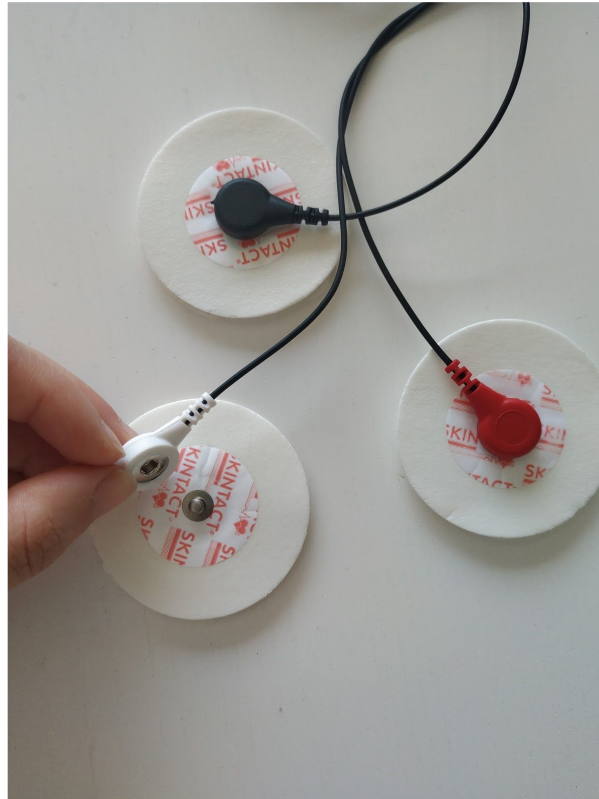


Ilustración 5. Conexión de electrodos

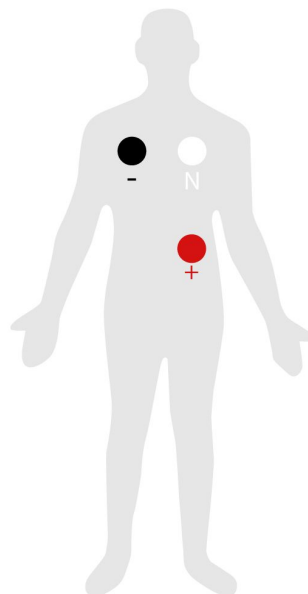


Ilustración 6. Colocación de los electrodos en el cuerpo

Una vez colocado el electrocardiograma, pasaremos al electromiograma. Al igual que anteriormente, primero se pondrán los electrodos, y después se colocarán en el músculo siguiendo el siguiente patrón.

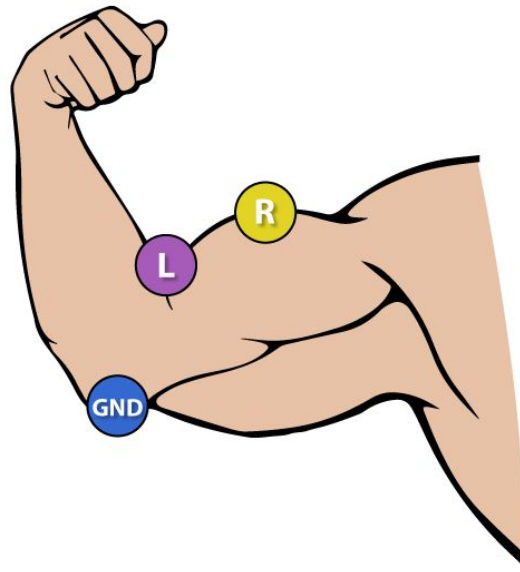


Ilustración 7. Colocación del electromiograma

A continuación, el sensor de flujo de aire, que deberá de colocarse en los orificios de la nariz como se muestra en la siguiente imagen.

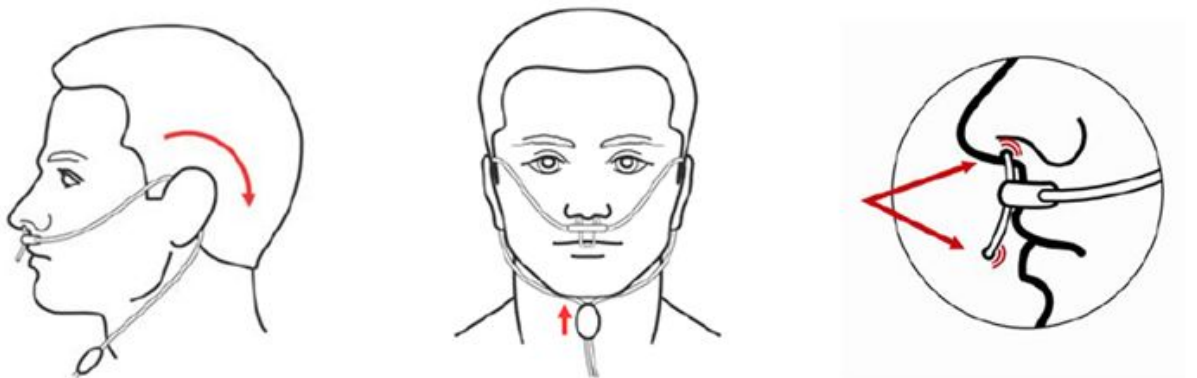


Ilustración 8. Colocación del sensor airFlow

El último sensor a conectar, será el de conductividad de la piel, que se ajustarán en los dedos índice y corazón, como se ve a continuación.



Ilustración 9. Colocación sensor de conductividad

Una vez tenemos todos los sensores conectados y colocados, sólo falta conectar el cable de alimentación al sistema en la clavija de la parte inferior y enchufarlo a la luz para que comience la recogida de datos.



Ilustración 10. Sistema de recogida de datos

ANEXO II. Manual de uso de la aplicación móvil

Este manual de usuario, describe las funciones básicas de la aplicación MyMirrorAPP, destinada a guiar al paciente en las terapias de exposición frente al espejo.

En primer lugar al abrir la aplicación, aparecerá la pantalla para realizar el login. El paciente debe de poner el usuario y la contraseña indicados por el terapeuta y pulsar sobre el botón “LOGIN”. En caso no rellenar correctamente alguno de estos campos, se mostrará un mensaje de error indicando el motivo por el que no se ha podido iniciar la sesión.

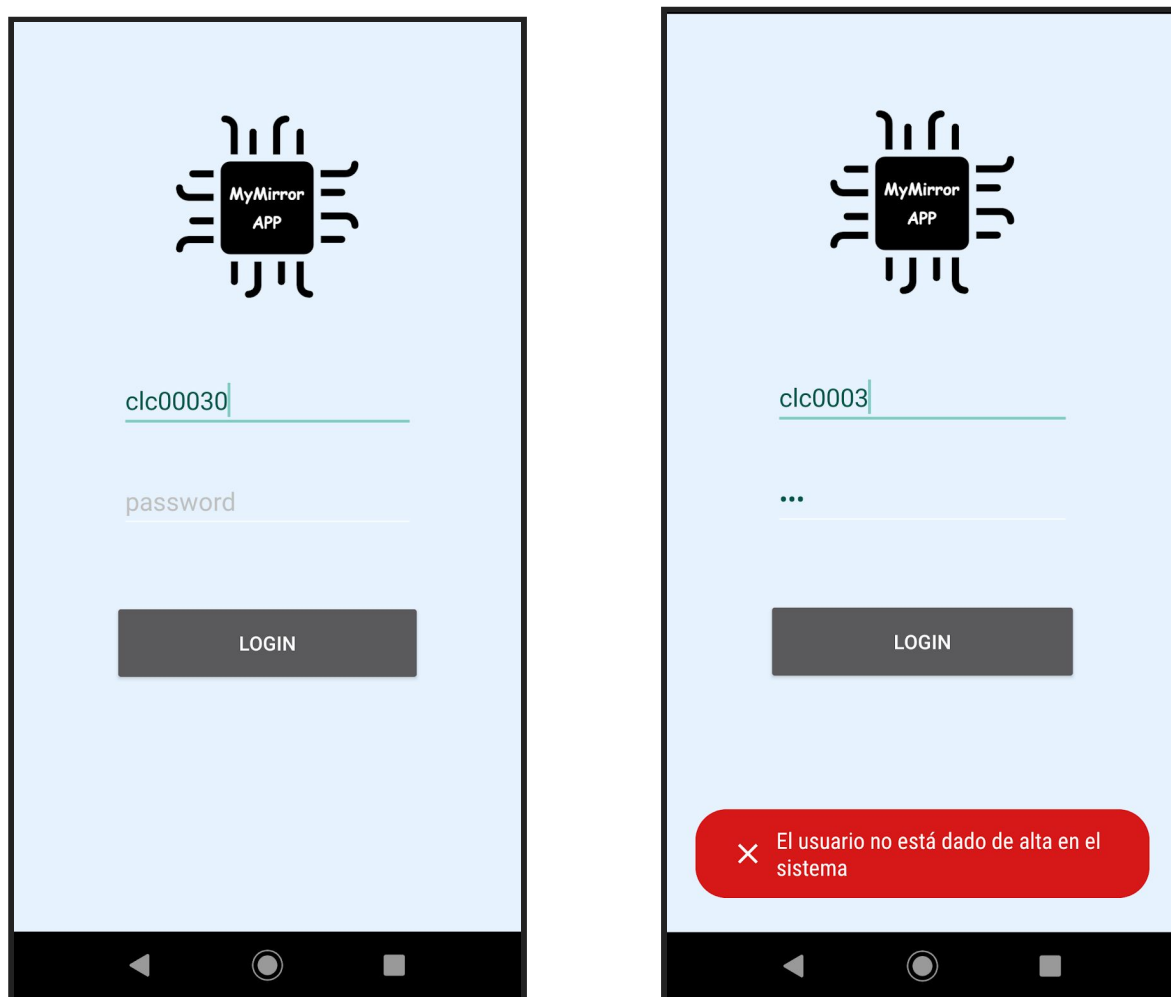


Ilustración 1. Aplicación móvil - Login

Una vez el paciente se ha autenticado correctamente, se muestra una pantalla de bienvenida. Desde aquí, el paciente podrá iniciar la terapia o, en caso necesario, revisar el manual de ayuda donde podrá encontrar las instrucciones para el uso del sistema de recogida de datos y colocación de los sensores, así como los pasos necesarios para realizar la sesión de manera correcta.

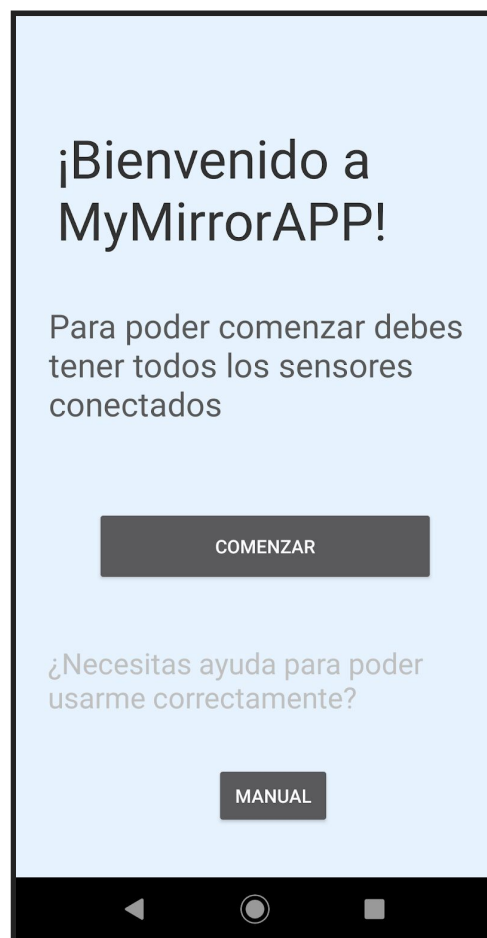


Ilustración 2. Aplicación móvil - Bienvenida

Lo primero que se realiza al comienzo de la terapia es la recogida de valores subjetivos, esto significa la realización de una serie de preguntas donde el paciente debe de deslizar el marcador e indicar el estado en el que se siente o se encuentra. Cada vez que se pulse sobre continuar, aparecerá la siguiente pregunta hasta que se realizan todas las cuestiones programadas.



Ilustración 3. Aplicación móvil - Preguntas subjetivas

Respondidas todas las preguntas anteriores, es hora de tomar los valores en reposo. El paciente debe de colocarse en un lugar donde se encuentre cómodo y relajado sin exponerse aún al espejo, y una vez preparado, pulsa sobre continuar.

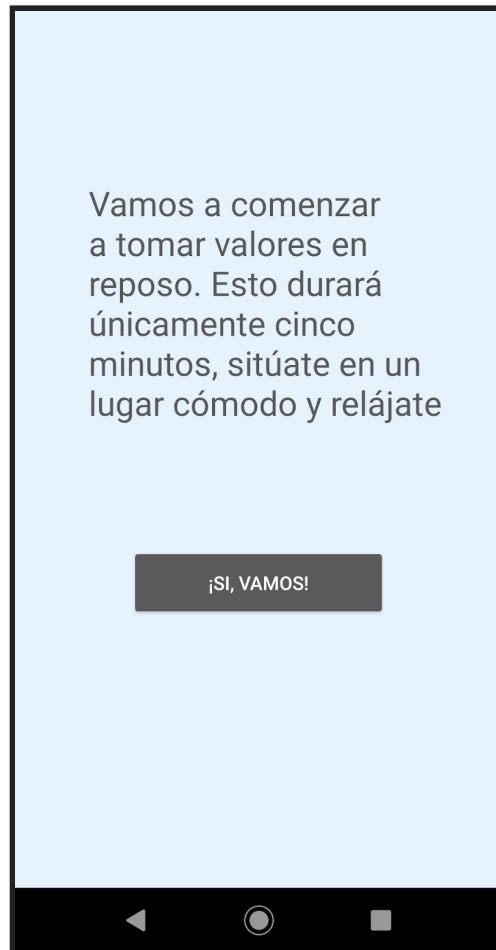


Ilustración 4. Aplicación móvil - Comienzo de recogida de datos en reposo

Se están recogiendo los valores fisiológicos del paciente, esta recogida durará cinco minutos y en la pantalla se mostrará un contador indicando el tiempo restante.



Ilustración 5. Aplicación móvil - Recogiendo datos en reposo

Una vez finalizado el tiempo de recogida de datos en reposo, se muestra en pantalla las instrucciones para que el paciente se coloque frente al espejo y comience la terapia. Al pulsar sobre continuar, se comenzará de nuevo con la recogida de variables fisiológicas.

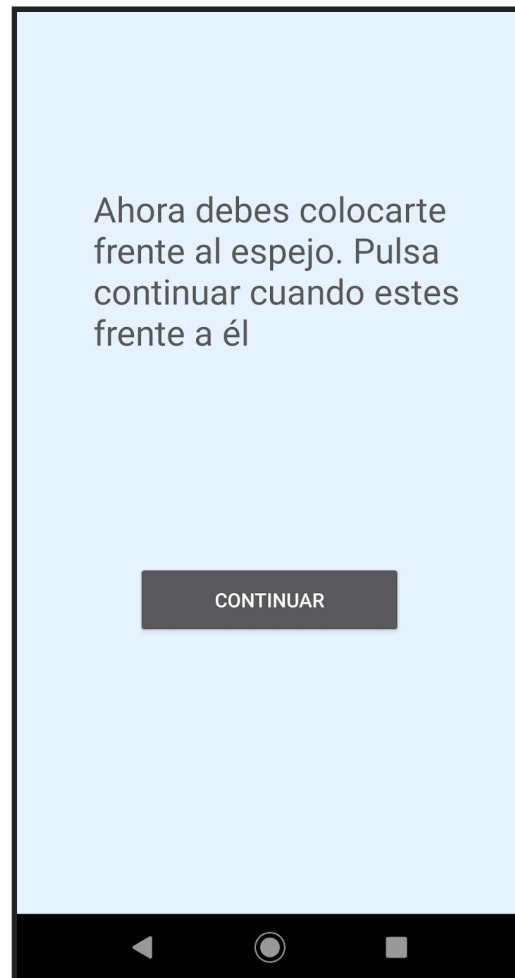


Ilustración 6. Aplicación móvil - Comienzo de recogida de datos

Mientras se están recogiendo los datos de los sensores, en la aplicación se muestra una pantalla informativa que, tras pasar un tiempo determinado por los terapeutas, aparecerá un botón para finalizar la sesión cuando el usuario considere necesario.

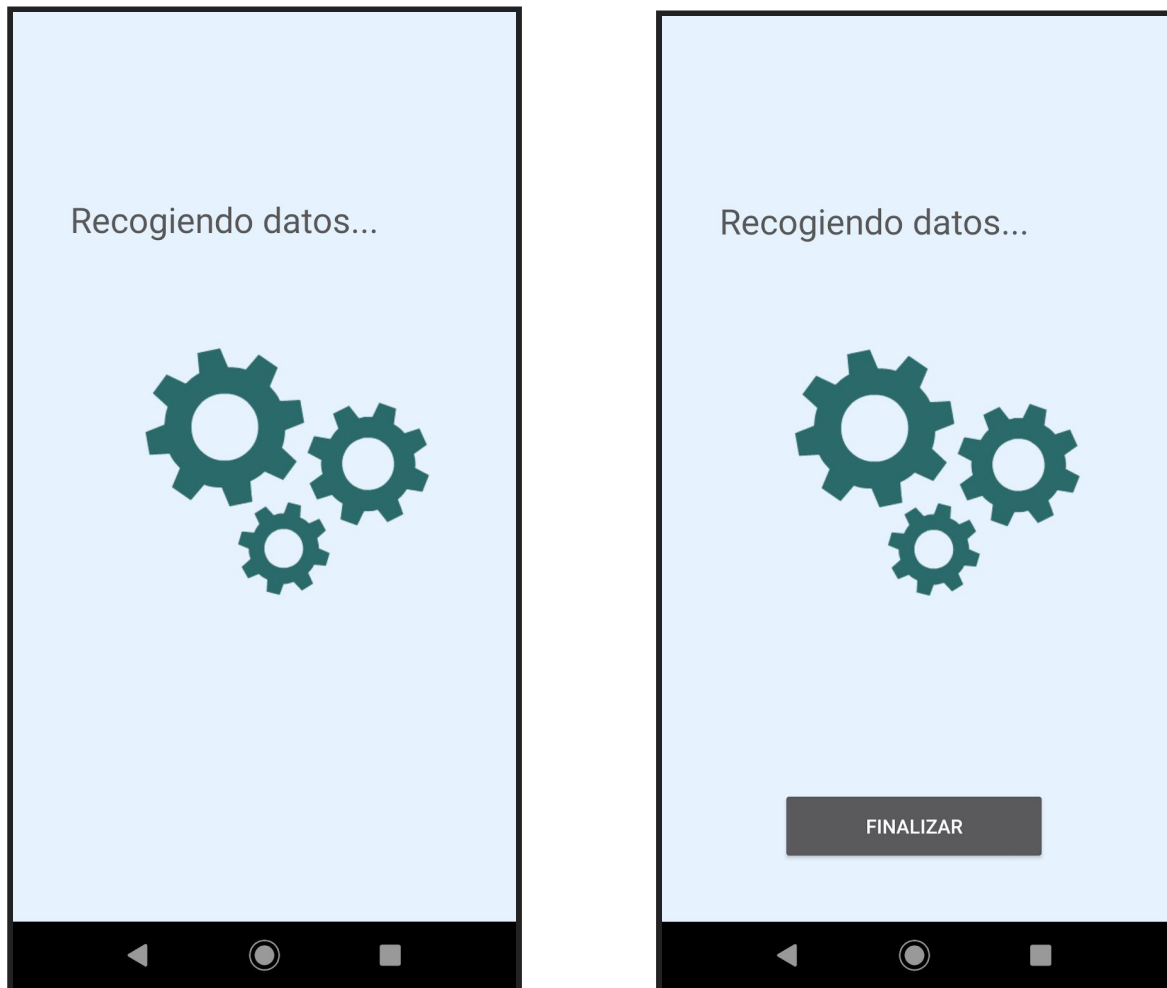


Ilustración 7. Aplicación móvil - Recogiendo datos frente al espejo

Cuando el usuario finaliza la sesión, se volverán a realizar mismas preguntas que al comienzo de la sesión, las cuales el paciente deberá responder. Al pulsar sobre continuar se muestra la siguiente pregunta hasta finalizar.



Ilustración 8. Aplicación móvil - Recogida de datos subjetivos

Contestada la última pregunta, se mostrará al paciente un aviso indicando que la sesión ha finalizado.

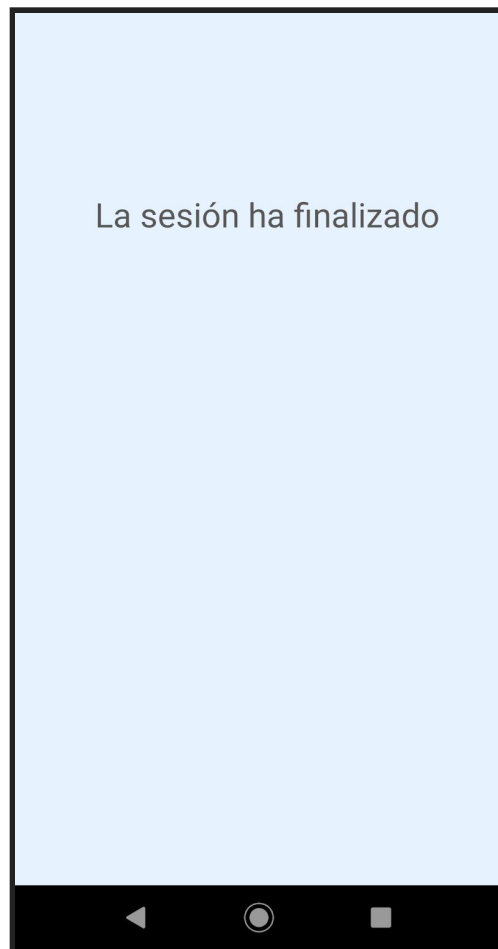


Ilustración 9. Finalización de la sesión

ANEXO III. Manual de uso de la aplicación web

En este manual, se mostrará la aplicación web destinadas a terapeutas para gestionar las terapias de exposición frente al espejo. Mediante esta página web podrán gestionar a los pacientes y controlar las sesiones que realizan.

En primer lugar, al abrir la aplicación web, aparecerá una vista para iniciar sesión. Es necesario introducir el nombre de usuario y la contraseña. En caso de que la contraseña o el nombre de usuario no sean correctos o no estén rellenos, se le notificará al usuario.

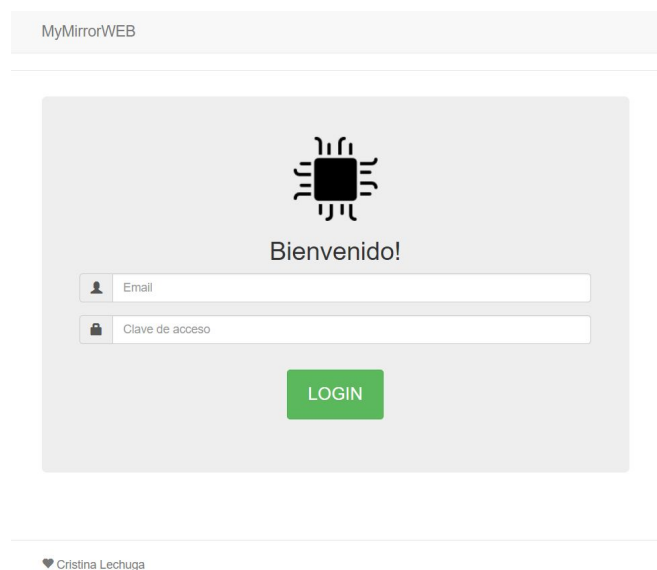


Ilustración 1. Aplicación web -Login

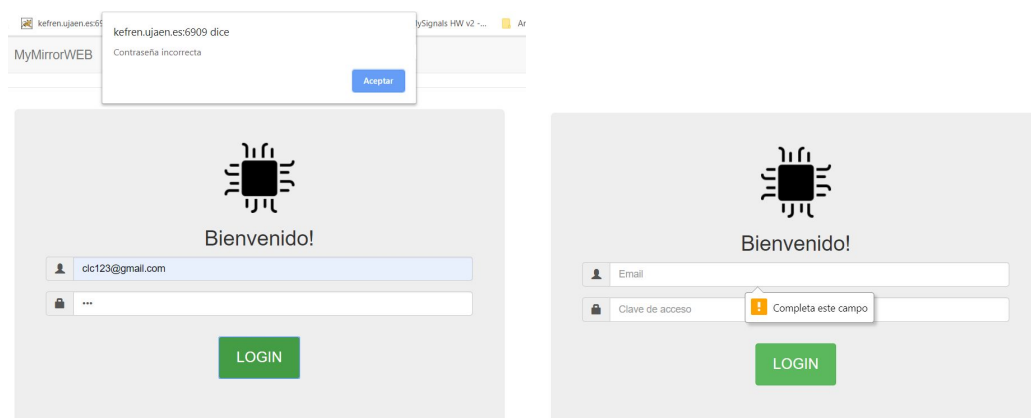


Ilustración 2. Aplicación web - Login con mensajes de error

Una vez el usuario ha iniciado la sesión, se muestra la vista del listado de las sesiones realizadas por los pacientes. En el listado, se indica el nombre de usuario que ha realizado la sesión, la fecha de inicio y fin de la sesión y el estado en el que esta se encuentra. Se podrá pulsar sobre el icono de “ver” para acceder a los datos recogidos de dicha sesión.

En la parte inferior de la vista, se encuentra la navegación por páginas y la selección del número de sesiones que se quieren visualizar en la pantalla.

En la parte superior de esta, se encontrará siempre el menú, donde el usuario podrá seleccionar lo que le interese.

MyMirrorWEB
Sesiones
Usuarios

Sesiones

Usuario	Inicio de la sesión	Finalización de la sesión	Estado	
clc	2019/10/08 - 20:58:29		Recogiendo datos frente al espejo	
clc	2019/10/08 - 20:44:37	2019/10/08 - 20:58:09	Finalizado	
clc00030	2019/10/08 - 20:38:18	2019/10/08 - 20:41:43	Finalizado	
clc	2019/10/08 - 20:28:20	2019/10/08 - 20:30:57	Finalizado	
clc	2019/10/07 - 20:59:04		Pendiente	
clc	2019/09/22 - 14:04:07		Pendiente	
clc	2019/09/18 - 12:18:21	2019/09/18 - 12:20:00	Finalizado	
clc00030	2019/09/15 - 20:46:51	2019/09/15 - 20:48:34	Finalizado	
clc00030	2019/09/14 - 15:14:59		Recogiendo datos frente al espejo	
clc00030	2019/09/14 - 15:10:35	2019/09/14 - 15:13:02	Finalizado	

« 1 2 »

5 20 50

Cristina Lechuga

Ilustración 3. Aplicación web - Listado de sesiones

Si el usuario pincha sobre la opción de ver los datos recogidos, se mostrará la siguiente pantalla, la cual está dividida en tres secciones: datos en reposo, datos frente al espejo y análisis de los datos.

En las secciones relativas a los datos recogidos, se podrá ver los datos subjetivos y los valores recogidos por los sensores. En cuanto a los datos subjetivos, se indica la pregunta que se le ha hecho al paciente y su respuesta, con respecto a los datos recogidos por los sensores, estos se mostrarán al pulsar sobre la opción “Mostrar datos” y se podrán descargar en formato CSV pinchando en el botón “Descargar datos”.

MyMirrorWEB
Sesiones
Usuarios

Detalle de la sesión

Datos en reposo

Datos subjetivos:

Pregunta	Respuesta
¿Cómo te sientes?	80
¿Te gusta tu cuerpo?	25
¿Estás enfadada?	46

Datos recogidos por los sensores:

Mostrar datos
Descargar datos

Datos frente al espejo

Datos subjetivos:

Pregunta	Respuesta
¿Cómo te sientes?	27
¿Te gusta tu cuerpo?	1
¿Estás enfadada?	77

Datos recogidos por los sensores:

Mostrar datos
Descargar datos

Ilustración 4. Aplicación web - Detalle general de una sesión

Al pulsar sobre “Mostrar datos” se desplegará una tabla que muestra todos los datos recogidos por los sensores. Se detalla el momento de recogida de los datos, y la medida en voltios recogida por cada uno de los sensores.

¿Te gusta tu cuerpo?				
50				
¿Estás enfadada?				
83				
Datos recogidos por los sensores:				
Ocultar datos Descargar datos				
Timestamp	Airflow	GSR	ECG	EMG
2019-09-14T15:15:03	0.02933	0.91398	2.55132	1.54448
2019-09-14T15:15:03	0.01955	0.826	2.1261	1.55914
2019-09-14T15:15:03	0.05375999999999995	0.89443	3.93939	1.54935
2019-09-14T15:15:04	0.1173	0.89443	2.76636	1.52981
2019-09-14T15:15:04	0.10753	0.8651	1.95503	1.52493
2019-09-14T15:15:05	0.12708	0.86999	1.99902	1.52493
2019-09-14T15:15:05	0.12219	0.8113400000000001	1.79863	1.52981

Ilustración 5. Aplicación web - Mostrando datos recogidos en una sesión

Al pulsar sobre “Descargar datos”, se descargará en el ordenador un fichero CSV con todos los valores recogidos por los sensores.

```

dataSensors: Bloc de notas
Archivo Edición Formato Ver Ayuda
timestamp,GSRVolt,airVolt,ECGVolt,EMGVolt
2019-09-14T15:15:03,0.91398,0.02933,2.55132,1.54448
2019-09-14T15:15:03,0.826,0.01955,2.1261,1.55914
2019-09-14T15:15:03,0.89443,0.05375999999999995,3.93939,1.54935
2019-09-14T15:15:04,0.89443,0.1173,2.76636,1.52981
2019-09-14T15:15:04,0.8651,0.10753,1.95503,1.52493
2019-09-14T15:15:05,0.86999,0.12708,1.99902,1.52493
2019-09-14T15:15:05,0.8113400000000001,0.12219,1.79863,1.52981
2019-09-14T15:15:06,0.81623,0.12708,2.22873,1.53959
2019-09-14T15:15:06,0.8846499999999999,0.13685,2.62952,1.52493
2019-09-14T15:15:07,0.88954,0.06354,2.71261,1.52981
2019-09-14T15:15:07,0.86999,0.08309,2.14076,1.52004
2019-09-14T15:15:08,0.87488,0.05865,3.64614,1.54935
2019-09-14T15:15:08,0.7966799999999999,0.09775,2.46823,1.52493
2019-09-14T15:15:09,0.81623,0.09286,3.00098,1.52004
2019-09-14T15:15:09,0.8651,0.10753,3.97849,1.53959
2019-09-14T15:15:09,0.86999,0.07331,2.90811,1.5249300000000001
2019-09-14T15:15:10,0.86022,0.05865,2.12121,1.5249300000000001
2019-09-14T15:15:10,0.8015599999999999,0.11241,1.71066,1.52004
2019-09-14T15:15:11,0.78201,0.14174,2.28739,1.52004
2019-09-14T15:15:11,0.8455499999999999,0.09286,3.74878,1.52004
2019-09-14T15:15:12,0.84066,0.08309,3.3870999999999998,1.52981
2019-09-14T15:15:12,0.86999,0.14662999999999998,3.93939,1.53959
2019-09-14T15:15:13,0.8797699999999999,0.12708,3.09384,1.55425
2019-09-14T15:15:13,0.79179,0.17595,1.7741900000000002,1.5249300000000001
2019-09-14T15:15:14,0.78201,0.1955,2.4780100000000003,1.52981
2019-09-14T15:15:14,0.78201,0.23949,2.2434000000000003,1.52981
2019-09-14T15:15:14,0.80645,0.22972,3.09384,1.52004
2019-09-14T15:15:15,0.8455499999999999,0.20038999999999998,3.905179999999
2019-09-14T15:15:15,0.87488,0.20038999999999998,3.12805,1.54448
2019-09-14T15:15:15,0.8113400000000001,0.12107,1.74076,1.52004

```

Ilustración 6. Aplicación web - Fichero CSV descargado

En la parte inferior de la pantalla de visualización de datos, se podrá observar un análisis básico de los datos, que consiste en un gráfico de barras donde se puede visualizar de una manera sencilla los valores medios recogidos por cada uno de los sensores, comparando este valor antes de la exposición frente al espejo y durante la exposición.

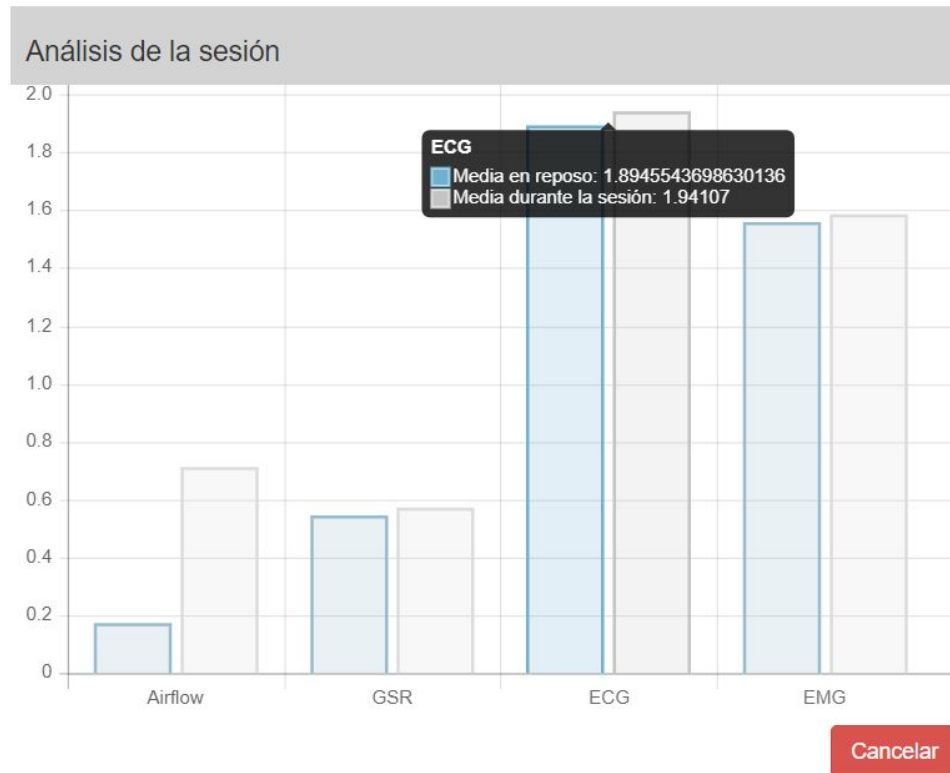


Ilustración 7. Aplicación web - Gráfica de barras con los valores medios recogidos

El acceso a la gestión de usuarios, se realiza a través del menú superior, al pinchar sobre “Usuarios” se abrirá el listado de pacientes que se han dado de alta en el sistema. En la tabla se mostrará el número de usuario, su nombre y apellidos y el nombre de usuario con el que inicia sesión en la aplicación móvil junto con el botón de “Editar usuario” que se encuentra representado por un lápiz.

En la parte inferior de la tabla, se encuentra la navegación por páginas y la selección del número de usuarios que se quieren visualizar en la pantalla.

En la parte superior de la tabla, se encuentra el botón para crear un nuevo usuario.

MyMirrorWEB Sesiones Usuarios

Usuarios

Nuevo usuario

ID	First Name	Last Name	User Name	
1	Cristina	Lechuga	clc00030	
2	Cris	Test	clc	
3	Jose David	Calada	jdca00022	
4	Ana	Montes	jana111	
5	Carmen	Rodríguez	srs111	

« 1 2 »

5 10 15

♥ Cristina Lechuga

Ilustración 8. Aplicación web - Listado de usuarios

Cuando el usuario pincha sobre el icono del lápiz, se abre la vista de edición de un usuario. En ella se ven los datos básicos del paciente y se podrá cambiar el estado del usuario, indicando si puede realizar la terapia “Activo”, mientras que si el usuario ya ha finalizado todas las sesiones, se pondrá en estado “Inactivo” para que no pueda volver a iniciar sesión en la aplicación web.

MyMirrorWEB

Sesiones

Usuarios

Editar Usuario

Nombre:

Cristina

Apellido:

Lechuga

DNI:

111223342G

Telefono:

666776677

Email:

crsxxx@red.ujaen.es

Usuario:

clc00030

Contraseña:

123456

Estado:

Activo

Cancelar

Guardar

♥ Cristina Lechuga

Ilustración 9. Aplicación web - Edición de un usuario

Si en la pantalla del listado de usuarios (Ilustración 8) el terapeuta pincha sobre “Nuevo usuario” se abrirá una vista para que rellene los datos básicos de un paciente, de esta manera se le creará una nueva cuenta y el paciente podrá iniciar sesión en la aplicación móvil y realizar la terapia.

MyMirrorWEB Sesiones Usuarios

Nuevo Usuario

Nombre:

Apellido:

DNI:

Telefono:

Email:

Usuario:

Contraseña:

Estado:

Cancelar

Guardar

♥ Cristina Lechuga

Ilustración 10. Aplicación web - Nuevo usuario

ANEXO IV. Documentación API REST



Select a spec

default

Api Documentation ^{1.0}

[Base URL: kefren.ujaen.es:6907/signals-1]
<http://kefren.ujaen.es:6907/signals-1/v2/api-docs>

Api Documentation

[Terms of service](#)

[Apache 2.0](#)

authentication-service Authentication Service

▼

POST

/authenticate/user

authenticate

Parameters

Try it out

Name	Description
userBean * required (body)	userBean

Example Value

Model

```
{
  "dni": "string",
  "email": "string",
  "firstName": "string",
  "id": 0,
  "lastName": "string",
  "password": "string",
  "phoneNumber": 0,
  "status": 0,
  "userName": "string"
}
```

Parameter content type

application/json

Responses

Response content type

application/json;charset=UTF-8

Code	Description
200	<i>OK</i>
Example Value Model <pre>{ "body": {}, "statusCode": "100 CONTINUE", "statusCodeValue": 0}</pre>	
201	<i>Created</i>
401	<i>Unauthorized</i>
403	<i>Forbidden</i>
404	<i>Not Found</i>

POST

/authenticate/userAdmin

authenticateWeb

Parameters

Try it out

Name	Description
userBean * required (body)	userBean
Example Value Model <pre>{ "dni": "string", "email": "string", "firstName": "string", "id": 0, "lastName": "string", "password": "string", "phoneNumber": 0, "status": 0, "userName": "string"}</pre>	
Parameter content type application/json	

Responses		Response content type
		application/json;charset=UTF-8
Code	Description	
200	<i>OK</i>	
	Example Value Model <pre>{ "body": {}, "statusCode": "100 CONTINUE", "statusCodeValue": 0}</pre>	
201	<i>Created</i>	
401	<i>Unauthorized</i>	
403	<i>Forbidden</i>	
404	<i>Not Found</i>	

basic-error-controller

Basic Error Controller

>

question-service

Question Service

▼

POST

/subjectiveData/save-subjectiveData

saveSubjectiveData

Parameters

Try it out

Name

Description

Name

subjectiveDataList * required

(body)

Description

subjectiveDataList

Example Value

Model

```
[
  {
    "id": 0,
    "idSession": 0,
    "question": "string",
    "response": "string",
    "status": 0
  }
]
```

Parameter content type

application/json

Responses

Response content type

application/json;charset=UTF-8

Code

Description

200

OK

Example Value

Model

```
{
  "body": {},
  "statusCode": "100 CONTINUE",
  "statusCodeValue": 0
}
```

201

Created

401

Unauthorized

403

Forbidden

404

Not Found

kefren.ujae.es:6907/signals-1/swagger-ui.html#/authentication-service/authenticateUsingPOST

4/19

session-service

Session Service



GET /session/average-data/{idSession} getAverageData

Parameters

Try it out

Name	Description
idSession * required integer(\$int64) (path)	idSession

Responses

Response content type

application/json;charset=UTF-8

Code	Description
200	<i>OK</i> Example Value Model <pre>{ "body": {}, "statusCode": "100 CONTINUE", "statusCodeValue": 0 }</pre>
401	<i>Unauthorized</i>
403	<i>Forbidden</i>
404	<i>Not Found</i>

POST /session/create createSession

Try it out

Parameters

NameDescription

sessionBean * required

sessionBean

(body)

Example Value Model

```
{
  "id": 0,
  "idUser": 0,
  "status": 0,
  "timeFinished": "2019-10-27T10:33:52.077Z",
  "timeStart": "2019-10-27T10:33:52.077Z",
  "userName": "string"
}
```

Parameter content type

application/json

Responses

Response content type

application/json;charset=UTF-8

CodeDescription

200

OK

Example Value Model

```
{
  "body": {},
  "statusCode": "100 CONTINUE",
  "statusCodeValue": 0
}
```

201

Created

401

Unauthorized

403

Forbidden

404

Not Found

GET

/session/downloadData/{status}/{idSession}downloadData

Parameters

Try it out

Name	Description
idSession * required integer(\$int64) (path)	idSession
status * required integer(\$int32) (path)	status

Responses

Response content type

text/csv

Code	Description
200	OK
401	Unauthorized
403	Forbidden
404	Not Found

PUT

/session/finishfinishSession

Parameters

Try it out

Name	Description
------	-------------

Name

sessionBean * required

(body)

Description

sessionBean

Example Value

Model

```
{
  "id": 0,
  "idUser": 0,
  "status": 0,
  "timeFinished": "2019-10-27T10:33:49.750Z",
  "timeStart": "2019-10-27T10:33:49.750Z",
  "userName": "string"
}
```

Parameter content type

application/json

Responses

Response content type

application/json;charset=UTF-8

Code

200

Description

OK

Example Value

Model

```
{
  "body": {},
  "statusCode": "100 CONTINUE",
  "statusCodeValue": 0
}
```

Code

201

Description

Created

Code

401

Description

Unauthorized

Code

403

Description

Forbidden

Code

404

Description

Not Found

Parameters

Try it out

Name	Description
requestParam * required [object] (query)	requestParam

Responses

Response content type

application/json;charset=UTF-8

Code	Description
200	<i>OK</i> Example Value Model <pre>{ "body": {}, "statusCode": "100 CONTINUE", "statusCodeValue": 0 }</pre>
401	<i>Unauthorized</i>
403	<i>Forbidden</i>
404	<i>Not Found</i>

GET /session/get-detail/{idSession} getSessionDetail

Parameters

Try it out

Name	Description
idSession * required integer(\$int64) (path)	idSession

Responses		Response content type
		application/json;charset=UTF-8
Code	Description	
200	OK	
	Example Value	Model
	<pre>{ "body": {}, "statusCode": "100 CONTINUE", "statusCodeValue": 0 }</pre>	
401	Unauthorized	
403	Forbidden	
404	Not Found	

PUT /session/startMirrorData startMirrorSession	
Parameters	Try it out
Name	Description
sessionBean * required (body)	sessionBean
	Example Value Model
	<pre>{ "id": 0, "idUser": 0, "status": 0, "timeFinished": "2019-10-27T10:33:47.585Z", "timeStart": "2019-10-27T10:33:47.585Z", "userName": "string" }</pre>
	Parameter content type
	application/json

Responses

Response content type

application/json;charset=UTF-8

Code	Description
200	OK Example Value Model <pre>{ "body": {}, "statusCode": "100 CONTINUE", "statusCodeValue": 0}</pre>
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

PUT

/session/startRelaxData

startSession

Parameters

Try it out

Name	Description
------	-------------

Responses	Response content type	application/json;charset=UTF-8
-----------	-----------------------	--------------------------------

Code	Description
200	<i>OK</i>
	Example Value Model
	<pre>{ "body": {}, "statusCode": "100 CONTINUE", "statusCodeValue": 0 }</pre>
201	<i>Created</i>
401	<i>Unauthorized</i>
403	<i>Forbidden</i>
404	<i>Not Found</i>

user-service User Service



POST /user/create-user createUser

Parameters

Try it out

Name

Description

userBean * required
(body)

userBean

Example Value Model

```
{
  "dni": "string",
  "email": "string",
  "firstName": "string",
  "id": 0,
  "lastName": "string",
  "password": "string",
  "phoneNumber": 0,
  "status": 0,
  "userName": "string"
}
```

Parameter content type

application/json

Responses

Response content type

/

Code

Description

Code	Description
200	<i>OK</i>
Example Value Model <pre>{ "body": {}, "statusCode": "100 CONTINUE", "statusCodeValue": 0}</pre>	
201	<i>Created</i>
401	<i>Unauthorized</i>
403	<i>Forbidden</i>
404	<i>Not Found</i>

GET

/user/get-all

getAllUser

Parameters

Try it out

Name	Description
requestParam * required [object] (query)	requestParam

Responses

Response content type

/

Code	Description
------	-------------

Code	Description
200	<i>OK</i>
Example Value Model <pre>{ "body": {}, "statusCode": "100 CONTINUE", "statusCodeValue": 0}</pre>	
401	<i>Unauthorized</i>
403	<i>Forbidden</i>
404	<i>Not Found</i>

GET

/user/get-data

getDataSensors

Parameters

Try it out

Name	Description
requestParam * required [object] (query)	requestParam

Responses

Response content type

/

Code	Description
------	-------------

Code	Description
200	<i>OK</i>
Example Value Model <pre>{ "body": {}, "statusCode": "100 CONTINUE", "statusCodeValue": 0}</pre>	
401	<i>Unauthorized</i>
403	<i>Forbidden</i>
404	<i>Not Found</i>

GET

/user/get-user-detail/{idUser} getUserDetail

Parameters

Try it out

Name	Description
idUser * required integer(\$int64) (path)	idUser

Responses

Response content type

/

Code	Description
------	-------------

Code	Description
200	<i>OK</i>
Example Value Model <pre>{ "body": {}, "statusCode": "100 CONTINUE", "statusCodeValue": 0}</pre>	
401	<i>Unauthorized</i>
403	<i>Forbidden</i>
404	<i>Not Found</i>

PUT

/user/update-user

updateUser

Parameters

Try it out

Name	Description
userBean * required (body)	userBean Example Value Model <pre>{ "dni": "string", "email": "string", "firstName": "string", "id": 0, "lastName": "string", "password": "string", "phoneNumber": 0, "status": 0, "userName": "string"}</pre> Parameter content type application/json

Responses

Response content type

/

CodeDescription

200OK

Models



Example ValueModel

ModelAndView

```
{
  "body": {},
  "statusCode": "100 CONTINUE",
  "statusCodeValue": 0
}
```

QuestionBean

201Created

```
{
  id: integer($int64)
  idSession: integer($int64)
  question: string
  response: string
  status: integer($int32)
}
```

403Forbidden

ResponseEntity

404Not Found

```
{...}
```

SessionBean

```
{
  id: integer($int64)
  idUser: integer($int64)
  status: integer($int32)
  timeFinished: string($date-time)
  timeStart: string($date-time)
  userName: string
}
```

UserBean

```
{
  dni: string
  email: string
  firstName: string
  id: integer($int64)
  lastName: string
  password: string
  phoneNumber: integer($int32)
  status: integer($int32)
  userName: string
}
```

View {...}

