



DISEÑO Y DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA MONITORIZACIÓN DEL BIENESTAR DE LA GENTE DE LA 3ª EDAD

TFM-PEC2



5 DE OCTUBRE DE 2019
JOSÉ MANUEL CASTELLANO DOMÍNGUEZ

Contenido

Estado del Arte:	2
Salud Conectada	4
Heart Rate OS – Android Watch	4
Health Wereable	5
Resumen aplicaciones vistas:	5
Objetivos y Alcance:.....	7
Planificación:	8
Bibliografía:.....	10

Estado del Arte:

Con la llegada del nuevo milenio, surgió el boom y auge de los dispositivos móviles e inteligentes. Este boom se inició con la aparición del primer *Iphone* lanzado por la compañía Apple y la presentación del sistema operativo Android por parte de Google, ambos durante el año 2007. Actualmente los sistemas operativos de *Android* (Google) e *IOS* (Apple) se ejecutan en incontables dispositivos como son teléfonos móviles, tabletas, pulseras, relojes, automóviles y televisores. Por poner algún ejemplo de la variedad de dispositivos tenemos, por el lado de Android, televisores de marcas como *Philips* y *Sony* o *smartwatches* donde se pueden encontrar cientos de modelos (*Huawei*, *Xaomi*, etc). Por el lado de *IOS*, se tiene la propia televisión diseñada por Apple y como *smartwatch* tiene su propio modelo de reloj denominado *Apple Watch*.



Fig.1 Algunos modelos de *Smartwatch* (Fuente: Computerhoy)

El boom no se limita únicamente simplemente a los dispositivos descritos anteriormente, sino a cualquier dispositivo que pueda estar conectado a una red. En la última década, el número de dispositivos interconectados entre sí, ha crecido exponencialmente. Este término se ha acuñado con el nombre de *IoT* (Internet of Things) y fue acuñado por Dave Evans perteneciente a Cisco. En [1] se indica en que el número de dispositivos interconectados superaba ya al de personas y que esta diferencia se iría incrementando a medida que fueran pasando los años. En el mencionado artículo, se indica que cada vez habrá más dispositivos con sensores que permitirán enviar transmitir dicha información para que pueda ser analizada o procesada por otros dispositivos.

Aquellos dispositivos que puedan ser sujetos a ser llevados o vestidos por una persona se les conoce como *wereables*. Este término fue acuñado [2] durante la década de los 90s, indicando que un *wearable* es una pieza de vestimenta que tiene capacidad de computación, es decir, es una pieza de ropa que actúa igual que un ordenador y que posee diferentes entradas para capturar eventos del entorno. Los *wereables* más conocidos son los *smartwatches*, aunque existen otros como camisetas, zapatillas, etc.

La mayoría de los *smartwatches* incorporan una serie de sensores que recogen información como el número de pasos, la frecuencia cardíaca o el número de horas de sueño que luego envían a otro dispositivo mediante una tecnología inalámbrica. La gran mayoría de los *smartwatches* actuales funcionan vinculándose a un *Smartphone* u teléfono inteligente mediante *Bluetooth* (lo que crea una red de área personal, o denominada PAN, formada por el teléfono inteligente y el reloj), aunque en los

smartwatches más recientes se permite realizar este vínculo gracias a una red *Wifi*. Este comportamiento se extrapola a la gran mayoría de los *wereables*. En el caso del presente proyecto, el *wearable* que se utiliza es una camiseta inteligente en vez de un reloj, aunque ambos tienen el mismo funcionamiento.

Por poner un ejemplo de lo indicado en el párrafo anterior, en el 2014 se presentó un estudio [3] que se pretendía construir una camiseta inteligente y una aplicación móvil, para el seguimiento de pacientes con reumatismo, que permite vincularse vía *Bluetooth* con la camiseta para así obtener y mostrar los datos en la pantalla del teléfono inteligente. En la figura 2 se muestra el esquema de conexión del mismo entre la camiseta y el teléfono inteligente.

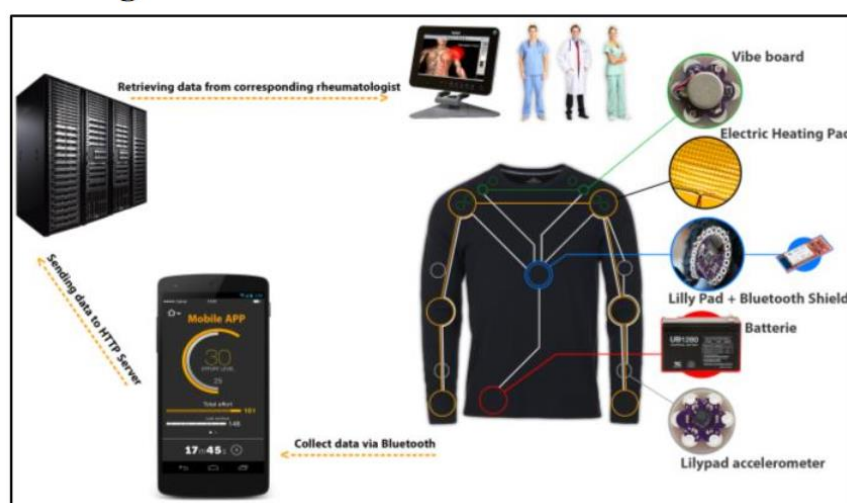


Fig. 2 - Esquema de conexión de camiseta inteligente. Fuente ([3])

Como se observa en la figura 2, la camiseta envía, vía *Bluetooth*, los datos obtenidos de los sensores instalados, la aplicación móvil los muestra y actúa de pasarela para publicarlos en la nube, desde donde el médico en cuestión puede consultarlos. Esta arquitectura funciona bien cuando se tiene el teléfono y la camiseta cerca, pero en el momento que se aleje, los datos dejan de ser obtenidos. Adicionalmente existe otro problema; el teléfono inteligente del usuario es el encargado de almacenar y subir los datos al servidor lo que hace que el consumo de batería sea elevado, más teniendo en cuenta de que se tiene que tener el *Bluetooth* activo para poderse vincular con la camiseta.

Si en vez del hardware, se centra el estudio sobre las aplicaciones para dispositivos móviles, basta con hacer una simple búsqueda en Google Play [4] para observar que existe gran variedad de aplicaciones que pueden ser utilizadas en los *wearable* o dependen de los datos del *wearable* para funcionar. En concreto, si se centra la investigación sobre aplicaciones relacionadas con la salud, se pueden encontrar cientos de las mismas. La gran mayoría de las aplicaciones que se encuentran en el market que implican el uso de un *wearable* están relacionadas con apps la detección de constantes relacionadas con la actividad deportiva, como son el número de pasos, la frecuencia cardíaca o el número de calorías quemadas. Una investigación de cada una de estas aplicaciones daría para escribir varios libros, por lo que simplemente se limitará a explicar las aplicaciones más relevantes encontradas respecto al ámbito que abarca el

presente TFM, que consiste el uso de un *wearable* para obtener constantes vitales de un paciente.

Salud Conectada

Esta aplicación [5], desarrollada por la compañía de seguros médicos Sanitas, permite conectarse a cualquier *wearable* via *Bluetooth* para obtener las constantes vitales del paciente y las muestra en el teléfono móvil. Adicionalmente permite apuntar las citas, recordatorios y un listado de medicamentos a tomar. El problema que tiene principalmente es la gran dependencia que tiene del *Bluetooth* para funcionar, de hecho, la principal queja de los usuarios es que se desvincula fácilmente el *wearable* del dispositivo móvil. Además, para obtener valores ambos dispositivos deben de estar cerca. Otro problema se ha detectado mientras se hizo una prueba de la aplicación para el presente estudio, debes de tener una cuenta en Sanitas para poderla utilizar y no se sugiere la creación de una desde la aplicación.

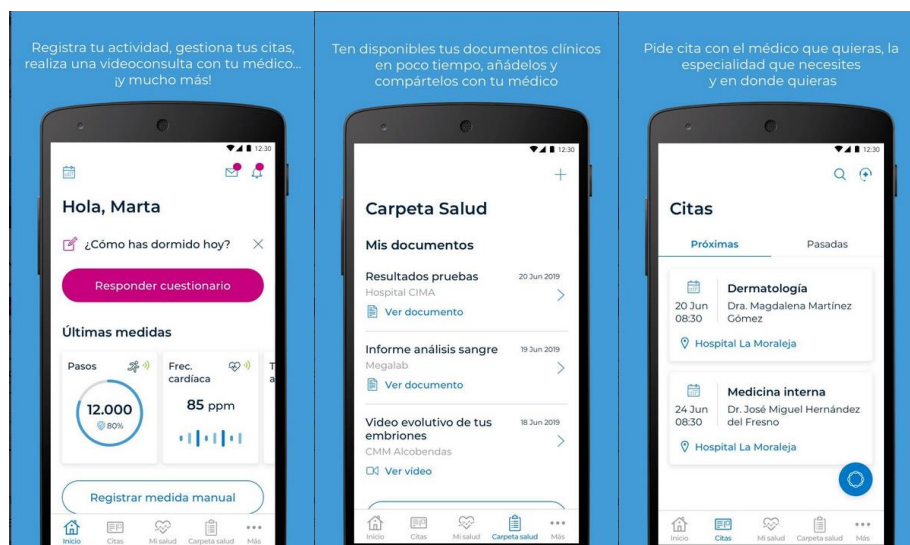


Fig.3 - Capturas aplicación Salud Conectada. Fuente ([5])

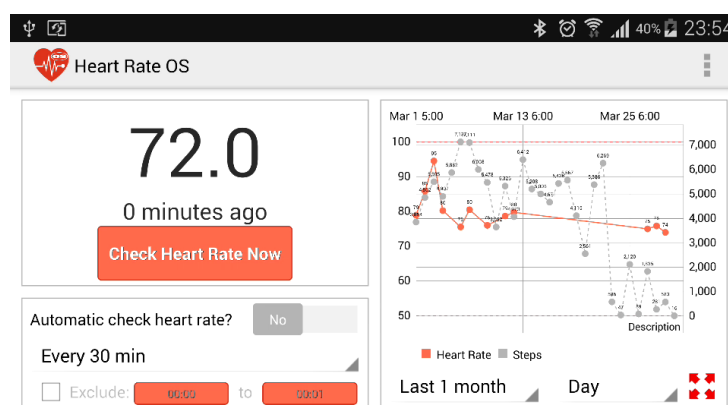


Fig.4 - Capturas aplicación Heart Rate OS. Fuente ([6])

Heart Rate OS – Android Watch

Esta aplicación [6], desarrollada por la LFApp y cuya aplicación puede visualizarse en la figura 4, permite medir la frecuencia cardiaca de la persona que lleva el reloj inteligente que esté vinculado al teléfono inteligente. La aplicación permite dos modos de funcionamiento. La primera de ellas es solicitando una medición manualmente,

mientras que la segunda es programar la aplicación para que haga mediciones cada media hora. Cada medición dura 1 minuto. Luego, los datos obtenidos se sincronizan con los de *Google Fit*.

El principal inconveniente que tiene esta aplicación es que la monitorización de la frecuencia cardiaca no es permanente y se puede programar únicamente cada 30 minutos. Además, las opciones de sincronización solo permiten sincronizar con *Google Fit*.

Health Wereable

Esta aplicación [7], desarrollada por Analog Devices Inc es una aproximación parecida a lo que se pretende abordar en el presente proyecto. Esta aplicación permite conectarse a un *smartwatch* específicamente creado [8] para suministrar una serie de constantes vitales a la aplicación móvil. Esta aplicación permite especificar entre que constantes se quieren realizar mediciones (permite frecuencia cardiaca, conductividad de la piel, temperatura, etc.).

La aplicación permite visualizar en tiempo real y mediante una serie de gráficas, las mediciones obtenidas a lo largo del tiempo y visualizar un histórico del mismo. La única diferencia respecto a las anteriores es que no existe opción para compartir y publicar los datos en la nube.

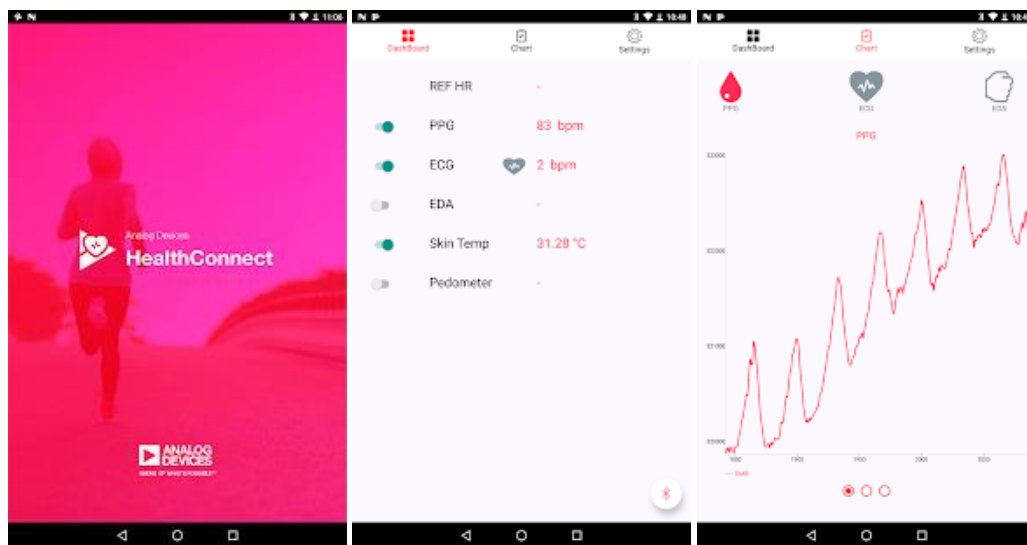


Fig.5 - Capturas aplicación Health Wereable. Fuente ([7])

Resumen aplicaciones vistas:

Las aplicaciones que se han visto presentan 3 características principales:

- Son apps nativas.
- Requieren de *Bluetooth* para vincularse con el *wearable*.
- En caso de detectar alguna anomalía en los datos, no se notifica al usuario.

Esto conlleva a una serie de características que tienen estas aplicaciones con el *wearable* asociado.

- Requieren estar cerca el dispositivo móvil del *wearable*.

- Se debe hacer un desarrollo por cada plataforma que se desee exportar la aplicación. Esto quiere decir que en caso de querer exportarlo a Android e IOS hay que realizar 2 desarrollos distintos.
- Si la app tiene opción de subir datos a la nube, utiliza la conexión a la red del dispositivo móvil, lo que utiliza los datos móviles del usuario e utiliza la batería del dispositivo.
- No existen mecanismos para avisar en caso de detectar un problema grave en las constantes vitales.
- No existen mecanismos que notifique al usuario que el *wearable* está a punto de quedarse sin batería.

Como se verá dentro del apartado de objetivos y el alcance, lo que se pretende con la nueva aplicación móvil es eliminar estas limitaciones que tienen las actuales aplicaciones.

Por último y para acabar el apartado, se abandona el terreno de las aplicaciones móviles para centrarnos en aplicaciones sanitarias relacionadas con el cuidado de pacientes mediante el uso de *wearables*.

En el año 2010, la universidad Rey Juan Carlos impulsó un proyecto para controlar las constantes vitales y tener localizados en todo momento a todos los pacientes del hospital. La idea principal en la que se basa el proyecto LOBIN [9], es tener una camiseta con un conjunto de sensores y que mediante Wifi comunique esta información al sistema central del Hospital La Paz, situado en Madrid. Principalmente esta camiseta está dirigida a aquellos pacientes con problemas de corazón, aunque la misma es capaz de medir otros factores como la temperatura del paciente o detección y alarma en caso de detectar que el paciente abandona la zona de la que le fue asignada.

El proyecto LOBIN se centra principalmente en la arquitectura del sistema (desde la implementación de la camiseta a la comunicación con el sistema central) y no en la aplicación en sí, pero si hay indicaciones de que como se tenía que desarrollar la aplicación y de lo que tenía que contener.



Fig.5 – Interfaz aplicación proyecto LOBIN. Fuente ([9])

La aplicación consiste en una aplicación Java con varias ventanas, las cuales cada una tiene una función muy específica. En la ventana principal se ve el árbol de pacientes y un mapa del hospital donde se localizan los pacientes. Cuando se pulsa sobre el paciente, se abre una nueva ventana con las constantes del paciente.

Objetivos y Alcance:

Como se ha visto en el apartado anterior, la mayoría de las aplicaciones para dispositivos móviles que se han desarrollado se han diseñado de forma nativa e utilizan *Bluetooth* para vincularse con el proyecto.

La idea del presente TFM es enfocar el desarrollo de la aplicación bajo otro punto de vista. La idea principal es desarrollar una aplicación para dispositivos móviles que pueda ser ejecutado bajo distintas plataformas, es decir, el desarrollo no va a ser nativo, ya que de lo contrario requeriría de un desarrollo por cada una de las plataformas. Por ello en primer lugar, la nueva aplicación móvil será una Web App que pueda ser fácilmente producida para distintas plataformas. Con ello se consigue que con un solo desarrollo conseguir tenerla para distintas plataformas. En principio, se desarrollará y se testeará para la plataforma *Android*, aunque con el mismo código implementado se podrá producir una versión para la plataforma *IOS*.

La Web App será implementada bajo el framework *IONIC*. La versión de *IONIC* sobre la que se va a trabajar es la 4.0 que trabaja bajo Angular 5.0. Como entorno de ejecución se utilizará *NODE* y gestión de paquetes *NPM*. Además, en la medida de lo posible, se van a utilizar componentes que permitan gestionar de manera eficiente y que permita la reutilización de código, como ejemplo de ello se puede mencionar el uso de los preprocesadores *SCSS* o el uso de *minify* para las entregas finales. Además, para el control de versiones y el reporte de los bugs detectados se utilizará como repositorio de versiones *Git*.

El requisito fundamental de esta app será obtener los datos que se van obteniendo de la camiseta, pero la principal diferencia es que las mismas serán obtenidas a través de un servidor en vez de ser obtenidas a través del *Bluetooth*.

Las principales funcionalidades que debe cumplir la aplicación son las siguientes:

- Debe permitir al usuario autenticarse en el sistema. Una vez autenticado en el sistema, los datos del resto de apartados deben mostrarse únicamente información de dicho usuario.
- Debe permitir al usuario registrar una cuenta.
- Debe listar el conjunto de camisetas que tiene el usuario registradas en la aplicación.
- Debe permitir añadir, modificar y eliminar camisetas.
- Debe mostrar el último valor obtenido por la camiseta para las siguientes constantes vitales.
 - Frecuencia cardiaca.
 - Sudoración.
 - Temperatura.
- Debe mostrar los datos de las constantes vitales en un periodo de tiempo mediante gráficas de las siguientes constantes vitales.
 - Frecuencia cardiaca.
 - Sudoración.
 - Temperatura.
- Debe permitir definir un umbral de notificación para una o más de las constantes vitales o un tiempo en el que no se reciban datos.

- Debe notificar al usuario si alguno de los umbrales definidos se supera.
- Debe notificar al usuario si la persona que lleva la camiseta ha sufrido una caída.
- Debe notificar al usuario si la batería de la camiseta es baja.
- Debe permitir al usuario compartir los datos en tiempo real o los datos del conjunto histórico de datos.

Todas estas funcionalidades deberán ser testeadas al finalizar sus respectivos desarrollos, así como las entregas que se hagan al cliente final. En este proyecto es imposible reunir a un conjunto de testers bajo un mismo sitio para la realización de las pruebas, por lo que se va a aplicar otro enfoque. Para conocer en todo momento los errores y excepciones que se puedan producir mientras los usuarios prueban la aplicación, se utilizará el servicio *Crashlytics* de *Firebase* [10]. Este servicio permite al desarrollador consultar los errores y excepciones producidos por la aplicación móvil y da información del dispositivo. Cuando un desarrollo este completo, simplemente se subirá la versión a *Google Play* como versión *Alpha* para los testers.

Por último y para la realización de las pruebas finales y la entrega del proyecto al cliente; dado que no se va a disponer del prototipo de la camiseta hasta principios del 2020, se generará un algoritmo en el servidor que genere la información como si se hubiera obtenido de la camiseta y se transfiera a la aplicación móvil. Estas camisetas serán registradas como camisetas virtuales y el origen de los datos será el propio servicio *backend* al que se conectaran los dispositivos móviles.

Planificación:

Para desarrollar la aplicación, se va a utilizar una aproximación PMBOK, enfocándonos en una serie de áreas que tal y como se indica en [11] y manteniendo un equilibrio entre todas ellas. Sobre todo, se va a enfocar en los aspectos de *calidad*, *tiempo* y *riesgos*, que son las áreas más importantes en el proyecto, pero sin dejar de lado el resto de los aspectos. Para conseguir dicho equilibrio se seguirá una filosofía basada en *SCRUM* [12] en la cual los *sprint* tendrán 2 semanas de duración. En cada uno de los *sprint* se abarcará el desarrollo una o 2 pantallas, según el coste de implementar dichas pantallas. Si durante un *sprint* se desarrolla más de 2 pantallas, aquellas tareas que requieran interactuar con los *early adopters*, se realizarán en el mismo periodo de tiempo. Estas tareas se indican con un asterisco. El desarrollo de cada una de las pantallas consistirá en los siguientes pasos:

- Entrevista con los *early adopters* para determinar cuáles son las necesidades para dicha pantalla. (*)
- Realización de un primer boceto en *Balsamiq*.
- Los *early adopters* testean y validan el boceto en *Balsamiq*. (*)
- Se implementa la pantalla junto con la funcionalidad definida.
- Se realiza un proceso de pruebas, corrigiendo los errores detectados. (*)
- Documentación

Existirán un total de 6 *sprint* distintos y habrá una entrega de código y documentación cada 2 *sprint*. La única excepción será el 6º *sprint* que no durará exactamente 2 semanas, sino una semana y media para adecuarlo al calendario de la entrega final del proyecto. Además de los 6 *sprint*, habrá un *sprint 0* que se iniciará con anterioridad a los 6 *sprint* y que tan sólo durará una semana. En este *sprint* inicial se hará una colección

inicial de todas las tareas y requisitos que debe cumplir el proyecto en general, que permita generar un *product backlog* inicial.

Al inicio del proyecto, no se conocen todas las características que va a tener la aplicación móvil ya que se desconoce todas las constantes vitales que van a medir la camiseta, por lo que se dejará margen de maniobra para poder definir requisitos adicionales, eliminar los que ya no sean necesarios o correcciones del proyecto a medida que se vayan avanzado. Durante los primeros *sprint* se irán especificando implementando las funcionalidades más críticas del programa (como por ejemplo la ventana en la que se visualizan las constantes vitales) y en los siguientes *sprint* se implementarán los requisitos menos importantes de la aplicación (notificaciones, dar de alta la camiseta, etc.). Se ha dejado hueco en los últimos *sprint* para cubrir posibles desviaciones en el proceso, cubrir nuevos requisitos a cubrir o corregir el *feedback* que provenga del cliente.

Cada 2 *sprint* habrá un hito que coincidirá con una entrega al cliente para que pueda evaluar una aplicación funcional y pueda ir reconduciendo el contenido de las siguientes entregas según su criterio. El 2º *sprint* después de cada hito (el *sprint* previo al siguiente hito), se definirá una tarea que consistirá en aplicar aquellas correcciones o modificaciones que vengan impulsadas por el cliente. El 3º hito (tras 6 *sprint*) se realizará la entrega final del proyecto y por tanto se procederá a pasar al mantenimiento del mismo.

En la figura 6, se expone una pequeña parte del diagrama de Gantt inicial con los *sprint* planificados. El mismo se puede encontrar completo para todo el ámbito del proyecto en el anexo A. Los *sprint* están representados en rojo y las tareas principales del proyecto en morado. Las subtareas que conforman las tareas principales se encuentran de color azul. Los hitos del proyecto corresponden con las entregas del proyecto (PEC 2, 3, 4 y 5) se encuentran de color amarillo. Por último, la defensa del TFM se encuentra de color verde, incluyéndose dentro de la planificación del proyecto.

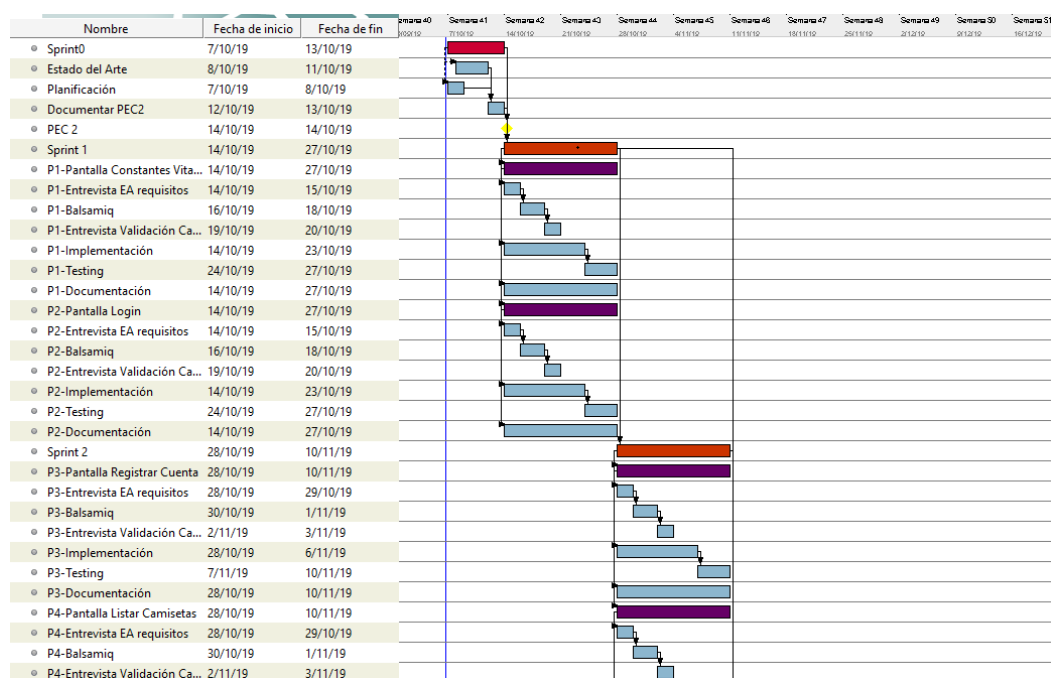


Fig.6 – Previsualización planificación del proyecto

Bibliografía:

- [1] - Dave Evans. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. CISCO White Paper [Internet], 2011 [Fecha de publicación 01/04/2011]. Disponible en https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf
- [2] - Man, S. Wearable computing: a first step toward personal imaging. Computer, vol. 30, no. 2, pp. 25-32, Feb. 1997.
- [3] - Conor O'Quigley, C. ; Sabourin, M. ; Coyle, S. ; Connolly, J. ; Condall, J. ; Curran, K. ; Corcoran, B. ; Diamond, D. Characteristics of a Piezo-Resistive Fabric Stretch Sensor Glove for Home-Monitoring of Rheumatoid Arthritis. 2014 11th International Conference on Wearable and Implantable Body Sensor Networks Workshops, pp. 23-26, June 2014
- [4] - Google LLC [Internet]. 2019. Google Play. [Consultado el: 07/10/2019]. Disponible en <https://play.google.com/>
- [5] - Google LLC [Internet]. 2019. Salud Conectada. [Consultado el: 07/10/2019]. Disponible en <https://play.google.com/store/apps/details?id=es.sanitas.hosp.saludconectada&gl=ES>
- [6] - Google LLC [Internet]. 2019. Heart Rate OS. Android Watch. [Consultado el: 07/10/2019]. Disponible en <https://play.google.com/store/apps/details?id=com.iwork.wearable.heartratesync2&gl=ES>
- [7] - Google LLC [Internet]. 2019. Health Wearable. [Consultado el: 07/10/2019]. Disponible en <https://play.google.com/store/apps/details?id=com.analogdevices.healthwearable&gl=ES>
- [8] - Broeders J. Transition from Wearable to Medical Devices. Analog Devices [Internet]. [Consultado el: 08/10/2019]. Disponible en <https://www.analog.com/en/technical-articles/transition-from-wearable-to-medical-device.html>
- [9] - López, Gregorio; Custodia, Victor; Moreno, José Ignacio. LOBIN: E-Textile and Wireless-Sensor-Network-Based Platform for Healthcare Monitoring in Future Hospital Environments. IEEE Transactions on Information Technology in Biomedicine, vol.14, issue 6, Nov. 2010
- [10] - Google LLC [Internet]. 2019. Crashlytics Firebase. [Consultado el: 10/10/2019]. Disponible en <https://firebase.google.com/docs/crashlytics/?hl=es-419>
- [11] - Rodríguez, José Ramón. El trabajo final como proyecto. FUOC. Fundació para la Universitat Oberta de Catalunya. [Consultado el: 02/10/2019]
- [12] - Scrum.org [Internet]. 2019. The home of the Scrum. [Consultado el: 09/10/2019]. Disponible en <https://www.scrum.org/>

[13] – Clarisó, Robert. Introducció al treball final. FUOC. Fundació para la Universitat Oberta de Catalunya [Consultado el 03/10/2019].

[14] – Rodríguez, José Ramón. La gestió del projecte a lo largo del treball final. FUOC. Fundació para la Universitat Oberta de Catalunya [Consultado el 04/10/2019]