

PROYECTO DE FINAL DE GRADO

Universidad Politécnica de Catalunya
Facultad de Informática de Barcelona
Ingeniería del Software



MÉTRICAS DE RENDIMIENTO AERÓBICO

Autor: Yeison Melo

Director: Pau Fonseca

Departamento de Estadística e Investigación Operativa

Junio, 2016

Agradezco el apoyo incondicional de mi familia
desde el momento en que empezó mi pasión por el mundo de la tecnología.
Pilar principal e inigualable.

Agradezco a mi madre por el apoyo y confianzas
en mis momentos de flaqueza
cuando solo quedaba yo por ir a dormir en casa.

Agradezco a amigos y compañeros que han estado a mi lado
durante estos años de desafío y superación personal. Ofreciendo
su apoyo tanto moral como intelectual.

Agradezco a los que nunca creyeron y no tuvieron
ningún problema en que se notara.

Tabla de ilustraciones	7
Glosario	9
1 Formulación del problema.....	11
1.1 Contexto.....	11
1.2 Introducción.....	11
1.2.1 Necesidad de métricas.....	12
1.2.2 Necesidad de flexibilidad	12
1.3 Resumen.....	12
1.4 Estado del arte.....	13
1.4.1 Introducción	13
1.4.2 Samsung S Health	13
1.4.3 Apple Watch	14
1.4.4 Microsoft	15
1.4.5 Polar Loop Crystal	15
1.4.6 Suunto.....	15
1.4.7 Garmin	16
1.4.8 Conclusión.....	16
1.5 Partes Interesadas	17
1.5.1 Usuario	17
1.5.2 Equipo del proyecto.....	17
1.5.3 Garmin	18
1.5.4 Corredoras de datos (data brokers).....	18
1.5.5 Anunciantes.....	18
1.6 Objetivos	18
1.6.1 Forever Fit Smartwatch	19
1.6.2 Forever Fit Web	19
1.6.3 Aclaraciones	19
2 Análisis de requisitos	20
2.1 Requisitos funcionales	20
2.1.1 <i>Smartwatch</i> Garmin Epix	20

2.1.2	Plataforma Web.....	22
2.1.3	Aclaraciones	24
2.2	Requisitos no funcionales.....	24
2.2.1	Requisitos de velocidad y latencia.....	24
2.2.2	Requisitos de usabilidad	24
2.2.3	Requisitos de seguridad	25
2.2.4	Requisitos de privacidad	25
2.2.5	Requisitos de integridad	25
2.2.6	Requisitos de comprensión y cortesía	25
2.2.7	Requisitos de precisión o exactitud	26
2.2.8	Requisitos de fiabilidad y disponibilidad.....	26
3	Especificación	27
3.1	Actores	27
3.2	Diagrama de casos de uso	27
3.3	Casos de uso	28
3.3.1	Casos de uso Forever Fit Smartwatch	28
3.3.1.1	CU#0001 Seleccionar test	28
3.3.1.2	CU#0002 Iniciar Vo2max Speed Test.....	29
3.3.1.3	CU#0003 Detener Vo2max speed Test	29
3.3.1.4	CU#0004 Continuar Vo2max speed Test	29
3.3.1.5	CU#0005 Iniciar One Mile Walk Test.....	29
3.3.1.6	CU#0006 Iniciar One Half Mile Run Test.....	30
3.3.1.7	CU#0007 Cancelar test en curso	30
3.3.1.8	CU#0008 Salir de la aplicación.....	30
3.3.2	Casos de uso Forever Fit Web	30
3.3.2.1	CU#0009 Registrar usuario.....	30
3.3.2.2	CU#0010 Iniciar sesión	31
3.3.2.3	CU#0011 Modificar datos del usuario	31
3.3.2.4	CU#0012 Cerrar sesión	32
3.3.2.5	CU#0013 Ver listado archivos .FIT	32
3.3.2.6	CU#0014 Subir archivo .FIT	32
3.3.2.7	CU#0015 Buscar archivos .FIT por nombre.....	33
3.3.2.8	CU#0016 Ordenar listado archivos .FIT.....	33
3.3.2.9	CU#0017 Eliminar registro archivo .FIT.....	34

3.3.2.10	CU#0018 Ver métricas archivo .FIT.....	34
4	Modelo Conceptual.....	35
4.1	Forever Fit Smartwatch.....	35
4.2	Forever Fit Web.....	36
5	Diseño.....	37
5.1	Arquitectura del sistema.....	37
5.1.1	Forever Fit Smartwatch.....	37
5.1.1.1	Pantallas.....	38
5.1.1.2	Diagrama de clases.....	40
5.1.2	Forever Fit Web.....	41
5.1.2.1	Modelo.....	43
5.1.2.2	Vista.....	44
5.1.2.2.1	Mapa de navegación.....	44
5.1.2.2.2	Pantallas.....	45
5.1.2.2.2.1	Inicio.....	45
5.1.2.2.2.2	Registrarme.....	46
5.1.2.2.2.3	Identificarme.....	47
5.1.2.2.2.4	Editar información usuario.....	48
5.1.2.2.2.5	Activities.....	49
5.1.2.2.2.6	Registros Activity.....	49
5.1.2.2.2.7	Error.....	50
5.1.2.3	Controlador.....	51
5.1.2.4	Diagrama de clases final.....	52
5.2	Acceso a los datos.....	53
5.2.1	Forever Fit Smartwatch.....	53
5.2.2	Forever Fit Web.....	54
6	Alcance.....	55
6.1	Introducción.....	55
6.2	Posibles Obstáculos.....	55
6.2.1	Restricción temporal.....	55
6.2.2	Tecnologías nuevas.....	56
6.2.3	Pruebas y Prototipo.....	56
6.3	Metodología y rigor.....	56
6.3.1	<i>Fita inicial</i>	56

6.3.1.1	Método de validación	57
6.3.2	<i>Fita de seguiment</i>	57
6.3.3	<i>Fita Final</i>	58
6.4	Análisis de alternativas.....	58
6.4.1	Exportación/Importación de ficheros	58
6.4.2	Base De Datos	59
6.4.2.1	Resumen:	60
6.4.3	IDE de desarrollo web	60
6.5	Identificación de Leyes y regulaciones	61
7	Planificación temporal.....	63
7.1	Introducción.....	63
7.1.1	Fita inicial	63
7.1.1.1	Fases	63
7.1.1.1.1	Fase Inicial.....	63
7.1.1.1.2	Fase de especificación	64
7.1.1.1.3	Fase de desarrollo	64
7.1.1.1.3.1	Iteración métricas.....	64
7.1.1.1.3.2	Iteración Generales.....	64
7.1.1.1.3.3	Fase final	65
7.1.1.1.4	Estimación en horas	65
7.1.1.1.5	Desviaciones y plan de acción.....	66
7.1.1.2	Recursos.....	66
7.1.1.3	Diagrama de Gantt.....	66
7.1.2	<i>Fita de seguiment</i>	67
7.1.2.1	Situación actual del proyecto.....	67
7.1.2.2	Estimación en horas	68
7.1.2.3	Resumen	69
7.1.3	<i>Fita Final</i>	69
8	Gestión económica.....	70
8.1	Introducción.....	70
8.1.1	Identificación costes	70
8.1.1.1	Costes directos	70
8.1.1.2	Costes indirectos	71
8.1.1.3	Plan de contingencia	71

8.1.2	Fita inicial	71
8.1.2.1	Costes directos	71
8.1.2.2	Costes Indirectos	73
8.1.2.3	Plan de contingencia	73
8.1.2.4	Presupuesto final.....	73
8.1.3	<i>Fita de seguiment</i>	74
8.1.3.1	Costes directos	74
8.1.3.2	Costes indirectos	75
8.1.3.3	Plan de contingencia	75
8.1.3.4	Presupuesto final.....	76
8.1.4	<i>Fita Final</i>	76
9	Sostenibilidad y compromiso social.....	77
9.1	Económica.....	77
9.2	Social.....	77
9.3	Ambiental.....	77
10	Justificación de las competencias técnicas.....	79
10.1	Descripción de los objetivos	79
10.2	Resumen del alcance.....	80
10.3	Relación con las asignaturas cursadas	80
10.4	Adecuación a las características de la especialidad de Ingeniería del Software	81
10.5	Competencias técnicas	82
11	Conclusiones	84
12	Trabajo Futuro.....	86
13	Referencias.....	87
14	Apéndice: Manual de utilización de la aplicación.....	90
14.1	Forever Fit Smartwatch	90
14.1.1	Uso general	90
14.1.1.1	Resumen	91
14.1.2	Vo2Max Speed test.....	91
14.1.3	One Mile Walk test	91
14.1.4	One Half Mile Run Test	92
14.2	Forever Fit Smartwatch	93

Tabla de ilustraciones

Ilustración 1. Interfaz Samsung health.....	14
Ilustración 2. Interfaz Apple watchIlustración.....	14
Ilustración 3. Dispositivos wereables Garmin.....	16
Ilustración 4. Diagrama casos de uso FFS.....	27
Ilustración 5. Diagrama casos de uso FFW.....	28
Ilustración 6. Modelo conceptual FFS.....	35
Ilustración 7. Modelo conceptual FFW.....	36
Ilustración 8. Diagrama arquitectura Forever Fit Smartwatch.....	37
Ilustración 9. . Captura del emulador del smartwatch Garmin Epix, estado inicial de las pantallas.....	38
Ilustración 10. . Vo2Max Speed Test en curso y en modo estimación continua.....	38
Ilustración 11. Pantalla para introducir FC Máxima del usuario.....	39
Ilustración 12. One Mile Walk Test en ejecución y estimación alcanzada.....	39
Ilustración 13. One Half Mile Run Test en ejecución y estimación alcanzada.....	39
Ilustración 14. Diagrama de clases de Forever Fit Smartwatch.....	40
Ilustración 15. Esquema arquitectura lógica Forver Fit Web.....	41
Ilustración 16. Estructura ficheros FFW: vistas, controladores y modelos.....	42
Ilustración 17. Diagrama de clases de modelos.....	43
Ilustración 18. Mapa de navegación Forever Fit Web.....	44
Ilustración 19. Pantalla inicio usuario no identificado.....	45
Ilustración 20. Pantalla inicio usuario identificado.....	46
Ilustración 21. Pantalla registro usuario.....	46
Ilustración 22. Pantalla registro usuario con error en los campos.....	47
Ilustración 23. Pantalla de identificación del usuario.....	47
Ilustración 24. Pantalla editar información usuario.....	48
Ilustración 25. Pantalla editar información usuario comprobación campos.....	48
Ilustración 26. Pantalla activities.....	49
Ilustración 27. Pantalla registros activity.....	50
Ilustración 28. Pantalla error.....	50
Ilustración 29. Diagrama de clases de controladores.....	51
Ilustración 30. Diagrama final de clases FFW.....	52
Ilustración 31. Mapa de código FFW.....	53
Ilustración 32. Estructura base de datos SQL FFW.....	54
Ilustración 33. Estructura MongoDB ejemplo de documento activities y documento activityRecord.....	54
Ilustración 34. Esquema metologia de trabajo.....	56
Ilustración 35. Pantalla de estado inicial de los diferentes test.....	90
Ilustración 36. Pantalla activities FFW.....	93
Ilustración 37. Pantalla detalle activity FFW.....	94
Ilustración 38. Pantalla editar información usuario FFW.....	94

Glosario

A continuación se presentan las definiciones de los conceptos más técnicos del proyecto o que se utilizarán con frecuencia:

Capacidad aeróbica [1] [2, p. 1]: consiste en la capacidad del organismo (corazón, vasos sanguíneos y pulmones) para funcionar eficientemente y llevar actividades sostenidas con poco esfuerzo, poca fatiga, y con una recuperación rápida (ejercicio aeróbico). Fisiológicamente, es la habilidad de producir un trabajo utilizando oxígeno como combustible. La capacidad aeróbica es una función del volumen máximo de oxígeno (VO₂ máx), el cual representa la capacidad máxima del organismo para metabolizar el oxígeno en la sangre (máximo transporte de oxígeno que nuestro organismo puede transportar en un minuto o más).

VozMax Speed: es una estimación de la velocidad que el deportistas alcanzarían al llegar a su Vo2Max, es decir la velocidad que alcanzaría al llegar a su máximo rendimiento.

Kit de Desarrollo de Software [3]: SDK (siglas en inglés de *software development kit*) se trata de un conjunto de herramientas de desarrollo de software que permiten al programador o desarrollador de software crear aplicaciones para un sistema en concreto.

Sprint: porción de tiempo de la fase de desarrollo dedicada a implementar un subconjunto de funcionalidades del sistema.

Smartwatch: dispositivo de digital portátil usado en la muñeca del usuario que ofrece funcionalidad y capacidades similares a las de un teléfono inteligente. Entre las funcionalidades más frecuentes se encuentra sensores para medir las constantes vitales (pulso, ritmo cardiaco..etc.), GPS, seguimiento de actividades etc. Para este proyecto en concreto se utilizara el modelo Garmin Epix, por lo tanto a menos que se indique lo contrario estaremos hablando de este modelo en concreto.

Weareble: se entiende por dispositivo *weareble* al conjunto de aparatos y dispositivos electrónicos que se incorporan en alguna parte de nuestro cuerpo (muñeca, cuello, zapatillas..etc.) interactuando con el usuario.

Application Programming Interface: API, se trata del conjunto de subrutinas, funciones, procedimientos y métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Forever Fit Smartwatch: FFS, hace referencia a la aplicación del smartwatch Garmin Epix que se desarrollará para el proyecto.

Forever Fit Web: FFW, hace referencia a la página web del sistema que se desarrollará para el proyecto.

Forever Fit: FF, hace referencia al conjunto de Forever Fit Smartwatch y Forever Fit Web.

Activity: se entiende por activity el conjunto de información extraída de los ficheros .FIT y que FFW utiliza para calcular las métricas. Un activity está formado por su información identificativa como puede ser su nombre, fecha de creación del fichero .fit, ..etc. y por un conjunto de activity records o registros de activity.

Activity Record o registro de activity: tupla que representa las constantes vitales del usuario en un momento dado y pertenece a un activity. Estas constantes vitales son las que utiliza el sistema para calcular las métricas: momento, frecuencia cardiaca, velocidad, distancia recorrida, latitud y longitud.

Compatibilidad con terceros: en el contexto de este proyecto, la compatibilidad hace referencia a que los archivos generados por el dispositivo smartwatch Garmin Epix puedan ser utilizados por otras aplicaciones además de la web que será desarrollada. Análogamente con el portal web se refiere a que pueda importar ficheros generados por otras aplicaciones.

Formulación del problema

1.1 CONTEXTO

Este proyecto se trata de un Trabajo de Final De grado de la especialidad de Ingeniería del Software para la Facultad de informática de Barcelona, Universidad Politécnico de Catalunya. La razón de este proyecto, en modalidad A, es sintetizar todos los estudios realizados a lo largo del grado.

Al final del proyecto se pretende conseguir un sistema que apoye el entrenamiento de deportistas utilizando el *smartwatch* Garmin Epix. La idea principal es que el sistema informe al usuario de su capacidad aeróbica, poder relacionar las variaciones de esta medida con el entrenamiento que se está realizando, exportar e importar estos datos con una plataforma web y que el sistema tenga compatibilidad con terceros.

Se ha seleccionado el dispositivo Epix ya que en el departamento del director del proyecto se disponía de uno, en un momento inicial.

1.2 INTRODUCCIÓN

El ejercicio físico enmarca las actividades corporales que se hacen para conservar, recuperar y/o mejorar el rendimiento, la salud general y el bien estar de las personas.

En los comienzos de la humanidad las capacidades físicas eran determinantes para la supervivencia del individuo desde la caza para asegurar el alimento, la resistencia a condiciones extremas o la lucha por la supervivencia y el liderazgo. A medida que hemos evolucionado los objetivos para la mejora de dichas capacidades se han ido ampliando y cambiando de prioridad. Situándonos en la época actual donde los principales objetivos del ejercicio físico son los relacionados con el bien estar y el ocio.

Es bien sabido que con el entrenamiento adecuado las capacidades físicas pueden mejorar, pero a medida que nos acercamos a nuestro límite genético máximo, es más difícil continuar mejorando y encontrar el entrenamiento que lleve al deportista al siguiente nivel. Para continuar aumentando las capacidades de rendimiento físico es de gran ayuda conectar con precisión y de forma cuantitativa causa y efecto, es decir, determinar cómo está afectando realmente el entrenamiento las capacidades del sujeto.

Vinculando entrenamiento y variación en la capacidad de rendimiento es posible valuar de forma continua la efectividad de los ejercicios que se están realizando y descartar los menos efectivos mientras mantenemos y aumentamos la práctica de los más efectivos.

Por otro lado, es importante que dicha relación entre entrenamiento y capacidad pueda ser realizada de forma automatizada, sencilla y lo más inmediata posible. De forma que los cambios y ajustes en el entrenamiento se puedan realizar en el tiempo mínimo.

1.2.1 NECESIDAD DE MÉTRICAS

Una vez identificada la necesidad de relacionar entrenamiento y rendimiento. El siguiente paso es establecer de forma cuantitativa dicha relación.

Las métricas son necesarias en todas partes. Ya sea para llevar a cabo un seguimiento de la evolución de ciertos valores, fijar objetivos o límites, las métricas son importantes. Con ellas sabemos dónde estamos, hacia dónde nos dirigimos, lo que podemos esperar, o si debemos rectificar; En resumen, las métricas hacen las cosas tangibles. Relacionando entrenamiento y métricas es posible establecer objetivos alcanzables, seguir la evolución del rendimiento y aplicar las modificaciones necesarias.

1.2.2 NECESIDAD DE FLEXIBILIDAD

Para utilizar de forma efectiva las métricas y realizar las modificaciones necesarias en función de estas es preciso que dichas medidas sean flexibles, es decir, tanto su toma como seguimiento sean rápidas, fáciles de ejecutar e interpretar y representativas del estado actual del deportista.

Pocas utilidades tendrán las métricas escogidas si el sujeto necesita grandes lapsos de tiempo para medirlas, podemos encontrar el caso en que ya no sean representativas del estado actual si se ha tomado demasiado tiempo para su procesamiento. De la misma forma deben ser las más sencillas posibles de interpretar para garantizar su uso generalizado sin necesidad de grandes conocimientos o profundizar demasiado en el trasfondo de los datos.

Las métricas elegidas para el sistema son por un lado el VO_2Max , el cual es estimador estrella para determinar el rendimiento aeróbico de los deportistas, y el Vo_2Max Speed para ofrecer una métrica orientada a corredores en el sistema. Finalmente, una vez obtenidas estas métricas se debe tener la seguridad de que estarán disponibles en todo momento. Por lo que deben ser persistentes y accesibles sin importar donde se encuentre el deportista.

1.3 RESUMEN

Una vez expuestos los puntos anteriores, podemos tener una primera visión de cuáles serán las funciones y objetivos del sistema a realizar. Se desea contar con un software apoyado en métricas capaces de medir el rendimiento aeróbico del deportista, detectar el efecto del entrenamiento sobre estas y ayudarlos a tener un seguimiento su evolución. Además, este sistema debe permitir un uso generalizado y accesible a los usuarios en cualquier momento.

Es vital entender a partir de ahora, que no se desea un sistema que simplemente recopile información sobre la actividad del usuario léase pasos, distancia recorrida en un día, ritmo cardiaco o diga que comer o ejercicios realizar. Lo que se desea es saber cuál es el nivel de rendimiento aeróbico del usuario, las variaciones que puede estar produciendo el entrenamiento sobre este, tener un seguimiento de su evolución y analizarlo. Por otro lado, también se busca esta información pueda ser exportada e importada con facilidad.

La estimación del Vo2Max y Vo2Max Speed se realizará con el dispositivo *smartwatch* Garmin Epix. Por otro lado, la importación e importación de los datos se realizará con una plataforma web que también será desarrollada, aumentando de esta forma la disponibilidad de los datos y su potencial de análisis. En cuanto a los ficheros generados por el sistema se utilizará el formato FIT, primeramente porque es el único formato con el que Garmin permite a terceros generar ficheros y segundo porque es de los archivos de seguimiento más completos en cuanto a seguimiento deportivo.

Es importante destacar que se pretende que la exportación e importación de datos entre la aplicación del dispositivo *smartwatch* y el portal web sea compatible con terceros, es decir, que se establezca un formato de datos que pueda ser seguido por otros desarrolladores. De esta forma será posible exportar los datos del *smartwatch*, en aplicaciones de terceros y/o importar datos en el portal web no necesariamente creados con la aplicación para el dispositivo Epix que se desarrollará.

1.4 ESTADO DEL ARTE

A continuación, se presenta el estado del arte de proyecto cuya finalidad es mostrar cómo se resuelve el problema actualmente o en qué estado se encuentra actualmente y por otro las tecnologías utilizadas en este proyecto para su resolución.

1.4.1 INTRODUCCIÓN

Actualmente existen numerosas aplicaciones y sistemas para el control de las actividades deportivas utilizando las últimas tecnologías. En el caso de nuestro proyecto dicho mercado queda enmarcado por las aplicaciones que usan dispositivos *smartwatch*, en concreto los dispositivos Garmin ya que será sobre un modelo de esta marca con el que se realizará la toma de datos y cálculo de métricas. También realizaremos el estudio sobre otras marcas populares, líderes en el mercado pero sin entrar en detalles sobre cada modelo en ninguno de los casos, esto es debido a que cada marca utiliza el mismo sistema operativo o funcionalidades casi idénticas en los diferentes modelos como se verá a continuación.

1.4.2 SAMSUNG S HEALTH

La empresa Coreana líder mundial en el sistema operativo Android [4], dispone los dispositivos Samsung Gear, los cuales funcionan con el sistema operativo Android Wear sobre el que se pueden ejecutar aplicaciones de diversas finalidades. Es posible instalar aplicaciones de desarrolladores en estos dispositivos pero suelen necesitarse conocimientos de desarrollador, acceso al SDK (Software Developer Kit), permisos de administrador, ..etc [5]. Además al ser una tecnología aun emergente las aplicaciones propias del SO son las que mejor explotan sus características [6].

S health es la aplicación que Android Wear dispone para el entrenamiento físico: establecer objetivos, realizar seguimiento de actividades diarias y alcanzar nuevos hitos. La aplicación permite tener un control de dieta, sueño, monitoreo de constantes vitales..etc e instalar algunas expansiones.



Ilustración 1. Interfaz Samsung health

Respecto al monitoreo de actividades y seguimiento de objetivos es una aplicación bastante completa. No obstante, más allá de estas funcionalidades no se detectan métricas que informen sobre las capacidades aeróbicas del usuario o poder comparar su rendimiento con el de otra persona. Por ejemplo, si el objetivo del día marcado era recorrer 25 kilómetros y dos usuarios lo han cumplido no es posible establecer cual tiene unas condiciones físicas superiores o si alguno no la ha podido cumplirlo, no tiene que ser por estar menos en forma.

Respecto a la persistencia de los datos estos se pueden mantener sincronizados siempre y cuando sea en dispositivos Samsung identificados con la misma cuenta.

1.4.3 APPLE WATCH

En el caso de la empresa Apple nos encontramos con el Applewatch, único *smartwatch* de la marca hasta el momento, funcionando con el sistema operativo Watch OS. Entre las funcionalidades que ofrece el dispositivo, en lo que respecta, a las relacionadas con la salud y forma física encontramos un escenario muy similar al de Samsung. Un dispositivo con pocas aplicaciones [7], aun menos relacionadas con salud y forma física para seguimiento de objetivos, dieta, actividad realiza durante, plan de entrenamiento el día y sincronización con otros productos Apple.



Ilustración 2. Interfaz Apple watch

En concreto la aplicación por defecto para la salud y forma física es Entreno. Ofrece datos actualizados y al instante sobre el tiempo que el usuario se ha ejercitado, la distancia recorrida, las calorías quemadas, el ritmo y la velocidad. De la misma forma que con las funcionalidades con la oferta de Samsung, se detectan las mismas carencias en el dispositivo y sus aplicaciones. No hay forma de saber el rendimiento aeróbico del usuario ni como está afectando a sus capacidades el entrenamiento o métricas precisas de comparación con otros deportistas.

1.4.4 MICROSOFT

La multinacional estadounidense cuenta con tres dispositivos *wearables*: Microsoft Band, Fiband Blaze y Fitbit Wristband, todos orientados al seguimiento deportivo y a la mejora del rendimiento de los deportistas. Fitbit Blaze es un *smartwatch* mientras que los otros dos son pulseras de seguimiento. Todos cuenta con el software Microsoft Health que es el encargado de manejar todas las funcionalidades y datos relacionados con la actividad física y salud de los usuarios.

Es posible sincronizar toda la información recopilada en la nube para ser administrada desde la aplicación online de Microsoft Health. Además esta plataforma cuenta con compatibilidad con terceros, mediante una API abierta por lo que otros fabricantes pueden utilizar el portal de Microsoft para importar sus datos. Microsoft Health también proporciona aplicaciones móviles para dispositivos con Windows Phone, iOS y Android. Como característica importante a destacar en el marco de este proyecto, encontramos que el dispositivo MS Band permite estimar el VO₂Max. Un punto importante en contra de estos dispositivos es que por el momento solo ejecutan las aplicaciones que traen de fábrica no es posible desarrollar nuevas ni instalarlas por el momento, por lo tanto las posibilidades de expansión son nulas.

1.4.5 POLAR LOOP CRYSTAL

Este fabricante se especializa en el desarrollo de dispositivos de seguimiento de actividad. Ofrece *wearables* muy completos para esta finalidad y una amplia gama que cuenta con pulseras de seguimiento y dispositivos *smartwatch* entre otros. De la misma forma que la mayoría de fabricantes ofrece la posibilidad de sincronizar toda la información recogida con su portal web para un mayor análisis y seguimiento de las actividades, en este caso a diferencia de Microsoft, dicha plataforma web no es compatible con terceros. La empresa ofrece índices de rendimiento aeróbico propios además del VO₂Max. Por otro lado las posibilidades de expansión son prácticamente nulas ya que los dispositivos solo permiten la ejecución de sus aplicaciones de fábrica, no es posible desarrollar ni instalar otras.

1.4.6 SUUNTO

En esta empresa encontramos un patrón de dispositivos bastante parecido a Garmin cuenta con más de 100 *smartwatches* con funcionalidades muy específicas para diferentes actividades deportivas como pueden ser pesca, natación, escalada etc. En los dispositivos se encuentran aplicaciones de seguimiento de actividad, trazado de mapas, métricas de rendimiento aeróbico incluyendo el VO₂Max según el modelo. El fabricante ofrece una aplicación móvil disponible en Android e iOS con la que poder sincronizar la información recopilada. Igual que en Polar Loop Crystal se presenta un sistema sin posibilidad de expansión permitiendo la ejecución únicamente de las aplicaciones de fábrica.

1.4.7 GARMIN

La empresa Garmin dedicada al desarrollo y fabricación de dispositivos GPS cuenta con más de 30 dispositivos *smartwatch* y pulseras inteligentes entre sus productos. A diferencia de las empresas anteriores donde los dispositivos han sido creados para las actividades del día a día dentro de las que el cuidado de la forma física tiene un pequeño papel, los *smartwatch* Garmin están totalmente dirigidos a actividades especializadas: aviación, navegación, natación, *running*..etc. . Como consecuencia de esta diversidad de finalidades los dispositivos pueden presentar diferencias, mayoritariamente de cuatro tipos: controles, sensores, características de la pantalla y aplicaciones.

Garmin cuenta con dispositivos que pueden apoyar al entrenamiento de los usuarios informando con métricas que pueden estimar el rendimiento aeróbico, mantener un seguimiento del mismo y exportarlo; estas son las funcionalidades principales deseadas en la aplicación a desarrollar, como anteriormente se ha expuesto.



Ilustración 3. Dispositivos wereables Garmin

Ahora bien, analizando los *smartwatch* Garmin nos encontramos con las siguientes situaciones:

1. Las aplicaciones de los dispositivos para la estimación del rendimiento aeróbico están definidas para cada uno de ellos, no es posible transferirlas entre dispositivos ni tampoco modificar su programación.
2. Actualmente la sincronización de métricas y datos de los *smartwatch* solo funciona con la comunidad online Garmin Connect. La sincronización puede realizarse mediante conexión USB con ordenadores en los que se encuentre instalada la aplicación Garmin Express o teléfonos móviles con la aplicación Garmin Connect™ Mobile mediante conectividad Bluetooth®. Pese a esto existen APIs y funciones en el SDK para desarrolladores que quieren exportar e importar datos.
3. Las estimaciones del rendimiento aeróbico se realizan en todos los dispositivos a partir del monitor de frecuencia cardiaca, aunque hay más formas de estimar el rendimiento aeróbico. Por lo tanto, los dispositivos sin monitor de frecuencia cardiaca no vienen con esta funcionalidad.

1.4.8 CONCLUSIÓN

Las empresas más populares en el sector de los dispositivos *smartwatch* presentan dispositivos dirigidos a propósitos generales de las actividades cotidianas con una pequeña dedicación al seguimiento deportivo, otras

compañías menos popular disponen de equipos totalmente dirigidos a este objetivo pero con variaciones. La sincronización de datos se realiza exclusivamente con dispositivos de la misma marca por general. Algunos fabricantes ofrecen las métricas deseadas.

En el caso de Microsoft encontramos dispositivos muy completos, contando con una plataforma web de análisis con compatibilidad para terceros, pero dentro de sus dispositivos la información recopilada es solo sincronizable así su propia plataforma web. Esta plataforma a su vez permite exportar la información en formato .csv entre otros. También es destacable que fabricantes como Polar o Suun no permitan el desarrollo de aplicaciones en sus dispositivos limitando drásticamente la expansión de los mismos.

La empresa Garmin al fabricar artículos especializados ofrece un conjunto de aplicaciones que pueden considerarse alternativas a las de este proyecto pero con las limitaciones anteriormente expuestas: no todos los dispositivos cuentan con las métricas deseadas, todos los dispositivos utilizan un único método que es compartido para obtenerla, la sincronización de datos se puede realizar actualmente solo con la plataforma Garmin Connect limitando al usuario a las prestaciones de análisis que esta puedan ofrecer.

La aplicación propuesta en el proyecto ofrecerá diferencias que cubrirán un sector del mercado aun por explotar en los dispositivos *smartwatch*: métodos de cálculo de las métricas diferentes a los ofrecidos hasta ahora por las compañías estudiadas y un formato de exportación e importación de datos abierto, esto dota al sistema de una completa compatibilidad con terceros. Por un lado la información generada por otras aplicaciones y dispositivos pueda ser analizada por la plataforma web desarrollada utilizando el formato establecido. De la misma forma la información generada por la aplicación en el dispositivo Garmin podrá ser analizada por otras aplicaciones.

1.5 PARTES INTERESADAS

Se entiende como partes interesadas aquellas personas u organizaciones que pueden verse afectadas por el proyecto o afectar las actividades del mismo. En los siguientes apartados se exponen cada una de ellas.

1.5.1 USUARIO

Los usuarios de la aplicación serán considerados principalmente deportistas y/o entrenadores. Se entiende como deportista la persona que realiza algún deporte, profesionalmente o por afición, por otro lado, se entiende como entrenador aquella persona que se dedica a la preparación y a la dirección técnica de un deportista.

Tanto los deportistas como entrenadores serán los más beneficiados directamente. Haciendo uso del sistema podrán realizar un seguimiento de cómo está afectando el entrenamiento actual al rendimiento del deportista en el caso del entrenador o a sí mismo en caso del deportista, y realizar las modificaciones convenientes. Tendrán la capacidad seguir la evolución de las métricas de rendimiento y el sistema las analizará para su interpretación.

1.5.2 EQUIPO DEL PROYECTO

El equipo del proyecto formado por el estudiante y el director/ponente es claramente una parte interesada. Es tarea de ambos trabajar y estudiar las diversas fases de desarrollo del proyecto. Del equipo se espera una inversión de tiempo y esfuerzo. Son los encargados de la creación del producto final y que este sea completado con éxito, es decir, satisfaciendo los requisitos y cumpliendo los objetivos.

1.5.3 GARMIN

La empresa Garmin es una parte interesada ya que el sistema propuesto utiliza el dispositivo Garmin Epix. De esta forma la empresa tiene la exclusividad del dispositivo *smartwatch* en que funcionara la aplicación. Un software exitoso aumentara las funciones de su dispositivo y por lo tanto su atractivo a los consumidores.

1.5.4 CORREDORAS DE DATOS (DATA BROKERS)

Una corredora de datos, también llamada agente de información o información de distribuidor, es una empresa que recopila información personal sobre los consumidores y vende esa información a otras organizaciones.

En nuestro caso, recopilaría información de los usuarios de la aplicación, en concreto los niveles de rendimiento de estos o la evolución de los mismos, estos datos pueden utilizarse para crear un perfil del tipo de deportista del que se trata y las necesidades que puede tener.

Por lo general, las corredoras crean perfiles de personas con fines de comercialización y los venden a empresas que quieran dirigir sus anuncios y ofertas o analizar grandes cantidades de información.

1.5.5 ANUNCIANTES

Empresas, entidades u organizaciones que tienen interés en utilizar el sistema para promocionarse. Dado que es una aplicación para personas que quieren realizar un seguimiento de rendimiento deportivo, empresas dedicadas a la suplementación deportiva, equipamiento o actividades relacionadas pueden interesadas.

1.6 OBJETIVOS

En este apartado se definen los objetivos del proyecto a realizar. El conjunto de estos objetivos constituyen la razón por la que el sistema será desarrollado y tienen como finalidad satisfacer las necesidades de las partes interesadas. Para especificar los diferentes objetivos se utilizará la siguiente plantilla:

Objetivo <i>#[Identificador del objetivo]</i>
Descripción: <i>[Descripción del objetivo]</i>
Partes interesadas: <i>[Cual/es parte/es interesada/s desea/n el objetivo en especificado]</i>

Para estructurar mejor los diferentes objetivos del proyecto, estos serán separados entre los que atañen a la aplicación del dispositivo *smartwatch* y a los que corresponden a la plataforma web.

1.6.1 FOREVER FIT SMARTWATCH

Objetivo #1

Descripción: Utilizando la aplicación en el *smartwatch* los usuarios podrán obtener una estimación de su VO₂Max para saber en qué nivel se encuentra su rendimiento aeróbico.

Partes interesadas: Usuario

Objetivo #2

Descripción: Utilizando la aplicación en el *smartwatch* los usuarios podrán obtener una estimación de la velocidad que alcanzarían al llegar a su VO₂Max para saber en qué nivel se encuentran sus capacidades como corredor.

Partes interesadas: Usuario

1.6.2 FOREVER FIT WEB

Objetivo #3

Descripción: Utilizando la aplicación web los usuarios serán capaces administrar su sesión de usuario en el portal para acceder a su información de forma segura.

Partes interesadas: Usuario

Objetivo #4

Descripción: Utilizando la aplicación en el *smartwatch* los usuarios podrán obtener una estimación de su VO₂Max para saber en qué nivel se encuentra su rendimiento aeróbico.

Partes interesadas: Usuario

Objetivo #5

Descripción: Utilizando la aplicación en el *smartwatch* los usuarios podrán obtener una estimación de la velocidad que alcanzarían al llegar a su VO₂Max para saber en qué nivel se encuentran sus capacidades como corredor.

Partes interesadas: Usuario

1.6.3 ACLARACIONES

Aunque todos los objetivos implican directamente solo a los usuarios como partes interesadas, en realidad cada una de ellas lo es para cada objetivo. Esto es debido a que el deseo del resto de partes interesadas, es que los usuarios tengan una experiencia lo más positiva posible. En el caso de Garmin su interés consiste en mejorar el atractivo de su dispositivo Epix, por lo tanto esto vendrá definido por la satisfacción del usuario al usar la aplicación. De forma análoga los corredores de datos y los anunciantes desean llegar al máximo de usuarios, por lo que su preocupación es que estos estén satisfechos con las funcionalidades. En el caso del equipo del proyecto un usuario satisfecho significará un proyecto finalizado correctamente.

Análisis de requisitos

En este apartado se describen cada uno de los requisitos del sistema, es decir, las necesidades que ha de satisfacer el producto final para considerarse válido. A continuación se describen tanto los requisitos funcionales como no funcionales del sistema deseado.

2.1 REQUISITOS FUNCIONALES

Esta sección se centra en describir los requisitos funcionales del sistema, es decir, las funcionalidades que se espera que el sistema final proporcione. Para definir los requisitos funcionales, se utilizará la siguiente plantilla:

Objetivo: [Objetivo que satisface]
Requisito Funcional: #[Identificador del requisito]
Descripción: [Descripción del requisito]
Partes interesadas: [Cual/es parte/es interesada/s desea/n el requisito en especificado]

Para estructurar mejor los diferentes requisitos del proyecto, estos serán separados entre los que atañen a la aplicación del dispositivo *smartwatch* y a los que corresponden a la plataforma web.

2.1.1 SMARTWATCH GARMIN EPIX

Objetivo: #1
Requisito Funcional: #1
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios podrán obtener una estimación de su VO ₂ Max mediante el test One Mile Walk Test.
Partes interesadas: Usuario

Objetivo: #1
Requisito Funcional: #2
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios podrán obtener una estimación de su VO ₂ Max mediante el test One Half Mile Run Test.
Partes interesadas: Usuario

Objetivo: #1
Requisito Funcional: #3
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios podrán reiniciar los test One Mile Walk Test y/o One Half Mile Run Test una vez acabado.
Partes interesadas: Usuario

Objetivo: #2
Requisito Funcional: #4
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios podrán obtener una estimación de la velocidad que alcanzarían al llegar a su VO ₂ Max mediante el test VO ₂ Max Speed.
Partes interesadas: Usuario

Objetivo: #2
Requisito Funcional: #5
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios podrán definir su frecuencia cardiaca máxima para poder obtener una estimación del test VO ₂ Max Speed precisa.
Partes interesadas: Usuario

Objetivo: #2
Requisito Funcional: #6
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios podrán detener el test VO ₂ Max Speed para continuarlo cuando deseen.
Partes interesadas: Usuario

Objetivo: #2
Requisito Funcional: #7
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios podrán continuar el test VO ₂ Max después de haberlo detenido.
Partes interesadas: Usuario

Objetivo: #2
Requisito Funcional: #8
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios tendrán una estimación continua del VO ₂ Max Speed Test una vez alcanzada la primera estimación, esta estimación continua será la media de todas las estimaciones a partir de la primera conseguida.
Partes interesadas: Usuario

Objetivo: #1 #2
Requisito Funcional: #9
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios podrán visualizar por pantalla mensajes informativos sobre el estado del test que está realizando.
Partes interesadas: Usuario

Objetivo: #1 #2
Requisito Funcional: #10
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios podrán visualizar por pantalla mensajes con instrucciones sobre cómo realizar el test escogido.
Partes interesadas: Usuario

Objetivo: #1 #2
Requisito Funcional: #11
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios deben poder detener la ejecución de cualquiera de los test.
Partes interesadas: Usuario

Objetivo: #1 #2
Requisito Funcional: #12
Descripción: Utilizando el <i>smartwatch</i> los usuarios deben ser capaces de iniciar la aplicación.
Partes interesadas: Usuario

Objetivo: #1 #2
Requisito Funcional: #13
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios deben ser capaces de salir de la aplicación.
Partes interesadas: Usuario

2.1.2 PLATAFORMA WEB

Objetivo: #3
Requisito Funcional: #14
Descripción: Utilizando la página web los usuarios podrán registrar para utilizar sus funcionalidades.
Partes interesadas: Usuario

Objetivo: #3
Requisito Funcional: #15
Descripción: Utilizando la página web los usuarios registrados podrán iniciar sesión mediante usuario y contraseña para utilizar sus funcionalidades.
Partes interesadas: Usuario

Objetivo: #3
Requisito Funcional: #16
Descripción: Utilizando la página web los usuarios registrados podrán cerrar su sesión para dejar de utilizar la aplicación.
Partes interesadas: Usuario

Objetivo: #4
Requisito Funcional: #17
Descripción: Utilizando la página web los usuarios registrados podrán obtener una estimación de su VO ₂ Max mediante el test One Mile Walk Test.
Partes interesadas: Usuario

Objetivo: #4
Requisito Funcional: #18
Descripción: Utilizando la página web los usuarios registrados podrán obtener una estimación de su VO ₂ Max mediante el test One Half Mile Run Test.
Partes interesadas: Usuario

Objetivo: #4
Requisito Funcional: #19
Descripción: En caso de que el test no haya recorrido la distancia mínima necesaria o no se haya podido realizar la estimación para los test One Mile Walk Test y/o One Half Mile Run test se mostrara un mensaje informativo.
Partes interesadas: Usuario

Objetivo: #5
Requisito Funcional: #20
Descripción: Utilizando la página web los usuarios registrados podrán obtener una estimación de la velocidad que alcanzarían al llegar a su VO ₂ Max mediante el test VO ₂ Max Speed.
Partes interesadas: Usuario

Objetivo: #5
Requisito Funcional: #21
Descripción: Utilizando la página web los usuarios registrados podrán obtener el resultado final de la estimación continua del VO ₂ Max Speed
Partes interesadas: Usuario

Objetivo: #5
Requisito Funcional: #22
Descripción: En caso de que el test no haya durado el tiempo mínimo necesario para el test VO ₂ Max Speed se mostrará el resultado N/A y para el resultado de la estimación continua.
Partes interesadas: Usuario

Objetivo: #4 #5
Requisito Funcional: #23
Descripción: Utilizando un navegador web los usuarios deben ser capaces de acceder a la página web.
Partes interesadas: Usuario

Objetivo: #4 #5
Requisito Funcional: #24
Descripción: Utilizando la aplicación en el <i>smartwatch</i> los usuarios podrán generar ficheros .FIT durante la ejecución de los test para su posterior exportación.
Partes interesadas: Usuario

Objetivo: #4 #5
Requisito Funcional: #25
Descripción: Utilizando la página web los usuarios registrados podrán subir archivos .FIT para su análisis.
Partes interesadas: Usuario

Objetivo: #4 #5
Requisito Funcional: #26
Descripción: Utilizando la página web los usuarios registrados podrán visualizar los archivos .FIT que han subido.
Partes interesadas: Usuario

Objetivo: #4 #5
Requisito Funcional: #27
Descripción: Utilizando FFW los usuarios registrados podrán borrar los archivos .FIT que hayan subido.
Partes interesadas: Usuario

Objetivo: #4 #5
Requisito Funcional: #28
Descripción: Utilizando la página web los usuarios registrados podrán visualizar el contenido de los archivos .FIT que hayan subido.
Partes interesadas: Usuario

2.1.3 ACLARACIONES

Igual que en el apartado 1.6 Objetivos, aunque todos los objetivos implican directamente solo a los usuarios como partes interesadas, en realidad cada una de ellas lo es para cada objetivo para más detalles ir a 1.6.3 Aclaraciones.

2.2 REQUISITOS NO FUNCIONALES

Esta sección se centran en describir los requisitos no funcionales del sistema, es decir, las propiedades y cualidades que este debe tener independientemente de sus funcionalidades como puede ser el tiempo de respuesta o la usabilidad. Para definir tales requisitos se utilizará la siguiente plantilla:

Requisito Funcional: <i>#[Identificador del requisito funcional]</i>
Descripción: <i>[Descripción del requisito funcional]</i>
Justificación: <i>[Razón por la es necesaria el requisito nombrado]</i>
Criterio de satisfacción: <i>[Cual/es parte/es interesada/s desea/n el requisito en especificado]</i>

2.2.1 REQUISITOS DE VELOCIDAD Y LATENCIA

Requisito Funcional: # 1
Descripción: el sistema debe tener una velocidad de respuesta adecuada.
Justificación: : el sistema debe de tener una velocidad de después adecuada para ofrecer al usuario un entorno fluido y sin esperas.
Criterio de satisfacción: la aplicación será desarrollado de forma que las funcionalidades se realicen en el tiempo mínimo posible y optimizando las tareas lo máximo posible.

2.2.2 REQUISITOS DE USABILIDAD

Requisito Funcional: # 2
Descripción: el diseño del sistema debe ser atractivo e invitar a la utilización del mismo.
Justificación: el diseño del sistema debe ser atractivo e invitar a la utilización del mismo para agradar al usuario y que se encuentre cómodo utilizando el sistema.
Criterio de satisfacción: se realizará el diseño de las diferentes pantallas del sistema teniendo en cuenta criterios de diseño que favorezcan el estilo del web. De la misma forma se realizará un diseño confluyente con el estilo actual de las aplicaciones web y de <i>smartwatch</i> .

Requisito Funcional: # 3
Descripción: la aplicación debe ser fácil de utilizar
Justificación: es importante que la aplicación sea sencilla de utilizar para ahorrar tiempo al usuario y agilizar sus tareas. Además la aplicación del <i>smartwatch</i> se utilizara a la vez que se realiza deporte por lo que el usuario debe de poder utilizarla y establecer sus parámetros rápidamente.
Criterio de satisfacción: se realizará un sistema intuitivo y que tenga un comportamiento similar al de aplicaciones similares del mercado para que el usuario tenga nociones sobre su utilización aun antes de instalarla. Además la aplicación contará con un manual de uso.

2.2.3 REQUISITOS DE SEGURIDAD

Requisito Funcional: # 4

Descripción: un usuario solo tendrá acceso a su información.

Justificación: para asegurar la privacidad de la información del usuario cada uno de estos solo podrá acceder a su respectiva información en el portal web mediante el uso de cuentas de usuario

Criterio de satisfacción: se accederá al portal web con diferentes cuentas de usuario y se comprobará que solo se puede acceder a la información respectiva del usuario identificado en la aplicación, también se comprobará que se mantenga la identificación después de cerrar la página web y demás situaciones de transición de sesiones entre diferentes cuentas.

2.2.4 REQUISITOS DE PRIVACIDAD

Requisito Funcional: # 5

Descripción: la información personal se realizará de forma que cumpla con la Ley Orgánica de Protección de Datos personales (LOPD).

Justificación: el sistema debe ofrecer al usuario la seguridad de que sus datos serán privados.

Criterio de satisfacción: se pedirá a un abogado experto en la LOPD que confirme que el sistema no viola la ley. Para más detalle de cómo se va a satisfacer este criterio ir al apartado 6.5 Identificación de Leyes y regulaciones.

2.2.5 REQUISITOS DE INTEGRIDAD

Requisito Funcional: # 6

Descripción: No se podrán informar datos en formatos erróneos tanto en la aplicación web como en el smartwatch.

Justificación: en los casos en que el usuario pueda introducir algún tipo de información de forma manual esta debe de estar en el formato esperado para evitar funcionamientos inesperados del sistema.

Criterio de satisfacción: en los diferentes casos en que el usuario debe introducir datos en el sistema se realizarán pruebas introduciendo datos en formatos erróneos y comprobando que el sistema controla estas situaciones de forma correcta.

2.2.6 REQUISITOS DE COMPRENSIÓN Y CORTESÍA

Requisito Funcional: # 7

Descripción: la información mostrada debe de tener un registro lingüístico formal con un vocabulario adecuado.

Justificación: los usuarios del sistema deben de sentir que se es trata con respeto y la forma adecuada.

Criterio de satisfacción: toda la información mostrada en el sistema será en un registro lingüístico apropiado que den al usuario el nivel de respeto necesario pero sin sobrecargar la comprensión con adornos innecesarios.

2.2.7 REQUISITOS DE PRECISIÓN O EXACTITUD

Requisito Funcional: # 8
Descripción: el formato de las fechas debe ser DD/MM/AAAA
Justificación: el formato DD/MM/AAAA es el más utilizado en Europa.
Criterio de satisfacción: todos los valores decimales de la aplicación tendrán el punto como separador.

Requisito Funcional: # 9
Descripción: el separador decimal debe ser el punto
Justificación: las Academias de la Lengua recomiendan el punto con motivo de promover una tendencia hacia la unificación [8].
Criterio de satisfacción: todos los valores decimales de la aplicación tendrán el punto como separador.

2.2.8 REQUISITOS DE FIABILIDAD Y DISPONIBILIDAD

Requisito Funcional: # 10
Descripción: el sistema debe de estar operativo como mínimo el 99% del tiempo.
Justificación: el usuario debe de poder utilizar el sistema en cualquier momento.
Criterio de satisfacción: el sistema estará disponible y funcional como mínimo el 99% del tiempo.

Requisito Funcional: # 11
Descripción: el portal web debe funcionar correctamente en los navegadores más utilizados: Google Chrome, Mozilla Firefox e Internet Explorer.
Justificación: el portal web debe ser accesible en el máximo número posible de navegadores o al menos en los más utilizados para adaptar la aplicación a las costumbres del usuario.
Criterio de satisfacción: se comprobará el correcto funcionamiento de la aplicación en los diferentes navegadores mencionados.

Especificación

En esta sección primeramente se definirán los actores del sistema y a continuación de estos el diagrama de sus casos de uso, finalmente estos últimos serán descritos detalladamente.

3.1 ACTORES

Los actores del sistema son las personas que interactuarán de forma directa con la aplicación, se trata de un rol desempeñado por un usuario. A continuación se describen los actores implicados en el sistema:

- Usuario smartwatch: este actor hace referencia a cualquier persona que utilice la aplicación en el *smartwatch*. Su principal objetivo será realizar los diferentes test que la aplicación proporciona para estimar su rendimiento aeróbico y/o la velocidad máxima que alcanzaría al llegar a su VO₂Max..
- Usuario no registrado: se trata de una persona que accede a la página web del sistema sin identificarse o sin estar registrado directamente, las funcionalidades de las que dispondrá serán limitadas debido a este hecho. Básicamente podrá ver la pantalla principal o identificarse o registrarse para pasar a ser un actor "usuario registrado"
- Usuario registrado: se trata de un usuario correctamente identificado en la página web por lo tanto podrá subir archivos .FIT, gestionarlos y obtener sus métricas de rendimiento aeróbico a partir de estos y cerrar su sesión.

3.2 DIAGRAMA DE CASOS DE USO

A continuación se presenta el diagrama de casos de uso de la aplicación cuya finalidad es representar de forma gráfica la relación entre los diferentes casos de uso y los diferentes roles de usuario.

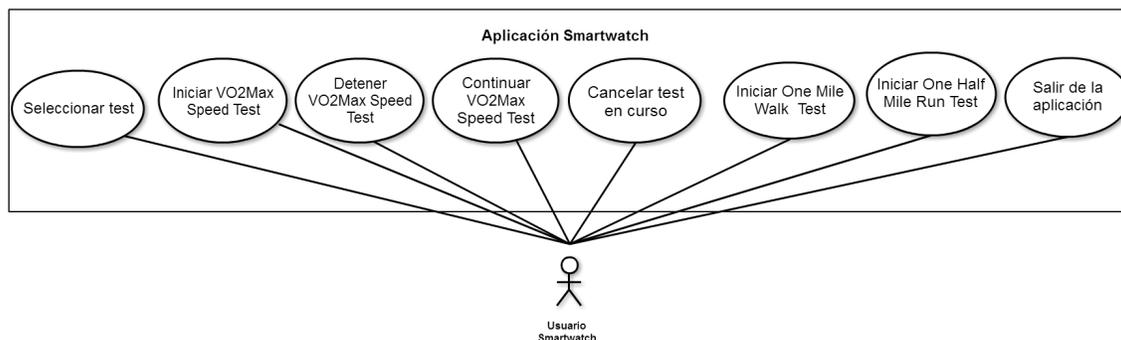


Ilustración 4. Diagrama casos de uso FFS



Ilustración 5. Diagrama casos de uso FFW

3.3 CASOS DE USO

En este sub apartado se encuentran detallados los casos de uso presentados en apartado anterior. Cada uno de estos casos de uso se centra en establecer cómo alcanzar una meta del usuario utilizando el sistema. Siguiendo el estilo del documento se presentan primero la descripción de casos de uso de FFS y a continuación los correspondientes a FFW.

En ambos casos se utilizará la siguiente plantilla para la descripción de los casos de uso:

[Condición que se ha satisfecho antes del comienzo del caso de uso]
[Acción que provoca el inicio del caso de uso]
[Secuencia Acciones entre el actor y el sistema]
[Alternativas y tratamiento de situaciones anormales]

3.3.1 CASOS DE USO FOREVER FIT SMARTWATCH

3.3.1.1 CU#0001 SELECCIONAR TEST

- Actor Principal:** Usuario smartwatch
Precondición: Ninguna
Disparador: El usuario desea seleccionar alguno de los test.
Escenario Principal:
1. El usuario desliza la pantalla hacia la izquierda o derecha hasta encontrarse en la del test que desea ejecutar.
 2. El sistema muestra un mensaje informativo para iniciar el test.
- Extensiones:**
- 1.a. El usuario desliza hacia la izquierda varias veces
 - 1.a.1. La aplicación se cierra ya que al deslizar hacia la izquierda desde el último test (One Half Mile Run Test) se cierra la aplicación.

3.3.1.2 CU#0002 INICIAR VO2MAX SPEED TEST

- Actor Principal:** Usuario smartwatch
- Precondición:** El usuario ha seleccionado el test VO2Max Speed.
La frecuencia cardiaca máxima del usuario está definida en la aplicación.
La frecuencia cardiaca en reposo del usuario está definida en la configuración del dispositivo.
Los componentes que capturan las constantes vitales del dispositivo están en funcionamiento.
El seguimiento de actividades del dispositivo está funcionando.
- Disparador:** El usuario quiere ejecutar el test Vo2Max Speed
- Escenario Principal:**
1. El usuario pulsa la pantalla.
 2. El sistema inicia el test.
 3. El sistema muestra un mensaje informativo de cómo realizar el test.

3.3.1.3 CU#0003 DETENER VO2MAX SPEED TEST

- Actor Principal:** Usuario smartwatch
- Precondición:** El test VO2Max Speed está en marcha
- Disparador:** El usuario quiere detener la ejecución del Vo2Max Speed Test
- Escenario Principal:**
1. El usuario pulsa la pantalla.
 2. El sistema detiene el test.
 3. El sistema muestra un mensaje informativo sobre como continuar el test.

3.3.1.4 CU#0004 CONTINUAR VO2MAX SPEED TEST

- Actor Principal:** Usuario smartwatch
- Precondición:** El test VO2Max Speed está detenido
- Disparador:** El usuario quiere reanudar la ejecución del Vo2Max Speed Test
- Escenario Principal:**
1. El usuario pulsa la pantalla.
 2. El sistema reanuda el test.
 3. El sistema muestra un mensaje informativo de cómo realizar el test.

3.3.1.5 CU#0005 INICIAR ONE MILE WALK TEST

- Actor Principal:** Usuario smartwatch
- Precondición:** El usuario ha seleccionado el test One Mile Walk.
El género del usuario está definido en el perfil del dispositivo.
El peso del usuario está definido en el perfil del dispositivo.
La edad del usuario está definido en el perfil del dispositivo.
Los componentes que capturan las constantes vitales del dispositivo están en funcionamiento.
El seguimiento de actividades del dispositivo está funcionando.
- Disparador:** El usuario quiere ejecutar el test Vo2Max Speed
- Escenario Principal:**
1. El usuario pulsa la pantalla.

2. El sistema inicia el test.
3. El sistema muestra un mensaje informativo de cómo realizar el test.

3.3.1.6 CU#0006 INICIAR ONE HALF MILE RUN TEST

- Actor Principal:** Usuario smartwatch
- Precondición:** El usuario ha seleccionado el test One Half Mile Run.
Los componentes que capturan las constantes vitales del dispositivo están en funcionamiento.
El seguimiento de actividades del dispositivo está funcionando.
- Disparador:** El usuario quiere ejecutar el test Vo2Max Speed
- Escenario Principal:**
1. El usuario pulsa la pantalla.
 2. El sistema inicia el test.
 3. El sistema muestra un mensaje informativo de cómo realizar el test.

3.3.1.7 CU#0007 CANCELAR TEST EN CURSO

- Actor Principal:** Usuario smartwatch
- Precondición:** Se está ejecutando alguno de los test del sistema.
- Disparador:** El usuario quiere cancelar la ejecución del test en curso.
- Escenario Principal:**
1. El usuario desliza la pantalla hacia la izquierda o la derecha.
 2. El sistema carga la nueva pantalla y cancela el test anterior.
- Extensiones:**
- 1.a. El usuario se encontraba en la pantalla del ultimo test y realiza un desplazamiento hacia la izquierda
 - 1.a.1. La aplicación se cierra.

3.3.1.8 CU#0008 SALIR DE LA APLICACIÓN

- Actor Principal:** Usuario smartwatch
- Precondición:** Estar ejecutando la aplicación
- Disparador:** El usuario quiere salir de la aplicación
- Escenario Principal:**
1. El usuario desliza la pantalla hacia la izquierda hasta salir.

3.3.2 CASOS DE USO FOREVER FIT WEB

3.3.2.1 CU#0009 REGISTRAR USUARIO

- Actor Principal:** Usuario no identificado.
- Precondición:** El usuario no está identificado en el sistema.
- Disparador:** El usuario desea crear una cuenta de usuario.
- Escenario Principal:**
1. El usuario entra en el sistema.
 2. El usuario selecciona la opción de "Regístrate" en el menú principal.
 3. El sistema muestra el formulario con los datos que deben informarse.

- 4. El usuario introduce los datos necesarios.
 - 5. El sistema válida los datos.
 - 6. El sistema registra al nuevo usuario.
 - 7. Se inicia la sesión del nuevo usuario.
- Extensiones:**
- 4.a. Alguno de los campos no son válidos:
 - 4.a.1. El sistema informa de los campos que no son válidos.
 - 4.a.2. Se vuelve al paso 3.
 - 4.b. Falta alguno de los campos:
 - 4.b.1. El sistema informar de los campos que faltan.
 - 4.b.2. Se vuelve al paso 3.

3.3.2.2 CU#0010 INICIAR SESIÓN

- Actor Principal:** Usuario no identificado.
- Precondición:** El usuario no está identificado en el sistema.
- Disparador:** El usuario desea iniciar sesión en el sistema.
- Escenario Principal:**
- 1. El usuario entra en el sistema.
 - 2. El usuario selecciona la opción de "Identificarme" en el menú principal.
 - 3. El sistema muestra el formulario con los datos que deben informarse.
 - 4. El usuario introduce los datos necesarios.
 - 5. El sistema válida los datos.
 - 6. El sistema comprueba los datos del usuario con la base de datos.
 - 7. El sistema inicia la sesión del usuario
- Extensiones:**
- 4.a. Alguno de los campos no son válidos:
 - 4.a.1. El sistema informar de los campos que no son válidos.
 - 4.a.2. Se vuelve al paso 3.
 - 4.b. Falta alguno de los campos:
 - 4.b.1. El sistema informar de los campos que faltan.
 - 4.b.2. Se vuelve al paso 3.
 - 6.a. El sistema tiene problemas interactuando con la base de datos
 - 6.a.1. El sistema lleva a una página de error.
 - 6.b. El la combinación usuario/contraseña introducida no coincide con ningún registro en la base de datos.
 - 6.b.1. El sistema muestra un mensaje informativo.

3.3.2.3 CU#0011 MODIFICAR DATOS DEL USUARIO

- Actor Principal:** Usuario identificado.
- Precondición:** El usuario está identificado en el sistema.
- Disparador:** El usuario desea modificar sus datos en el sistema.
- Escenario Principal:**
- 1. El usuario selecciona la opción de "Hola [NombreUsuario]" en el menú principal
 - 2. El sistema muestra el formulario con los datos que pueden modificarse.
 - 3. El usuario introduce los datos que desea modificar.
 - 4. El sistema válida los datos.
 - 5. El sistema registra los cambios en la base de datos.
- Extensiones:**
- 4.a. Alguno de los campos no son válidos:

- 4.a.1. El sistema informar de los campos que no son válidos.
- 4.a.2. Se vuelve al paso 2.
- 4.b. Falta alguno de los campos:
 - 4.b.1. El sistema informar de los campos que faltan.
 - 4.b.2. Se vuelve al paso 2.
- 5.a. El sistema tiene problemas interactuando con la base de datos
 - 5.a.1. El sistema lleva a una página de error.

3.3.2.4 CU#0012 CERRAR SESIÓN

- Actor Principal:** Usuario identificado.
Precondición: El usuario está identificado en el sistema.
Disparador: El usuario desea cerrar su sesión en el sistema.
Escenario Principal:
 1. El usuario selecciona la opción de "Salir"
 2. El sistema cierra la sesión del usuario.

3.3.2.5 CU#0013 VER LISTADO ARCHIVOS .FIT

- Actor Principal:** Usuario identificado.
Precondición: El usuario está identificado en el sistema.
Disparador: El usuario desea ver su lista de archivos .FIT
Escenario Principal:
 1. El usuario pulsa el la opción "Activities" en el menú principal
 2. El sistema interactúa con la base de datos y recupera el listo de activities pertenecientes al usuario
 3. El sistema carga la pantalla Activities con un listado de los archivos .FIT que el usuario ha subido al sistema.

- Extensiones:**
 - 2.a. El sistema tiene algún problema interactuando con la base de datos:
 - 2.a.1. El sistema lleva a una página de error.
 - 3.a. El usuario no ha subido ningún archivo al sistema
 - 3.a.1. La lista de ficheros subidos permanece vacía.

3.3.2.6 CU#0014 SUBIR ARCHIVO .FIT

- Actor Principal:** Usuario identificado.
Precondición: El usuario está identificado en el sistema.
 El usuario está en la pantalla de Activities.
Disparador: El usuario desea subir un archivo .FIT
Escenario Principal:
 1. El usuario pulsa el botón archivo.
 2. El sistema muestra una ventana para seleccionar el fichero del sistema de archivos del usuario.
 3. El usuario selecciona un archivo.
 4. El usuario presiona el botón "Subir"
 5. El sistema procesa el archivo seleccionado.
 6. El sistema guarda la información del archivo seleccionado.
 7. El sistema actualiza la tabla de archivos .FIT presente en el sistema con la información del nuevo fichero.**Extensiones:**
 - 5.a. El archivo no tiene el formato correcto:

- 5.a.1. El sistema lleva a una página de error.
- 5.b. El usuario no ha seleccionado ningún archivo
 - 5.b.1. El sistema lleva a una página de error.
- 5.c. El sistema tiene problemas interactuando con la base de datos
 - 5.c.1. El sistema lleva a una página de error.

3.3.2.7 CU#0015 BUSCAR ARCHIVOS .FIT POR NOMBRE

- Actor Principal:** Usuario identificado.
- Precondición:** El usuario está identificado en el sistema.
El usuario está en la pantalla de Activities.
- Disparador:** El usuario desea buscar en la lista de registros de ficheros .FIT subidos por nombre
- Escenario Principal:**
1. El usuario escribe el criterio de búsqueda por nombre en el recuadro "Buscar por nombre".
 2. El usuario pulsa el botón aceptar.
 3. El sistema recupera los registros del usuario que cumplen la condición.
 4. El sistema actualiza el lista de registros de archivos .FIT con el resultado obtenido.
- Extensiones:**
- 3.a. El sistema tiene problemas interactuando con la base de datos
 - 3.a.1. El sistema lleva a una página de error.
 - 3.b. El sistema no recibe ninguna cadena de texto desde el cuadro de texto.
 - 3.b.1. El sistema recupera todos los registros pertenecientes al usuario.
 - 4.a. El sistema no devolvió ningún resultado
 - 4.a.1. La lista queda vacía.

3.3.2.8 CU#0016 ORDENAR LISTADO ARCHIVOS .FIT

- Actor Principal:** Usuario identificado.
- Precondición:** El usuario está identificado en el sistema.
El usuario está en la pantalla de Activities.
- Disparador:** El usuario desea filtrar la lista de registros de ficheros .FIT subidos por nombre o fecha de creación
- Escenario Principal:**
1. El usuario selecciona el título de alguna de las dos columnas
 2. El recupera los registros de la base de datos ordenados por el criterio marcado
 3. El sistema actualiza la lista de registros.
- Extensiones:**
- 2.a. El sistema tiene problemas interactuando con la base de datos
 - 2.a.1. El sistema lleva a una página de error.

3.3.2.9 CU#0017 ELIMINAR REGISTRO ARCHIVO .FIT

- Actor Principal:** Usuario identificado.
- Precondición:** El usuario está identificado en el sistema.
El usuario está en la pantalla de Activities.
El usuario tiene al menos el registro de un activity.
- Disparador:** El usuario desea subir un archivo .FIT
- Escenario Principal:**
1. El usuario pulsa el link "Borrar" de la columna acciones correspondiente al registro del activity.
 2. El sistema borra el registro de la base de datos del sistema.
 3. Se actualiza la tabla desapareciendo el registro correspondiente a la información del fichero .FIT borrado.
 - 4.
- Extensiones:**
- 2.a. El sistema tiene problemas interactuando con la base de datos
 - 2.a.1. El sistema lleva a una página de error.

3.3.2.10 CU#0018 VER MÉTRICAS ARCHIVO .FIT

- Actor Principal:** Usuario identificado.
- Precondición:** El usuario está identificado en el sistema.
El usuario está en la pantalla de activities.
El usuario tiene al menos el registro de un activity.
- Disparador:** El usuario desea subir un archivo .FIT
- Escenario Principal:**
1. El usuario pulsa el link del nombre del activity del que quiere ver las métricas en el listado.
 2. El sistema recupera de la base de datos toda la información correspondiente al activity seleccionado de la base de datos.
 3. El sistema analiza los datos
 4. El sistema lleva a la pantalla "Activity Record" donde aparecen la estimación de las diferentes métricas del sistema, un resumen del activity y todos los registros que tiene.
- Extensiones:**
- 2.a. El sistema tiene problemas interactuando con la base de datos.
 - 2.a.1. El sistema lleva a una página de error.
 - 3.a. El sistema tiene problemas analizando los datos.
 - 3.a.1. El sistema lleva a una página de error.

Modelo Conceptual

Este capítulo se centra en realizar un esquema conceptual previo a la implementación. El objetivo es dar una idea global sobre los diferentes conceptos relevantes con las que se pretende que cuente el sistema y como se plantean las relaciones entre estos. El modelo se centrará principalmente en los datos, independientemente de la forma final en que se implemente, la cual dependerá de las tecnologías utilizadas.

4.1 FOREVER FIT SMARTWATCH

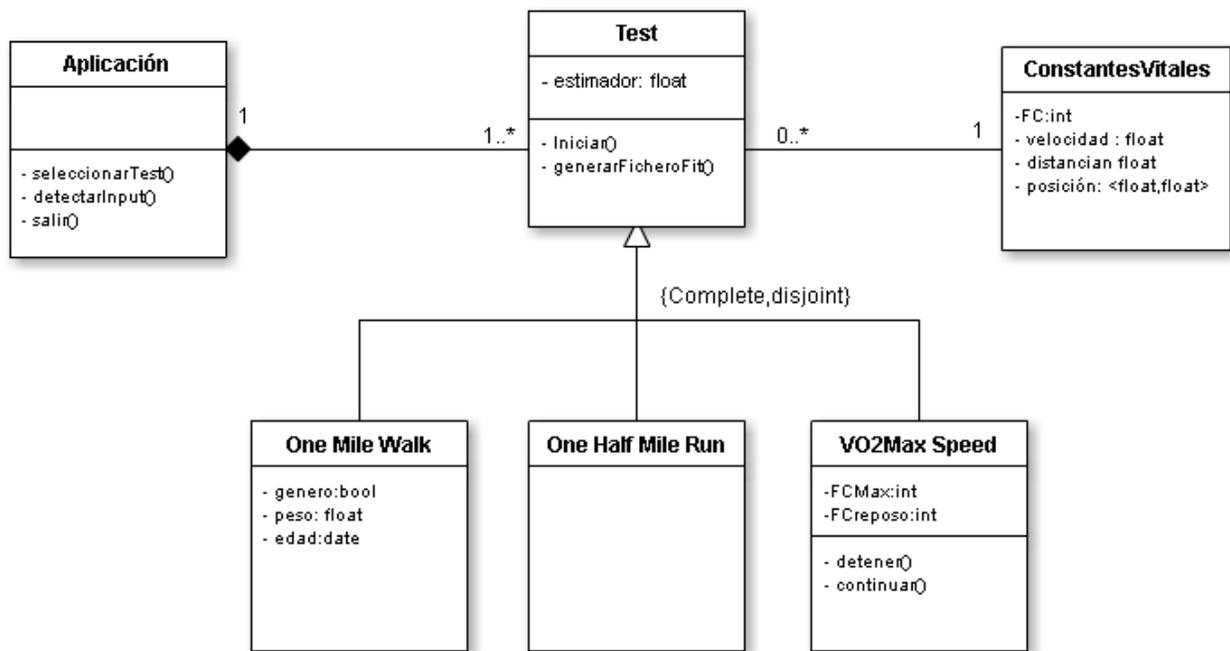


Ilustración 6. Modelo conceptual FFS

Para la aplicación del dispositivo Garmin, el sistema se presenta con una herencia entre los diferentes tipos de test debido a que es evidente que estos tendrán aspectos importantes en común, respecto a la implementación o funcionalidades que necesitarán. A partir del estudio previo de los diferentes test, en el caso de los One Mile Walk Test and VO₂Max Speed Test ya se tienen en cuenta características identificativas, como por ejemplo parámetros clave para realizar la estimación.

También se identifica una entidad principal "Aplicación" que se encarga de gestionar los diferentes test y el concepto de "ContantesVitales" para referirnos al conjunto de funcionalidades propias del sistema Garmin para acceder a los inputs del exterior.

4.2 FOREVER FIT WEB

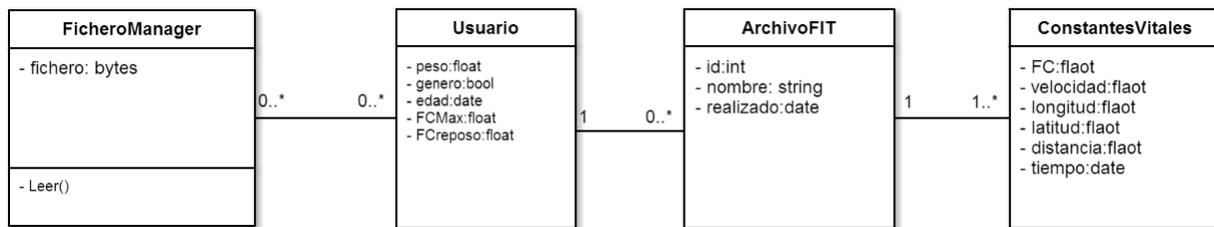


Ilustración 7. Modelo conceptual FFW

Para la aplicación web la idea principal es tener una entidad que represente al usuario, encargada de todo el sistema de autenticación, relacionado con una otra encargada de gestionar la manipulación de los ficheros FIT. Esta entidad se encargaría de leer los archivos decodificarlos y codificarlos en un formato entendible para los humanos.

Una vez cargada toda la información de los ficheros .FIT esta sería representada por la entidad "ArchivoFit" en la base de datos, que tendría toda la información relevante respecto a los tests. A su vez esta entidad estaría formada por un conjunto de registros que serían las constantes vitales momento a momento generadas durante la ejecución del test. A partir de estas dos entidades se calcularían las métricas para cada actividad.

Este capítulo pertenece a la fase de diseño del software, se establecen las estructuras de datos, la arquitectura general del sistema, su interfaz y al acceso a la base de datos. La finalidad de este capítulo es modelar el software que será construido.

5.1 ARQUITECTURA DEL SISTEMA

En este apartado se describen las arquitecturas lógicas tanto para FFS como para FFW. En ambos casos se detallará la arquitectura utilizada, su funcionamiento y los esquemas pertinentes para aclarar cómo es representada la estructura planteada en el apartado 4. Modelo Conceptual, con las tecnologías elegidas para implementar el sistema.

5.1.1 FOREVER FIT SMARTWATCH

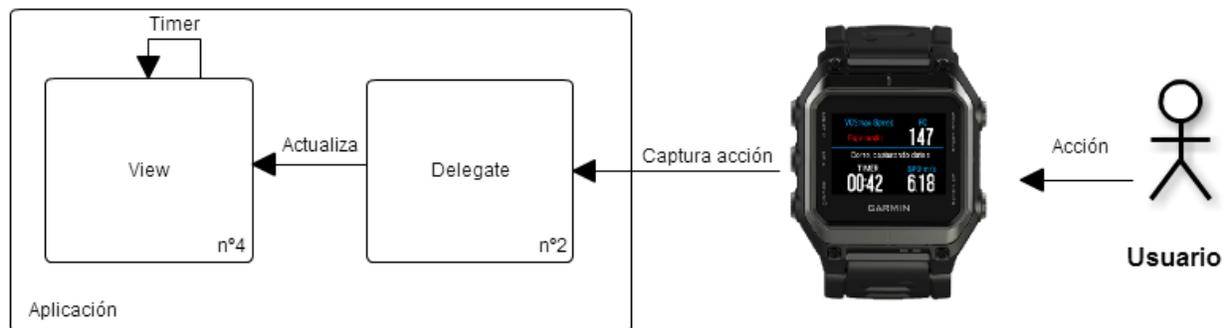


Ilustración 8. Diagrama arquitectura Forever Fit Smartwatch

Utilizando el diagrama superior se presenta la arquitectura de FFW. Dicha estructura está realizada siguiendo las prestaciones que el sistema operativo del fabricante del dispositivo Epix ofrece para desarrollar aplicaciones. Durante la ejecución de la aplicación hay siempre una pareja de *view-delegate* que está al mando en un momento determinado. La clase *delegate* se encarga de capturar las acciones del usuario como puede ser pulsar un botón, deslizar la pantalla ..etc. Las clases *view* o *vista* se encargan de la lógica que tenga dicha pantalla y de dibujar lo que visualiza el usuario.

En el caso particular de FFS la aplicación cuenta con 2 *delegates*, el primario que se encarga de capturar las acciones sobre las pantallas de los 3 test que tiene la aplicación y uno secundario que maneja las acciones sobre la pantalla para definir la frecuencia cardiaca máxima del usuario, la cual es necesaria para el Vo2Max Speed Test. En el caso de las vistas el sistema cuenta con cuatro, una por cada test de la aplicación y una cuarta para introducir la frecuencia cardiaca máxima del usuario.

5.1.1.1 PANTALLAS

Forever Fit Smartwatch está compuesto por 4 pantallas que entre las se puede navegar deslizando horizontalmente de izquierda a derecha.



Ilustración 9. . Captura del emulador del smartwatch Garmin Epix, estado inicial de las pantallas

Pantalla 1: se encarga de gestionar la estimación de la métrica VO₂Max, al pulsar la pantalla el test es iniciado, si se vuelve a pulsar el test se detiene. Es el único test que permite ser detenido ya que el resto deben realizarse sin interrupciones recomendándose haber elegido un espacio adecuado para esto previamente. El reloj es una cuenta atrás de 12 minutos ya que es el tiempo mínimo que necesita el test para realizar una primera estimación. A partir de esta primera estimación suceden dos cosas primero, la métrica se estima de forma continua y el resultado mostrado es la media de las estimaciones, segundo el reloj dejar de ser una cuenta atrás y marca el tiempo que lleva el test en ejecución.

Al deslizar la pantalla hacia la izquierda o derecha el test se cancela. En el caso de deslizar la pantalla hacia la derecha se mantiene la pantalla actual. En el caso de deslizar hacia la izquierda **se pasa a la pantalla número 3.**



Ilustración 10. . Vo2Max Speed Test en curso y en modo estimación continua

Pantalla 2: esta pantalla se encarga de recoger un dato que es necesario para que el test de la pantalla 1 tenga la precisión adecuada, la frecuencia cardiaca máxima del usuario. Para introducir dicho dato se ha de pulsar la pantalla primero. Esta pantalla se mostrara siempre que se realice un desplazamiento **desde la pantalla 3 hacia la pantalla 2**, pero no desde la 1 hacia la 2 como se ha especificado en la descripción de la pantalla 1.



Ilustración 11. Pantalla para introducir FC Máxima del usuario

Pantalla 3: se encarga de realizar el One Mile Walk Test, al pulsar la pantalla el test es ejecutado y muestra la estimación al recorrer la distancia marcada. Una vez obtenida la estimación el test se detiene. Esta pantalla es la pantalla de inicio del sistema. Se ha elegido que fuera esta y no la pantalla 1 por la relación de transiciones que hay con la pantalla número 2.



Ilustración 12. One Mile Walk Test en ejecución y estimación alcanzada

Pantalla 4: se encarga de realizar el One Half Mile Run Test, al pulsar la pantalla el test es ejecutado y muestra la estimación al recorrer la distancia marcada. Una vez obtenida la estimación el test se detiene. Esta pantalla es la última de la aplicación por lo tanto al realizar un desplazamiento hacia la izquierda se cierra.



Ilustración 13. One Half Mile Run Test en ejecución y estimación alcanzada

5.1.1.2 DIAGRAMA DE CLASES

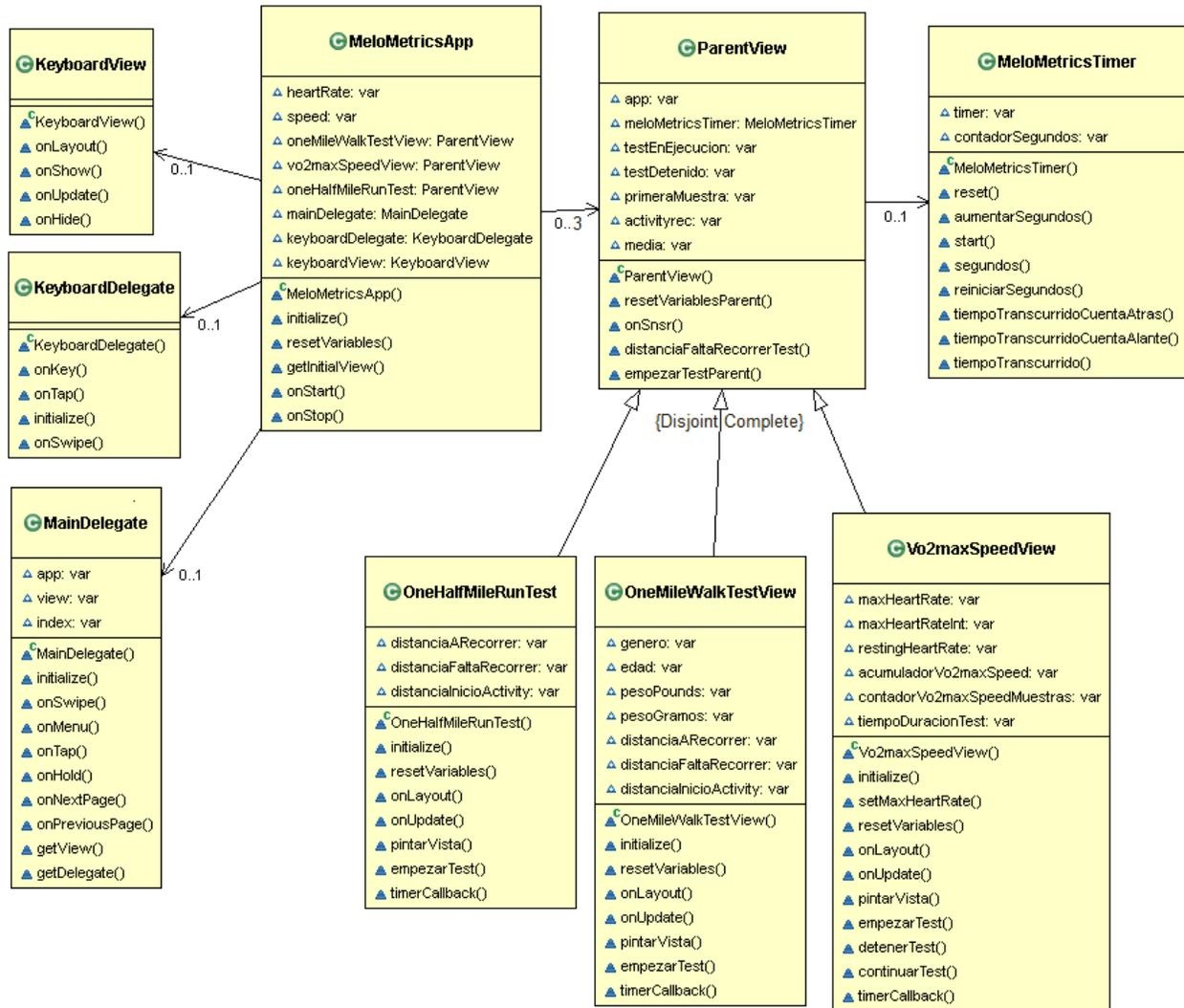


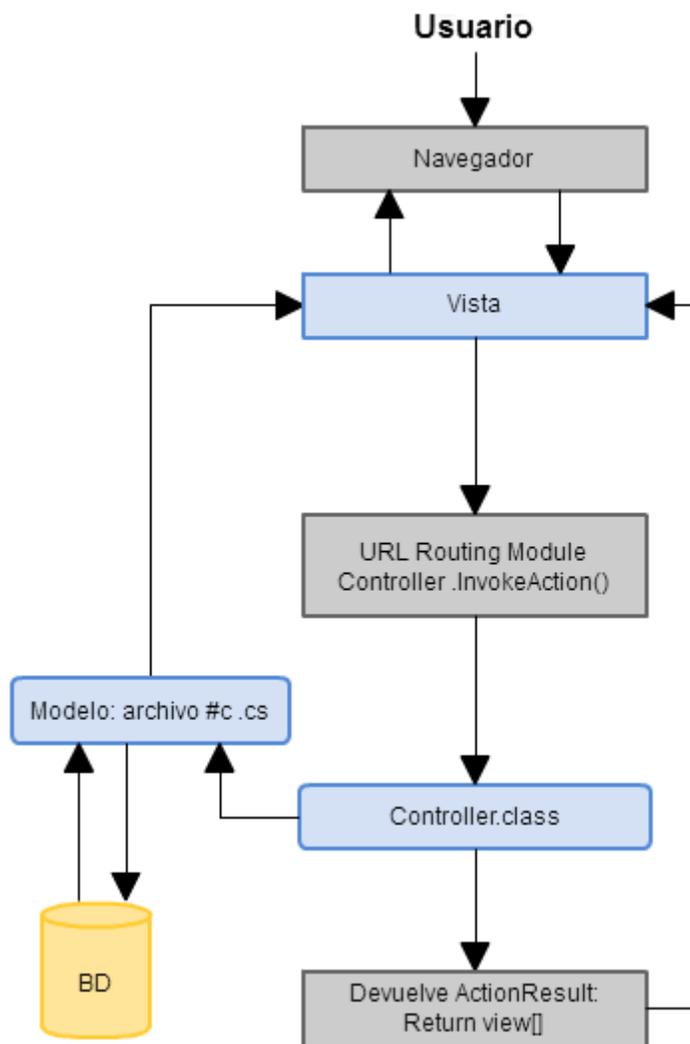
Ilustración 14. Diagrama de clases de Forever Fit Smartwatch

La clase principal se trata de *MeloMetricsApp* es el punto de arranque de la aplicación y sirve como punto de conexión entre el resto de clases del sistema. Para los diferentes test del sistema se ha empleado herencia con una clase *ParentView* de la que descienden un hijo por cada test que ofrece la aplicación, con este mecanismo añadir nuevos test al sistema se agiliza al tener una clase padre con las características que estos suelen tener en común. Cada test cuenta con su propio reloj para avanzar la ejecución del test en cuestión. *MainDelegete* se encarga de capturar los deslizamientos de pantalla para cambiar de test, empezar un test o detenerlo en el caso del VO2Max Speed Test.

Por último están las clases encargadas de registrar un parámetro que es necesario que usuario introduzca por pantalla para realizar el test Vo2Max Speed, *KeyBoardView* y *KeyBoardViewDelegate*.

5.1.2 FOREVER FIT WEB

Forever Fit Web utiliza el patrón Modelo-Vista-Controlador como arquitectura. Aunque es popularmente conocido como solo un patrón de diseño, se aplicará durante toda la implementación de la página web por lo que pasa a ser su arquitectura. La razón principal para elegir esta arquitectura es que, desde el principio del proyecto se estableció que el portal web sería desarrollado con el asp.net. Este framework está centrado en una arquitectura MVC desde hace varias versiones, aunque todavía ofrece la posibilidad de una arquitectura basada en Web Forms [9]. Además MVC es un patrón para aplicaciones con arquitecturas bien estructuradas, orientadas a pruebas y fáciles de mantener, por lo que es una opción ideal para el proyecto [10]. Además esta estructura permite una separación de conceptos y funcionalidades que facilitan el desarrollo. A continuación un ejemplo ilustrativo de cómo funciona la arquitectura del software:



El usuario visualiza la vista final a través del navegador.

El usuario genera acciones en el navegador como puede ser pulsar un link o un botón. El resultado es una petición a una URL.

Esta petición a URLs son traducidas por el módulo correspondiente e invocan al método/action y controlador asociados. El controlador puede recibir información de la vista. Por ejemplo: `home/setUserName?id=2&name=test`

Controller: Home

Action: SetUserName

Parámetros Opcionales método GET en este caso:

Id = 2

name=test

El controlador se encarga de ejecutar su implementación. Normalmente implican acciones de consulta o modificaciones al modelo. En caso de cambios en el modelo este se encarga de notificar a la vista para que actualice la información y de interactuar con la base de datos..

El controlador devuelve un ActionResult que puede contener html, texto plano, variables de entorno etc. La vista se encarga de consultar esta información, consultar el modelo y renderizarla hacia el código de la web. De esta forma el navegador solo tiene que interpretar el código final.

Ilustración 15. Esquema arquitectura lógica Forever Fit Web

De esta forma toda el sistema queda separada en 3 tipos de entidades principales cada una de ellas con un objetivo claro. La arquitectura MVC bajo el framework asp.net utiliza una estructura de archivos y nomenclatura que un desarrollo ordenado y simple de seguir:

Raíz[directorio]

- **Controller** [directorio]
 - Archivos Controlador .CS
- **Models** [directorio]
 - Archivos Modelo .CS
- **Views** [directorio]
 - **Views**[directorio]
 - Archivos Vista aspx/ascx/cshtml

Como sus respectivos nombres indican, se encuentran 3 directorios principales para los controladores, modelos y vistas. Las vistas son asociados automáticamente al controlador con el mismo nombre de la carpeta en que se encuentra y de forma análoga una vista se asocia, al método/action del controlador correspondiente, con su mismo nombre. Finalmente los modelos son asociados a las vistas y controladores que tengan su mismo nombre. Una petición lanzada con la URL: /users/login seguiría la siguiente secuencia de accesos generalmente:



Para Forever Fit Web la estructura principal de ficheros siguiendo la arquitectura establecida se muestran en las siguientes ilustraciones:

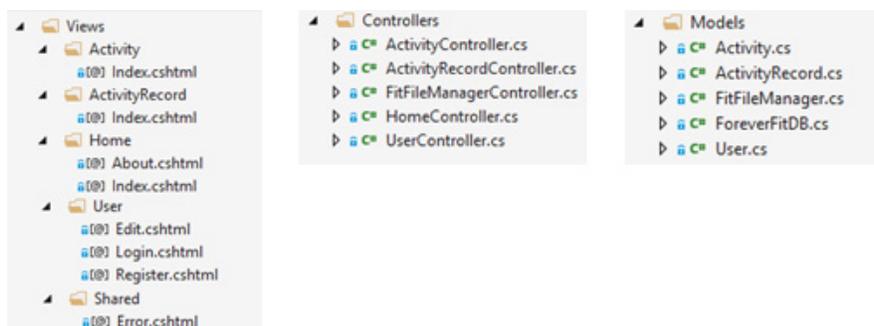


Ilustración 16. Estructura ficheros FFW: vistas, controladores y modelos

En los siguientes subapartados se detallarán las diferentes partes de la arquitectura que forman el sistema de FFW, es decir, las vistas, controladores y modelos para finalizar con el diagrama de clases general del sistema y la navegación entre ellas.

5.1.2.1 MODELO

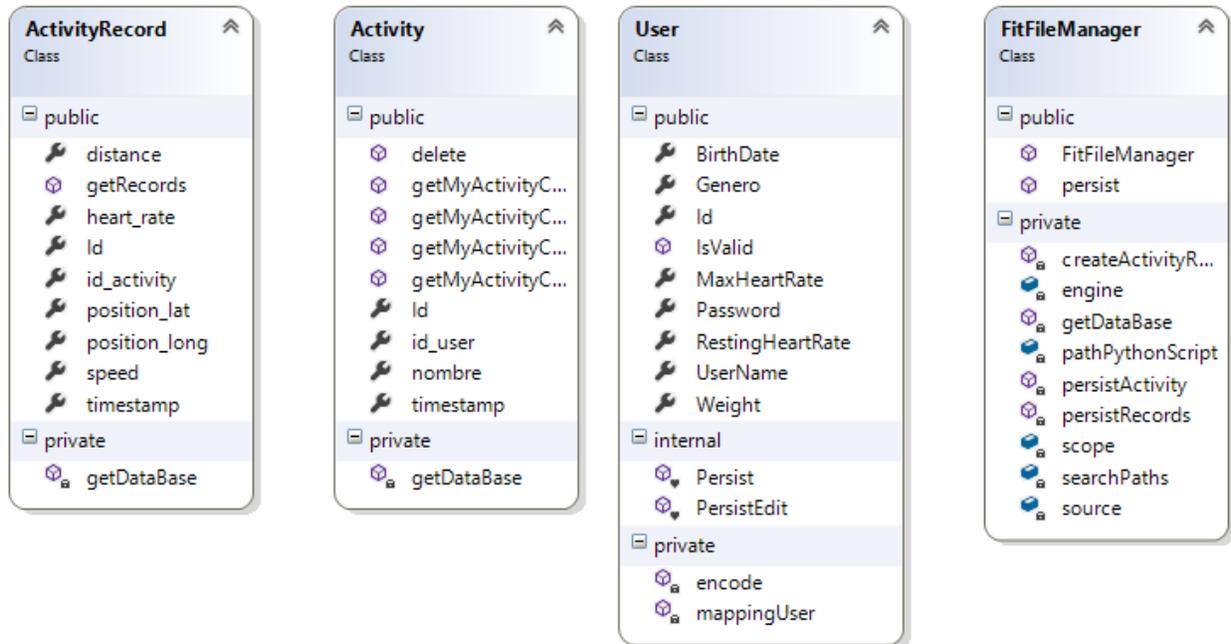


Ilustración 17. Diagrama de clases de modelos

ActivityRecord: esta clase modela un registro de todo el conjunto de registros de un activity. Los registros son independientes entre ellos pero dependientes del activity al que pertenecen. También se encarga de recuperar los registros de un activity de la base de datos.

Activity: esta clase modela la información identificativa de un activity. Notar que no hay una relación directa de agregación desde la clase de ActivityRecord como cabría pensar. Con esto se pretende por un lado mantener la separación entre modelos que fomenta la arquitectura y por otro lado mejorar la eficiencia del sistema ya que los registros de un activity (los cuales normalmente sobrepasan la centena) solo serán cargados bajo demanda, al consultar su detalle. La clase también se encarga de recuperar los activities de la base de datos.

User: esta clase se encarga de modelar y administrar a los usuarios. Se encarga también de realizar las comunicaciones con la base de datos SQL del sistema para identificar, persistir o recuperar información del usuario.

FitFileManager: esta clase primeramente se encarga de subir el fichero .FIT en el directorio del servidor. Posteriormente se encarga ejecutar el script programado en Python que se encarga de descodificar los archivos .FIT, los cuales están codificados en bytes y preparar el resultado en formato de cadena de texto. Finalmente se encarga de acceder a la base de datos para persistir tanto el nuevo activity como sus registros.

Notar que solo los modelos del sistema establecen interacciones directas con la base de datos tal y como indica la arquitectura MVC, manteniendo hacia una separación uniforme de funcionalidades y dependencias en el sistema. También remarcar que los métodos de acceso a base de datos de los modelos son estáticos o de la clase ya estos no dependen de una instancia en concreto sino del conjunto.

5.1.2.2 VISTA

En la arquitectura MVC las vistas abarcan los componentes muestran la interfaz visual al usuario y con la cual este interactúa. Generalmente es la representación de un modelo. Por ejemplo una vista sería una pantalla en que viéramos la información del usuario identificado en la aplicación.

Notar que estos archivos por si solos no son clases, por lo que no se presenta su estructura en forma diagrama de clases. Se tratan de archivos que combinan HTML, CSS y código C# para generar la el código final que el navegador interpretará para mostrar la vista al usuario. Cada uno corresponde a una pantalla del sistema. A partir de aquí se presenta el mapa de navegación entre estas vistas y la descripción detallada de cada una de ellas.

5.1.2.2.1 MAPA DE NAVEGACIÓN

A continuación el mapa e navegación de la aplicación se ha realizado un diseño que facilite el acceso al mayor número de funcionalidades independientemente de la vista en que se encuentre el usuario.

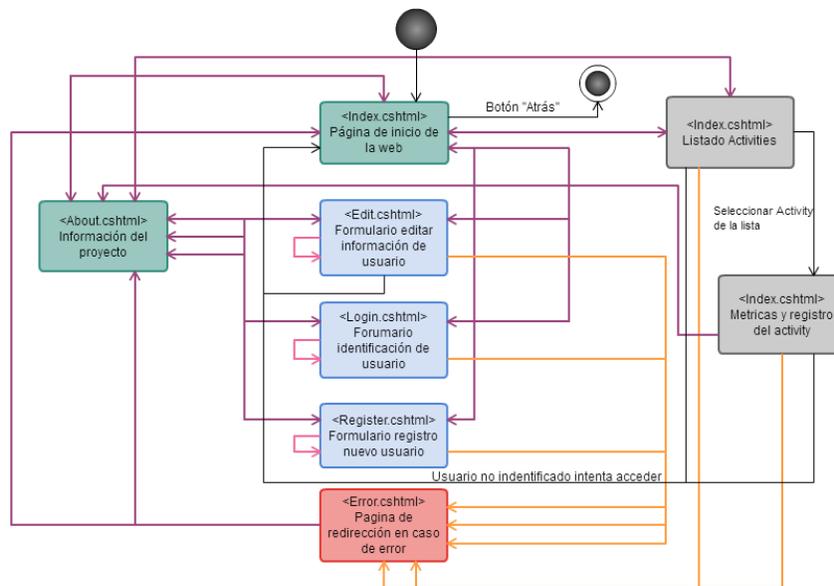


Ilustración 18. Mapa de navegación Forever Fit Web

Los links de Inicio, Acerca de, Registrarme, Registrarme, Editar y Listado Activities se encuentra en el menú principal de la aplicación el cual es fijo y visible siempre. A continuación la trama del mapa de navegación:

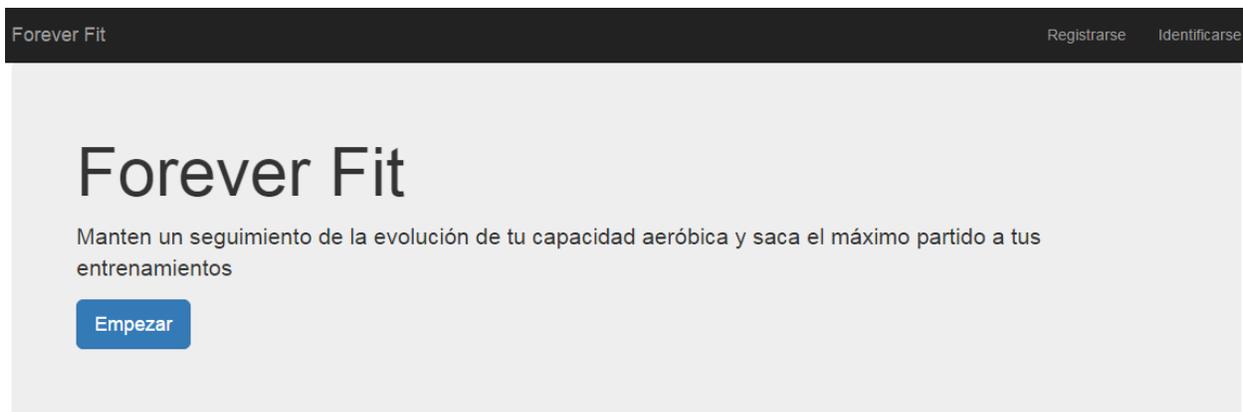
-  Navegación bidireccional entre pantallas del menú principal. Se puede ir de origen a destino y viceversa.
-  Redireccionamiento automático en caso de error al intentar cargar la pantalla por ejemplo un problema de conexión a la base de datos al intentar subir un archivo de un formato incorrecto.
-  Redireccionamiento automático a la misma pantalla por comprobación de formularios, por ejemplo falta algún dato o no está en el formato correcto.

5.1.2.2.2 PANTALLAS

En este apartado se encuentra la pantallas que forman el sistema y la descripción de su contenido, funcionalidades y propósito.

5.1.2.2.2.1 INICIO

Pantalla inicio de la aplicación al entrar. Es publica por lo tanto visible para los usuarios no identificados en la aplicación. Ofrece una breve descripción del objetivo de Forever Fit y detalla cómo se deben ejecutar los diferentes test del sistema en Forever Fit Smartwatch. Se puede acceder a las pantallas de registro de usuario, identificación o *Acerca de*.



One Mile Walk Test

En este predictor del VO2Max se tienen en cuenta: genero, peso y edad del usuario. Conjuntamente con el tiempo que se ha tardado en recorrer una milla y el ritmo cardiaco en dicho momento. En este test no se debe correr sino caminar tan rápido como sea posible. Supone un test suave sin necesidad de realizar un esfuerzo máximo por parte del usuario.

One Half Mile Run Test

Este test consiste en recorrer 1.5 millas en el menor tiempo posible para tener una estimación de su VO2MAX, por lo tanto es una prueba de esfuerzo. Está permitido correr o caminar. Debido a que requiere un esfuerzo importante el usuario debe ser cuidadoso y no sobrepasar sus capacidades. En este test no se tiene en cuenta el tiempo que se tarda en recorrer la distancia marcada.

Vo2max Speed

Predictor de la velocidad máxima alcanzable al llegar al VO2Max, este test requiere una **duración mínima de 12 minutos**, apartir de entonces la estimación será continua por lo que la estimación final puede ser ligeramente diferente a la inicial. Para este test es necesaria la fruencia cardiaca en reposo del usuario y su frecuencia cardiaca máxima, mientras mas precisos sean estos datos mayor será la estimación.

[Acerca de Forever Fit](#) | © 2016 - My ASPNET Application

Ilustración 19. Pantalla inicio usuario no identificado

Al usuario identificarse en la aplicación los links del menú superior principal del sistema pasa de tener los links de *Registrarme* o *Identificarme* a los de *Hola [nombre usuario]* para editar su información de usuario y *Salir* para cerrar su sesión. Además se añade el link *Activities*. El comportamiento de este menú es el mismo en toda la aplicación por lo que en las siguientes pantallas no se describirá otra vez para evitar redundancia innecesaria.

Forever Fit

Manten un seguimiento de la evolución de tu capacidad aeróbica y saca el máximo partido a tus entrenamientos

One Mile Walk Test

En este predictor del VO2Max se tienen en cuenta: genero, peso y edad del usuario. Conjuntamente con el tiempo que se ha tardado en recorrer una milla y el ritmo cardiaco en dicho momento. En este test no se debe correr sino caminar tan rápido como sea posible. Supone un test suave sin necesidad de realizar un esfuerzo máximo por parte del usuario.

One Half Mile Run Test

Este test consiste en recorrer 1.5 millas en el menor tiempo posible para tener una estimación de su VO2MAX, por lo tanto es una prueba de esfuerzo. Está permitido correr o caminar. Debido a que requiere un esfuerzo importante el usuario debe ser cuidadoso y no sobrepasar sus capacidades. En este test no se tiene en cuenta el tiempo que se tarda en recorrer la distancia marcada.

Vo2max Speed

Predictor de la velocidad máxima alcanzable al llegar al VO2Max, este test requiere una **duración mínima de 12 minutos**, apartir de entonces la estimación será continua por lo que la estimación final puede ser ligeramente diferente a la inicial. Para este test es necesaria la fruencia cardiaca en reposo del usuario y su frecuencia cardiaca máxima, mientras mas precisos sean estos datos mayor será la estimación.

Acerca de Forever Fit | © 2016 - My ASPNET Application

Ilustración 20. Pantalla inicio usuario identificado

5.1.2.2.2 REGISTRARME

Este pantalla incluye principalmente el formulario para dar de alta un nuevo usuario en el sistema. En caso de error interactuando con la base de datos o interno en la aplicación redireccionará, a la página de error del sistema. En caso de un registro exitoso redireccionará a la página de inicio con la sesión del usuario iniciada.

Registrar

Usuario

Usuario	<input type="text"/>
Contraseña	<input type="password"/>
Genero	<input type="text" value="Hombre"/>
Peso	<input type="text"/>
Fecha de nacimiento	<input type="text" value="dd/mm/aaaa"/>
Frecuencia cardiaca máxima	<input type="text"/>
Frecuencia cardiaca en reposo	<input type="text"/>
	<input type="button" value="Crear"/>

Acerca de Forever Fit | © 2016 - My ASPNET Application

Ilustración 21. Pantalla registro usuario

En caso de que algún campo este incorrecto se o que el nombre de usuario ya esté en uso se redirecciona a la misma página mostrando los mensajes pertinentes.

The screenshot shows the 'Registrar Usuario' page. At the top, there is a dark navigation bar with 'Forever Fit' on the left and 'Regístrame' and 'Identificarme' on the right. Below the navigation bar, the title 'Registrar' is displayed in a large font, followed by 'Usuario' in a smaller font. The registration form consists of several fields: 'Usuario' (text input), 'Contraseña' (text input), 'Genero' (dropdown menu with 'Hombre' selected), 'Peso' (text input), 'Fecha de nacimiento' (text input with placeholder 'dd/mm/aaaa'), 'Frecuencia cardiaca máxima' (text input), and 'Frecuencia cardiaca en reposo' (text input). Each text input field has a red error message below it: 'El campo Usuario es obligatorio.', 'El campo Contraseña es obligatorio.', 'El campo Peso es obligatorio.', 'El campo Fecha de nacimiento es obligatorio.', 'El campo Frecuencia cardiaca máxima es obligatorio.', and 'El campo Frecuencia cardiaca en reposo es obligatorio.'. A 'Crear' button is located at the bottom of the form. At the bottom of the page, there is a footer with the text 'Acerca de Forever Fit | © 2016 - My ASP.NET Application'.

Ilustración 22. Pantalla registro usuario con error en los campos

5.1.2.2.2.3 IDENTIFICARME

Esta página es la entrada al sistema para identificarse y poder utilizar todas las funcionalidades de FFW. En caso de algún problema con los campos introducidos en el formulario se redirecciona a la misma pantalla mostrando los mensajes informativos pertinentes. En caso de error interactuando con la base de datos o interno en la aplicación, redireccionará a la página de error del sistema. En caso de una identificación exitosa redireccionará a la página de inicio.

The screenshot shows the user login page. At the top, there is a dark navigation bar with 'Forever Fit' on the left and 'Regístrame' and 'Identificarme' on the right. Below the navigation bar, the text 'Utiliza una cuenta para identificarte' is displayed. The login form consists of two text input fields: 'Usuario' (containing 'majorika') and 'Contraseña' (containing '****'). A 'Entrar' button is located below the password field. At the bottom of the page, there is a footer with the text 'Acerca de Forever Fit | © 2016 - My ASP.NET Application'.

Ilustración 23. Pantalla de identificación del usuario

5.1.2.2.2.4 EDITAR INFORMACIÓN USUARIO

En caso de que un usuario no identificado intente acceder será redireccionado a la pantalla de identificación. En esta pantalla se encuentra un formulario que permite modificar la mayoría de propiedades del usuario. En caso de error interactuando con la base de datos o interno en la aplicación redireccionará, a la página de error del sistema. En caso de que se realicen los cambios de forma exitosa se redirecciona a la misma pantalla.

Forever Fit Activities Hola majorika!

View

User

Usuario	<input type="text" value="majorika"/>
Genero	<input type="text" value="Mujer"/>
Peso	<input type="text" value="70"/>
Fecha de nacimiento	<input type="text" value="04/06/2016"/>
Frecuencia cardiaca máxima	<input type="text" value="180"/>
Frecuencia cardiaca en reposo	<input type="text" value="59"/>

Acerca de Forever Fit | © 2016 - My ASP.NET Application

Ilustración 24. Pantalla editar información usuario

En caso de que algún campo este incorrecto se o que el nombre de usuario ya esté en uso se redirecciona a la misma página mostrando los mensajes pertinentes.

Forever Fit Activities Hola majorika!

View

User

Usuario	<input type="text" value="majorika"/>
Genero	<input type="text" value="Mujer"/>
Peso	<input type="text" value="70"/>
Fecha de nacimiento	<input type="text" value="04/06/2016"/>
Frecuencia cardiaca máxima	<input type="text"/>
Frecuencia cardiaca en reposo	<input type="text"/>

Acerca de Forever Fit | © 2016 - My ASP.NET Application

Ilustración 25. Pantalla editar información usuario comprobación campos

5.1.2.2.2.5 ACTIVITIES

Esta página es solo accesible para los usuarios identificados. En caso de que un usuario no identificado intente acceder será redireccionado a la pantalla de identificación. Primeramente se encuentran los componentes para subir nuevos archivos .FIT al sistema y persistirlos. Debajo de estos componentes un campo de texto para realizar una búsqueda entre los Activities del sistema por nombre. En la tabla se muestra el listado activities del usuario. Las columnas de *Fecha Realización* y *Nombre* permiten efectuar una ordenación ascendente o descendente al pulsarla su título. Al hacer pulsar el nombre de un activity nos envía a la pantalla del detalle de este activity. El listado cuenta con un sistema de paginación de 13 elementos por página.

En caso de error, al intentar subir un archivo, cargar la página, ordenar la tabla, interactuando con la base de datos o interno en la aplicación, redireccionará a la página de error del sistema.

Nombre	Fecha Realización	Acciones
54L94545.FIT	21/04/2015 7:47:40	Eliminar
54L94545.FIT	21/04/2015 7:47:40	Eliminar
54LF5237.FIT	21/04/2015 13:52:37	Eliminar
54MH5016.FIT	22/04/2015 15:50:16	Eliminar
54MH5016.FIT	22/04/2015 15:50:16	Eliminar

Ilustración 26. Pantalla activities

5.1.2.2.2.6 REGISTROS ACTIVITY

Esta página es solo accesible para los usuarios identificados. En caso de que un usuario no identificado intente acceder será redireccionado a la pantalla de identificación. Esta pantalla es la encargada de mostrar toda la información del activity: un resumen sobre sus datos básicos, el resultado de las métricas del sistema una vez analizados los datos y una tabla con todos los registros que forman el activity. Cada registro está formado por un momento, latitud, longitud, distancia, velocidad y ritmo cardiaco. Esta información es la recogida por el smartwatch durante la realización de un test. Dichos registros están paginados para consultarlos de forma más cómoda, 13 elementos por página. En caso de error interactuando con la base de datos o interno en la aplicación, redireccionará a la página de error del sistema.

Momento	Latitud	Longitud	Distancia	Velocidad	Ritmo Cardiaco
21/04/2015 7:47:40	591842447	73171662	541,71	5,944	125
21/04/2015 7:47:41	591843013	73171327	547,36	5,645	125
21/04/2015 7:47:44	591844531	73170152	563,17	5,421	125
21/04/2015 7:47:46	591845648	73169478	574,35	5,328	124
21/04/2015 7:47:47	591846174	73169111	579,74	5,44	124
21/04/2015 7:47:54	591849710	73165861	618,13	5,514	124
21/04/2015 7:47:56	591850643	73164811	628,89	5,458	124
21/04/2015 7:48:01	591852773	73161495	657,22	5,617	126
21/04/2015 7:48:03	591853494	73159988	668,55	5,598	126
21/04/2015 7:48:07	591854669	73157548	686,93	3,798	126
21/04/2015 7:48:11	591855557	73155862	700,09	3,434	123
21/04/2015 7:48:17	591857263	73152586	725,5	4,721	122
21/04/2015 7:48:21	591859093	73149439	751,02	5,421	122

Página 1 de 12

1 2 3 4 5 6 7 8 9 10 ... » »»

Ilustración 27. Pantalla registros activity

5.1.2.2.2.7 ERROR

Cuando ocurre algún error en el sistema el usuario es redireccionado a esta pantalla informativa.

Forever Fit Activities	Hola majorikal Salir
<h2>Problema al cargar la pagina Error</h2> <p>La página solicitada no se pudo encontrar, pongase en contacto con su webmaster o inténtelo de nuevo. Pulsa el botón Atrás para navegar a la página que has venido de previamente</p> <p>O simplemente puede pulse el siguiente botón:</p> <p style="text-align: center;">Llévame a casa</p>	
<p>Acerca de Forever Fit © 2016 - My ASPNET Application</p>	

Ilustración 28. Pantalla error

5.1.2.3 CONTROLADOR

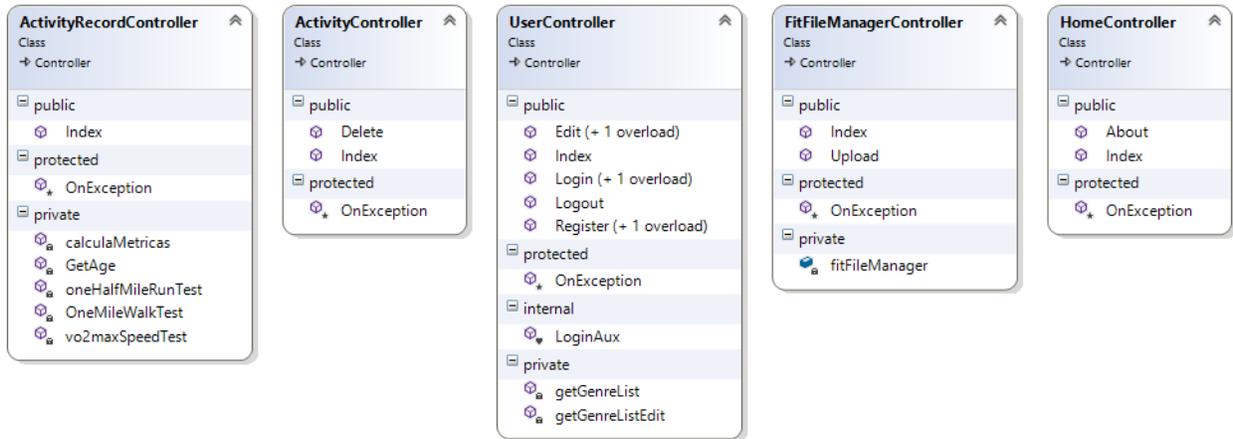


Ilustración 29. Diagrama de clases de controladores

ActivityRecordController: este controlador se encarga de manejar las peticiones que se envían hacia la pantalla de registros de un activity. Notar que solo hay una petición posible, que es ver el detalle de un activity, se maneja a través vez del método *Index*. El resto de métodos son privados y se utilizan para el cálculo de las métricas u obtener información intermedia.

ActivityController: se encarga de administrar las peticiones enviadas hacia la pantalla de activities. Estas peticiones pueden ser dos *Index* para ver la pantalla con el listado de activities o *Delete* para borrar algún activity del listado.

UserController: este controlador se encarga de manejar las peticiones que se envían hacia las pantallas de usuario como son Identificarme, Regístrame, Identificar usuario o Editar información usuario. Se aprecia que tiene algunos métodos con doble definición (*overload*) debido a que uno es para entrar en la página y el otro es para cuando se envía la información de los formulario, métodos GET y POST respectivamente.

FitFileManagerController: la única función de este controlador es recibir la petición desde la pantalla de Activity para subir un archive .FIT. Es la acción que se llama al presionar el botón *aceptar* de la pantalla del listado de activities. La acción desencadena todo el proceso de lectura y persistencia del fichero .FIT.

HomeController: se encarga de manejar las peticiones hacia las pantallas de Inicio y Acerca de. Solo muestra información no tiene ninguna lógica interna remarcable, aparte de ver si hay una sesión de usuario activa para mostrar el menú principal correspondiente.

Notar que todos los controladores cuenta con un método *OnException* para redireccionar a la pantalla de Error en caso de algún conflicto en el sistema.

5.1.2.4 DIAGRAMA DE CLASES FINAL

A partir del esquema conceptual visto en el apartado 4.2 Forever Fit Web y de la arquitectura elegida para la implementación del sistema se define el diagrama de clases final.

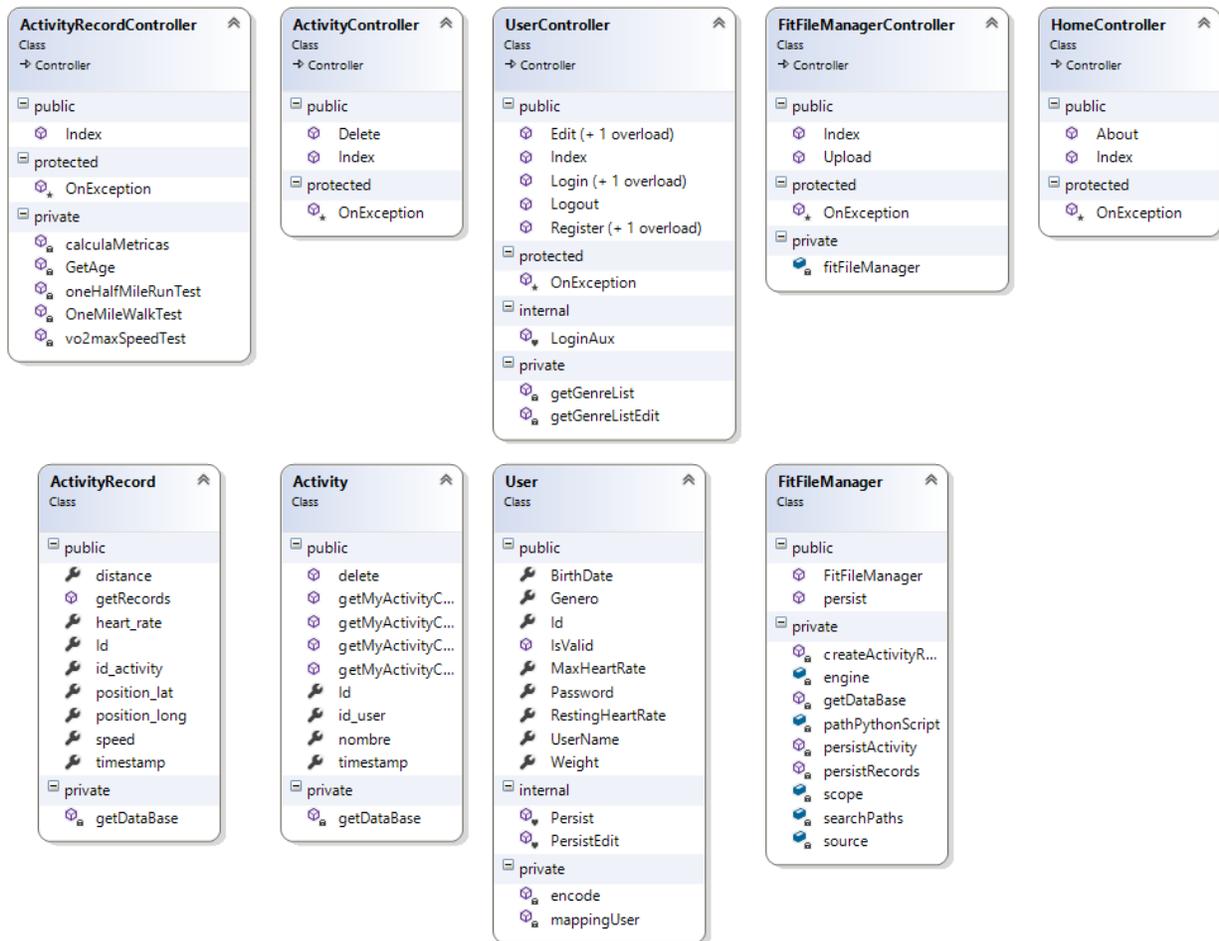


Ilustración 30. Diagrama final de clases FFW

Notar que las clases no tienen ninguna relación directa entre ellas ya que la arquitectura MVC lo evita, por ejemplo el controlador de usuarios no contiene un vector de modelos de usuario pero en cambio sí usa el modelo usuario.

Las vistas están preparadas para enviar y recibir instancias del modelo (el cual tienen asignado por la nomenclatura de la arquitectura explicada en el apartado 5.1.2. Forever Fit Web) y recibir las notificaciones de este, cuando es actualizado. Por lo tanto aunque el controlador internamente trabaje con otros modelos las entidades que recibe o devuelve deben de corresponder con el modelo al que está asociado la vista que llama al método y que capture lo que devuelva. Para entender mejor las relaciones de uso se presenta el siguiente mapa código:

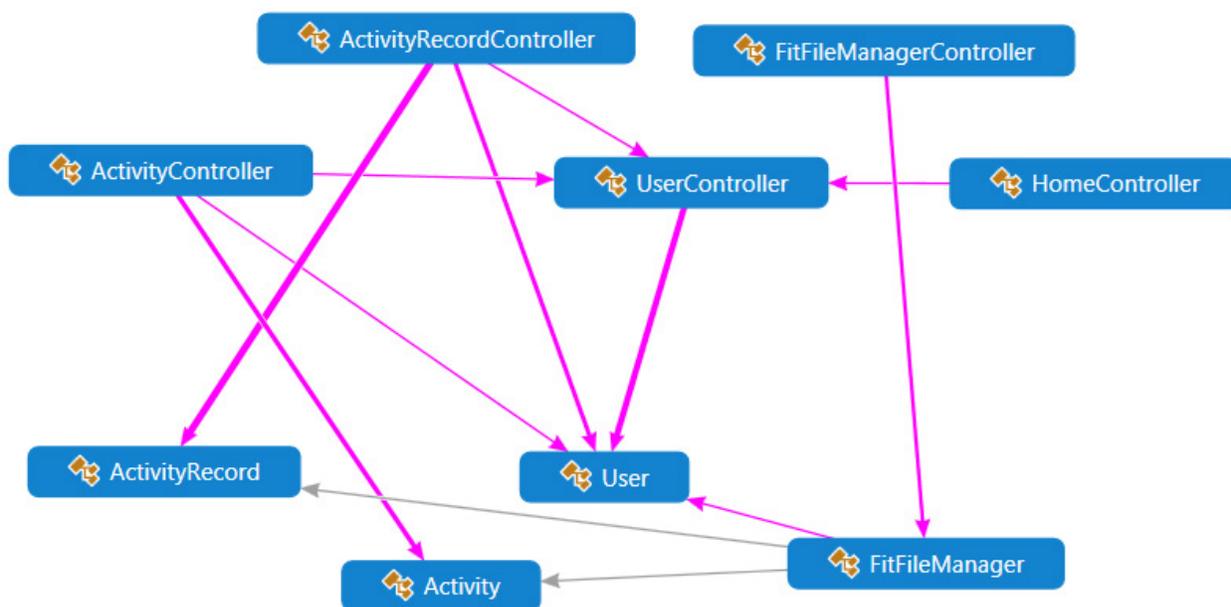


Ilustración 31. Mapa de código FFV

En la parte superior del diagrama se encuentran los controladores mientras que en la inferior están los modelos del sistema.

Las llamadas entre controladores se realizan solo con UserController presente en alguno de los extremos de la comunicación, esto es debido a que como se ha visto anteriormente ciertas pantallas necesitan comprobar que hay un usuario identificado en el sistema. Por la misma razón son también populares las llamadas al modelo User además de que se necesitan de propiedades de esta clase para calcular algunas métricas.

FitFileManager presenta llamadas de un color diferente hacia Activity y ActivityRecord ya que no los utiliza directamente sino que los referencia para realizar la persistencia de estos en la base de datos.

5.2 ACCESO A LOS DATOS

En esta sección se trata el acceso a los datos que realiza el sistema, se describe como es realizada la persistencia de la información y como es recuperada. En el análisis de las alternativas del proyecto, en concreto el apartado 6.4.2. Base De Datos, se profundiza en el porqué de estas elecciones para el proyecto.

5.2.1 FOREVER FIT SMARTWATCH

La persistencia de FFS se realiza únicamente a través de ficheros .FIT. Cada vez que el usuario inicia un test la aplicación se encarga de empezar la generación del fichero .FIT, la generación de este fichero es administrada y llevada a cabo por el sistema operativo de Garmin y a través de instrucciones de la API FFS se encarga de iniciar su creación, detenerla y guardar el resultado final.

El sistema operativo Garmin se encarga de dejar los archivos .FIT que son generados por el sistema en una carpeta donde el usuario puede recuperarlos posteriormente conectando el dispositivo a un ordenador mediante un cable USB.

5.2.2 FOREVER FIT WEB

El acceso a los datos de FFW se realiza utilizando dos bases de datos diferentes. Para acceder a los datos del usuario que utiliza el sistema se utiliza una base de datos SQL, la cual contiene una única tabla con la estructura mostrada a continuación.

```
CREATE TABLE [dbo].[User] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Username] NVARCHAR (50) NOT NULL,  
    [Password] NVARCHAR (MAX) NOT NULL,  
    [RegDate] DATETIME DEFAULT (getdate()) NOT NULL,  
    [Genero] BIT DEFAULT ((0)) NOT NULL,  
    [BirthDate] DATE DEFAULT (getdate()) NOT NULL,  
    [MaxHeartRate] FLOAT (53) DEFAULT ((186.0)) NOT NULL,  
    [RestingHeartRate] FLOAT (53) DEFAULT ((56.0)) NOT NULL,  
    [Weight] FLOAT (53) DEFAULT ((70.0)) NOT NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

Ilustración 32. Estructura base de datos SQL FFW

Para preservar la privacidad del usuario, la contraseña de este es guardada encriptada en el sistema utilizando un algoritmo de hash seguro (SHA1). La tabla es utilizada para persistir y consultar la información de los diferentes usuarios del sistema. La interacción con la base de datos SQL se realiza únicamente a través del modelo usuario como se ha detallado anteriormente en este capítulo.

En el caso de la persistencia relacionada con los ficheros .FIT, se utiliza una base de datos NoSQL orientada a documentos, MongoDB. La estructura de los documentos de la base de datos es la siguiente:

```
{  
  "_id" : ObjectId("5766de5cf23adc1f64c1dff2"),  
  "id_user" : "12",  
  "nombre" : "54L94545.FIT",  
  "timestamp" : ISODate("2015-04-21T07:47:40.000Z")  
},  
{  
  "_id" : ObjectId("5766de5cf23adc1f64c1e081"),  
  "id_activity" : "5766de5cf23adc1f64c1dff2",  
  "timestamp" : ISODate("2015-04-21T07:58:35.000Z"),  
  "position_lat" : 592016561,  
  "position_long" : 73402356,  
  "distance" : 3899.64990234375,  
  "speed" : 0,  
  "heart_rate" : 115  
},
```

Ilustración 33. Estructura MongoDB ejemplo de documento activities y documento activityRecord

La estructura está formada por dos colecciones, primeramente la colección *activity* que se encarga de registrar la información básica del fichero .FIT y la colección *activityRecord* que tiene todos los registros de los activities del sistema. Notar que la relación entre *activity* y *activityRecord* está normalizada ya que cada *activityRecord* guarda el identificador del *activity* al que pertenece, es decir una clave foránea.

La pantalla *activity* consulta la primera colección mientras que la pantalla del detalle de un *activity* consulta la segunda a partir del identificador del *activity* seleccionado.

6.1 INTRODUCCIÓN

El proyecto abarca el desarrollo de un sistema que apoye el entrenamiento de deportistas o personas que los asesoran, por un lado una aplicación que se ejecutará en el dispositivo *smartwatch* Garmin Epix que se encarga de ofrecer métricas que estimen el rendimiento aeróbico del usuario, por otra parte esta información podrá ser exportada a una plataforma web. Esta web también será desarrollada, el usuario solo tendrá acceso a su propia información y contará con funcionalidades para interpretar mejor la información importada.

Existen diferentes métodos para estimar el VO₂Max pero no se ofrecerán todos. Tanto el número de métodos ofrecidos como la profundidad de su análisis en el portal web, serán en función del tiempo disponible y de las posibilidades de estimarlos con las tecnologías de las que se dispone

La realización del proyecto constara de diversas fases bien diferenciadas para enmarcarlo en un TFG de aproximadamente 4 meses de duración:

1. La primera fase será previa al desarrollo , se definirá el problema a resolver, su contexto, estado del arte, la planificación temporal y económica inicial y sostenibilidad. Esta primera parte corresponde a la documentación generada durante la asignatura de GEP. En paralelo a las siguientes fases esta documentación se verá incrementada para formar la memoria final del proyecto..
2. La segunda fase enmarca todas las actividades de desarrollo de las aplicaciones del proyecto tanto para el *smartwatch* como la versión web a donde se exportaran las métricas del usuario. Como se detalla más adelante, en la página 56 Metodología y rigor, esta fase estará formada por varios *sprints* siguiendo una metodología *scrum*.
3. La última fase corresponderá a preparar la defensa final del proyecto y completar la memoria.

6.2 POSIBLES OBSTÁCULOS

A continuación, se exponen las situaciones más destacables que pueden poner el riesgo el cumplimiento de los objetivos del proyecto.

6.2.1 RESTRICCIÓN TEMPORAL

Al ser un proyecto de final de grado el desarrollo del sistema propuesto tiene un tiempo limitado de 4 meses aproximadamente. Como se ha mencionado anteriormente se desea obtener una aplicación con funcionalidades, aunque sean pocas, completas y a partir de su finalización añadir nuevas al sistemas, optando siempre por la calidad en frente de la cantidad. Una planificación temporal detallada será esencial para detectar estancamientos en el desarrollo y hacer los cambios pertinentes para resolver dichas desviaciones.

6.2.2 TECNOLOGÍAS NUEVAS

Los *smartwatch* Garmin funcionan ejecutando un lenguaje orientado a objetos propio, Monkey C. Esto hace que toda la información relacionada al respecto se reduzca a la ofrecida por la propia empresa. No se cuenta con una amplia comunidad, foros independientes o desarrolladores que compartan problemas y soluciones en proyectos similares.

De la misma forma que con Monkey C, toda la información relacionada a la comunicación con el exterior se reduce a la información oficial de Garmin. Se ofrecen diversas posibilidades de comunicación con el exterior (Garmin Connect API, Connect IQ SDK o Ant Wirelles Networks) pero es necesario un estudio de cada una para encontrar cual se adapta mejor a las necesidades del proyecto. Se espera un coste alto en tiempo para esta fase del proyecto.

6.2.3 PRUEBAS Y PROTOTIPO

Las pruebas de la aplicación se realizarán con el dispositivo *smartwatch* Garmin Epix, no obstante el SDK cuenta con un emulador con el que poder realizar las pruebas inmediatas durante el desarrollo. El emulador no puede ejecutar todas las funcionales del dispositivo real como por ejemplo algunos métodos de comunicación por lo tanto, las pruebas pueden llevar periodos importantes de tiempo.

Por un lado las acciones de tener que reinstalar la aplicación en el *smartwatch* para ver los cambios realizados. Por otro las pruebas en el dispositivo requerirán realizar actividad física para poder recoger la información necesaria a través de los sensores, esto es debido a que las estimaciones más fidedignas del rendimiento aeróbico precisan de esto y no se realizan simplemente introduciendo los datos manualmente [2, p. 5 Tabla 2].

6.3 METODOLOGÍA Y RIGOR

6.3.1 FITA INICIAL

El desarrollo del sistema se realizarán utilizando una metodología ágil, *scrum*. La elección de esta metodología es debido a que permite obtener resultados pronto, donde los requisitos son cambiantes o pocos definidos. Esto se adapta perfectamente al proyecto ya que se cuenta con poco más de 4 meses para su finalización y pueden encontrarse obstáculos, sobre todo relacionados con las tecnologías desconocidas ver sección 6.2.2.

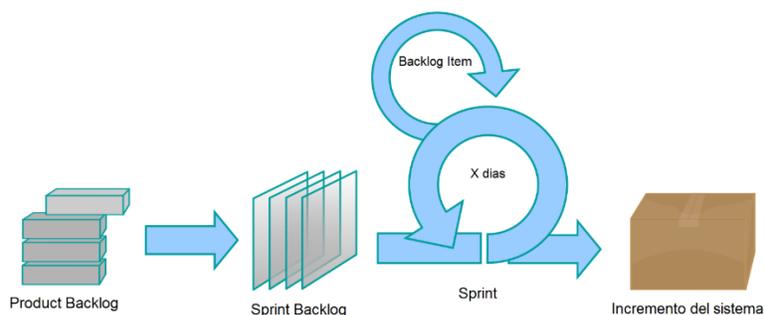


Ilustración 34. Esquema metologia de trabajo

El *product backlog* estará formado por todas las funcionalidades de sistema priorizadas. Los *sprints* dedicados a la estimación del rendimiento aeróbico estarán formados un *sprint backlog* que cumpla los siguientes objetivos:

1. Obtener cálculo de métrica.
2. Guardar el historial de la métrica.
3. Consultar historial métrica.
4. Exportar métrica
5. Recuperar métrica.

Cabe esperar que una vez acabado el primer *sprint* dedicado al cálculo de una métrica en el esto de *sprints* de este tipo, los objetivos 1-5 necesitarán un periodo de tiempo menor ya que el camino a recorrer será similar.

El resto de *sprints* se dedicarán al análisis de los datos recuperados en el portal web, las funcionalidades de autenticación en este, mejor el aspecto visual ..etc.

6.3.1.1 MÉTODO DE VALIDACIÓN

Dentro de cada *sprint* se contará con un periodo para realizar las pruebas de validación que confirmen que las funcionalidades se han desarrollado correctamente. Mediante el emulador del dispositivo se podrá probar el funcionamiento de las funcionalidades de forma más inmediata. Las pruebas finales siempre se realizará en el *smartwatch* Garmin Epix del que se dispone para el proyecto. El producto incremental resultado de cada *sprint* también será facilitado al director del proyecto para su validación.

Por otro lado la tienda de aplicaciones de Garmin permite subir y validar apps en menos de dos días por lo que se publicarán en ella diferentes versiones de la aplicación para asegurar que cumple con los estándares del fabricante.

6.3.2 FITA DE SEGUIMENT

Como inicialmente se planteó, se ha utilizado una metodología ágil en el desarrollado del proyecto, pero con modificaciones en la finalidad de los *sprints* y en su orden. Inicialmente se estableció que los *sprints* dedicados a la estimación de métricas fueran desde la implementación completa de la misma en la aplicación del *smartwatch*, pasando por su exportación del dispositivo y posterior importación en la aplicación web. Esto ha cambiado debido a dos razones principales:

- El simulador ofrecido por el fabricante no emula las características del dispositivo que permiten generar los archivos necesarios con los datos para ser exportados. Inicialmente ya se contaba con este inconveniente al realizar un estudio de las prestaciones del simulador del dispositivo como de la API del software Garmin. El simulador solo permite enviar las instrucciones de creación pero no generar el archivo en sí. Tampoco permite modificar o consultar los valores del fichero mientras lo está creando y está cargado en memoria.
- Aunque se contaba con un dispositivo *smartwatch* real para realizar las pruebas de la aplicación y por lo tanto generar los ficheros necesarios para la exportación, no ha estado disponible en ningún momento desde el inicio hasta la *fita de seguiment*.

De esta forma los *sprints* iniciales del proyecto se han centrado únicamente en completar la aplicación del dispositivo *smartwatch*. Se ha implementado la estimación de las diferentes métricas y probado el funcionamiento de la aplicación utilizando el simulador. La simulación de constantes vitales, inputs y outputs son correctas y fidedignas por los que en este aspecto el simulador ha cumplido su tarea en substitución del dispositivo real.

El simulador ha permitido tanto la generación de constantes vitales aleatorias así como desde archivos. Se han podido probar los diferentes test de la aplicación con los mismos inputs a partir de archivos .FIT con dicha información, este punto ha sido importante para detectar incongruencias en las estimaciones obtenidos. Cabe destacar que estos archivos son los mismos que el simulador no permite generar. Han sido obtenidos a partir de usuarios que han compartido la recogida de sus constantes vitales a través de internet [11].

Al haber acabado el desarrollo de la aplicación del *smartwatch* sin contar con él dispositivo, los siguientes *sprints* han pasado a desarrollar la aplicación web, sin cambios en la metodología ágil. Las pruebas de importación de archivos se han realizado con los mismo ficheros .FIT antes mencionados [11]. Para comprobar que la aplicación estaba importando los datos correctamente se han decodificado, vuelto a codificar en un formato entendible y posteriormente comparado con la información almacenada en la base de datos.

6.3.3 FITA FINAL

La metodología de trabajo del proyecto no ha sufrido cambios desde los mencionados en el momento de *la fita inicial*. La parte final del proyecto se ha dedicado a completar la memoria del proyecto y a implementar los últimos detalles de del sistema. En cuanto a las pruebas en el dispositivo real, no han sido posibles no se ha dispuesto del dispositivo en ningún momento, como en un principio se aseguró cuando la propuesta de proyecto fue publicada. Pese a esto, como también se ha mencionado anteriormente, la parte por probar en el dispositivo real compete a corroborar que las instrucciones para crear los archivos .FIT desde FFS se ejecutan correctamente. Lo que respecta al resto de pruebas se puede tomar con fiabilidad los resultados obtenidos con el simulador.

6.4 ANÁLISIS DE ALTERNATIVAS

A continuación se plantean los problemas más destacados el desarrollo de la aplicación, las diferentes alternativas para resolverlos y la resolución final conjuntamente con el razonamiento que ha llevado a su elección.

6.4.1 EXPORTACIÓN/IMPORTACIÓN DE FICHEROS

El manejo de ficheros ha sido desde el inicio del proyecto uno de los puntos más inciertos respecto a su implementación, se partió desde diferentes posibilidades para enfocar la resolución del problema intentado realizar la implementación más sencilla. El paso por estas fases implico una fuerte inversión de tiempo en investigar las posibilidades que ofrecía el sistema, aunque desde un primer momento este punto del proyecto se previó esta demora :

1. Inicialmente se pretendía generar un fichero en el *smartwatch* con los resultados de las métricas para ser exportado a la plataforma web, esta idea fue descartada debido a que el sistema operativo de Garmin no permite la generación de ficheros libremente en sus dispositivos.
2. El sistema genera ficheros .FIT que contienen información sobre las constantes vitales del usuario o posición entre otros datos durante el entrenamiento. Por lo que la siguiente alternativa consistió en intentar modificar la información de estos ficheros en tiempo de ejecución para escribir la información deseada. Esta acción tampoco es permitida la información de estos ficheros solo tiene acceso de lectura en tiempo de ejecución, además una vez creado el sistema tampoco permite su lectura y/o escritura.
3. Finalmente la solución ha sido utilizar los ficheros .FIT generados durante la realización de la actividad, exportar todos estos datos y volver a estimar las métricas en la plataforma web con los datos una vez importados.

La exportación de dichos ficheros también ha supuesto desafío importante ya que estos no se almacenan en un formato legible sino en bytes. Se ha tenido que incluir un módulo en Python [12] en el sistema que realiza la conversión a un fichero legible antes de persistirlo en la base de datos. Aunque el modulo realizaba la conversión de formato ha sido necesario estudiar el SDK de los ficheros .FIT [13] para entender y utilizar el modulo. Además se han tenido que realizar modificaciones en el mismo para adaptarlo a las necesidades del proyecto (filtrar información necesaria, modificar inputs y outputs, flujo de ejecución, ..etc) y que funcionara coordinadamente con las tecnologías ya utilizadas.

Aunque no era la idea inicial ni la más sencilla de implementar, como se detalla más adelante en el apartado 6.4.2, ha servido para aumentar el valor de la aplicación tanto a nivel funcional como de aprendizaje.

6.4.2 BASE DE DATOS

El siguiente apartado está enfocada a la persistencia y recuperación de datos en la plataforma web del sistema. La idea inicial respecto al almacenamiento de datos era contar con una base de datos relacional SQL o MySQL que se encarga de esta tarea, ya que se contemplaba solo la persistencia del perfil del usuario y las métricas obtenidas. Finalmente la información respectiva a los usuarios se persiste en una base de datos relaciona SQL mientras que la información referente a los ficheros .FIT importada se almacena en una base de datos no relacionado, en concreto se utilizó MongoDB un sistema de base de datos orientado a documentos.

La información relacionado con los usuarios sigue la idea inicial:

- Por un lado una base de datos SQL cumple de sobras las necesidades para persistir esta información.
- Las bases de datos relacionales presentan menos vulnerabilidades de seguridad y al estar almacenado información personal de los usuarios es importante esta característica [14].
- En futuras ampliaciones del sistema, este tipo de base de datos es más adecuado para realizar funciones de agregación con nuevas entidades como por ejemplo relacionar los usuarios con los diferentes dispositivos que poseen [15].

La información sobre las métricas ha cambiado respecto la idea inicial debido a que:

- Inicialmente solo se planteaba almacenar la información del resultado final de la métrica, eso sería entonces información del usuario y unos 3 valores más. Esto fue imposible por las razones comentadas en el apartado 6.4.1.

La aplicación almacena 6 constantes vitales diferentes (distancia, ritmo cardiaco, velocidad, longitud, latitud, fecha/ hora) del usuario que se toman en un intervalo de entre 1 milisegundo y 1 segundo. Por lo tanto en un solo test que se tarde unos 5 minutos en realizar estaríamos hablando de unas 300 filas con 1800 valores como mínimo. El test VO₂Max Speed necesita 12 minutos para realizar la estimación, 720 filas con 4320 valores en total y no hemos tenido ni siquiera en cuenta los valores necesarios para relacionar los registros con sus propio *activity* o con el usuario.

Manejando este volumen de registros una base de datos SQL convencional vería afectado su rendimiento en poco tiempo en una aplicación funcionando con un volumen decente de usuarios. La solución a este problema ha sido utilizar una base de datos no relacional orientada a documentos, en concreto MongoDB. Estas bases de datos se caracterizan por su flexibilidad y alto rendimiento en operaciones de lecturas [15] [16] [17] [18]. Lo cual es perfecto para el volumen de datos que se recopilara de los ficheros generados por el sistema Garmin del *smartwatch*. Además estos datos solo serán consultados en la aplicación para mostrarse al usuario o hacer el cálculo de las métricas, nunca modificados.

Por otro lado esta decisión, además de aumentar la escalabilidad de lectura de la aplicación, también mejora en cuanto a lo que a nivel de análisis de las actividades del deportista respecta. Con las constantes actualmente almacenadas de cada segundo de cada test ampliar la aplicación web para analizar estos datos o añadir nuevas métricas será mucho más fácil, ya que en lugar de almacenar un resultado final estamos persistiendo toda la información intermedia.

6.4.2.1 RESUMEN:

- Persistencia relacionada con la información de los usuarios
 - o Sigue la idea inicial, base de datos relacional SQL Server
 - o Características suficientes para almacenar esta información
 - o Mayor seguridad en base de datos SQL
 - o Mejor rendimiento con funciones de agregación con futuras entidades nuevas
- Persistencia relacionada con las métricas:
 - o Cambios importantes, se pase de una idea inicial de manejar pocos valores a una gran cantidad
 - o Base de datos no SQL orientada a objetos MongoDB
 - o Gran flexibilidad y rendimiento en lecturas
 - o Mucho potencial de expandir el análisis con la información que se almacena

6.4.3 IDE DE DESARROLLO WEB

Desde un primer momento fue establecido que la aplicación web se desarrollaría con Microsoft .NET Framework utilizando Orchard Project, el cual básicamente consiste en un proyecto ya creado con diversos módulos y paneles de control para agilizar el desarrollo web. Finalmente la idea de utilizar Orchard Project fue descartada debido que

se deseaba tener un control muy preciso sobre la página web a desarrollar y sus componentes. Este tipo de entornos suelen añadir bastante código extra y características que pueden ser útiles para proyectos más superficiales o en los que no se necesita generar un producto tan específico, como puede ser la creación de un blog de noticias o una página personal.

Por lo tanto el entorno web se ha desarrollado desde cero utilizando la aplicación Visual Studio Express Web 2013. Existen otros IDEs que se hubieran podido elegir para este fin pero Visual Estudio al igual que el framework .NET pertenecen a Microsoft y por lo tanto es el IDE más popular para desarrollar con .NET, además cuenta con versión gratuita.

Dentro de Visual Studio Express existía la posibilidad de realizar el proyecto utilizando los lenguajes de programación Visual Studio o C#. Se ha elegido C# primeramente por la familiaridad con el lenguaje. El proyecto está realizado siguiendo el patrón Modelo-Vista-Controlador ya que permite reutilizar código y separar conceptos lo que facilita la tarea de desarrollo y mantenimiento del sitio web. [19]

6.5 IDENTIFICACIÓN DE LEYES Y REGULACIONES

El sistema desarrollado realiza una monitorización de las actividades de los usuarios, generación de ficheros con esta información e importación de estos en la plataforma web para persistirlos en la base de datos para su posterior consulta y análisis. Por lo tanto al enmarcar la aplicación a un nivel europeo se debe asegurar que todo el sistema cumple la Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal (LOPD), cuya finalidad es regular el tratamiento de los datos y ficheros personales, independientemente del medio en el que son utilizados, los derechos de los ciudadanos sobre ellos y las obligaciones de quienes los crean o tratan. [20] [21]

A continuación se detallan las acciones a realizar para asegurar que el sistema cumple con la LOPD [22]:

- **Deber de información:** el usuario será informado de modo expreso, preciso e inequívoco de los datos que serán recogidos en cada momento de la ejecución de la aplicación en el *smartwatch*, de los ficheros generados a partir de la monitorización, de sus persistencia en la base de datos a través de la aplicación web y su análisis. También será informado de la finalidad con la que es monitorizada su actividad. Dicha información estará disponible de forma explícita a través de la tienda online de aplicaciones de Garmin y al visitar la página web desarrollada.
- **Consentimiento de los afectados:** además de estar informado de trato de sus datos los usuarios deben estar completamente de acuerdo, por ello se establecerán cláusulas que impliquen que al instalar la aplicación en el dispositivo el usuario está aceptando la monitorización de su actividad. Además al registrarse en la aplicación web el usuario deberá volver a aceptar todas las condiciones de tratamiento de datos a partir de un formulario.
- **Datos especialmente protegidos y seguridad de los mismo:** se asegura que la información persistida en la base de datos de la aplicación web será solo accedida por la persona que cuente con el usuario y contraseña de la cuenta registrada, con la que dicha información fue ingresada en el sistema o el administrador de la aplicación. Por lo que respecta a la utilización del *smartwatch*, se establecerá que es del propietario de este la responsabilidad de la utilización del dispositivo, por el mismo u otras personas. La contraseña de las cuentas de usuario serán encriptadas en la base de datos para asegurar protección.

- **Deber de secreto:** el administrador/res de la aplicación estará/n comprometido/s a guardar el secreto de toda la información del usuario monitorizada y almacenada en el sistema. El usuario será informado de esto también y se deberá contar con su consentimiento.
- **Comunicación de datos, acceso a los datos por cuenta de terceros y transferencia internacional de datos:** en el caso de implantar la aplicación en el mercado y contar con anunciantes y/o corredores de datos en el portal web. Se añadirá la información respecto al tratado de los datos que estas terceras personas realizarán y exactamente qué información y como será compartida con estos. En todo momento el usuario deberá ofrecer su consentimiento y se asegurara la confidencialidad de su identidad, en caso contrario también será informado y deberá ofrecer su consentimiento.

Planificación temporal

7.1 INTRODUCCIÓN

El proyecto se enmarca en un periodo de aproximadamente cuatro meses, comprendido desde finales de febrero del 2016 a junio del mismo año. Destacar que la realización del proyecto será llevada a cabo por una sola persona, y, por lo tanto, no hay opción a efectuar tareas en paralelo.

7.1.1 FITA INICIAL

El proyecto en total tiene una primera planificación de 488 horas distribuidas en 4 meses con una carga de trabajo de aproximadamente 4 horas diarias, contando fines de semana.

7.1.1.1 FASES

A continuación, se presentan las diferentes fases que acotaran el proyecto. Dichas fases se presentan en orden de dependencia, aunque cambios inesperados o redefiniciones en el proyecto, pueden llevar realizar modificaciones en fases preliminares y posteriores, a la que nos encontremos en el proyecto. Durante las fases el director/ponente tendrá a su disposición la documentación generada hasta el momento y también los prototipos desarrollados del sistema hasta el momento.

7.1.1.1.1 FASE INICIAL

La fase inicial del proyecto corresponde al período de la asignatura de Gestión de Proyectos, cuya finalidad es generar un informe previo de apoyo que contenga sus objetivos, alcance planificación etc. GEP tiene una duración de 4 semanas dividida en 6 entregas, las dos últimas corresponden a las entregas de toda la documentación generada anteriormente y a su presentación oral.

A la vez que se generan la documentación para la asignatura de Gestión de Proyectos la comunicación con el director/ponente será vital para asegurar que tanto él como el estudiante tienen la misma visión del sistema que se desea crear o al menos llegar a un punto común. Una vez acabada GEP, la documentación generada hasta el momento será complementada paralelamente al resto de fases del proyecto para generar la memoria final del mismo.

7.1.1.1.2 FASE DE ESPECIFICACIÓN

La primera tarea será definir los requisitos que deberá cumplir el sistema. Conjuntamente se realizará una especificación detallada de la aplicación pantallas, flujo de acciones, distribución de controles ..etc. En esta fase también será necesario estudiar las tecnologías utilizadas para saber de qué funcionalidades se dispone y cuáles pueden ser utilizadas para las necesidades del proyecto. Una vez definidas todas las funcionalidades y requisitos del sistema, se generará el *product backlog* de acuerdo con la metodología *scrum* utilizada para el desarrollo del proyecto.

7.1.1.1.3 FASE DE DESARROLLO

Una vez detallada la aplicación y con el *product backlog*, la fase de desarrollo estará dividida en diferentes *sprints* o iteraciones que estarán formados por un subconjunto de las tareas en el *product backlog*. Cada *sprint* contará con un periodo para establecer la prioridad de las tareas dentro del *sprint*, determinar posibles dependencias o fijar los recursos necesarios para su producción

Al final de cada iteración el producto final se verá incrementado con las funcionalidades implementadas. También se realizarán pruebas para asegurar que el producto final generado hasta el momento funciona correctamente. En el caso que la iteración afecte a funcionalidades ya implementadas del sistema el conjunto de pruebas será más exhaustivo para comprobar que la unión se realizado sin alternaciones indeseadas en el producto. Las pruebas se realizarán con el emulador que facilita Garmin para los desarrolladores y posteriormente con el dispositivo Garmin Epix. El director/ponente tendrá a su disposición tanto el producto estable con las iteraciones acabadas como una versión parcial con la iteración en curso.

7.1.1.1.3.1 ITERACIÓN MÉTRICAS

Los *sprints* dedicados la implementación de métricas serán los que dotaran al sistema de mayor funcionalidades, debido al patrón base especificado en el apartado 6.3 Metodología y rigor para los *sprints* de este tipo, donde una métrica es implementada en todo el sistema desde la aplicación del *smartwatch* pasando por su sincronización con el exterior y su posterior recuperación en el portal web.

Tanto el lenguaje de programación para los dispositivos Garmin como el método con el que se exportarán los datos al portal web son desconocidos. Por esto se prevé una fuerte inversión de tiempo en la primera iteración de este tipo, hasta familiarizarse con estas características de la tecnología utilizada.

7.1.1.1.3.2 ITERACIÓN GENERALES

El resto de iteraciones se encargaran, por un lado a realizar un análisis más profundo de los datos recogidos con el objetivo de mejorar la comprensión de la evolución de las métricas. También se enmarca en este tipo las iteraciones destinadas a mejorar la usabilidad y la apariencia del sistema.

Todas las iteraciones de este tipo no son de máxima prioridad y sus tareas tampoco entrarían dentro de los requisitos mínimos del sistema por lo que en caso de estancamientos que lleven a falta de tiempo en el proyecto, estas iteración podría ser reducidas o suprimida para dar prioridad a otras fases.

7.1.1.1.3.3 FASE FINAL

La fase final de proyecto se dedicará primeramente a realizar una prueba final completa de todo el sistema para acabar definitivamente las iteraciones de desarrollo. El tiempo restante se dedicará a completar la memoria del proyecto y a preparar la presentación final del mismo.

7.1.1.1.4 ESTIMACIÓN EN HORAS

En esta sección se encuentra cada una de las fases anteriormente detalladas, desglosadas en sus diferentes tareas y la carga horaria que se ha determinado para ellas. Del sumatorio del tiempo estimado para cada tarea y la duración del proyecto se desprende la carga de trabajo diaria de 4 horas = $488/(4*30)$.

Tareas	Horas
Fase Inicial (GEP)	160
Definición de la contextualización y alcance	40
Planificación temporal	24
Gestión Económica y sostenibilidad	28
Presentación Preliminar	20
Plec de condiciones	24
Presentación y documentación final	24
Fase de especificación, análisis y diseño	28
Análisis de requisitos	8
Especificación casos de uso	8
Modelo conceptual	3
Arquitectura y diseño	9
Iteración 1	164
Obtener cálculo de métrica	28
Guardar historial métrica	12
Consultar historial métrica	20
Exportar métrica	60
Recuperar métrica	44
Iteración 2	48
Obtener cálculo de métrica	16
Guardar historial métrica	4
Consultar historial métrica	4
Exportar métrica	12
Recuperar métrica	12
Iteración análisis	40
Investigar estadísticas	12
Implementar estadísticas	28
Fase Final	48
Memoria	20
Presentación	28
Total	488

7.1.1.1.5 DESVIACIONES Y PLAN DE ACCIÓN

Debido a que la mayoría de tecnologías empleadas son desconocidas hasta el momento y los retrasos propios del desarrollo de software, en caso de desviaciones importante en la planificación definida se aumentara el número de horas dedicadas al proyecto. No obstante, si a pesar de esto no se consigue superar las desviaciones presentadas, se sacrificarán horas destinadas a la iteración de análisis y en caso muy extremo y como última opción se tomarán horas de la iteración 2. Se espera que las desviaciones se presenten durante las iteraciones 1 y 2, mayoritariamente durante la primera.

Además, la planificación se ha realizado de forma que la finalización del proyecto sea 5 días antes de la presentación final por lo que se cuenta con estos 5 días de reserva en caso de imprevistos.

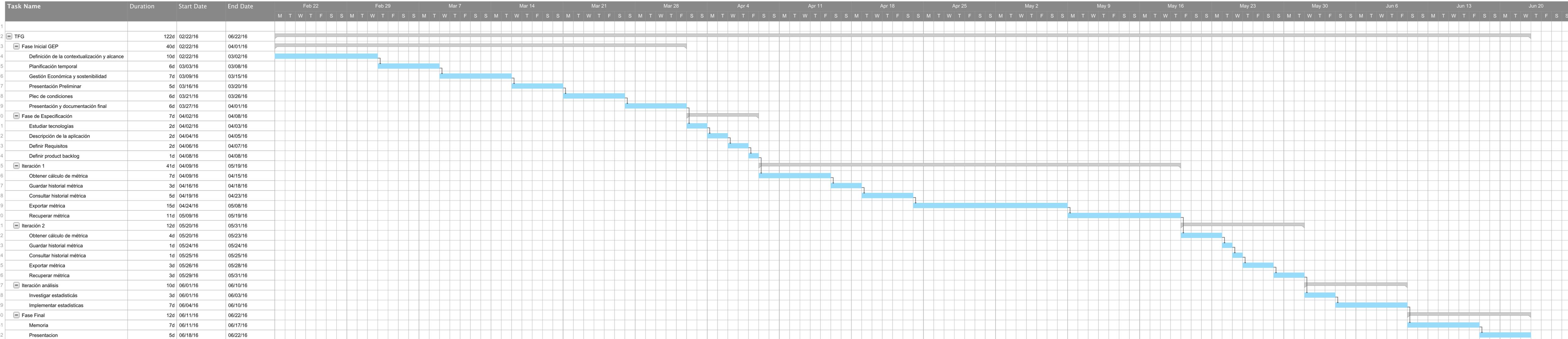
7.1.1.2 RECURSOS

En este apartado se describen los diferentes recursos necesarios para la realización del proyecto y sus funciones.

- Equipo portátil Asus k550cc-xx1092h con el siguiente conjunto de aplicaciones:
 - Sistema operativo Windows 8,1 Pro
 - Suite Microsoft Office 2013
 - Adobe Reader XI Pro
 - Adobe Photoshop CS6
 - Eclipse Mars 4.4.2 con
 - Garmin IQ SDK
 - Google Drive Desktop
 - Cuenta Google Gmail
 - Navegador Google Chrome 50.0
 - Microsoft Visual Studio Web 2013
 - MongoDB 3.2.6
 - Python 3.4.3
- Dispositivo *smartwatch* Garmin Epix

7.1.1.3 DIAGRAMA DE GANTT

En la página siguiente se encuentra el diagrama de Gantt de la planificación del proyecto antes de la *fit a de seguiment*, realizado con la aplicación online Smartsheet [23].



7.1.2 FITA DE SEGUIMIENTO

Respecto a la planificación inicial del proyecto, han habido cambios importantes hasta llegar a la *fita de seguiment*. Primeramente el número total de días del proyecto ha pasado de 122 a 115; manteniendo la idea de acabar el desarrollo y la memoria, una semana antes (13/06/2016) de la entrega final (20/06/2016) y por lo tanto contar con 5 días más en caso de imprevistos. La fase inicial coincidente con la asignatura de GEP no ha sufrido cambios. La fase dedicada a la especificación ha tenido cambios pero sutiles ya que se ha terminado antes de lo planificado inicialmente.

Las iteraciones previstas inicialmente han sufrido cambios primeramente por la no disponibilidad del dispositivo *smartwatch* y segundo por una planificación incorrecta del tiempo necesario a desarrollar el portal web:

- Primeramente de los *sprints* dedicados al desarrollo de métricas se han eliminado las tareas destinadas a exportar e importar los datos en la página web, ya que no se ha dispuesto del dispositivo en ningún momento como se planteó inicialmente. Debido a este hecho y que no hay una fecha exacta de cuándo podría estar disponible, se ha dedicado un *sprint* final para realizar las pruebas antes de realizar la memoria y la defensa del proyecto en caso de que fuera posible.
- Se ha pasado de 2 iteraciones dedicadas al cálculo de métricas a 3 ya que se deseaba dotar a la aplicación de 3 tipos diferentes: una relacionado con la velocidad máxima alcanzable al llegar al VO₂Max del deportista (test VO₂Max Speed) y dos para estimar el VO₂Max, pero requiriendo un esfuerzo más suave (One Mile Walk Test) y otro requiriendo un esfuerzo superior (One Half Mile Run Test).
- Se han añadido 3 iteraciones destinadas al desarrollo del portal web que en un momento inicial no se habían explicitado en el diagrama Gantt, ya que se consideraban implícitas en las tareas de exportación e importación de las métricas en la iteración 1. Además inicialmente se pensó que el desarrollo del portal web sería más fácil pero, a medida que se han estudiado las tecnologías o cambios importantes en cómo se persistiría la información por ejemplo, han llevado a dedicar sprints exclusivamente a estas tareas.

Estos cambios quedan reflejados en las horas totales del proyecto, inicialmente estimadas en 488 que se han visto incrementadas a 575, casi 100 horas más. Respecto a la planificación inicial esto implica un aumento de 1 hora más diaria de trabajo contando fines de semana, 5 horas en total. Cabe remarcar que la distribución horario por días no ha sido uniforme durante estos, debido a otras responsabilidades pero el sumatorio total por tarea y por iteración si se ha complicado hasta el momento sin grandes desviaciones.

7.1.2.1 SITUACIÓN ACTUAL DEL PROYECTO

Actualmente el proyecto se encuentra a punto de iniciar la penúltima iteración y está al día con la planificación presentada en el diagrama de Gant que se encuentra al final de esta sección, la aplicación del *smartwatch* funciona sin problemas y correctamente en el emulador hasta el momento. Se ha logrado la decodificación e importación de ficheros .FIT en la plataforma web, su persistencia y recuperación. Las próximas iteraciones serán dedicadas a re-estimar las métricas en el portal web, implementar un seguimiento gráfico de la información importada y mejorar la usabilidad del portal web. En caso de disponer del dispositivo el último sprint se dedicará a realizar pruebas con el mismo.

7.1.2.2 ESTIMACIÓN EN HORAS

En esta sección se encuentra cada una de las fases anteriormente detalladas, desglosadas en sus diferentes tareas y la carga horaria que se ha determinado para ellas

Tareas	Horas
Fase Inicial (GEP)	200
Definición de la contextualización y alcance	50
Planificación temporal	30
Gestión Económica y sostenibilidad	35
Presentación Preliminar	25
Plec de condiciones	30
Presentación y documentación final	30
Fase de especificación	25
Estudiar tecnologías	10
Descripción de la aplicación	5
Definir Requisitos	5
Definir product backlog	5
Iteración 1	60
Diseño pantalla	15
Estimar métrica	25
Implementar temporizador	15
Ajustar flujo de pantallas	5
Iteración 2	50
Diseño pantalla	5
Estimar métrica	25
Implementar temporizador	10
Ajustar flujo de pantallas	10
Iteración 3	40
Diseño pantalla	5
Estimar métrica	20
Implementar temporizador	5
Ajustar flujo de pantallas	10
Iteración 4	45
Web inicial y autenticación	20
Importar ficheros .FIT	20
Diseñar esquema DB	5
Iteración 5	40
Persistencia MongoDB	20
Consultar y recuperar Activities	20
Iteración 6	60
Estimar métricas	20
Seguimiento gráfico	20
Optimizar web	20
Iteración 7	20
Pruebas finales	20
Fase Final	35
Memoria	25
Presentación	10
Total	575

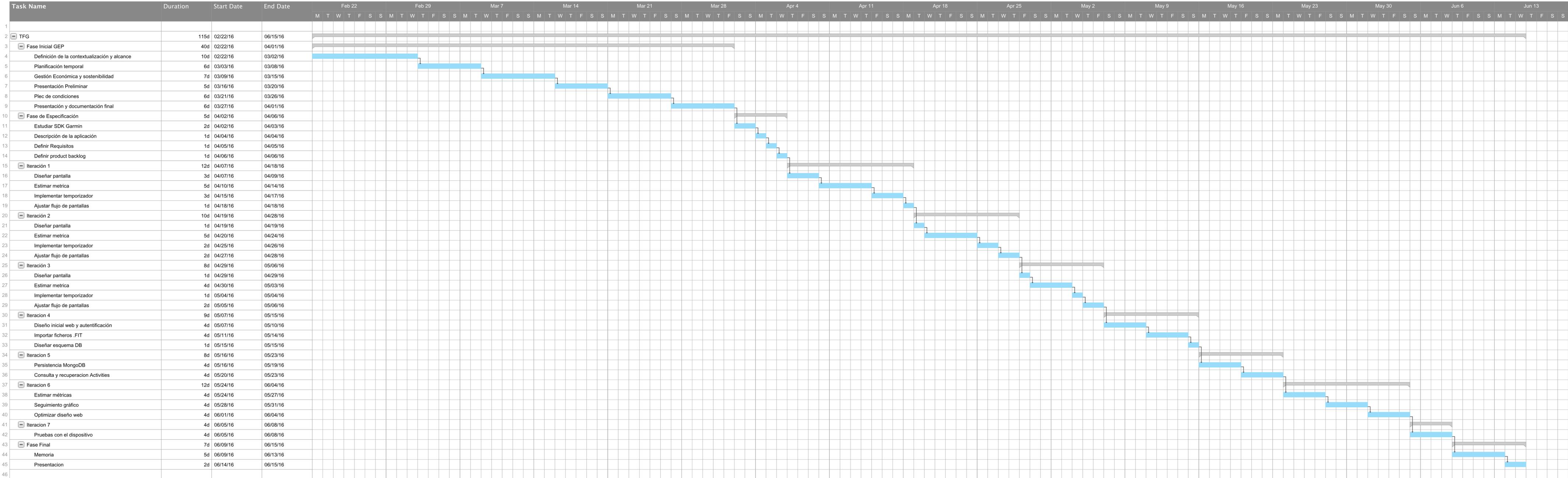
7.1.2.3 RESUMEN

Los cambios realizados hasta el momento no ha limita ni reducido los objetivos del proyecto, en todo caso han sido ampliados como se ha visto con el portal web. Se han realizado ajustes para adaptar la planificación a cambios inesperados y se sigue contemplando la posibilidad de realizar pruebas con el dispositivo real. El número de horas total dedicadas al proyecto se ha visto incremento en casi 100 horas más debido a que durante la primera planificación no se tenía una visión completa de la envergadura del proyecto. Los efectos que este aumento de tiempo implica en la gestión económica se encuentra en el apartado 8.1.

En la página siguiente se encuentra el diagrama de Gantt de la planificación del proyecto, con los cambios en la planificación expuestos, realizado con la aplicación online Smartsheet [23].

7.1.3 FITA FINAL

La planificación temporal final ha sido prácticamente la misma que la marcada en *la fita de suiment*. Tanto la duración de los *sprints* como las dedicaciones se han seguido sin desviaciones importantes. Debido a que ha sido imposible disponer en del dispositivo Epix, para realizar las pruebas en un smartwatch real, la iteración 7 que estaba dedicada principalmente a estar labor, se ha centrado en probar la aplicación utilizando el emulador y a probar la aplicación web. De esta forma la estimación de horas y tareas permanece como fue definida en *la fita de seguiment*. Destacar que retrasos en el desarrollo y la documentación han llevado a activar el plan de contingencia marcada, alargando así el proyecto 5 días más respecto a la fecha de finalización ideal.



Gestión económica

8.1 INTRODUCCIÓN

Presentado el planteamiento del problema conjuntamente con la solución propuesta y su planificación temporal, el siguiente paso es analizar la dimensión económica del proyecto, demostrando su viabilidad financiera.

Se ha mencionado anteriormente que este proyecto se trata de un TFG sin inversión económica real y desarrollado por una única persona. Pese a esto el análisis de los recursos necesarios para su realización y la estimación total del coste se hará como si se tratara de un proyecto competitivo real, con el equipo de personas ideal para su desarrollo y por lo tanto con el debido efecto en el coste final, las horas de los recursos humanos a precio de mercado.

8.1.1 IDENTIFICACIÓN COSTES

En los siguientes subapartados se identificarán los tipos de costes generales del proyecto y posteriormente la estimación de cada uno de ellos para las diferentes fases que ha tenido el proyecto.

8.1.1.1 COSTES DIRECTOS

Los costes directos comprenden todos los gastos generados por las actividades del proyecto y que sin este no existirían, en este caso se trata de los recursos humanos, es decir, los roles que forman el equipo necesario para que el desarrollo propuesto sea realizado con éxito. A continuación, la remuneración por hora correspondiente a los roles identificados, se ha utilizado un estudio de remuneración del año 2016 [24]. El precio por hora de cada rol se calculado de la siguiente forma: $\text{sueldoMínimoRol}/(\text{DíaLaborables2016} \cdot 8)$, estableciendo que el año 2016 tiene 250 días laborales y jornadas diarias de 8 horas.

Rol	Remuneración (€/h)	Sueldo Mínimo €
Jefe de proyecto	17,5	35000
Analista	12	24000
Diseñador	9	18000
Programador	9	18000
Resp. de pruebas	11	22000

8.1.1.2 COSTES INDIRECTOS

Conexión Internet: Durante el desarrollo del proyecto se utilizará en el equipo una conexión a internet de fibra óptica a una velocidad de 50Mbps a 60 euros mensuales. Dado que dicha conexión no es de uso exclusivo para el proyecto se considerará que corre a cargo de este un 15% del precio de la conexión.

Transporte: dos abonos trimestrales de 85€ cada uno del transporte metropolitano de Barcelona, se dedicará una proporción de 1% respecto al precio de los abonos. Los desplazamientos relacionados con el proyecto que necesitarán de la utilización de los abonos serán muy escasos y el segundo de ellos solo será utilizado durante su primer mes de vida útil para el proyecto.

Amortización del Hardware: el equipo hardware necesario para el proyecto, primeramente, se trata de un equipo portátil con una vida útil de 5 años, para estimar la amortización de este supondremos un uso del 60% de sus recursos dedicados al proyecto durante 4 meses por lo tanto procederemos de la siguiente forma:

$$\text{precioDelEquipo}/(5*365)=\text{amortizaciónDiaria}$$

$$\text{amortizacionDelProyecto}=\text{amortizaciónDiaria} * 4*30.$$

Amortización Software: todo el software necesario para el proyecto es libre o se cuenta con una licencia gratuita al ser utilizado por estudiantes de la UPC, se estimará un uso generalizado del 90% entre todas las aplicaciones dedicadas al proyecto de sus ejecuciones durante los 4 meses.

Impresiones en papel: las impresiones en papel serán casi inexistentes ya que todas las entregas y revisiones serán realizadas en formato digital no obstante se considerará una estimación de 50 impresiones a 20 céntimos de euro cada una.

8.1.1.3 PLAN DE CONTINENCIA

Los costes de los planes de contingencia comprenden el aumento en el presupuesto del proyecto, en el caso de que por retrasos en sus realización se deba de alargar su periodo de duración. De esta manera es utilizando el margen de días que hay entre la fecha de finalización ideal del proyecto y la de entrega final del mismo.

8.1.2 FITA INICIAL

8.1.2.1 COSTES DIRECTOS

A partir de las tareas en el diagrama de Gantt para la fase inicial, realizado en la planificación temporal del apartado 7.1 y el coste de los roles identificados en este mismo capítulo, se obtiene la dedicación en horas de cada rol en las diferentes tareas del proyecto y el coste generado.

Tareas	Horas	Recurso	Coste
Fase Inicial (GEP)	160		2800
Definición de la contextualización y alcance	40	Jefe de proyecto	700
Planificación temporal	24	Jefe de proyecto	420
Gestión Económica y sostenibilidad	28	Jefe de proyecto	490
Presentación Preliminar	20	Jefe de proyecto	350
Plec de condiciones	24	Jefe de proyecto	420
Presentación y documentación final	24	Jefe de proyecto	420
Fase de	28		363,9
Análisis de requisitos	8	Analista 100%	96
Especificación casos de uso	8	Analista 100%	96
Modelo conceptual	3	Analista 100%	72
Arquitectura y diseño	9	Diseñador 30%	24,3
		Analista 70%	75,6
Iteración 1	164		1508,8
Obtener cálculo de métrica	28	Programador 90%	226,8
		Resp. de pruebas 10%	30,8
Guardar historial métrica	12	Programador 90%	97,2
		Resp. de pruebas 10%	13,2
Consultar historial métrica	20	Programador 90%	162
		Resp. de pruebas 10%	22
Exportar métrica	60	Programador 90%	486
		Resp. de pruebas 10%	66
Recuperar métrica	44	Programador 90%	356,4
		Resp. de pruebas 10%	48,4
Iteración 2	48		441,6
Obtener cálculo de métrica	16	Programador 90%	129,6
		Resp. de pruebas 10%	17,6
Guardar historial métrica	4	Programador 90%	32,4
		Resp. de pruebas 10%	4,4
Consultar historial métrica	4	Programador 90%	32,4
		Resp. de pruebas 10%	4,4
Exportar métrica	12	Programador 90%	97,2
		Resp. de pruebas 10%	13,2
Recuperar métrica	12	Programador 90%	97,2
		Resp. de pruebas 10%	13,2
Iteración análisis	40		401,6
Investigar estadísticas	12	Analista	144
Implementar estadísticas	28	Programador 90%	226,8
		Resp. de pruebas 10%	30,8
Fase Final	48		840
Memoria	20	Jefe de proyecto	350
Presentación	28	Jefe de proyecto	490
Total	488		6355,9

Horas = Nº Días Tarea * Horas de trabajo por día

Horas de trabajo por día = 4, ver apartado 7.1.1.1.4

Coste = horas de la tarea * (% de implicación del rol/100) *precio hora del rol

De la tabla anterior se desprende que el precio promedio por hora del proyecto es de 12,91€ aproximadamente, en cuanto a recursos humanos. Las tareas de gestión y documentación son llevadas a cabo por el jefe de proyecto, las decisiones de diseño y funcionalidades por el diseñador y el analista respectivamente, finalmente las tareas de programación son realizadas por el desarrollador y las pruebas por el responsable de pruebas.

8.1.2.2 COSTES INDIRECTOS

Recurso	Valor Unitario €	% proyecto	Valor Proyecto
Conexión a internet	180	15	27
Transporte	170	1	1,7
Equipo Portátil	700/5*365	4*30 día	46,02739726
Garmin Epix	460/5*365	4*30 días	30,24657534
Amortización Software	0	90	0
Impr. Papel	10	100	10
Total			114,9739726

8.1.2.3 PLAN DE CONTINGENCIA

Anteriormente, durante la planificación temporal apartado 7.1, se ha estimado acabar el proyecto 5 días antes de la presentación final, por lo tanto en el peor de los casos en que el desarrollo del proyecto continuara durante este período de tiempo se aproximan los costes directos e indirectos siguientes:

$$\text{costesDirectosDelProyectoPorDia} = \text{TotalCostesDirectos} / 4 * 30$$

$$\text{costesIndirectosDelProyectoPorDia} = \text{TotalCostesIndirectos} / 4 * 30$$

Aumento en los costes:

Coste por día	Valor por día	5 días imprevistos
Directo	52,26583333	261,3291667
Indirectos	0,958116438	4,790582192
Total	53,22394977	266,1197489

8.1.2.4 PRESUPUESTO FINAL

A continuación, tenemos la estimación final de la gestión económica del proyecto, también se encuentra contabilizado el caso en que son necesarios los 5 días previos a la presentación final para continuar con el desarrollo.

Cotes	Valor	Total v1	Total v2
Directos	6271,9		
Indirectos	114,9739726	6652,993721	6386,873973
Extras	266,1197489		

En el valor total v1 se aprecia el peor de los casos que presenta el aumento extra en los costes estimados al alargar el desarrollo del sistema por 5 días, en la versión total v2 se presenta el caso en el que se cumple con la planificación temporal deseada sin contratiempos.

8.1.3 FITA DE SEGUIMENT

8.1.3.1 COSTES DIRECTOS

En el apartado 7.1.2 se han expuesto los cambios en la planificación inicial del proyecto, la reestructuración de tareas que se ha realizado y su justificación en el momento de la *fita de seguiment*. A partir de esto y el coste por hora de los roles identificados para cada tarea se obtiene la estimación de los costes directos del proyecto.

Tareas	Horas	Recurso	Coste
Fase Inicial (GEP)	200		3500
Definición de la contextualización y alcance	50	Jefe de proyecto	875
Planificación temporal	30	Jefe de proyecto	525
Gestión Económica y sostenibilidad	35	Jefe de proyecto	612,5
Presentación Preliminar	25	Jefe de proyecto	437,5
Plec de condiciones	30	Jefe de proyecto	525
Presentación y documentación final	30	Jefe de proyecto	525
Fase de	32		429,9
Análisis de requisitos	10	Analista 100%	120
Especificación casos de uso	10	Analista 100%	120
Modelo conceptual	3	Analista 100%	90
Arquitectura y diseño	9	Diseñador 30%	24,3
		Analista 70%	75,6
Iteración 1	60		552
Diseño pantalla	15	Programador 90%	121,5
		Resp. de pruebas 10%	16,5
Estimar métrica	25	Programador 90%	202,5
		Resp. de pruebas 10%	27,5
Implementar temporizador	15	Programador 90%	121,5
		Resp. de pruebas 10%	16,5
Ajustar flujo de pantallas	5	Programador 90%	40,5
		Resp. de pruebas 10%	5,5
Iteración 2	50		460
Diseño pantalla	5	Programador 90%	40,5
		Resp. de pruebas 10%	5,5
Estimar métrica	25	Programador 90%	202,5
		Resp. de pruebas 10%	27,5
Implementar temporizador	10	Programador 90%	81
		Resp. de pruebas 10%	11
Ajustar flujo de pantallas	10	Programador 90%	81
		Resp. de pruebas 10%	11
Iteración 3	40		368
Diseño pantalla	5	Programador 90%	40,5
		Resp. de pruebas 10%	5,5
Estimar métrica	20	Programador 90%	162
		Resp. de pruebas 10%	22
Implementar temporizador	5	Programador 90%	40,5
		Resp. de pruebas 10%	5,5
Ajustar flujo de pantallas	10	Programador 90%	81
		Resp. de pruebas 10%	11
Iteración 4	45		414
Web inicial y autenticación	20	Programador 90%	162
		Resp. de pruebas 10%	22
Importar ficheros .FIT	20	Programador 90%	162
		Resp. de pruebas 10%	22

	Diseñar esquema DB	5	Programador 90%	40,5
			Resp. de pruebas 10%	5,5
Iteración 5		40		368
	Persistencia MongoDB	20	Programador 90%	162
			Resp. de pruebas 10%	22
	Consultar y recuperar Activities	20	Programador 90%	162
			Resp. de pruebas 10%	22
Iteración 6		60		552
	Estimar métricas	20	Programador 90%	162
			Resp. de pruebas 10%	22
	Seguimiento gráfico	20	Programador 90%	162
			Resp. de pruebas 10%	22
	Optimizar web	20	Programador 90%	162
			Resp. de pruebas 10%	22
Iteración 7		20		216
	Pruebas finales	20	Resp. de pruebas 90%	198
			Programador 10%	18
Fase Final		35		612,5
	Memoria	25	Jefe de proyecto	437,5
	Presentación	10	Jefe de proyecto	175
Total		582		7472,4

Horas = N° Días Tarea * Horas de trabajo por día

Horas de trabajo por día = 5, ver apartado 7.1.2.2

Coste = horas de la tarea * (% de implicación del rol/100) * precio hora del rol

8.1.3.2 COSTES INDIRECTOS

Los costes indirectos del proyecto permanecen igual ya que la amortización se hizo sobre 4 meses y esto no ha cambiado.

8.1.3.3 PLAN DE CONTINGENCIA

Los costes en caso de necesitarse los 5 días extras de la planificación.

$$\text{costesDirectosDelProyectoPorDia} = \text{TotalCostesDirectos} / 4 * 30$$

$$\text{costesIndirectosDelProyectoPorDia} = \text{TotalCostesIndirectos} / 4 * 30$$

Aumento en los costes:

Coste por día	Valor por día	5 días imprevistos
Directo	62,27	311,35
Indirectos	0,958116438	4,790582192
Total	63,22811644	316,1405822

8.1.3.4 PRESUPUESTO FINAL

A continuación, tenemos la estimación final de la gestión económica del proyecto, también se encuentra contabilizado el caso en que son necesarios los 5 días previos a la presentación final para continuar con el desarrollo.

Cotes	Valor	Total v1	Total v2
Directos	7472,4	7903,514555	7587,373973
Indirectos	114,9739726		
Extras	316,1405822		

En el valor total v1 se aprecia el peor de los casos que presenta el aumento extra en los costes estimados al alargar el desarrollo del sistema por 5 días, en la versión total v2 se presenta el caso en el que se cumple con la planificación establecida en el momento de la *fita de seguiment* sin contratiempos.

8.1.4 FITA FINAL

Como se ha introducido anteriormente en el apartado , se ha mantenido la planificación establecida en la *fita de seguiment* sin cambios significativos. Por lo tanto la planificación económica también sigue siendo la misma que la establecida para la *fita de seguiment*. Destacar que retrasos en el desarrollo y la documentación han llevado a activar el plan de contingencia marcada, alargando así el proyecto 5 días más respecto a la fecha de finalización ideal.

Sostenibilidad y compromiso social

En esta sección se estudia la viabilidad económica del proyecto, así como el impacto ambiental y social que supone su desarrollo.

9.1 ECONÓMICA

En el estudio del arte del proyecto se ha detectado un importante sector de mercado que cubrirá el proyecto por lo que se espera que tenga usuarios deseosos de obtener la aplicación final. Además, actualmente el incremento por la práctica de las actividades deportivas y por lo tanto el interés en el rendimiento aeróbico es constante. Comparado con otros proyectos de mercado podemos afirmar que la estimación de los costes es bastante competitiva ya que como hemos visto se plantea un desarrollo con los sueldos mínimos por perfil necesario. No obstante, al tratarse de un TFG el coste humano real será cero y por lo tanto solo tendríamos que tener en cuenta los costes indirectos relacionados.

Respecto a la planificación temporal el proyecto podría ser realizado en un periodo de tiempo menor pero solo sería posible aumentando el número de personas en el proyecto y por lo tanto los costes. El proyecto tiene identidad propia y no se espera cooperación con otros proyectos en el futuro próximo. Económicamente el proyecto merece un 6 ya que, aunque el sector de mercado que puede abarcar es grande se necesita atraer la atención de los usuarios a este y dar a conocer el producto, esto seguramente implicaría más gastos en promocionarlo.

9.2 SOCIAL

La sociedad actual en donde cada vez es más importante tener hábitos sanos y realizar actividades deportivas hacen que el proyecto tenga un gran valor social generalizado. La aplicación puede ser utilizada en diferentes ámbitos desde deportistas principiantes, personas que se están recuperando de algún problema de salud hasta deportistas de elite. También es de utilidad para entrenadores o personas que se encargan de realizar un seguimiento deportivo. El sistema asegura métricas que informan de su capacidad aeróbica a los usuarios y de la evolución que están teniendo. Siempre accesible desde su dispositivo *smartwatch* o portal web. En este ámbito el proyecto se valora con un 8.

9.3 AMBIENTAL

El impacto medioambiental del proyecto es prácticamente nulo. Los desplazamientos llevados a cabo serán realizados en transporte público por lo que no se genera contaminación debido al uso de vehículos privados. El consume eléctrico de los equipos es mínimo ya que estamos hablando de un dispositivo *smartwatch* y un único equipo portátil. La generación de papel se ha estimado una cantidad de 50 impresiones de las que muy

posiblemente no se use ni siquiera la tercera parte ya que todas las entregas y presentaciones relacionadas con el proyecto son en formato digital, en algún caso hojas de apuntes durante alguna reunión con el director/ponente podrían generarse.

Se podría considerar hasta una mejora en la sostenibilidad y recursos utilizados en el seguimiento deportivo, debido a que utilizando el sistema, no hace falta la utilización de papel para tomar notas de medidas o cálculos de métricas ya que todo es captado y almacenado por el sistema en formato digital. Por todas estas razones se valora el proyecto con un 9, la contaminación generada es casi inexistente.

Justificación de las competencias técnicas

Este capítulo se entra en la descripción del proyecto a realizar, las competencias técnicas seleccionadas por el alumno, la justificación de estas y la justificación de que el proyecto descrito se adecua a las características de la especialidad escogida.

10.1 DESCRIPCIÓN DE LOS OBJETIVOS

A continuación se presentan los objetivos principales del proyecto, se pretende desarrollar una aplicación que apoye a personas que practican deporte. El valores más importantes del sistema serán las métricas que se obtenga a partir del dispositivo *smartwatch* y la posterior exportación que se realice de ellas en un formato estándar que pueda ser interpretado por otras aplicaciones. En función de las métricas y su evolución el deportista podrá relacionar el efecto que el entrenamiento está teniendo sobre su rendimiento y modificarlo en consecuencia.

- Aplicación multiplataforma, divida en dos partes: aplicación *smartwatch* y aplicación web.
- La aplicación del *smartwatch* debe funcionar como mínimo en el dispositivo Garmin Epix.
- La aplicación del *smartwatch* debe de utilizar las constantes reales del usuario ritmo cardiaco, velocidad, tiempo de ejercicio ..etc.
- La aplicación del *smartwatch* debe acceder a los sensores del dispositivo.
- La aplicación del *smartwatch* debe ofrecer métodos para calcular el rendimiento aeróbico del deportista.
- La aplicación del *smartwatch* debe poder gestionar el historial de las métricas del usuario.
- La aplicación del *smartwatch* debe poder exportar el historial de métricas almacenadas en el dispositivo.
- La aplicación del *smartwatch* debe poder exportar el historial de métricas almacenadas en un formato comprensible para las personas.
- La aplicación del *smartwatch* debe permitir borrar los historiales de las métricas almacenadas.
- La aplicación web debe permitir importar la información de las métricas exportadas des del dispositivo *smartwatch*.
- La aplicación web debe ser accesible desde cualquier dispositivo con conexión a Internet y un navegador.
- La aplicación web debe funcionar en la última versión del navegador Google Chrome como mínimo.
- El usuario debe identificarse en la aplicación web antes de acceder a los datos de la cuenta, se trata de una aplicación con autenticación antes del inicio de sesión.
- La información importada por el usuario a la aplicación web debe ser persistente, por lo tanto se ha de implementar una base de datos para almacenar dichos datos.
- La aplicación web debe presentar la información importada en un formato amigable que ayude a interpretar sus variaciones y filtrarla.
- La aplicación web debe permitir gestionar los historiales del usuario.
- La aplicación web debe permitir gestionar el perfil del usuario.

De las acciones de gestionar se desprenden las acciones de crear, editar y borrar.

10.2 RESUMEN DEL ALCANCE

El desarrollo del sistema estará dividido en diferentes fases. La primera estará destinada a plantear el problema a resolver, el alcance del proyecto, la planificación temporal y una estimación de costes en el mercado real. En las siguientes fases esta documentación inicial irá creciendo para convertirse en la memoria final del proyecto. La siguiente fase enmarcará el desarrollo de la aplicación, se realizará utilizando una metodología ágil en concreto *scrum*. Por lo tanto una vez definido el *product backlog* del proyecto se establecerán los diferentes *sprints* con las funcionalidades que se implementarán en cada uno de ellos. Cada *sprint* cuenta con validación de las nuevas funcionalidades desarrolladas. Las pruebas se realizarán con el emulador del *smartwatch* que el SDK del fabricante ofrece y posteriormente con el dispositivo Garmin Epix.

Las principales restricciones del proyecto por un lado son las temporales ya que se trata de un TFG de una duración de 4 meses aproximadamente y por lo tanto la planificación debe realizarse en este marco temporal. Dicha planificación cuenta con métodos de contingencia en caso de desviaciones temporales que puedan poner en peligro el éxito del proyecto. La segunda restricción importante es la que respecta las nuevas tecnologías que se utilizarán ya que son desconocidas por el estudiante. Los dispositivos *smartwatch* Garmin utilizan un lenguaje de programación propio Monkey C, además se trata de un sistema bastante cerrado por lo que la información relacionada se limita a la ofrecida por el fabricante.

El proyecto abarca la realización de su memoria, métricas para medir el rendimiento aeróbico del usuario, exportarlas e importarlas en una plataforma web. Existen diferentes métodos para calcular dicha métrica pero no se ofrecerán todos, de la misma forma el número de métodos ofrecidos será en función del tiempo disponible y de las posibilidades de estimarlos con las tecnologías de las que se dispone. La fase final se dedicará a acabar la memoria y a preparar la defensa del proyecto.

10.3 RELACIÓN CON LAS ASIGNATURAS CURSADAS

Los conocimientos de la asignatura de Ingeniería de Requisitos serán útiles para la definición de los objetivos de las partes interesadas, la formulación correcta de los requisitos y la argumentación de su satisfacción. De Gestión de proyectos de software será útil para conocer las metodologías clásicas y ágiles así como decidir cuál es la ideal para el proyecto así como los diferentes ciclos de vida del mismo. Proyecto de Ingeniería del Software apoyará al proyecto en los conocimientos aportados respecto a la planificación, costes, términos, entregas, actos de reunión, presentaciones orales, ..etc.

Para la persistencia del sistema los conocimientos relacionados con bases de datos obtenidos durante las asignaturas de Diseño de Bases de Datos y Conceptos Avanzados de Bases de Datos Especializadas serán de ayuda para definir esquemas, tipo de base de datos ..etc. Los patrones de diseño aprendidos durante la asignatura de Arquitectura del Software serán útiles en el desarrollo de la plataforma web para desarrollar las diferentes capas de esta y decidir cuál es más indicado o si son necesarios. Para la interpretación de los datos importados en la plataforma web y como se mostrarán al usuario los conocimientos adquiridos durante Minería de Datos y Probabilidad y Estadística serán fundamentales. Finalmente las habilidades adquiridas durante *Academic Skills for Developing a Project* serán de ayuda tanto para el desarrollo del proyecto como para la comprensión del SDK de Garmin ya que toda la información del dispositivo del *smartwatch* ofrecida por el fabricante se encuentra en inglés.

10.4 ADECUACIÓN A LAS CARACTERÍSTICAS DE LA ESPECIALIDAD DE INGENIERÍA DEL SOFTWARE

El proyecto propuesto se trata de desarrollar una aplicación multiplataforma desde cero y por un único desarrollador, por lo tanto es necesario que esta persona tenga los conocimientos sobre todas las diferentes fases del proceso de desarrollo de software. A continuación se mencionan diferentes necesidades que presenta el desarrollo del sistema planteado y las capacidades de la especialidad de ingeniería del software que ayudarán a satisfacerlas:

- Se ha escogido una mitología *scrum* para organizar la implementación del producto final, por lo que ha sido necesario conocer las diferentes metodologías de desarrollo de software para elegir la más adecuada al proyecto.
- La aplicación necesitará de una base de datos que almacene la información de los diferentes usuarios en la plataforma web por lo tanto es necesario saber especificar, diseñar, implementar, evaluar y administrar bases de datos.
- La plataforma web deberá ser accesible desde cualquier dispositivo con conexión a internet, por lo que son necesarias capacidades de desarrollo y mantenimiento de aplicaciones de red. Por otro lado también poder establecer estándares de comunicación entre aplicaciones de diferentes plataformas ya que la aplicación web debe importar información desde el *smartwatch*.
- Se necesita definir una planificación temporal de todo el proyecto y establecer métodos de contingencia en caso de desviaciones temporales por lo que las capacidades de identificar, evaluar y gestionar los riesgos potenciales asociados a la construcción de software que pudiesen presentarse son esenciales.
- Se han planificado tareas de pruebas para las funcionalidades desarrolladas durante los *sprints*, por lo que es necesario poder definir controles de calidad y diseñar pruebas durante la producción del software.
- Durante todo el proyecto se realiza una memoria del mismo por lo que documentar soluciones de software de forma correcta es necesario.

Hemos visto que el proyecto se adecua perfectamente a las características de la especialidad de software. Al ser un proyecto realizado por una sola persona, es imprescindible que este sea un ingeniero del software ya que puede asegurar tener todos los conocimientos necesarios. Lo importante es conocer bien las herramientas disponibles para resolver los problemas que se presenten y ser capaz de elegir la que mejor se adapten a las necesidades del proyecto.

10.5 COMPETENCIAS TÉCNICAS

En esta sección se mencionan las competencias técnicas del proyecto y su nivel de consolidación escogido, conjuntamente se justifica el porqué de la elección de estas y como se pretende lograr al nivel de consolidación marcado.

CES1.2: Dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles. [Bastante]

En el proyecto se debe realizar por una lado una aplicación en el dispositivo *smartwatch* y también un portal web para exportar los datos, ambas plataformas deberán exportar e importar datos entre ellas. En el caso del dispositivo Garmin Epix se deberá estudiar en profundidad el SDK que ofrece el fabricante para seleccionar las funcionalidades que sirvan para los objetivos del proyecto. El portal web será desarrollado utilizando la tecnología .net que también deberá ser estudiada.

Por lo tanto, el desarrollo del sistema planteado necesita conocer los estándares de las tecnologías Garmin y .net para decidir las estrategias a seguir, que lleven a la resolución del problema planteando en función de las prestaciones que ofrecen. También se necesita establecer un estándar de intercambio de archivos o comunicación entre ambas tecnologías. Estas razones hacen que el proyecto tenga bastante relación con esta competencia.

CES1.4: Desarrollar, mantener, evaluar y mantener servicios y aplicaciones distribuidas con soporte de red. [Un Poco]

El portal web de la aplicación en si es una aplicación distribuida con soporte de red, ya que se podrá acceder desde cualquier dispositivo con conexión a internet y navegador web. El servidor ofrecerá los servicios a partir de las peticiones de los clientes y deberá acceder a bases de datos para recuperar la información almacenada.

CES1.7: Controlar la calidad y diseñar pruebas en la producción del software. [Bastante]

La aplicación en el dispositivo Garmin, se usará con seguridad mientras los usuarios realizan deporte por lo que su funcionamiento debe ser sencillo e intuitivo. Además se desea que sea utilizado por un amplio público y no solo deportistas o entrenadores de alto nivel. Para esto se han especificado pruebas con usuarios reales para asegurar que en la mayoría considera la aplicación fácil de utilizar. De la misma forma el portal web será probado para garantizar un buen nivel de usabilidad tanto en las funcionalidades como en la interpretación del análisis de las métricas.

CES2.1: Definir y gestionar los requisitos de un sistema de software. [Bastante]

Antes de empezar las fases de desarrollo del sistema se realizará una de análisis previo en la que se especificara detalladamente los diferentes requisitos, funcionales y no funcionales, que debe cumplir la aplicación así como sus respectivos criterios de satisfacción. A partir de estos se crearan los diferentes elementos del *product backlog* para los diferentes *sprints*. También está planificada la realización de una descripción completa del sistema y sus funcionalidades por lo que esta competencia queda bastante abarcada dentro del proyecto.

CES2.2: Diseñar soluciones apropiadas en uno o más dominios de la aplicación, utilizando métodos de ingeniería del software que integren aspectos éticos, sociales, legales y económicos. [En Profundidad]

Dado que tanto en el *smartwatch* como en la aplicación web se capturarán y almacenarán datos de los usuarios se asegurara de que todo el sistema siga la LOPD. También se garantizará que solo el usuario identificado correctamente tenga acceso a sus datos vía web. Por lo tanto se deberán implementar soluciones de identificación de usuario, control de sesiones e informar al usuario con todo detalle de cómo se gestionara su información.

Conclusiones

Antes de la realización de este proyecto ya había tenido contacto en mi carrera laboral con proyectos de grandes dimensiones pero solo en la fases de implementación o implementación dentro de un sprint. El proyecto me ha servido primeramente para concienciarme de la gran cantidad del gran esfuerzo que implica comenzar solo con una idea y llevarla hasta un producto final acabado, explorando todas las fases intermedias.

El desarrollo del proyecto también me ha servido para poner a prueba las capacidades de planificación y reacción, como ha sucedido con un uno de los más importantes imprevistos que se han presentado. Al presentarse la idea del proyecto por el departamento de estadística se aseguraba que se dispondría de un dispositivo smartwatch real para realizar las pruebas de la aplicación. Siguiendo esta premisa fue realizada toda una planificación temporal de principio a fin. Sin embargo dicho dispositivo nunca estuvo disponible por terceras razones en el departamento.

Este hecho implicó una reestructuración de toda la planificación y un punto de inflexión en la metodología de desarrollo. No obstante puedo decir que estoy muy complacido en cómo se he manejado la situación, consiguiendo no poner en peligro el resultado final, realizando tanto un desarrollo como una planificación con el que obtener un producto final contando con el dispositivo real o sin él.

En cuanto a conocimientos respecta, al iniciar el proyecto no tenía ninguna noción sobre desarrollo utilizando el framework asp.net y mucho menos Monkey C, que es el lenguaje de programación utilizado por el sistema de Garmin para sus smartwatch. Los conocimientos de asp.net los considero muy enriquecedores ya que es una tecnología actual y que tiene una demanda importante en el mercado laboral.

En cuando a Monkey C destaco haber puesto a prueba mis capacidades resolutivas, ya que se trata de un lenguaje con unas funcionalidades muy limitadas y muy poco explotado, funcionalidades que en otros lenguajes hubieran sido casi instantáneas, con el sistema de Garmin muchas veces me he encontrado teniendo que reestructurar toda la aplicación o pasar días intentando buscar la solución. Personalmente no recomendaría el sistema Garmin en cuanto a desarrollo de aplicaciones y creo que la empresa debe replantearse si deberían pasar a un lenguaje más actual, explotado y sobretodo con funcionalidades. En el caso de seguir adelante con su propio producto, Monkey C, es necesaria una fuerte actualización de toda la API; En caso contrario dudo que el atractivo de sus aumente o tan siquiera se mantenga con otros competidores avanzando a pasos agigantados en comparación con ellos, en lo que a tecnología de *wearables* se refiere y a facilidades para desarrolladores.

Inicialmente el proyecto partió con la idea de persistir los resultados finales de las métricas utilizando una base de datos SQL tradicional. A medida que avanzó el proyecto y debido a las limitaciones con el trato de ficheros del smartwatch Epix, se acabó teniendo que persistir un volumen muchísimo mayor de información. En lugar de almacenar los resultados finales, es decir las métricas, se han persistido todas las constantes vitales recopiladas por smartwatch durante los entrenamiento, lo que implica un numero de tuplas del orden de centena o millar sin exagerar para cada ejecución de un test. Este hecho hizo ver rápidamente las limitaciones de las bases de datos SQL, al observar con un volumen medianamente grande de usuarios, la eficiencia de la base de datos sería muy baja.

La solución a esto fue pasar a una base de datos NoSql orientada a objetos, en concreto MongoDB. Este tipo de base de datos habían sido presentadas en la asignatura de Conceptos Avanzados de Bases de Datos. Fue muy satisfactorio aplicar tales conocimientos avanzando, viéndome obligado a salir del marco de las bases de datos SQL

con éxito ya que muchas veces parece que no haya vida más allá del SQL, en lo que a bases de datos se refiere. Este cambio que nació de una adversidad finalmente se convirtió en una ventaja ya que dotó a la base de datos del sistema de un gran potencial de análisis con la información almacenada pueden extraerse nuevas métricas o estimadores del VO₂Max y realizar un verdadero trabajo de minería de datos.

En definitiva, el proyecto ha servido para consolidar conocimientos, ponerlos en práctica y adquirir nuevos. Los objetivos establecidos han sido cumplidos y se ha obtenido un producto final satisfactorio, un sistema que apoye el entrenamiento de deportistas relacionándolo con métricas le informen de su rendimiento aeróbico. También se ha conseguido importar estos datos a una plataforma web para realizar un seguimiento de dichas métricas, dotando además a la base de datos del sistema web de un gran potencial de análisis, al almacenar las constantes vitales de cada momento del entrenamiento. Todo ello compatible con terceros ya que el portal web permite analizar cualquier fichero .FIT y de la misma forma los ficheros .FIT generados por el smartwatch podrán ser leídos por otras aplicaciones al ser un formato de archivo abierto.

Trabajo Futuro

En cuanto al trabajo futuro de la aplicación primeramente es necesario probar Forever Fit Smartwatch en un dispositivo real ya que a pesar de que se conataba con uno en el departamento del propuso el proyecto, nunca se pudo tener acceso a él. La parte más crítica a probar sería la correspondiente a la generación de los ficheros .FIT durante la ejecución de los test. El resto de funcionalidades el simulador es más que valido como prueba fidedigna.

Por otro lado toda la parte relacionada al análisis de datos, hay mucho potencial por explotar. La base de datos de Forever Fit Web contiene prácticamente todas las constantes vitales del usuario momento a momento durante la ejecución del test. Toda esta información es susceptible a un fuerte análisis y trabajo de minería de datos para apoyar el entrenamiento del deportista.

Referencias

- [1] «wikipedia,» 29 02 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Capacidad_aer%C3%B3bica. [Último acceso: 29 03 2016].
- [2] F. T. Ltd, «Automated Fitness Level (VO₂max) Estimation with Heart Rate and Speed Data,» 07 11 2014. [En línea]. Available: https://www.firstbeat.com/app/uploads/2015/09/white_paper_VO2max_11-11-2014.pdf. [Último acceso: 29 03 2016].
- [3] «wikipedia,» 09 01 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Kit_de_desarrollo_de_software. [Último acceso: 29 03 2016].
- [4] P. Palau, «andro4all,» 15 11 2013. [En línea]. Available: <http://andro4all.com/2013/11/samsung-dispositivos-android>. [Último acceso: 10 02 2016].
- [5] «europe-samsung,» 01 03 2015. [En línea]. Available: <https://www.europe-samsung.com/smsdev/Home/Articulo/225>. [Último acceso: 29 02 2016].
- [6] «samsung,» 01 09 2015. [En línea]. Available: <http://www.samsung.com/es/galaxy/gear-s2/apps/>. [Último acceso: 29 02 2016].
- [7] A. G. Morato, «hipertextual,» 05 01 2016. [En línea]. Available: <http://hipertextual.com/2016/01/faltan-apps-para-apple-watch>. [Último acceso: 04 203 2016].
- [8] «RAE,» Real Academia Española y Asociación de Academias de la Lengua Española, 2010. [En línea]. Available: <http://aplica.rae.es/orweb/cgi-bin/z.cgi?t=0227102043562554148940345&s=6>. [Último acceso: 15 05 2016].
- [9] S. Koirala, «Codeproject,» 9 2014. [En línea]. Available: <http://www.codeproject.com/Articles/821275/Webforms-vs-MVC-and-Why-MVC-is-better>. [Último acceso: 09 06 2016].
- [10] F. Ast, «msdn,» 13 08 2014. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/dd381412\(v=vs.108\).aspx](https://msdn.microsoft.com/es-es/library/dd381412(v=vs.108).aspx). [Último acceso: 5 06 2016].
- [11] S. Clement, «github,» [En línea]. Available: <https://github.com/SteveClement/GarminForerunner920XTPProfile>. [Último acceso: 27 4 2016].
- [12] D. Cooper, «Github,» [En línea]. Available: <https://github.com/dtcooper/python-fitparse>. [Último acceso: 11 05 2016].
- [13] D. Innovations, «Thisisant,» [En línea]. Available: <https://www.thisisant.com/resources/fit>. [Último acceso: 08 05 2016].

- [14] C. Buckler, «Sitepoint,» [En línea]. Available: <http://www.sitepoint.com/sql-vs-nosql-choose/>. [Último acceso: 2016 05 11].
- [15] C. Buckler, «Sitepoint,» [En línea]. Available: <http://www.sitepoint.com/sql-vs-nosql-differences/>. [Último acceso: 11 05 2016].
- [16] s. IT, «DB-Engines,» [En línea]. Available: <http://db-engines.com/en/ranking>. [Último acceso: 05 05 2016].
- [17] MongoDB, «MongoDB,» [En línea]. Available: <https://www.mongodb.com/what-is-mongodb>. [Último acceso: 01 05 2016].
- [18] MongoDB, «MongoDB,» [En línea]. Available: <https://www.mongodb.com/mongodb-architecture>. [Último acceso: 2016 05 01].
- [19] Microsoft, «Msdn,» [En línea]. Available: [https://msdn.microsoft.com/es-es/library/dd381412\(v=vs.108\).aspx](https://msdn.microsoft.com/es-es/library/dd381412(v=vs.108).aspx). [Último acceso: 01 05 2016].
- [20] «Agencia Española de Protección de Datos,» 2014. [En línea]. Available: https://www.agpd.es/portalwebAGPD/canaldocumentacion/informes_juridicos/reglamento_lopd/index-ides-idphp.php. [Último acceso: 21 05 2016].
- [21] «Noticias Juridicas,» [En línea]. Available: http://noticias.juridicas.com/base_datos/Admin/lo15-1999.html. [Último acceso: 20 05 2016].
- [22] «Cuida tus datos,» 2008. [En línea]. Available: <http://www.cuidatusdatos.com/infocomocumplirlpd.html>. [Último acceso: 21 05 2016].
- [23] «(<https://app.smartsheet.com/>,» [En línea]. Available: smartsheet.
- [24] «pagepersonnel,» 01 01 2016. [En línea]. Available: http://www.pagepersonnel.es/sites/pagepersonnel.es/files/er_tecnologia16.pdf. [Último acceso: 10 03 2016].
- [25] trainingpeaks, «Matt Fitzgerald,» 04 07 2009. [En línea]. Available: <http://home.trainingpeaks.com/blog/article/the-best-way-to-monitor-your-run-training-daily>. [Último acceso: 03 14 2016].
- [26] J. Daley, «Shapesense,» [En línea]. Available: <http://www.shapesense.com/fitness-exercise/calculators/vo2max-calculator.shtml>. [Último acceso: 02 05 2016].
- [27] J. Griffing, «ExRx,» [En línea]. Available: <http://exrx.net/Calculators/Rockport.html>. [Último acceso: 2016 05 01].
- [28] G. M. Kline, «Estimation of VO₂ Max from a one mile track walk, gender, age, and body weight. Medicine and Science in sports and Exercise,» *Medicine and Science in sports and exercise*, vol. 19, nº 3, pp. 253-259, 1987.

- [29] F. Dolgener, «Uni,» [En línea]. Available: http://www.uni.edu/dolgener/Advanced_exercise_physiology/Electronic%20Articles/Rockport%20Walking%20Test.pdf. [Último acceso: 02 05 2016].
- [30] J. Griffing, «ExRx,» [En línea]. Available: <http://www.exrx.net/Calculators/OneAndHalf.html>. [Último acceso: 02 05 2016].
- [31] A. Cade, «Udel,» [En línea]. Available: http://www.udel.edu/armyrotc/current_cadets/cadet_resources/manuals_regulations_files/FM%2021-20%20-%20Physical%20Fitness%20Training.pdf. [Último acceso: 02 05 2016].
- [32] D. o. t. A. Headquarters, «Physical Fitness Training,» Washington, DC, 1992.

Apéndice: Manual de utilización de la aplicación

14.1 FOREVER FIT SMARTWATCH

Esta sección pretende guiar al usuario en la utilización de la aplicación y sus prestaciones. Se presentan los diferentes test que la forman, como ejecutarlos e interpretar sus resultados.

14.1.1 USO GENERAL

La aplicación está formada por tres pantallas cada una de las cuales representa un test. Solo se permite la ejecución de un test a la vez por lo que al cambiar de pantalla se pierde el estado del test en la pantalla anterior. Para cambiar de test se debe deslizar hacia la derecha o la izquierda con el dedo y para salir de la aplicación continuar deslizando hacia la izquierda hasta pasar todas las pantallas.



Ilustración 35. Pantalla de estado inicial de los diferentes test

Para iniciar un test se debe pulsar la pantalla y realizar las actividades. Antes, durante y después de la ejecución de los test se muestran mensajes informativos al usuario sobre qué debe hacer o el estado de la estimación. En caso de finalizar el test con éxito se mostrará el resultado de la estimación por pantalla y se generará un fichero .FIT con la información necesaria para exportar el test e importarlo a la aplicación web. En el caso del Vo2Max Speed Test es posible detener y continuar el test pulsando la pantalla una vez está en ejecución. Para este test también es necesario que el usuario introduzca su frecuencia cardíaca máxima para asegurar una estimación precisa, esto lo puede realizar en la pantalla número 2.

14.1.1.1 RESUMEN

Resumen de las instrucciones generales de utilización:

1. Abrir aplicación
2. Deslizar derecha o izquierda para seleccionar test.
3. Pulsar pantalla para iniciar test o para introducir la frecuencia cardiaca máxima en el caso de la pantalla 2.
4. Pulsar pantalla para detener o continuar test en el caso de Vo2Max Speed Test.
5. Al finalizar el test: se muestra la estimación y genera archivo .fit para la aplicación web.
6. Deslizar hacia la izquierda hasta pasar todas las pantallas para salir.

14.1.2 VO2MAX SPEED TEST

Este test sirve para estimar la velocidad a la que llegaría el corredor al llegar a su VO₂Max, es decir la velocidad que alcanzaría al llegar a su máximo rendimiento aeróbico. El test empezará a realizar las estimaciones una vez alcanzados los 12 minutos en ejecución. A partir de este momento la estimación será continua y la mostrado será una media de todas las estimaciones hasta el momento.

El dispositivo toma una muestra de la frecuencia cardiaca (FC) y la velocidad cuando son alcanzados 12 minutos de entrenamiento. Entonces el ritmo cardiaco es convertido en un porcentaje de ritmo cardiaco de reserva. La FC de reserva es la diferencia entre máximo y en reposo. Para realizar esta acción es necesario que la frecuencia cardiaca máxima del usuario sea introducido en el dispositivo, a través de la pantalla creada para esta tarea, antes de empezar a ejecutar el test. Mientras más exacto sea este tu numero mejor será la estimación. La siguiente formula es usada para calcular la FC del usuario como un porcentaje de reserva:

$$\begin{aligned} \% \text{ FC en reserva} &= (\text{FC actual} - \text{FC en reposo}) / \text{FC de reserva} \\ &= \\ &= (\text{FC actual} - \text{FC en reposo}) / (\text{FC máximo} - \text{FC reposo}) \end{aligned}$$

Entonces la velocidad alcanzada es dividida por %FC en reserva, para producir una estimación de la velocidad máxima que el usuario podría mantener durante en 12 minutos, la cual es cercano a la velocidad que alcanzaría al llegar a su Vo₂Max [25].

14.1.3 ONE MILE WALK TEST

Este test sirve para aproximar el VO₂Max del usuario. Para realizar este test se debe encontrar una ubicación adecuada en la que poder caminar una milla (1.60934 km), ya que se debe recorrer la distancia sin interrupciones ni paradas, e iniciar el test presionando la pantalla. El test se detendrá automáticamente cuando se haya recorrido la distancia necesaria. En este test no se debe correr sino caminar tan rápido como sea posible. En caso de querer detener el test es suficiente con presionar la pantalla y volver a repetir la acción para continuarlo. A continuación se describe como es calculada la métrica:

$$VO_2Max = 132.853 - 0.0769W - 0.3877A + 6.315G - 3.2649T - 0.1565H$$

Donde:

W = peso (en Libras)

A = edad (en años)

G = Facto Genero (G = 0 femenino, G = 1 masculino)

T = Tiempo de ejecución del test (en minutos)

H = Ritmo cardiaco al terminar (latidos/minute)

A diferencia del test *One Half Mile Run* este test ofrece al usuario la posibilidad tener una estimación de su VO₂Max sin necesidad de realizar un esfuerzo tan drástico, ya que el factor determinante no es únicamente el tiempo que se tarda en recorrer la distancia marcada. [26] [27] [28] [29]

14.1.4 ONE HALF MILE RUN TEST

Este test sirve para aproximar el VO₂Max del usuario. Consiste en recorrer 1,5 millas (2.41402 km), en el menor tiempo posible por lo tanto es una prueba de esfuerzo, más dura que la del test anterior además se debe recorrer la distancia sin interrupciones ni paradas. Está caminar solo en caso de agotamiento. Debido a que requiere un esfuerzo importante el usuario debe ser cuidadoso y no sobrepasar sus capacidades. A continuación se describe como es calculada la métrica:

$$VO_2Max = 483/T + 3.5$$

Donde:

T = Tiempo de ejecución del test (en minutos)

Al igual que los test anteriores se recomienda realizar estiramientos antes y después de realizarlo con un descanso posterior adecuado. Se mostrara la estimación del VO₂Max una vez recorrida la distancia marcada. [26] [30] [31] [32]

14.2 FOREVER FIT SMARTWATCH

Esta sección pretende guiar al usuario en la utilización de la aplicación y sus prestaciones. Primeramente para acceder a las funcionalidades del sistema el usuario debe estar registrado e identificado en el sistema.



Forever Fit Registrarse Identificarse

Forever Fit

Manten un seguimiento de la evolución de tu capacidad aeróbica y saca el máximo partido a tus entrenamientos

[Empezar](#)

One Mile Walk Test

En este predictor del VO2Max se tienen en cuenta: genero, peso y edad del usuario. Conjuntamente con el tiempo que se ha tardado en recorrer una milla y el ritmo cardiaco en dicho momento. En este test no se debe correr sino caminar tan rápido como sea posible. Supone un test suave sin necesidad de realizar un esfuerzo máximo por parte del usuario.

One Half Mile Run Test

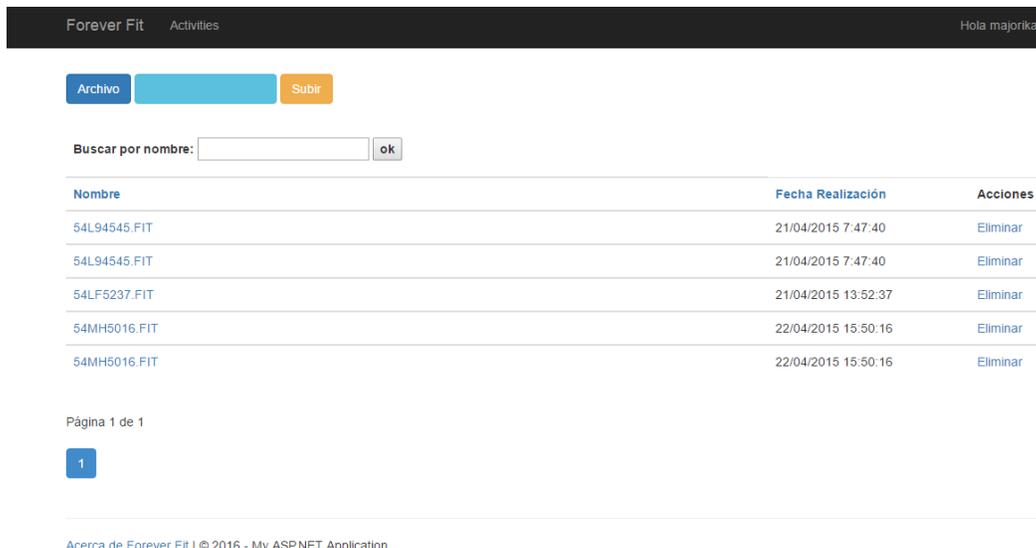
Este test consiste en recorrer 1.5 millas en el menor tiempo posible para tener una estimación de su VO2MAX, por lo tanto es una prueba de esfuerzo. Está permitido correr o caminar. Debido a que requiere un esfuerzo importante el usuario debe ser cuidadoso y no sobrepasar sus capacidades. En este test no se tiene en cuenta el tiempo que se tarda en recorrer la distancia marcada.

Vo2max Speed

Predictor de la velocidad máxima alcanzable al llegar al VO2Max, este test requiere una **duración mínima de 12 minutos**, apartir de entonces la estimación será continua por lo que la estimación final puede ser ligeramente diferente a la inicial. Para este test es necesaria la frecuencia cardiaca en reposo del usuario y su frecuencia cardiaca máxima, mientras mas precisos sean estos datos mayor será la estimación.

[Acerca de Forever Fit](#) | © 2016 - My ASP.NET Application

En la pantalla de actividades se muestran un listado con los ficheros .FIT subidos por el usuario. Sobre esta lista se encuentra el componente para subir dichos archivos, solo está permitido subir archivos de este tipo de lo contrario seremos redireccionados a la página de error. Al subir el archivo la lista de actividades se actualizará automáticamente, dicha lista esta página y muestra 13 elementos por página.



Forever Fit Activities Hola majorikal

[Archivo](#) [Subir](#)

Buscar por nombre:

Nombre	Fecha Realización	Acciones
54L94545.FIT	21/04/2015 7:47:40	Eliminar
54L94545.FIT	21/04/2015 7:47:40	Eliminar
54LF5237.FIT	21/04/2015 13:52:37	Eliminar
54MH5016.FIT	22/04/2015 15:50:16	Eliminar
54MH5016.FIT	22/04/2015 15:50:16	Eliminar

Página 1 de 1

[1](#)

[Acerca de Forever Fit](#) | © 2016 - My ASP.NET Application

Ilustración 36. Pantalla actividades FFW

En el listado de actividades nos encontramos con las columnas de nombre y "Fecha Realización", esta última columna hace referencia a la fecha en que se creó el fichero .FIT original en el smartwatch no a la de subida del archivo. Si pulsamos el link del nombre del activity seremos dirigidos a la página del detalle del mismo. En esta página se muestra un resumen de la información del activity, el resultado de las métricas y el listado de todos los registros del activity con una paginación de 13 elementos.

Forever Fit Activities Hola majorikal Salir

Resumen

Realizado: 21/04/2015 7:47:40
 Duración: 11 min
 Distancia: 3916.6 m.
 Muestras: 146

One Mile Walk Test

89,17

One Half Mile Run Test

86,06

Vo2max Speed

N/A 12 minutos

N/A Test completo

Momento	Latitud	Longitud	Distancia	Velocidad	Ritmo Cardiaco
21/04/2015 7:47:40	591842447	73171662	541,71	5,944	125
21/04/2015 7:47:41	591843013	73171327	547,36	5,645	125
21/04/2015 7:47:44	591844531	73170152	563,17	5,421	125
21/04/2015 7:47:46	591845648	73169478	574,35	5,328	124
21/04/2015 7:47:47	591846174	73169111	579,74	5,44	124
21/04/2015 7:47:54	591849710	73165861	618,13	5,514	124
21/04/2015 7:47:56	591850643	73164811	628,89	5,458	124
21/04/2015 7:48:01	591852773	73161495	657,22	5,617	126
21/04/2015 7:48:03	591853494	73159988	668,55	5,598	126
21/04/2015 7:48:07	591854669	73157548	686,93	3,798	126
21/04/2015 7:48:11	591855557	73155862	700,09	3,434	123
21/04/2015 7:48:17	591857263	73152586	725,5	4,721	122
21/04/2015 7:48:21	591859093	73149439	751,02	5,421	122

Página 1 de 12

Ilustración 37. Pantalla detalle activity FFW

La aplicación cuenta con una página para editar la información del usuario, esta información es utilizada por los test ha sido que debe de ser lo más precisa posible para asegurar una estimación correcta de las métricas.

Forever Fit Activities Hola majorikal

View

User

Usuario
Genero
Peso
Fecha de nacimiento
Frecuencia cardiaca máxima
Frecuencia cardiaca en reposo

Acerca de Forever Fit | © 2016 - My ASP.NET Application

Ilustración 38. Pantalla editar información usuario FFW