



**Universidad**  
Zaragoza

## Proyecto Fin de Carrera

APPOWER : Aplicación móvil para la medición de potencia en entrenamientos de fuerza.

Autor

Carlos Alberó Castellón

Directores

Álvaro Alesanco Iglesias  
Nuria Garatachea Vallejo

Universidad de Zaragoza  
Escuela de Ingeniería y Arquitectura  
2015



# Agradecimientos

*Quisiera agradecer a mis profesores Álvaro y Nuria la oportunidad de trabajar en este proyecto, así como toda la ayuda que me han ofrecido.*



# APPOWER: Aplicación móvil para la medición de potencia en entrenamientos de fuerza

## RESUMEN

Este Proyecto Fin de Carrera ha consistido en el desarrollo de una aplicación móvil capaz de realizar e interpretar medidas de fuerza y potencia durante el entrenamiento de fuerza, en concreto para dos ejercicios, press de banca y sentadillas.

Mediante los datos que proporciona el acelerómetro de los teléfonos móviles, y realizando una serie de cálculos y tratamiento de señal, se consigue presentar por pantalla las variables más relevantes en estos ejercicios. Además, se lleva a cabo un reconocimiento de repeticiones, de manera que estos datos se analizan para cada una de ellas.

En todo entrenamiento de fuerza, resulta fundamental llevar un control y seguimiento de los ejercicios que se realizan en cada sesión y de ello depende en gran medida la consecución de los objetivos del deportista.



# Índice general

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación . . . . .	4
1.2	Objetivos . . . . .	4
1.3	Estado del arte . . . . .	5
1.3.1	Aplicaciones móviles . . . . .	5
1.3.2	Acelerómetro en dispositivos móviles . . . . .	6
1.3.3	Codificador lineal . . . . .	8
1.3.4	Dispositivos actuales. . . . .	9
1.4	Materiales y tecnologías utilizadas . . . . .	9
1.5	Organización de la memoria . . . . .	11
<b>2</b>	<b>Descripción del sistema</b>	<b>13</b>
2.1	Máquina de Smith o Multipower . . . . .	13
2.2	Ejercicios . . . . .	13
2.2.1	Sentadillas . . . . .	13
2.2.2	Press de banca . . . . .	14
2.3	Phonegap . . . . .	15
2.4	Acceso al acelerómetro. . . . .	16
2.5	Detección de repeticiones . . . . .	17
2.6	Velocidad y Potencia . . . . .	19
<b>3</b>	<b>Desarrollo de la Aplicación</b>	<b>23</b>
3.1	Calibrado . . . . .	24
3.2	Entrenamiento . . . . .	27



3.2.1	Entrenamiento controlado . . . . .	29
3.2.2	Entrenamiento libre . . . . .	30
3.3	Historial . . . . .	30
3.4	Validación básica de los datos . . . . .	31
<b>4</b>	<b>Conclusiones y líneas futuras</b>	<b>37</b>
4.1	Conclusiones . . . . .	37
4.2	Desarrollos futuros . . . . .	38
4.3	Cronograma de Tareas . . . . .	38
	<b>Bibliografía</b>	<b>41</b>
<b>A</b>	<b>Acrónimos</b>	<b>43</b>
<b>B</b>	<b>Ejecución de la aplicación</b>	<b>45</b>
<b>C</b>	<b>Guía de uso</b>	<b>47</b>
C.1	Calibrado . . . . .	48
C.2	Entrenamiento . . . . .	49
C.2.1	Entrenamiento controlado . . . . .	49
C.2.2	Entrenamiento libre . . . . .	50
C.3	Historial . . . . .	51
<b>D</b>	<b>Funciones más importantes</b>	<b>53</b>
D.1	Acceso y tratamiento de la señal de aceleración . . . . .	53
D.2	Guardar archivo XML . . . . .	58
D.3	Media . . . . .	60
D.4	Interpolación . . . . .	61
D.5	Gráficos . . . . .	61

# Índice de figuras

1.1	Ejemplo de Curva de Potencia. . . . .	3
1.2	Tipos de aplicaciones. . . . .	6
1.3	Ejes del acelerómetro. . . . .	7
1.4	Encoder Lineal. . . . .	8
1.5	Máquina de Smith o Multipower. . . . .	11
2.1	Sentadilla en Multipower. . . . .	14
2.2	Press de Banca en Multipower. . . . .	14
2.3	Encapsulado con Phoengap. . . . .	15
2.4	Aceleración resultante de una repetición . . . . .	17
2.5	Esquema de detección de repeticiones. . . . .	18
2.6	Aceleración a su paso por los filtros de umbral y paso bajo. . . . .	19
2.7	Aceleración y velocidad. . . . .	20
2.8	Fuerza y Potencia. . . . .	21
3.1	Esquema de la aplicación. . . . .	23
3.2	Diagrama UML de la aplicación. . . . .	24
3.3	Esquema básico de Calibrado. . . . .	25
3.4	Curva de Potencia. . . . .	25
3.5	Entrenamiento Sentadilla. . . . .	27
3.6	Gráfica que expresa la potencia durante el entrenamiento. . . . .	30
3.7	Aceleración, velocidad y correlación con móvil 1. . . . .	33
3.8	Fuerza, potencia y correlación con móvil 1 . . . . .	33
3.9	Aceleración, velocidad y correlación con móvil 2. . . . .	34

3.10	Fuerza, Potencia y correlación con móvil 2. . . . .	34
4.1	Diagrama de Gantt. . . . .	39
C.1	Pantalla de inicio. . . . .	47
C.2	Pantalla inicial de Calibrado. . . . .	48
C.3	Pasos para completar el Calibrado. . . . .	48
C.4	Pantalla de selección de entrenamiento . . . . .	49
C.5	Pantallas Entrenamiento Controlado . . . . .	49
C.6	Pantallas Entrenamiento Libre . . . . .	50
C.7	Acceso al historial. . . . .	51
C.8	Consulta de entrenamientos pasados . . . . .	52
C.9	Consulta de Curva de potencia. . . . .	52

# Índice de códigos

3.1	Código que calcula potencia y valores de pico en la calibración. . . .	26
3.2	Código que genera la potencia y detección de repeticiones en el entrenamiento . . . . .	28
3.3	Ejemplo de archivo XML. Ejercicio con 3 repeticiones. . . . .	31
B.1	Código para ejecutar la aplicación desde nuestra código HTML. . .	45
D.1	Código con las funciones principales de la calibración . . . . .	53
D.2	Código con las funciones principales del entrenamiento . . . . .	55
D.3	Código que genera un fichero XML y lo salva. . . . .	58
D.4	Código encargado de generar avisos sonoros. . . . .	60
D.5	Código que realiza la interpolación lineal para generar la curva de potencia. . . . .	61
D.6	Código correspondiente a los gráficos de la aplicación . . . . .	62



# Capítulo 1

## Introducción

La fuerza en el ámbito deportivo se entiende como la capacidad de producir tensión que tiene el músculo al activarse o, como se entiende habitualmente, al contraerse [1]. En el entrenamiento de fuerza se hace uso de resistencias con el fin de lograr la contracción muscular. Como objetivos de este entrenamiento podemos tener la mejora de la fuerza resistencia, fuerza velocidad, fuerza máxima o la hipertrofia.

La fuerza velocidad o potencia tiene como objetivo el desarrollo muscular para conseguir una mayor explosividad, es decir, lograr altos índices de fuerza en el menor tiempo posible. Es determinante en cualquier deporte, ya que influye en sprints, arrancadas, etc. El entrenamiento dirigido a la mejora de este tipo de fuerza trabajará a pocas repeticiones, con cargas altas y de manera explosiva, dejando largos descansos entre series.

La fuerza resistencia busca aumentar la capacidad del deportista para que este sea capaz de mantener las contracciones musculares durante un tiempo prolongado y sin una disminución del rendimiento de trabajo. Es fundamental en multitud de deportes, en los que hay que mantener una fuerza a un nivel constante durante un periodo largo de tiempo, como ocurre en el boxeo, la gimnasia, la marcha, etc. Se trabaja realizando ejercicios a un número alto de repeticiones, normalmente mayor de 20, con cargas no muy pesadas y poco descanso entre series.

La fuerza máxima corresponde con la máxima contracción muscular voluntaria

que el deportista es capaz de realizar. Es determinante en halterofilia, saltos, lanzamientos, etc. Se entrena con cargas muy altas, a más repeticiones que la fuerza velocidad, entre 3 y 8 habitualmente, con un tiempo de descanso menor y de forma menos explosiva.

Por último, la hipertrofia, donde no se busca una mejora de las capacidades físicas del deportista, sino una mejora estética mediante el aumento de tamaño de los músculos. La principal característica se encuentra en los tiempos de ejecución del ejercicio, donde tanto la contracción como la extensión muscular van a durar más tiempo que en cualquier otro tipo de entrenamiento [2].

La efectividad del entrenamiento para el desarrollo de la fuerza depende de la aplicación de una carga adecuada. La carga de un entrenamiento se refiere a los factores que componen dicho entrenamiento, volumen e intensidad. El primero se corresponde con la parte cuantitativa del mismo, número de repeticiones por serie, cantidad de ejercicios por entrenamiento, número de días de entrenamiento, etc. La intensidad tiene que ver con la parte cualitativa, es decir, cantidad de peso que se va a movilizar, velocidad de ejecución, tiempo de descanso, etc.

La velocidad de ejecución es un elemento determinante de la intensidad debido a que tanto las exigencias neuromusculares como los efectos del entrenamiento dependen en gran medida de ella. Cuanto mayor sea la velocidad ante una resistencia, mayor será la intensidad, por tanto, lo importante en la velocidad como factor de intensidad es que sea la máxima posible o casi para la resistencia que se desplaza [3].

En los últimos años, se han desarrollado diferentes dispositivos que permiten un cálculo preciso de la fuerza, velocidad y potencia mecánica aplicadas en ejercicios con resistencias, tales como sentadillas o press de banca. Estos avances han demostrado que la relación entre la cantidad de peso cargado y la velocidad o de otra forma, entre fuerza y potencia dependen, además del tipo de ejercicio o músculos implicados, de las características neuromusculares de cada persona [4].

Resulta muy interesante la utilización de este tipo de dispositivos para la cuantificación del entrenamiento en términos de potencia y fuerza. Una de sus aplicaciones es la determinación de la curva de potencia del deportista, que

establece la relación entre la velocidad y la resistencia que mueve una persona realizando un ejercicio determinado. Ofrece una información muy valiosa para deportistas y personas que practican algún deporte puesto que permite conocer la carga óptima con la que se obtiene la potencia máxima en el ejercicio.

Esta curva se calcula realizando una serie de tests en un determinado ejercicio, comenzando con resistencia baja e incrementándola progresivamente. En ella se puede observar el aumento gradual de la potencia hasta alcanzar el valor de pico máximo y su posterior descenso. El rango de la curva de potencia va desde la ausencia de resistencia hasta el valor de 1RM<sup>1</sup>.

En la Figura 1.1 podemos observar un ejemplo de una curva de potencia, en este caso calculada en sentadillas. Se han completado 6 tests, con el 15 %, 30 %, 40 %, 50 %, 60 % y 70 % de su 1RM. De esta curva podemos extraer que la máxima potencia para el ejercicio de sentadillas el deportista la alcanza con una carga del 50 % de su 1RM. Además, si el deportista entrena con un determinado peso, por ejemplo con el 70 % de su 1RM, conoce el valor de potencia que debe desarrollar, en este caso 300 W para trabajar al máximo. Esta es una buena manera de controlar la calidad del entrenamiento, puesto que se conoce a priori la potencia que el deportista debe alcanzar y mantener durante el ejercicio.

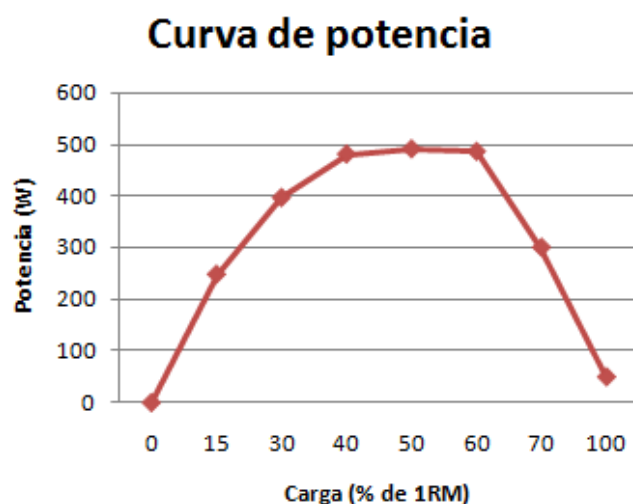


Figura 1.1: Ejemplo de Curva de Potencia.

<sup>1</sup>cantidad máxima de peso que una persona puede llegar a cargar en determinado ejercicio para realizar una repetición.



El objetivo principal de este proyecto será el desarrollo de una herramienta capaz de realizar medidas de fuerza y potencia en el entrenamiento y calcular la curva de potencia en el ejercicio en forma de aplicación móvil, utilizando los datos que aporta el acelerómetro.

## 1.1 Motivación

El control y seguimiento del entrenamiento es una de las claves en el éxito y la consecución de los objetivos para cualquier plan de preparación física. Dentro del entrenamiento de fuerza, esto cobra mayor importancia si cabe, puesto que se trata de ejercicios muy cortos y específicos.

Para el deportista, y sobre todo para su entrenador, tener una herramienta capaz de medir las variables más relevantes del entrenamiento resulta de gran ayuda de cara a la evaluación del progreso y la planificación. En el caso del entrenamiento de fuerza, estas variables son la fuerza y la potencia.

Cada vez son más las aplicaciones móviles de uso en el ámbito deportivo, que facilitan la monitorización de los entrenamientos del usuario. Se trata de una herramienta muy cómoda, ya que se puede llevar encima en cualquier momento, y que no requiere de ningún tipo de instalación. Además, permite la interacción a tiempo real con el usuario.

Desde este proyecto, se quiere presentar una aplicación para móvil, sencilla y capaz de interactuar con el deportista mediante señales auditivas, que permita llevar un seguimiento y medición en el entrenamiento de fuerza.

## 1.2 Objetivos

El principal objetivo de este proyecto es el de proporcionar tanto al deportista como al entrenador una herramienta en forma de aplicación móvil que le permita llevar un registro y un control sobre sus ejercicios de entrenamiento de la fuerza.

Para alcanzar este objetivo, se desarrollará una aplicación multiplataforma que se pueda utilizar en cualquier dispositivo sin importar el sistema operativo

del mismo. Se ha elegido la plataforma de desarrollo Phonegap, que permite la implementación de este tipo de aplicaciones híbridas.

Esta aplicación deberá ser capaz de generar la curva de potencia del usuario para los dos ejercicios elegidos, sentadilla y press de banca. Esta curva se almacenará y se utilizará en la parte de control del entrenamiento.

Se implementará un detector de repeticiones, con objeto de poder separar los datos obtenidos para cada una de ellas, además de la posibilidad de poder llevar una cuenta mediante una señal sonora. Con este detector, se llevará a cabo el control del entrenamiento, determinando la potencia y fuerza desarrollada en cada una de las repeticiones, pudiendo avisar al usuario si se están realizando de manera adecuada.

Por último, se permitirá el almacenaje de los entrenamientos en los ficheros del dispositivo, con el objeto de poder realizar consultas de cada uno de ellos y observar el progreso del deportista.

## 1.3 Estado del arte

### 1.3.1 Aplicaciones móviles

A la hora de desarrollar una app para dispositivos móviles existen tres alternativas. Figura 1.2:

- **Aplicaciones nativas:** Una aplicación nativa es aquella que está escrita en el lenguaje propio de cada dispositivo, por ejemplo, en Java para dispositivos con sistema operativo Android o en Objective-C para dispositivos con sistema operativo iOS. Este tipo de aplicaciones tiene la ventaja de tener el completo acceso a las funcionalidades del hardware del dispositivo.
- **Aplicaciones web:** Las aplicaciones web son, básicamente, sitios web que parecen y funcionan como aplicaciones nativas. Sin embargo, solo pueden ser accedidas mediante el navegador instalado en el dispositivo, y no tienen acceso a las funcionalidades hardware del dispositivo.

- **Aplicaciones híbridas:** Se encuentran en medio de las dos primeras, ya que son fundamentalmente webs encapsuladas en código nativo. Esto les permite el acceso a las funciones hardware del dispositivo que no tienen las aplicaciones web, y un desarrollo común para todos los sistemas operativos, principal desventaja en las aplicaciones nativas.



Figura 1.2: Tipos de aplicaciones.

En nuestro caso se ha escogido la tercera opción, una aplicación híbrida. La razón principal es la ventaja que supone el no tener que aprender y manejar los distintos lenguajes de programación que conllevaría desarrollar la aplicación nativa para los distintos sistemas operativos actuales. Además, permiten el acceso al hardware del dispositivo, en nuestro caso, al acelerómetro. Como simplemente hacemos uso de este sensor, y no hay instrucciones ni cálculos muy complejos, una aplicación de este tipo es la elección adecuada.[5]

### 1.3.2 Acelerómetro en dispositivos móviles

Un acelerómetro es un sensor capaz de convertir aceleraciones, cambios de velocidad, en una señal eléctrica. Existen de varios tipos: mecánicos,

piezoeléctricos, capacitivos, etc. En el caso de los teléfonos móviles, los más utilizados son estos últimos, cuyo funcionamiento se basa en un condensador con una placa fija y otra móvil. Con el movimiento de la placa móvil, la capacidad del condensador varía, quedando así registrada la magnitud de dicho movimiento.

El acelerómetro que contienen los teléfonos móviles permite a día de hoy el desarrollo de multitud de aplicaciones, muchas de ellas de uso en el ámbito deportivo para medir distancias, calorías, velocidad, etc. Normalmente se utiliza junto a otros sensores, tales como el giroscopio, que registra la velocidad angular del dispositivo y el magnetoscopio, que funciona como brújula para determinar la dirección. En nuestro caso, únicamente utilizaremos el acelerómetro, ya que los ejercicios, por realizarse en la máquina Smith o Multipower, se desarrollarán en un mismo plano.

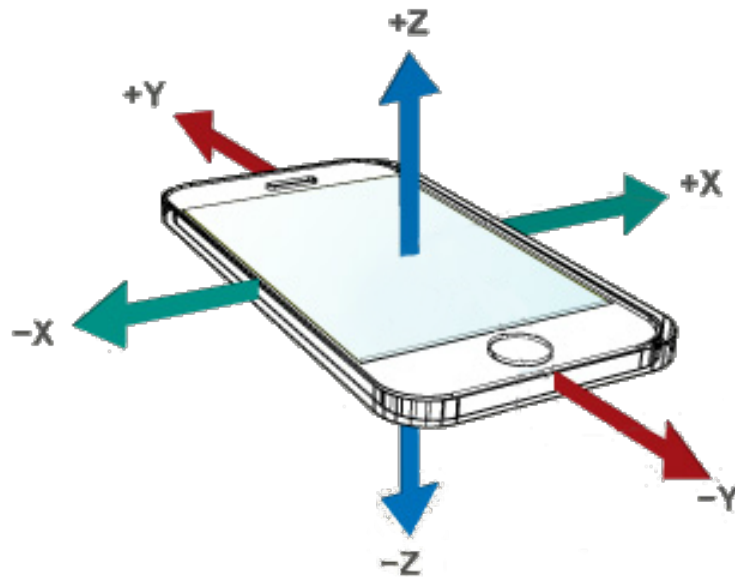


Figura 1.3: Ejes del acelerómetro.

Los datos que genera el acelerómetro a partir de los movimientos registrados del dispositivo son la aceleración que este sufre en cada uno de los tres ejes del espacio tridimensional, como se aprecia en la Figura 1.3. Hay que distinguir entre la aceleración dinámica que es la que aparece al mover el dispositivo y aceleración estática, que es la que produce la fuerza de la gravedad.

### 1.3.3 Codificador lineal

Un codificador lineal o *encoder* lineal es un sensor que codifica posición mediante un dinamómetro. Es el dispositivo de referencia a la hora de realizar medidas de fuerza y potencia en este tipo de ejercicios, debido a su gran precisión. El funcionamiento del dinamómetro se basa en la ley de Hooke. A partir de los cambios en la elasticidad de un muelle con una determinada calibración, permite calcular el peso de un cuerpo o realizar la medición de una fuerza. En los dispositivos utilizados para el ámbito deportivo, un cable se engancha a la carga externa, es decir a la barra con la que se realiza el ejercicio. En el otro extremo, fijado al dinamómetro en el suelo, se produce la elongación y compresión del mismo a la vez que se va realizando el ejercicio (Figura 1.4 ).

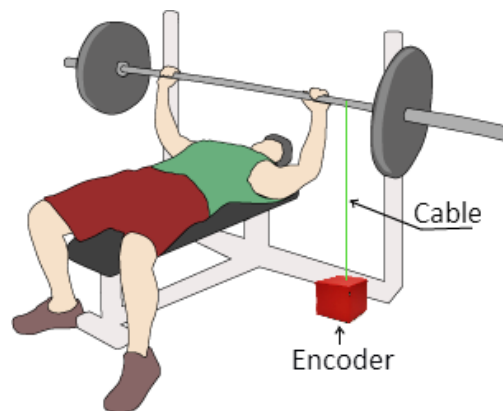


Figura 1.4: Encoder Lineal.

Estos dispositivos permiten, por tanto y mediante cálculos, obtener variables como la potencia y el trabajo mecánico, la fuerza o la velocidad, todas interesantes para el entrenamiento y la evaluación de la fuerza. El error que producen estos dispositivos es muy pequeño, menor cuanto más vertical es el movimiento del ejercicio. Es por ello que se recomienda el uso de una máquina de peso guiado sobre los pesos libres, ya que de esta manera el recorrido es completamente vertical.

La resolución que presentan los codificadores lineales es muy alta, y la tasa de muestreo puede llegar hasta los 1000 Hz.

Como aspectos negativos, se puede comentar que al ser dispositivos relativamente nuevos, resultan caros. Además, requieren de una cierta instalación

y son dependientes de otro dispositivo mayor como sería un ordenador, que es dónde se encuentra el software para la interpretación de los datos.

### 1.3.4 Dispositivos actuales.

A continuación, se presentan algunos de los dispositivos conocidos en el mercado cuya finalidad es similar a la que se le quiere dar a nuestra aplicación.

- **T-Force Encoder:** Es un sistema basado en un codificador lineal, cuyo funcionamiento se explica en el apartado 1.3.3. El codificador debe estar conectado en todo momento a un ordenador, donde está alojado el software que interpreta los datos recogidos. Es el dispositivo utilizado en las pruebas de laboratorio para dar validez a los datos ofrecidos por el teléfono.
- **Chronojump:** Dispositivo similar al anterior, basado en un codificador lineal, de este tipo está entre los más económicos, rondando su precio los 500 euros.
- **SmartCoach Power Encoder (Keiser):** Otro codificador lineal, pero de gama alta, su precio está por encima de los 2000 euros.
- **MyoTest:** Dispositivo basado en un acelerómetro, que puede ser utilizado para la medición de variables del entrenamiento en multitud de disciplinas. Su tamaño es similar al de un teléfono móvil. Su precio está entorno a los 800 euros.
- **Liza:** Aplicación móvil desarrollada exclusivamente para dispositivos iOS, que hace uso del acelerómetro para realizar medidas de fuerza y potencia en ejercicios de entrenamiento de fuerza. Última actualización en 2011.

## 1.4 Materiales y tecnologías utilizadas

Respecto a los materiales y tecnologías utilizados en este proyecto, ha sido necesario manejar los siguientes:

- **Eclipse:** Entorno de programación, desarrollo y compilación que se ha utilizado para la implementación del código de la aplicación, así como para las pruebas sobre emuladores.
- **Phonegap:** Plataforma de desarrollo de aplicaciones móviles multiplataforma, de código abierto y similar al desarrollo web. Se ha utilizado la versión 4.1.2.
- **HTML5:** Lenguaje de marcado para la elaboración de páginas web. Se utiliza de manera integrada para el desarrollo de aplicaciones en Phonegap.
- **Javascript:** Lenguaje de programación interpretado utilizado para la elaboración de la aplicación.
- **Java:** Lenguaje de programación utilizado para desarrollar la aplicación. Se ha utilizado la versión 1.8.0.
- **Android:** Sistema operativo basado en Linux, de código abierto, para el cual se ha desarrollado la aplicación.
- **Jquery Mobile:** Entorno de trabajo optimizado para dispositivos móviles. Se ha utilizado la versión 1.4.5.
- **Matlab:** Lenguaje de alto nivel y entorno interactivo para el cálculo numérico, la visualización y la programación. Se ha utilizado para el desarrollo y comparación de gráficas en la versión 7.10
- **XML:** Lenguaje de marcas utilizado para almacenar datos de forma legible. Todos los archivos almacenados por la aplicación tienen esta extensión.
- **T-Force Encoder:** Sistema de medición dinámica de la fuerza y la potencia, utilizado en las pruebas de laboratorio.
- **Máquina de Smith o Multipower:** Máquina consistente en una barra fija a unos raíles verticales, que permite el movimiento exclusivamente en ese plano. Puede ser utilizada para realizar gran cantidad de ejercicios y se muestra en la Figura 1.5.



Figura 1.5: Máquina de Smith o Multipower.

## 1.5 Organización de la memoria

La memoria está estructurada de la siguiente forma:

- En el **Capítulo 1** se encuentra una breve introducción al proyecto, junto con la motivación, objetivos, materiales y herramientas utilizadas.
- En el **Capítulo 2** se describe brevemente el sistema, los ejercicios elegidos y la máquina sobre la que se van a realizar, la plataforma de programación y el acceso y tratamiento de los datos que proporciona el acelerómetro.
- En el **Capítulo 3** se explica con mayor profundidad cada una de las partes de la aplicación.
- El **Capítulo 4** contiene los resultados, conclusiones y posibles líneas de desarrollo futuras.

Los anexos que se encuentran al final de la memoria se distribuyen de la siguiente manera:

- En el **Anexo A** se encuentran los acrónimos utilizados en la redacción del documento.
- En el **Anexo B** se muestra la parte de código nativo necesaria para correr la aplicación.



- En el **Anexo C** se encuentra la guía de usuario de la aplicación.
- En el **Anexo D** aparece parte de código Javascript correspondiente a las funciones más representativas del proyecto.

# Capítulo 2

## Descripción del sistema

### 2.1 Máquina de Smith o Multipower

Se ha establecido que los ejercicios se realicen en la máquina de Smith o Multipower (Figura 1.5). Consiste en una barra fija a unos raíles, que permiten su movimiento solo en el plano vertical. Dependiendo del modelo, estos pueden tener una ligera inclinación. Se puede realizar en ella multitud de ejercicios. En nuestro caso nos hemos centrado en sentadillas y press de banca.

Se ha elegido esta máquina ya que solo permite el movimiento en un plano, facilitando así la interpretación de los datos del acelerómetro. Cuando se producen movimientos en otras direcciones, habría que hacer uso de otros sensores como son el giroscopio o el magnetoscopio.

### 2.2 Ejercicios

La aplicación se va a desarrollar principalmente para dos ejercicios, sentadillas y press de banca.

#### 2.2.1 Sentadillas

Ejercicio básico para el entrenamiento de los músculos de las piernas. Trabaja directamente los músculos de muslo, cadera y glúteos y fortalece los huesos,

ligamentos y tendones de piernas y caderas. El movimiento básico se inicia siempre en posición erguida. El ejercicio se realiza flexionando rodillas y cadera para hacer bajar el cuerpo hacia el suelo sin perder la verticalidad, volviendo luego a la posición erguida. Existen variedades en función de la profundidad del movimiento y de dónde se coloque el peso. Figura 2.1.

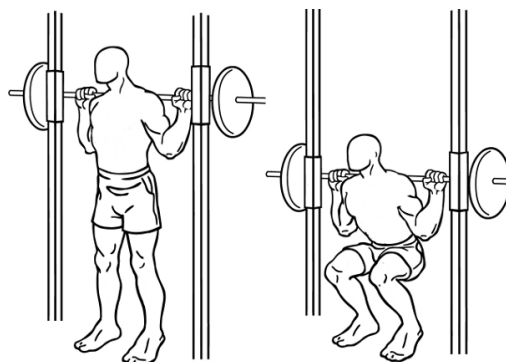


Figura 2.1: Sentadilla en Multipower.

### 2.2.2 Press de banca

Ejercicio principal en el entrenamiento del músculo pectoral. También implica la utilización de tríceps y deltoides. El ejercicio se realiza tumbado en un banco, levantando y bajando el peso por encima del pecho con una separación entre manos igual a la de los hombros tal y como se puede ver en la Figura 2.2.

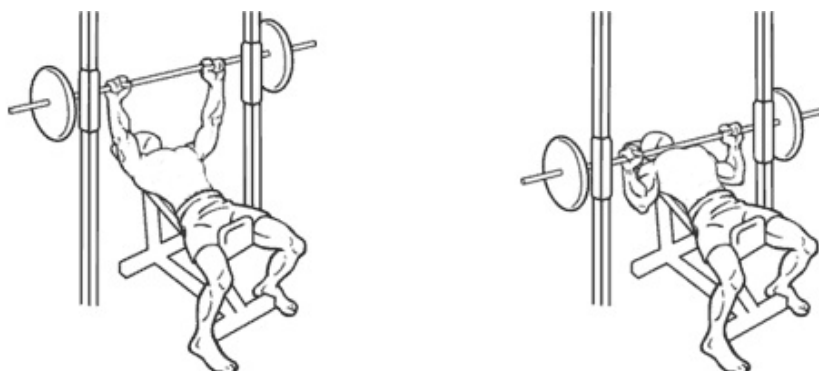


Figura 2.2: Press de Banca en Multipower.

## 2.3 Phonegap

Phonegap es un entorno de trabajo para el desarrollo de aplicaciones móviles perteneciente a Adobe Systems que permite el desarrollo de aplicaciones para dispositivos móviles mediante el uso de herramientas genéricas tales como HTML, Javascript y CSS. El resultado es una aplicación híbrida, es decir, que no se trata de una aplicación nativa al dispositivo, ni tampoco de una aplicación web, si no de una solución intermedia.

La principal ventaja de este tipo de aplicaciones es que pueden ser usadas en cualquier dispositivo inteligente, independientemente del sistema operativo que utilice. Además, mediante una serie de APIs <sup>1</sup>, permite el acceso a funciones y elementos internos del dispositivo, tales como el acelerómetro, la cámara, la red, el almacenamiento, etc.

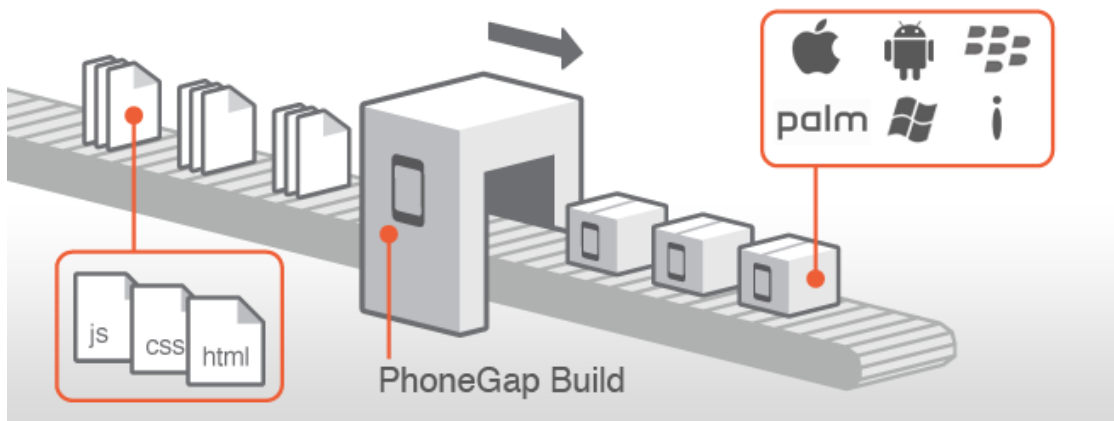


Figura 2.3: Encapsulado con Phoengap.

El código de una aplicación desarrollada en Phonegap consta de dos partes diferenciadas. Por un lado la parte nativa, que son los plugins que permiten el acceso a las funciones propias de cada dispositivo, y la función encargada de lanzar la aplicación desde el código web. Esta parte encapsula el código principal y hace de puente entre el dispositivo y la aplicación.

Por otro lado tenemos el programa principal, dividido entre código HTML5 para la parte gráfica y de navegación, con sus correspondientes hojas de estilo

<sup>1</sup>Application Programming Interface, conjunto de reglas y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas

CSS, y las funciones Javascript, que permiten la interactividad con el usuario. Permite también trabajar con JQuery Mobile, lo que consigue dar un aspecto más atractivo y mayor funcionalidad a la aplicación.

## 2.4 Acceso al acelerómetro.

Como ya se ha mencionado anteriormente, Phonegap permite el acceso a las funciones hardware de los dispositivos mediante el uso de una serie de plugins. Estos plugins son partes de código nativo (en el caso de Android, Java) que permiten a nuestras funciones Javascript comunicarse con el hardware del dispositivo. De esta manera se consigue acceder a la cámara, al sistema de archivos, al acelerómetro, a la localización por GPS, etc.

En el caso del acelerómetro, el plugin que da acceso a este sensor, nos permite consultar la aceleración en un determinado instante de tiempo, o como en nuestro caso, obtener los datos en tiempo real actualizados a una frecuencia modificable. El plugin ha sido modificado para que proporcione directamente el módulo de la aceleración, en lugar de en sus tres componentes. La fórmula utilizada es la que aparece en la Ecuación 2.1.

$$a_m = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (m/s^2) \quad (2.1)$$

A partir de la aceleración, podemos obtener directamente la fuerza, simplemente realizando el producto por el peso de la resistencia utilizada en el ejercicio como se puede ver en la Ecuación 2.2.

$$F = Ma \quad (N) \quad (2.2)$$

Para calcular la velocidad, al no tener una aceleración constante, debemos realizar la integral aproximada, en nuestro caso mediante la Regla del Trapecio, la cual podemos ver en la Ecuación 2.3.

$$\int_a^b f(x)dx \approx (b-a) \frac{f(a) + f(b)}{2} \quad (2.3)$$

Siendo  $f(a)$  la aceleración en el instante de tiempo actual y  $f(b)$  la aceleración de la muestra anterior, con  $a$  y  $b$  los tiempos de recogida del dato correspondiente. Estos tiempos también son proporcionados por el plugin, en formato epoch.<sup>2</sup>

## 2.5 Detección de repeticiones

Una de las principales funciones de la aplicación es la de detectar y contar las repeticiones completadas por el usuario durante la realización de un ejercicio. Esto permite obtener datos aislados de cada una de las repeticiones además de poder realizar un análisis gráfico para observar la evolución del ejercicio. Se ofrece también la posibilidad de recibir una señal sonora al completar las repeticiones.

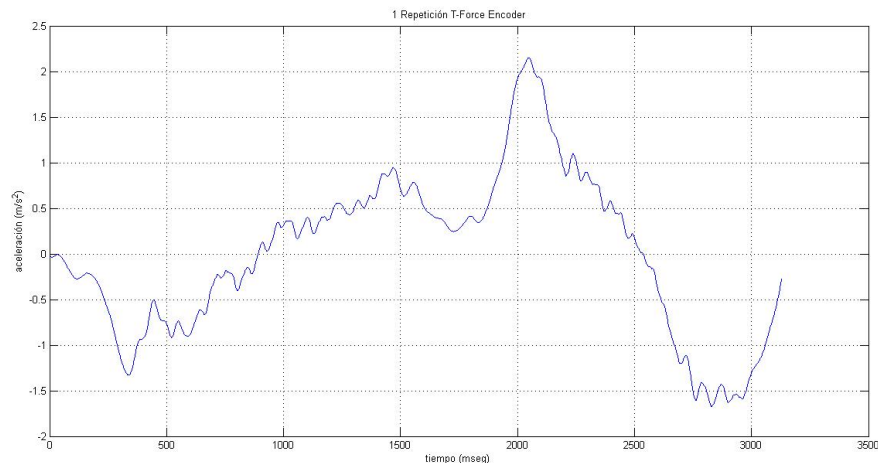


Figura 2.4: Aceleración resultante de una repetición

Para poder detectar repeticiones a partir de la señal de aceleración que obtenemos del dispositivo deberemos realizar a esta un pequeño tratamiento. Lo primero que se debe conocer es la forma de onda que tiene la aceleración durante una repetición, como ejemplo, tenemos la Figura 2.4. En nuestro caso, siempre vamos a partir desde la posición erguida, por lo que el primer movimiento será el de bajada. Consideraremos como nuestro eje de movimiento el eje vertical. Durante la bajada, se produce una aceleración negativa al comenzar el movimiento,

<sup>2</sup>sistema para la descripción de instantes de tiempo: se define como la cantidad de segundos transcurridos desde la medianoche UTC del 1 de enero de 1970, sin contar segundos intercalares.

y una aceleración positiva debida al frenado cuando se llega abajo. En la subida, se observa el pico de aceleración positiva en el impulso inicial, y una aceleración negativa correspondiente al frenado producido por la llegada al final del movimiento.

Para detectar las repeticiones, nos basamos en los pasos por cero o cambios de signo de la señal. Con el fin de evitar el ruido estático que nos encontramos y que puede hacer cambiar de signo la señal en valores muy pequeños, realizamos un primer filtrado con un umbral de aceleración. Toda aceleración positiva menor que  $1 \text{ m/s}^2$  y negativa mayor que  $-1 \text{ m/s}^2$  se aproximará a cero. Posteriormente, la señal resultante se filtra con un filtro paso bajo, con el fin de eliminar valores atípicos producidos por el ruido. A partir de esta señal, y teniendo en cuenta que cada repetición pasa por dos cambios de signo, del negativo al positivo en la bajada y del positivo al negativo en la subida, cuando se detecte el segundo cambio de signo, se determinará que se ha completado una repetición. En la Figura 2.5 observamos el esquema de funcionamiento del sistema de detección de repeticiones.

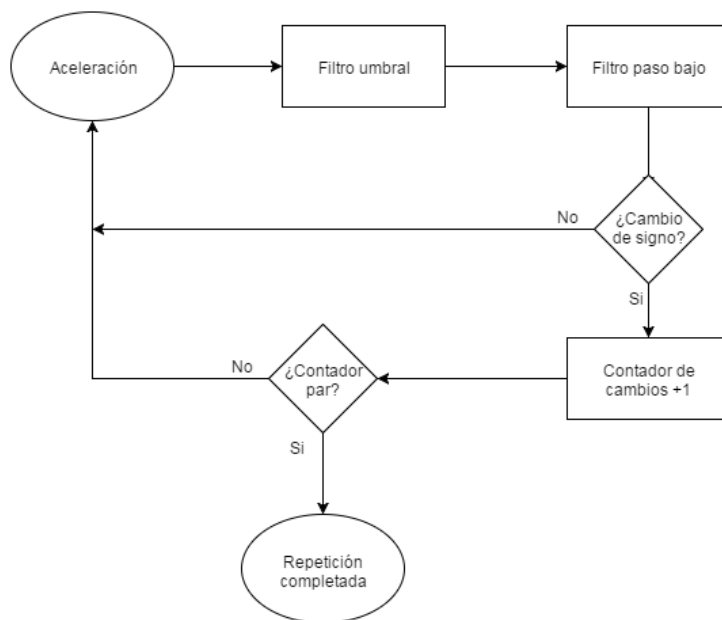


Figura 2.5: Esquema de detección de repeticiones.

En la figura 2.6 podemos ver la señal de aceleración a la salida de los filtros, primero del umbral, y segundo del paso bajo, correspondiente a un ejercicio de tres repeticiones. Como se puede observar en la señal resultante, hay 6 cambios de

signo. Como sabemos, durante una repetición la aceleración sufre dos cambios de signo, de negativo a positivo, y de positivo a negativo, cada vez que contamos dos cambios de signo se valida una repetición. En este caso tenemos hasta tres.

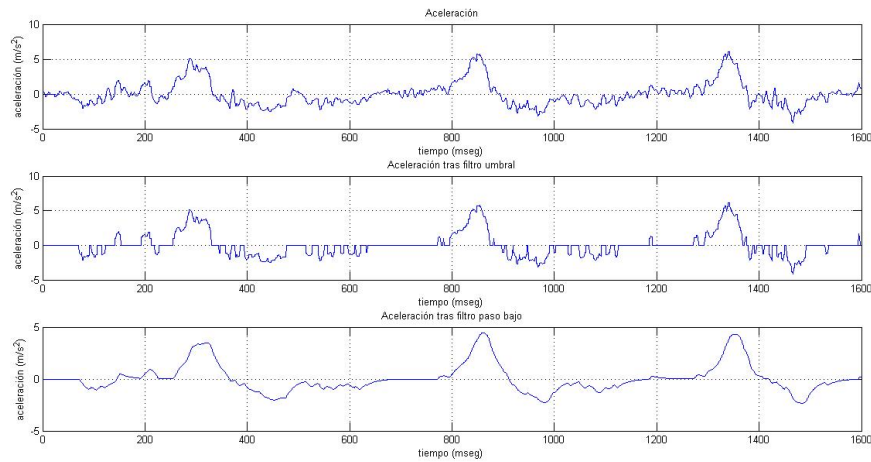


Figura 2.6: Aceleración a su paso por los filtros de umbral y paso bajo.

## 2.6 Velocidad y Potencia

La potencia se define como fuerza por velocidad. A partir del valor de aceleración que obtenemos del dispositivo, y conocida la carga en el ejercicio, obtenemos directamente la fuerza como el producto de las dos (Ecuación 2.2).

Para el cálculo de la velocidad se hace uso de la Regla del Trapecio para aproximar la integral de la aceleración (Ecuación 2.3). Adaptando esta ecuación a nuestro caso, en el que estamos calculando velocidad, debemos actualizar el valor anterior con la llegada de una nueva muestra. El tiempo de muestreo será la diferencia de tiempo en milisegundos entre la llegada de la muestra actual y la llegada de la muestra anterior. El cálculo de la velocidad quedaría como se puede ver en la Ecuación 2.4.

$$V(t) = V(t - 1) + (t_{muestreo}(mseg)) \frac{a_m(t - 1) + a_m(t)}{2} \quad (m/s) \quad (2.4)$$

La señal que se obtiene al aplicar esta ecuación es la representada en la Figura



2.7. Corresponde a un ejercicio en el que se completaron tres repeticiones, que se pueden apreciar tanto en la señal de aceleración como en la de velocidad.

Se observa como los picos de una y otra señal no coinciden exactamente, ya que la máxima velocidad se alcanza justo en el instante antes de empezar a frenar, mientras que valor máximo de aceleración llega en el impulso de comienzo de la subida.

Hay que tener en cuenta, que en los valores de estas dos señales se ha eliminado la componente estática de la gravedad. Para la fuerza y la potencia si se debe tener en cuenta, ya que en ambas influyen aceleración dinámica y estática al intervenir el peso de una resistencia en su cálculo.

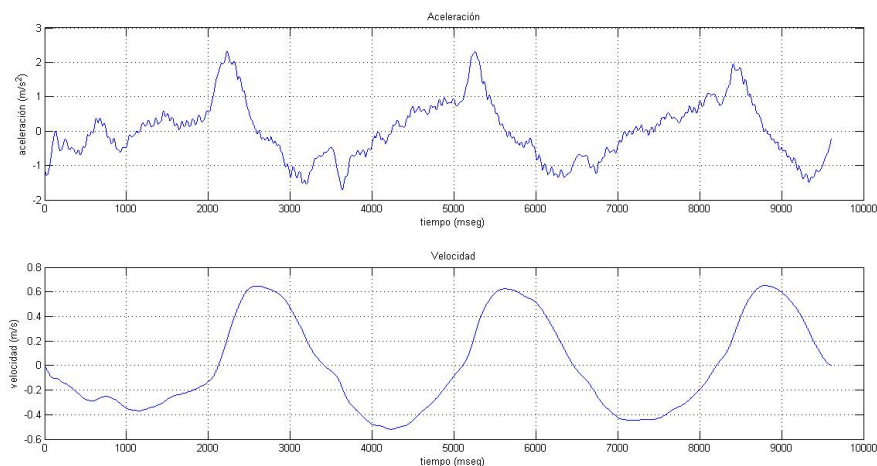


Figura 2.7: Aceleración y velocidad.

Para obtener el valor de la potencia, una vez se tiene velocidad y fuerza, simplemente hay que hacer el producto de estas dos. En la Figura 2.8 podemos ver las señales de fuerza y potencia del mismo ejercicio anterior, en el que se completaron tres repeticiones.

Para el cálculo de la curva de potencia, utilizaremos el valor de pico máximo obtenido en cada uno de los tests que se realicen. Normalmente este se dará en la primera repetición, puesto que la tendencia de la velocidad es ir disminuyendo con cada repetición adicional debido a la fatiga.

En la evaluación del entrenamiento, el valor de interés será el valor de pico de cada una de las repeticiones. Es este valor el que se comparará con el

correspondiente de su curva de potencia.

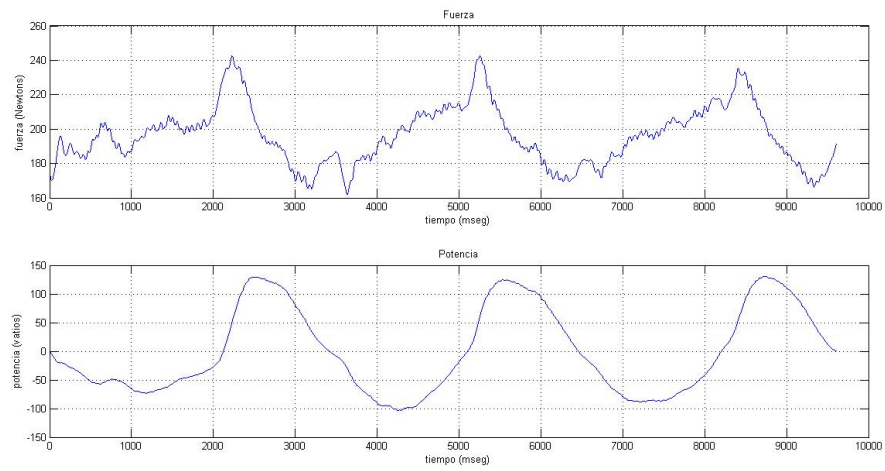


Figura 2.8: Fuerza y Potencia.



# Capítulo 3

## Desarrollo de la Aplicación

El funcionamiento básico de la aplicación es el que se ilustra en la Figura 3.1. Las opciones que encuentra el usuario desde la página de inicio, son consulta del historial, calibrado, entrenamiento o acceso a los datos del perfil.

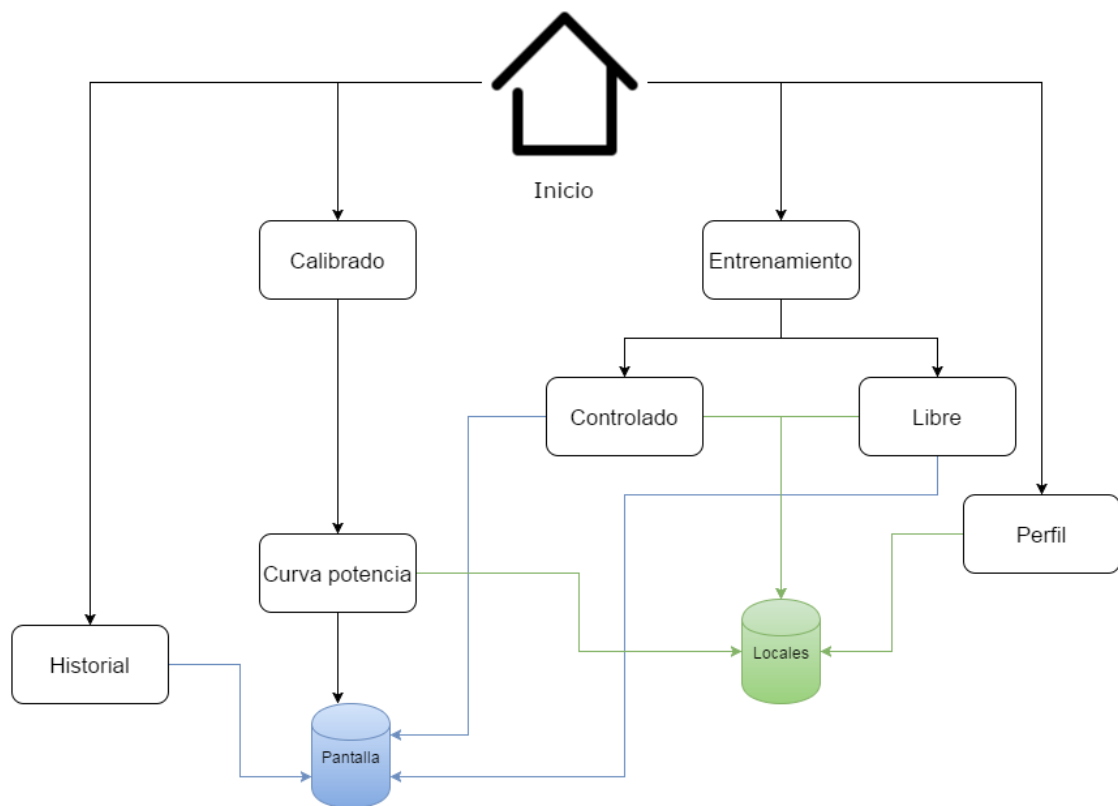


Figura 3.1: Esquema de la aplicación.

Para mostrar de manera sencilla el funcionamiento del código, este se ha

representado mediante un diagrama UML, Figura 3.2. Hay que tener en cuenta, que estamos trabajando con Phonegap, programando código web, y por tanto no orientado a objetos, lo que quiere decir que no hay clases como tales. En el diagrama se han presentado las principales funciones implicadas y su relación entre ellas.

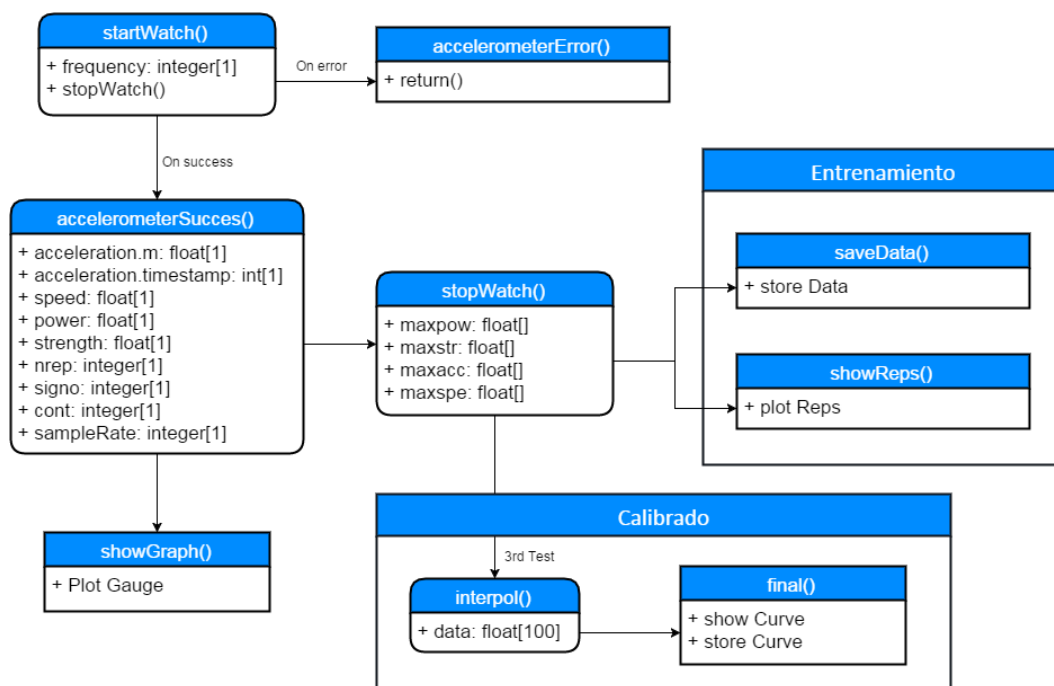


Figura 3.2: Diagrama UML de la aplicación.

## 3.1 Calibrado

En este apartado, se trata de generar y almacenar la curva de potencia del usuario para el ejercicio seleccionado. El esquema principal se ilustra en la Figura 3.3. El usuario deberá elegir el ejercicio que quiere calibrar e introducir el valor de su 1RM, que deberá haber sido calculado con anterioridad. Una vez se tienen estos datos, se procederá a realizar tres pruebas o test. En cada una ellas se cargará el 40 %, 60 % y 80 % de 1RM respectivamente. Se han elegido esos valores ya que normalmente el valor de pico máximo de potencia se consigue en este rango. Con estos tres valores ya podemos tener una idea de la tendencia de la curva, que completaremos con una interpolación, aunque sería interesante en un futuro desarrollo permitir la incorporación de más valores. Se realizará el

ejercicio seleccionado, subiendo lo más explosivamente posible un mínimo de dos repeticiones.

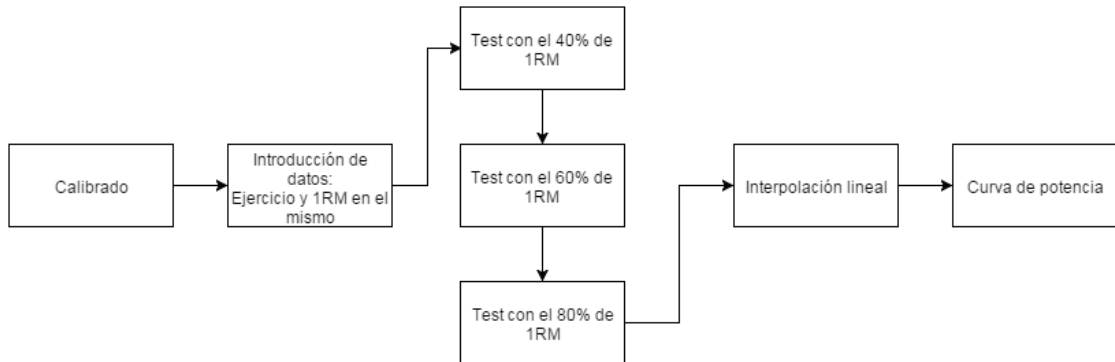


Figura 3.3: Esquema básico de Calibrado.

Al finalizar cada una de las tres pruebas, la aplicación guardará el mayor valor de potencia registrado en cada una de ellas. Una vez que se hayan completado las tres, se procederá a la generación de la curva de potencia mediante una interpolación lineal de los datos obtenidos.

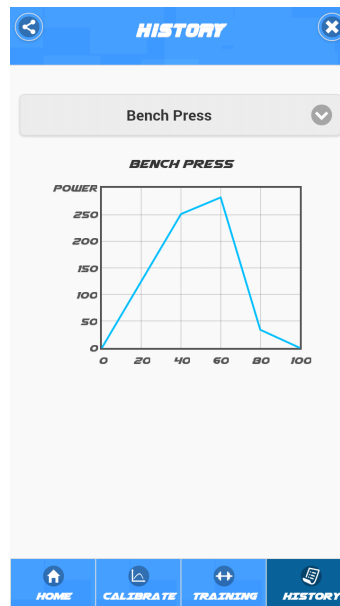


Figura 3.4: Curva de Potencia.

En la Figura 3.4 observamos la curva de potencia para el ejercicio press de banca generada por la aplicación. Esta tiene un rango de 0 a 100, donde el 100 se corresponde con el valor de 1RM. Normalmente esta curva adopta la forma de

la función gaussiana. Donde mayor potencia es capaz de desarrollar el deportista es en los valores intermedios de resistencia. Con valores más altos, el movimiento es más lento, por lo que la potencia disminuye. Con valores bajos de resistencia, aunque la velocidad de movimiento es mayor, al bajar el peso de la resistencia la potencia también lo hace.

En el código 3.1, se presenta una parte de código Javascript, encargado de generar los datos de fuerza y potencia a partir de la aceleración, y de obtener los valores de pico de estas dos variables.

```
function accelerometerSuccess(acceleration) {  
  if (contInicio > 200){ //Tiempo que tarda en inicializarse  
    accelerationD = acceleration.m - meanGrav; //modulo  
      de la aceleración menos la componente gravedad  
    acc.push(accelerationD);acc.shift();//almacenamiento  
      de los dos últimos valores en un vector de dos  
      componentes.  
5    timeAux = timeCal;  
    timeCal = acceleration.timestamp;  
    sampleRate = timeCal - timeAux; //tasa de muestreo  
      entre las dos últimas muestras  
    if (timeAux == 0) {sampleRate = 0;}  
    speed = speed + sampleRate*(acc[0] + acc[1])  
      /(2*1000); //cálculo de la velocidad  
10    strength = Math.abs((accelerationD + 9.8) * res);//  
      cálculo de la fuerza  
    power = strength * speed; //cálculo de la potencia  
    pm.push(power);fm.push(strength);//almacenamiento  
      de potencia y fuerza en sendos vectores  
  }  
  
  else {if (contInicio == 200){ //Cuando se han  
    completado 200 muestras calcula la media  
15    meanGrav = mean(meanAcc); //que será la gravedad  
      a restar y suena el aviso para  
    inicioSonar()  
    contInicio++; //comenzar el ejercicio
```

```
    }  
    else {meanAcc.push(acceleration.m);  
    contInicio++;  
    }  
  }  
}
```

Código 3.1: Código que calcula potencia y valores de pico en la calibración.

## 3.2 Entrenamiento

Es la parte que permite al usuario obtener los valores de potencia y fuerza durante el ejercicio. Se ha separado en dos apartados, entrenamiento controlado y entrenamiento libre.

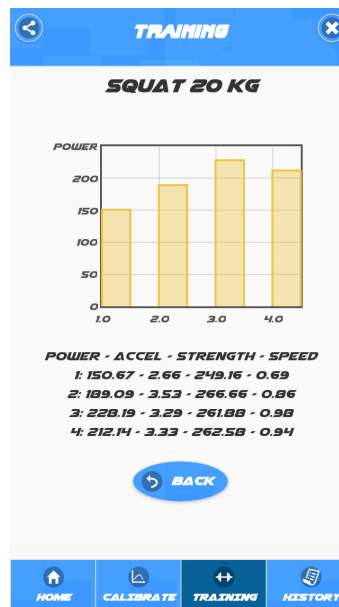


Figura 3.5: Entrenamiento Sentadilla.

En la Figura 3.5 se puede ver un resultado de un entrenamiento dentro de la aplicación. Como se aprecia, viene dado mediante un gráfico de barras. En este caso, se han completado cuatro repeticiones. Debajo del gráfico, se muestran los datos de cada una de estas repeticiones, en concreto, los valores máximos de potencia, aceleración, fuerza y velocidad.



En el código 3.2 podemos ver una parte del código correspondiente a la función encargada de extraer la aceleración del dispositivo y su transformación a potencia, a la vez que realiza la detección de repeticiones:

```
function accelerometerSuccess(acceleration) {  
    if (contInicio > 200) { //tiempo de inicialización  
        accTotal = acceleration.m - meanGrav; //elimina la  
            comp gravedad  
        if ((accTotal <1) && (accTotal>-1)) {accTotalReps =  
            0;} //paso por umbral  
5        else {accTotalReps = accTotal;}  
        timeAux = time1;  
        time1 = acceleration.timestamp;  
        sampleRate = time1 - timeAux; //tasa de muestreo  
        if (timeAux == 0) {sampleRate = 0;}  
10        acc.push(accTotal);acc.shift();  
        speed = speed + (sampleRate)*(acc[0] + acc[1])  
            /(2*1000); //cálculo de velocidad  
        accAcum = accAcum + (accTotalReps - accAcum)/20;  
            //filtro paso bajo  
        if (signo == 0 && sign(parseInt(accAcum)) != 0){  
            signo = sign(parseInt(accAcum));};  
        if (sign(parseInt(accAcum)) != 0){ //detector de  
            cambios de signo  
15        if (sign(parseInt(accAcum)) != signo) {  
            signo = sign(parseInt(accAcum));  
            if (cont%2 == 0){ //repeticio completada  
                sonar(); //sonido aviso  
                nrep += 1; //suma una repeticion  
20                repsdata = [nrep,Math.max.apply(null,repow)  
                    ]; //calculo potencia max  
                repow = [];  
            }  
            cont++;  
        }  
25    }  
}
```

```
        mod = Math.abs((accTotal + 9.8)*weight); //cálculo
            fuerza
        mod2 = mod*speed; //cálculo potencia
        repow.push(mod2); //Acumulación de potencia en
            vector
    }
30     else {if (contInicio == 200){ //fin inicialización
        meanGrav = mean(meanAcc); //Gravedad media
        inicioSonar()
        contInicio++; //comenzar el ejercicio
    }
35     else {meanAcc.push(acceleration.m);
        contInicio++;
    }
    }
}
```

Código 3.2: Código que genera la potencia y detección de repeticiones en el entrenamiento

### 3.2.1 Entrenamiento controlado

En este apartado, se hará uso de la curva de potencia obtenida en la parte de calibración. Con ella se conocen los valores máximos de potencia que el usuario es capaz de desarrollar en el ejercicio con cualquier carga. Esto se utilizará con el fin de mantener un nivel aceptable de potencia durante el entrenamiento. Para ello se deberá seleccionar un umbral, en porcentaje de la 1RM, que se considerará como el límite por encima del cual se aceptará la repetición como válida.

Se brindará la opción de entrenar con sonido, pudiendo recibir el deportista una señal sonora en el momento de completar cada repetición, diferente en función de si dicha repetición se considera válida o no, en función del umbral previamente establecido. Durante el entrenamiento, en la pantalla del teléfono se mostrará un velocímetro con la representación de la potencia a tiempo real tal y como aparece

en la Figura 3.6. Una vez finalizado el mismo, se presentará mediante un gráfico de barras el resultado. Cada una de estas barras representa a las distintas repeticiones.

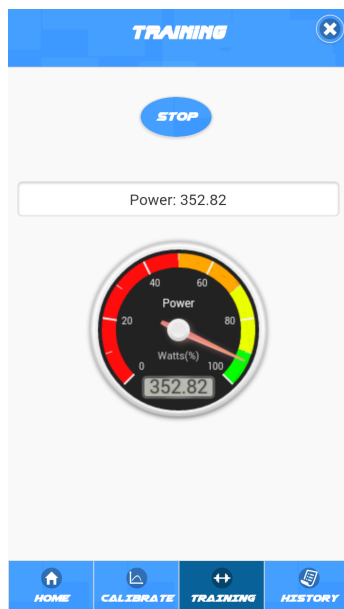


Figura 3.6: Gráfica que expresa la potencia durante el entrenamiento.

### 3.2.2 Entrenamiento libre

En esta opción, el desarrollo del ejercicio será libre, sin hacer uso de la curva de potencia. El usuario deberá elegir el ejercicio y el peso con que se va a trabajar y podrá activar la opción de sonido, que marcará con una señal sonora el conteo de las repeticiones. Mientras se realiza el ejercicio se mostrará por pantalla el mismo velocímetro marcando la potencia actual. Una vez terminado este se representará en forma de gráfico de barras, con la opción de almacenarlo en la memoria del dispositivo.

## 3.3 Historial

Los archivos de entrenamientos guardados, así como la curva de potencia para cada ejercicio y los datos del perfil del usuario se almacenarán con formato .xml. En el siguiente código tenemos un ejemplo de un archivo xml:

```
<User Name="Carlos">
  <Exercise Weight="20">
    <Squat>
5      <Rep: 1 Power="187.45" Acceleration="3.66" Strength="
        269.19" Speed="0.80" />
      <Rep: 2 Power="215.70" Acceleration="3.22" Strength="
        260.33" Speed="0.91" />
      <Rep: 3 Power="256.57" Acceleration="3.52" Strength="
        266.47" Speed="1.05" />
    </Squat>
  </Exercise>
10 </User>
```

Código 3.3: Ejemplo de archivo XML. Ejercicio con 3 repeticiones.

Estos archivos se almacenan en la raíz del dispositivo, creando un sistema de carpetas. En la carpeta inicial de la aplicación, se guardarán los datos del perfil del usuario y se creará una carpeta propia para el mismo. En el interior de esta, se alojará cada una de las curvas de potencia de los dos ejercicios y una subcarpeta que contendrá los ejercicios que se deseen salvar.

## 3.4 Validación básica de los datos

En esta sección se aborda en mayor profundidad la validación de los datos obtenidos por la aplicación, en concreto los valores de aceleración, fuerza, velocidad y potencia. Para ello se dispone en el laboratorio un codificador lineal de los que ya se ha hablado en el apartado 1.3.3. Se han realizado pruebas con dos teléfonos móviles sujetos a la barra, a la vez que el codificador lineal realiza la toma de datos para un ejercicio de sentadilla en el que se completaron tres repeticiones. El móvil 1 corresponde a un Jiayu G4S, mientras que el móvil 2 es un Vodafone Smart Prime 6 (Alcatel).

Lo primero que hay que destacar, es la frecuencia de muestreo, que en el

codificador lineal alcanza los 1000Hz, es decir, se actualiza la muestra cada milisegundo. En los móviles empleados en el estudio, la frecuencia máxima media se encuentra en torno a los 100Hz, tomando muestras cada 10 milisegundos. Este valor, varía de un dispositivo a otro en función de las especificaciones de su acelerómetro.

La resolución de los acelerómetros que llevan los teléfonos es bastante alta. En el caso del móvil 1, cuyo acelerómetro es el modelo de 3 ejes BMA056 de Bosch, esta es de  $0.0039 \text{ m/s}^2$ .

El problema es que el valor de la aceleración es extremadamente inestable, y en ocasiones deja de corresponderse con la realidad. Esto se debe a que los acelerómetros en general son muy ruidosos, efecto que se agrava teniendo en cuenta el ruido eléctrico y mecánico que genera el resto de componentes del teléfono. Por ello, resulta inevitable realizar un filtrado paso bajo de los datos.

En las Figuras 3.7 y 3.8 se pueden observar las señales de aceleración, velocidad, fuerza y potencia del móvil 1 y el encoder. Se aprecia también el índice de correlación entre las señales, que ha sido calculado como se indica en la Ecuación 3.1, siendo  $\sigma_{xy}$  la covarianza entre las dos señales y  $\sigma_x$  y  $\sigma_y$  la desviación típica de cada una de ellas y que se calculan tal y como se indica en las Ecuaciones 3.2 y 3.3.

$$R = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \quad (3.1)$$

$$\sigma_{xy} = \frac{\sum (x_i y_i)}{N} - \bar{x} \bar{y} \quad (3.2)$$

$$\sigma_x = \sqrt{\sum \frac{x_i^2}{N} - \bar{x}^2} \quad (3.3)$$

El índice de correlación entre dos señales está entre -1 y 1, siendo -1 correlación inversa, 0 ausencia de correlación y 1 correlación total. En nuestro caso, observamos como en todo momento este se encuentra por encima de 0.9. En el caso de la velocidad y la potencia, la correlación está muy próxima a 1, lo que indica una

alta correlación.

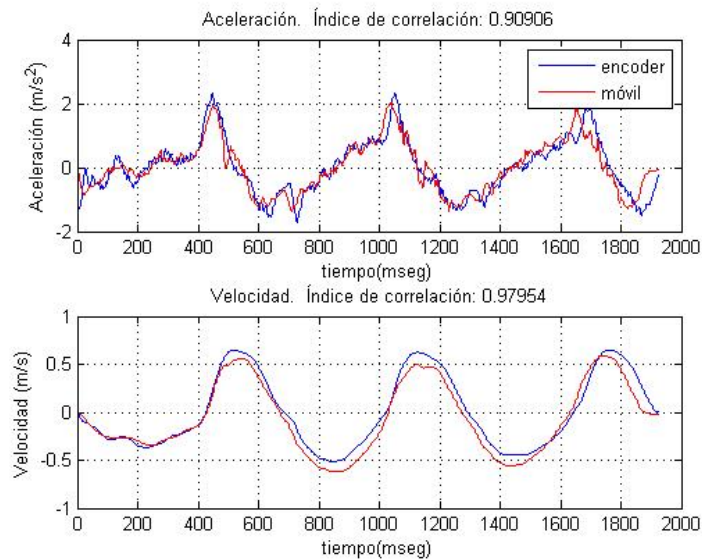


Figura 3.7: Aceleración, velocidad y correlación con móvil 1.

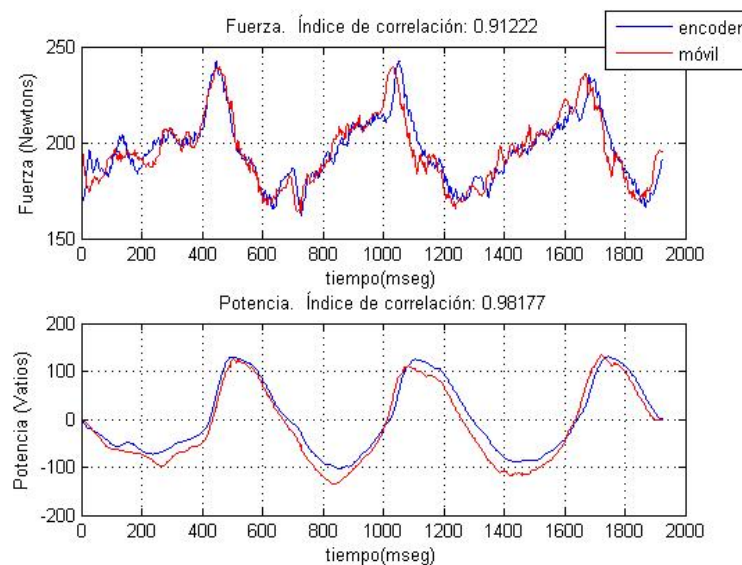


Figura 3.8: Fuerza, potencia y correlación con móvil 1

En las figuras 3.9 y 3.10 observamos las mismas gráficas anteriores correspondientes a aceleración, velocidad, fuerza y potencia, pero esta vez utilizando el móvil 2. Los resultados obtenidos son similares a los anteriores. Como podemos ver, las señales del móvil y el encoder tienen una forma muy similar. El índice de correlación en todas ellas se encuentra por encima de 0.9, en concreto

en potencia y velocidad por encima de 0.98, lo que se traduce en una correlación muy alta.

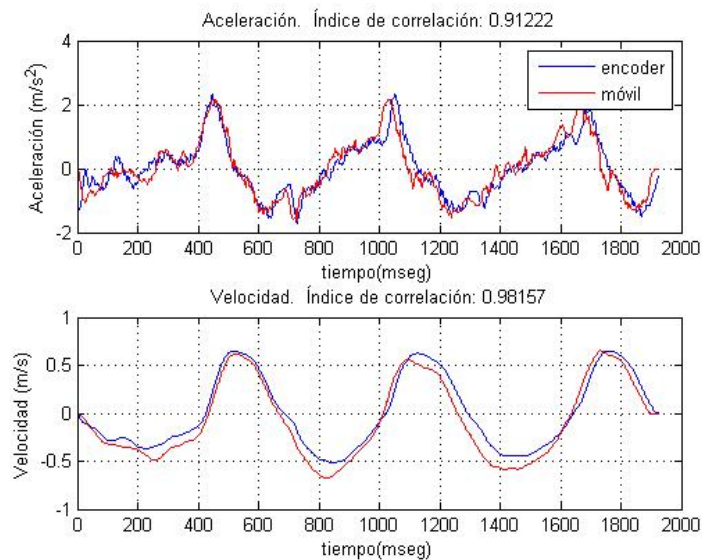


Figura 3.9: Aceleración, velocidad y correlación con móvil 2.

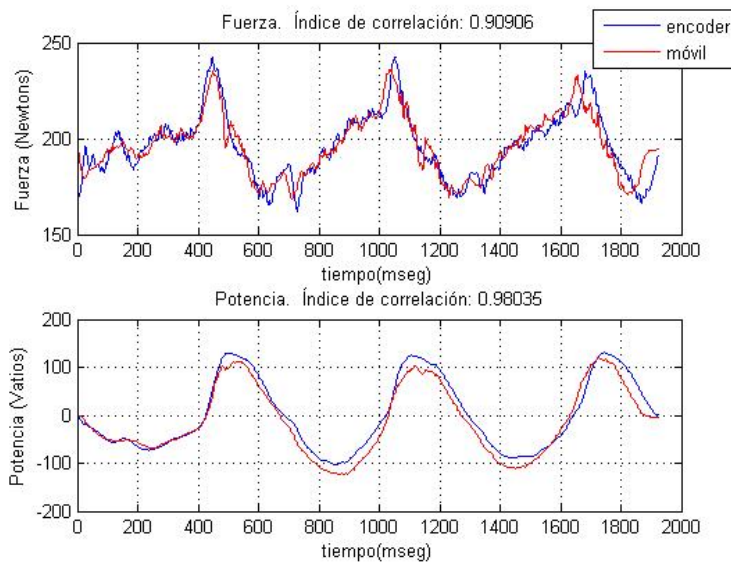


Figura 3.10: Fuerza, Potencia y correlación con móvil 2.

Como se puede observar comparando las señales, las muestras no acaban de corresponder del todo. Esto se produce debido a que el sincronismo a la hora de iniciar ambos sistemas (móvil y codificador) no es el ideal. Adicionalmente, las frecuencias de muestreo son diferentes para cada uno. Por tanto, ha sido necesario

el diezmado de la señal con mayor frecuencia y la interpolación de la de menor para poder realizar esta comparación.

Otro de los problemas presentados durante la fase de pruebas, fue la eliminación del factor de la gravedad a la hora de calcular la velocidad. Al ser un valor inestable, y que parece variar en función del dispositivo. La solución que se propuso y corroboró con buenos resultados fue la de dejar unos segundos el móvil en la posición de partida a modo de inicialización, donde se consultan los datos de aceleración, se hace una media y se sustrae el valor resultante, que corresponde con el valor estático de la gravedad.

Como conclusión, se puede ver que el sistema no obtiene exactamente los mismos valores que el sistema de referencia pero sí muy próximos, con un elevado índice de correlación. Queda claro que la finalidad del acelerómetro en los teléfonos móviles no es la de registrar movimientos pequeños con precisión, estos fueron introducidos como medida de seguridad ante posibles caídas (Cuando detecta una aceleración muy brusca, el teléfono se apaga). Un acelerómetro con mayor potencia y tasa de muestreo y mejor aislamiento respecto al ruido aportaría unos resultados mejores.

Destacar que las pruebas realizadas han sido muy básicas, únicamente se ha probado con dos móviles Android de gama media, y en pocos ejercicios. Así mismo, para dar completa validez al sistema, haría falta un banco de pruebas más complejo, con más dispositivos y un mayor número de ejercicios observados, situación que se deja para futuros estudios o desarrollos.





# Capítulo 4

## Conclusiones y líneas futuras

### 4.1 Conclusiones

La realización de este Proyecto Fin de Carrera ha supuesto una profundización y comprensión de la programación de aplicaciones móviles. Al haber desarrollado el código con Phonegap, ha habido una parte muy extensa en cuanto al aprendizaje de los lenguajes HTML, Javascript y CSS y su ensamblado, además de todo el entorno de desarrollo Android, así como usos y aplicaciones del acelerómetro. Se ha estudiado también parte de los principios, fundamentos y objetivos del entrenamiento de la fuerza.

Como resultado, se ha alcanzado el objetivo principal del proyecto, el cual era el desarrollo de una herramienta de utilidad para deportistas y entrenadores, capaz de medir valores de potencia durante la realización de ejercicios con pesas. Esta herramienta se ha implementado como una aplicación para Android de uso sencillo y con una interfaz atractiva y nada compleja.

La aplicación ha sido probada y comparada con los datos que ofrece un codificador lineal. A pesar de que las pruebas han sido limitadas y haría falta una validación mayor de los datos, los resultados iniciales parecen afirmar que se puede utilizar este tipo de herramientas para entrenamientos profesionales.

Como comentario personal, afirmar que la experiencia ha sido muy positiva, intensa debida a la gran cantidad de horas dedicadas pero muy beneficiosa, en

cuanto a todo lo aprendido durante estos meses y al resultado obtenido. Desde el comienzo ha sido un trabajo con motivación al ser un tema interesante para mí, y finalmente el resultado resulta satisfactorio.

## 4.2 Desarrollos futuros

A continuación se presentan una serie de mejoras o desarrollos del trabajo realizado, por resultar interesantes de cara al futuro.

- Preparar un banco de pruebas más completo realizando un mayor número de pruebas y con más móviles para llevar a cabo una evaluación más exhaustiva.
- Complementar el acelerómetro con el uso de otros sensores como el giroscopio o el magnetoscopio, de cara a poder realizar el ejercicio con pesos libres, y de movimientos en distintos planos.
- Generar aplicación cliente servidor, para poder ofrecer a un entrenador la sincronización con los datos de entrenamiento de todos sus deportistas de manera sencilla.
- Permitir la compartición de resultados y entrenamientos en las principales redes sociales.
- Desarrollar el código necesario para poder utilizar la aplicación en el resto de dispositivos con diferentes sistemas operativos tales como iOS, Windows Phone, etc.
- Integrar la aplicación en nuevos dispositivos como por ejemplo pulseras o relojes inteligentes, que evitarían el uso de adaptadores y soportes para colocar el móvil en la barra.

## 4.3 Cronograma de Tareas

La distribución del tiempo de cada tarea del proyecto se muestra en el diagrama de Gantt que aparece en la Figura 4.1. La primera de estas tareas fue la

acotación temática, para delimitar los objetivos, la manera de desarrollo y sobre que plataformas se debía implementar. La siguiente tarea, la de aprendizaje, ha sido la mas extensa de todas, ya que ha sido necesario obtener los conocimientos suficientes de los diferentes lenguajes de programación (HTML, CSS, Javascript), así como comprender el desarrollo para sistemas Android y profundizar en los conceptos básicos del entrenamiento de fuerza.

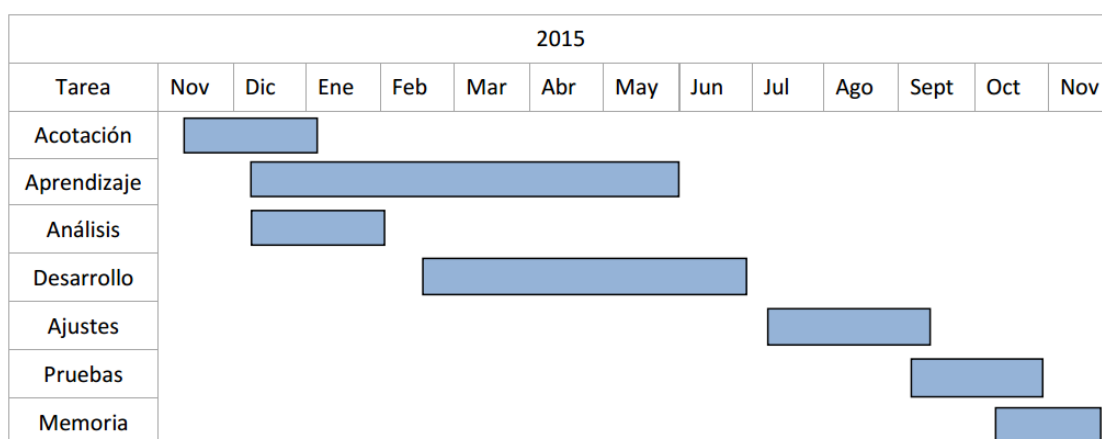


Figura 4.1: Diagrama de Gantt.

La tarea de análisis se realizó para observar la problemática del proyecto y como abordarla, así como afrontar el diseño y la orientación del proyecto.

En el desarrollo se implementó el código básico de la aplicación, a medida que se iban haciendo pruebas sobre un emulador.

En la tarea de ajustes se completaron los detalles y deficiencias del código. Posteriormente, se procedió a la fase de pruebas en laboratorio y validación de los datos obtenidos por la aplicación. Finalmente, se ha realizado la redacción de la memoria del proyecto.



# Bibliografía

- [1] G. GOLDSPIK, *Strength and Power in sport*, Blackwell Scientific Publication, London, 1992.
- [2] JUAN JOSÉ GONZÁLEZ BADILLO y ESTEBAN GOROSTIAGA AYESTARÁN, *Fundamentos del entrenamiento de la fuerza. Aplicación al alto rendimiento deportivo*, Capítulos 1 y 5, 2002.
- [3] JUAN JOSÉ GONZÁLEZ BADILLO y MIKEL IZQUIERDO REDÍN, *La carga de entrenamiento y el rendimiento en fuerza y potencia muscular*, Málaga, 2006.
- [4] FERNANDO J. NACLERIO, ALFONSO JIMÉNEZ, BRENT A. ALVAR, y MARK D. PETERSON, *Assessing strength and power in resistance training*, Journal of Human Sport and Exercise, Universidad de Alicante, 2009
- [5] GABOR PALLER, *Advantages and limitations of PhoneGap for sensor processing*, 2013.



# Anexo A

## Acrónimos

- **API:** Application Programming Interface.
- **HTML:** HyperText Markup Language.
- **JSON:** JavaScript Object Notation.
- **XML:** eXtensible Markup Language.
- **APP:** Application.
- **iOS:** iPhone Operating System.
- **API:** Application Programming Interface.
- **CSS:** Cascading Style Sheets.
- **UML:** Unified Modeling Language.
- **1RM:** One Repetition Maximum.





# Anexo B

## Ejecución de la aplicación

Para hacer posible la ejecución de nuestro código HTML, debemos crear un fichero en lenguaje nativo, en este caso Java, que informe al dispositivo desde donde debe arrancar la aplicación. En este caso se hará desde la página index.html.

Además, se da la instrucción de utilizar una pantalla de carga al iniciar y el tiempo que debe durar esta.

```
package com.phonegap.appower;

5 import android.os.Bundle;
import org.apache.cordova.*;

public class CordovaApp extends CordovaActivity
{
10     @Override
    public void onCreate(Bundle savedInstanceState)
    {

        super.setIntegerProperty("splashscreen", R.drawable.
            splash);
15     super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.html",
            1500);
    }
}
```

```
}
```

Código B.1: Código para ejecutar la aplicación desde nuestro código HTML.

# Anexo C

## Guía de uso

A continuación se explica de forma detallada y visual el funcionamiento de la aplicación, mostrando capturas de pantalla en cada uno de los pasos.

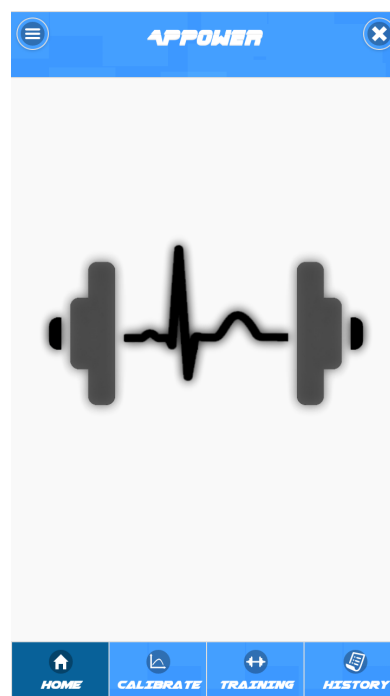


Figura C.1: Pantalla de inicio.

En la Figura C.1 se puede observar la pantalla de inicio de la aplicación. En la barra de navegación que hay en la parte superior, el usuario puede elegir entre tres tareas. Calibrado, entrenamiento y consulta del historial.

## C.1 Calibrado

En esta sección, la primera página es como la que aparece en la Figura C.2. El usuario debe seleccionar el ejercicio que quiere calibrar e introducir el valor de su 1RM para dicho ejercicio.

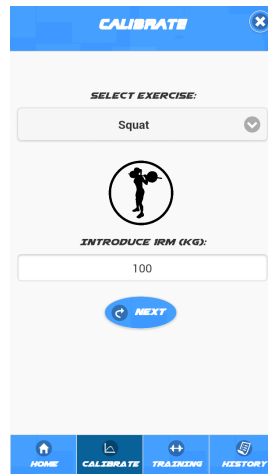
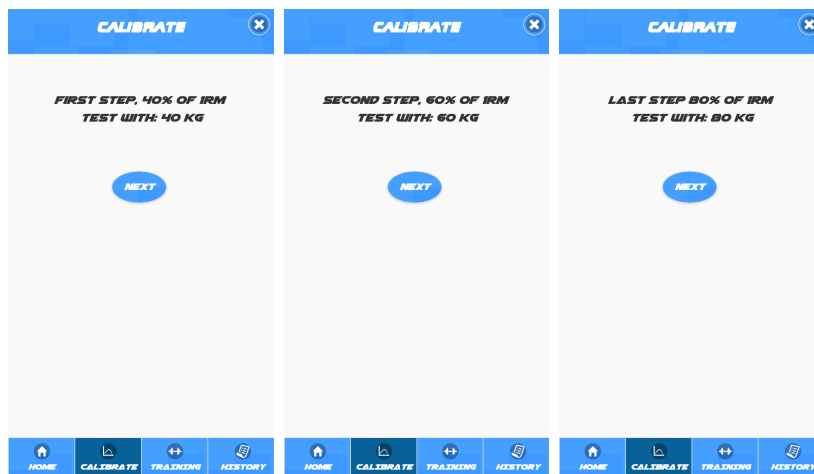


Figura C.2: Pantalla inicial de Calibrado.



(a) Prueba con 40 %

(b) Prueba con 60 %

(c) Prueba con 80 %

Figura C.3: Pasos para completar el Calibrado.

A continuación, tendremos que completar tres pruebas, con el 40, 60 y 80 por ciento de la 1RM introducida respectivamente. Pasamos por tres pantallas, Figura C.3, donde se indica el peso correspondiente que hay que cargar en cada prueba.

## C.2 Entrenamiento

Si se escoge la opción de entrenamiento, nos encontraremos con una pantalla para seleccionar si se quiere que este sea controlado o libre, como la de la Figura C.4.

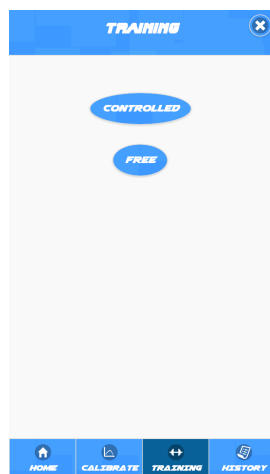
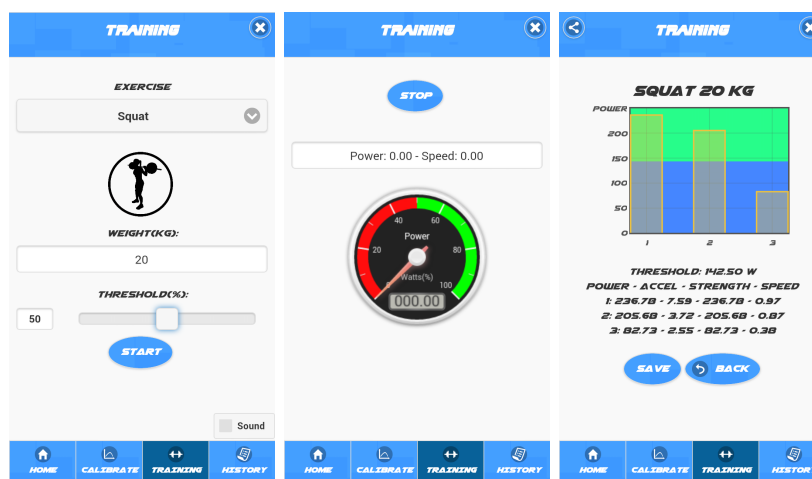


Figura C.4: Pantalla de selección de entrenamiento

### C.2.1 Entrenamiento controlado



(a) Selección

(b) Ejecución

(c) Resultados

Figura C.5: Pantallas Entrenamiento Controlado

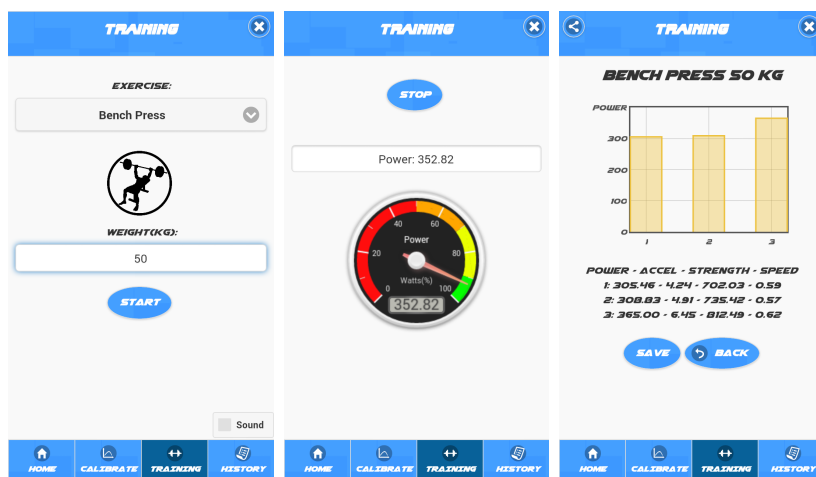
La característica de la parte de entrenamiento controlado es la opción de

imponer un umbral, para determinar si la potencia que se está desarrollando durante el ejercicio es la prevista. En la Figura C.5.a se muestra la pantalla de selección de ejercicio y peso con el que se va a realizar. También se deberá introducir el valor del umbral en porcentaje, en este caso se ha elegido 50. Si se activa la pestaña de sonido, se recibirá un sonido al completar cada repetición. Sonido de error, si está por debajo del umbral, o sonido normal en caso de encontrarse por encima.

En la Figura C.5.b vemos una captura de pantalla durante la ejecución del ejercicio. En el velocímetro se aprecian los dos colores en los rangos desde cero hasta el valor umbral y desde el umbral hasta 100.

En la última imagen, Figura C.5.c se presenta por pantalla el resultado del ejercicio, un gráfico de barras donde cada una de estas representa una repetición sobre un fondo diferenciado por el valor del umbral. En el caso de la imagen, las dos primeras repeticiones estarían dentro de este umbral, mientras que la última no ha alcanzado el valor mínimo de potencia.

## C.2.2 Entrenamiento libre



(a) Selección

(b) Ejecución

(c) Resultados

Figura C.6: Pantallas Entrenamiento Libre

Este apartado es muy similar al anterior, lo único que cambia es que no hay

posibilidad de utilizar un valor umbral de control de potencia. La opción de sonido activa las señales sonoras en cada repetición, a modo de ayuda para el conteo de las mismas. En la Figura C.6 vemos las tres pantallas por las que se pasa en esta sección, correspondiendo a un ejercicio en el cual se han completado tres repeticiones.

### C.3 Historial

El usuario puede acceder y ver entrenamientos pasados así como consultar sus actuales curvas de potencia en sentadillas y press de banca. Aparecerá una pantalla de selección como la mostrada en la Figura C.7.

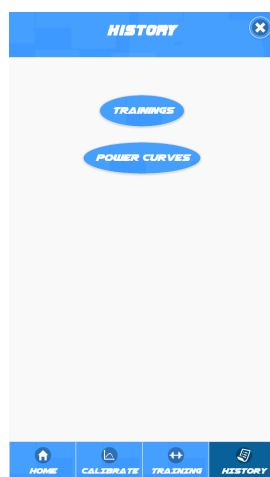


Figura C.7: Acceso al historial.

Si se elige la primera opción, consulta de entrenamientos, se accederá a una pantalla donde aparece una lista con todos los entrenamientos que se han guardado en el dispositivo (Figura C.8.a). Pulsando en cualquiera de estos, accederemos a la siguiente pantalla (Figura C.8.b), donde se mostrará el detalle del entrenamiento, desglosado por repeticiones. Se ofrece la posibilidad de borrar el ejercicio guardado.

Si el usuario accede al historial de curvas de potencia, verá una pantalla como la que aparece en la Figura C.9. En ella, podrá elegirse el ejercicio para el cual se quiere ver la curva de potencia.



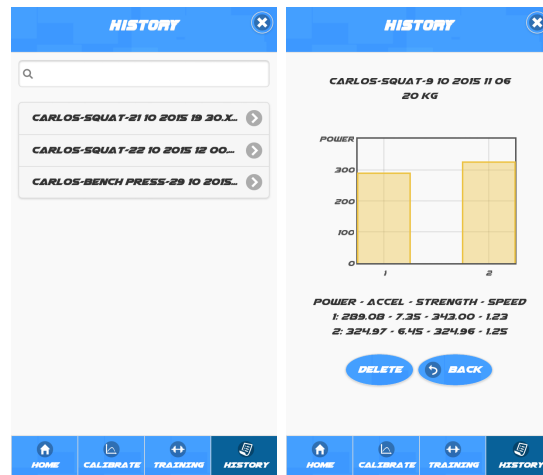


Figura C.8: Consulta de entrenamientos pasados



Figura C.9: Consulta de Curva de potencia.

# Anexo D

## Funciones más importantes

Como el código de la aplicación es muy extenso, en este anexo se mostrarán solo aquellas funciones que se consideran más relevantes. Si se quiere consultar el código completo, se puede hacer a través del siguiente enlace: [https://www.dropbox.com/sh/gklmtwuvcbw3g34/AAB0fSYsUi-Fi\\_MQc-BSEqYfa?dl=0](https://www.dropbox.com/sh/gklmtwuvcbw3g34/AAB0fSYsUi-Fi_MQc-BSEqYfa?dl=0)

### D.1 Acceso y tratamiento de la señal de aceleración

En el siguiente código, se pueden ver las principales funciones de acceso al acelerómetro, y cálculo de los valores de potencia y fuerza en la calibración.

```
function startWatch() {  
    var freq = { frequency: 1 }; //frecuencia de  
    //actualización cada mseg  
    sensorAcc = navigator.accelerometer.watchAcceleration  
        (accelerometerSuccess, accelerometerError, freq);  
}  
5  
  
function accelerometerError() { //función de error  
    alert('Failed to load accelerometer!');  
}  
  
10 function accelerometerSuccess(acceleration) {
```

```
if (contInicio > 200){ //inicialización
    accelerationD = acceleration.m - meanGrav; //
        aceleración resultante
    acc.push(accelerationD);acc.shift();
    timeAux = timeCal;
15    timeCal = acceleration.timestamp;
    sampleRate = timeCal - timeAux; //tasa de muestreo
    if (timeAux == 0) {sampleRate = 0;}
    speed = speed + sampleRate*(acc[0] + acc[1])
        /(2*1000); //cálculo de la velocidad
    strength = Math.abs((accelerationD + 9.8) * res); //
        cálculo fuerza
20    power = strength * speed; //cálculo potencia
    pm.push(power);fm.push(strength);
    document.getElementById("Pm").value= accelerationD.
        toFixed(2) + "-" + speed.toFixed(2);
    gauge.setValue(power); //Actualización del
        velocímetro
    gauge.draw();
25 }
    else {if (contInicio == 200){
        meanGrav = mean(meanAcc);
        inicioSonar()
        contInicio++;
30    }
        else {meanAcc.push(acceleration.m);
            contInicio++;
        }
    }
35 }

function mean(numbers) { //función que calcula la media
    de un vector
    var total = 0;
    for (i = 0; i < numbers.length; i += 1) {
40        total += numbers[i];
    }
}
```

```
    return total / numbers.length;
}

45
function stopWatch() {
document.getElementById("AccBut").disabled = false;
cont = 0; timeCal = 0; speed = 0; contInicio = 0;
if (sensorAcc) {
50   navigator.accelerometer.clearWatch(sensorAcc); //deja
        de mirar el acelerómetro
        sensorAcc = null;
        aux = (Math.max.apply(null, pm));
        maxpot = "Max Power: " + aux.toFixed(3); //cálculo
        de los valores máximos de fuerza y potencia
        document.getElementById("Pm").value= maxpot;
55   aux2 = (Math.max.apply(null, fm));
        maxstr = "Max Strength: " + aux2.toFixed(3);
        fm = []; pm = []; speed=0; acc=[0,0]; meanAcc = [];
        meanGrav = 0;
        xml()
        }
60 }
}
```

Código D.1: Código con las funciones principales de la calibración

Para la parte de entrenamiento, las funciones cambian un poco, tal y como podemos ver en el siguiente código.

```
function startWAcc() {
    var options = { frequency: 1 };
    sensorAcc = navigator.accelerometer.
        watchAcceleration(accelerometerWSuccess,
            accelerometerError, options);
}

5

function accelerometerError() { //función de error
    alert('Failed to load accelerometer!');
}
```

```

    }

10  function accelerometerSuccess(acceleration) { //función
        encargada de calcular la potencia y detectar reps en
        entrenamiento
    if (contInicio > 200) { //tiempo de inicialización
        accTotal = acceleration.m - meanGrav; //elimina la
            comp gravedad
        if ((accTotal <1) && (accTotal>-1)) {accTotalReps =
            0;} //paso por umbral
        else {accTotalReps = accTotal;}
15     timeAux = time1;
        time1 = acceleration.timestamp;
        sampleRate = time1 - timeAux; //tasa de muestreo
        if (timeAux == 0) {sampleRate = 0;}
        acc.push(accTotal);acc.shift();
20     speed = speed + (sampleRate)*(acc[0] + acc[1])
            /(2*1000); //cálculo de velocidad
        accAcum = accAcum + (accTotalReps - accAcum)/20;
            //filtro paso bajo
        if (signo == 0 && sign(parseInt(accAcum)) != 0){
            signo = sign(parseInt(accAcum));};
        if (sign(parseInt(accAcum)) != 0){ //detector de
            cambios de signo
25     if (sign(parseInt(accAcum)) != signo) {
            signo = sign(parseInt(accAcum));
            if (cont%2 == 0){ //repeticio completada
                sonar(); //sonido aviso
                nrep += 1; //suma una repeticion
                repsdata = [nrep,Math.max.apply(null,repow)
                    ]; //calculo potencia max
30     repow = [];
            }
            cont++;
        }
    }
35     mod = Math.abs((accTotal + 9.8)*weight); //cálculo

```

```
        fuerza
        mod2 = mod*speed; //cálculo potencia
        repow.push(mod2); //Acumulación de potencia en
        vector
        document.getElementById("InputTest").value = "Power
        : " + mod2;
        gauge.setValue(mod2); //actualización del
        velocímetro
40    gauge.draw();
    }

    else {if (contInicio == 200){ //Cuando se han
        completado 200 muestras calcula la media
        meanGrav = mean(meanAcc); //que será la gravedad
        a restar y suena el aviso para
        inicioSonar()
45    contInicio++; //comenzar el ejercicio
    }
    else {meanAcc.push(acceleration.m);
        contInicio++;
    }
50    }
}

function stopWAcc() {

55    speed = 0; accAcum = 0; nrep = 0; cont = 1; signo = 0;
        time1 = 0;
    if (sensorAcc) {
        navigator.accelerometer.clearWatch(sensorAcc); //
        deja de mirar el acelerómetro
        sensorAcc = null;
        speed = 0;
60    pow = [];str = [];spee = [];accel = [];speed=0;acc
        =[0,0];accAux =[];
    };
}
}
```

---

Código D.2: Código con las funciones principales del entrenamiento

## D.2 Guardar archivo XML

Todos los archivos almacenados de forma local en el dispositivo lo hacen con la extensión `.xml`. Para su generación en javascript, se ha utilizado la librería *Marknote*. Además, para poder almacenar y consultar archivos de manera local, es necesaria la utilización de un plugin específico. En el código que sigue a continuación, vemos un ejemplo de generación y posterior salvado en el directorio local de un entrenamiento.

```
function endTrainFree() {
    createFileTrainingFree();
    window.requestFileSystem(LocalFileSystem.PERSISTENT,
        0, gotEndFree, fail); //Intenta acceder al sistema
        de archivos locales
}

5
function gotEndFree(fileSystem) {
    var dir = "APPower/" + usFree + "/training";           //
        Directorio en el que vamos a guardar
    fileSystem.root.getDirectory(dir.toString(), {create:
        true}, gotEndDirFree, fail);
}

10
function gotEndDirFree(dirEntry) {
    var d = new Date();           //Cálculo de la fecha
        para añadir al nombre
    var temp = d.getMinutes(); if(temp<10){temp = "0"+temp;};
    var aux = d.getDate() + " " + (d.getMonth() + 1) + " " + d.
        getFullYear() + " " + d.getHours() + " " + temp;
15
    dirEntry.getFile(usFree + "-" + exerciseFree + "-" +
        aux.toString() + ".xml",{create:true},
        gotEndFileFree);
```

```
}

function gotEndFileFree(fileEntry) {
    fileEntry.createWriter(gotFileWriterEndFree, fail);
}

function gotFileWriterEndFree(writer) {
    writer.onwrite = function(evt) {
        console.log("Se escribió satisfactoriamente");
    };
    writer.write(docTrain.toString());
}

function createFileTrainingFree() {
    docTrain = new marknote.Document();           //creación
        del archivo xml
    rootElement = new marknote.Element("User");    //
        Elemento raíz, usuario
    rootElement.setAttribute("Name", usFree);      //
        Atributo del elemento raíz
    docTrain.setRootElement(rootElement);
    EjElement = new marknote.Element("Exercise"); //
        Nuevo elemento, ejercicio
    EjElement.setAttribute("Weight", weight);     //
        Atributo de ejercicio, peso
    rootElement.addChildElement(EjElement);       //Se
        añade ejercicio como hijo de la raíz
    ExElement = new marknote.Element(exerciseFree); //
        Nuevo elemento, tipo de ejercicio
    EjElement.addChildElement(ExElement);        //Se hace
        hijo del elemento anterior
    var aux = auxDataB.length;
    for(var i = 1; i <= aux; i++){
        TrainD = new marknote.Element("Rep: " + i); //Se
            añaden elementos hijo tantos como repeticiones
        TrainD.setAttribute("Power", auxDataB[i-1][1].toFixed
            (2)); //Atributos de repetición: pot, fuerz, acel.y
```



```
        vel.  
        TrainD.setAttribute("Acceleration", auxData2B[i-1][1].  
            toFixed(2));  
        TrainD.setAttribute("Strength", auxData3B[i-1][1].  
            toFixed(2));  
45    TrainD.setAttribute("Speed", auxData4B[i-1][1].toFixed  
            (2));  
        ExElement.addChildElement(TrainD);  
    }  
}
```

Código D.3: Código que genera un fichero XML y lo salva.

## D.3 Media

Cada vez que iniciamos un ejercicio, o al completar una repetición, siempre y cuando se haya activado la opción de sonido, la aplicación generará un sonido a modo de aviso. Para ello, ha sido necesaria la utilización de un plugin para el acceso a los recursos multimedia del dispositivo. A continuación, podemos ver dos funciones que lanzan esas señales sonoras.

```
function inicioSonar() {  
    media = new Media("/android_asset/www/medias/beep2.ogg");  
    //sonido de inicio de ejercicio  
    media.play();  
5 }  
  
function sonar() {  
    media2 = new Media("/android_asset/www/medias/beep.ogg");  
    //sonido de repetición completada  
    if (sound == "Yes"){  
10    media2.play();};  
}
```

Código D.4: Código encargado de generar avisos sonoros.

## D.4 Interpolación

El código siguiente es el encargado de generar la curva completa de potencia a partir de los datos obtenidos en los tres tests realizando una interpolación lineal.

```
function interpol(points){  
    var x = new Array();  
    x[0] = 0;  
    for(i = 1; i<40; i++){  
5        x[i] = [0 + ((i - 0)*(points[0] - 0))/(40 - 0)];}  
    x[40] = [points[0]];  
    for(i = 41; i<60; i++){  
        x[i] = [ (points[0] + ((i - 40)*(points[1] - points  
            [0])))/(60 - 40)];}  
    x[60] = [ points[1]];  
10    for(i = 61; i<80; i++){  
        x[i] = [ (points[1] + ((i - 60)*(points[2] - points  
            [1])))/(80 - 60)];}  
    x[80] = [points[2]];  
    for(i = 81; i<100; i++){  
        x[i] = [ (points[2] + ((i - 80)*(0 - points[2]))  
            /(100 - 80)];}  
15    x[100] = 0;  
    return x;  
}
```

Código D.5: Código que realiza la interpolación lineal para generar la curva de potencia.

## D.5 Gráficos

En la aplicación aparecen gráficos de la curva de potencia, de cada entrenamiento y durante el entrenamiento a modo de velocímetro. Para su representación se han utilizado las librerías `graphs.js` y `gauge.js`. A continuación se ilustra una parte del código correspondiente a la inicialización de estos gráficos.

```
//Velocímetro
function graph(){ //gráfico velocímetro
  gauge = new Gauge({
    renderTo      : 'myCanvas', //valor donde dibuja en la hoja
                    HTML
5    width        : 200,      //tamaño del gráfico
    height        : 200,
    glow          : true,
    units         : 'Watts(%)', //unidades, en este caso
                    potencia
10    title        : 'Power',
    minValue      : 0,
    maxValue      : 100,
    majorTicks    : ['0','20','40','60','80','100'],
    minorTicks    : 2,
    strokeTicks   : false,
15    highlights  : [ //colores en el gráfico
      { from : 0, to : 50, color : '#FF0D0D' },
      { from : 50, to : 70, color : '#FFA600' },
      { from : 70, to : 90, color : '#EBFF00' },
      { from : 90, to : 100, color : '#04FF00' },
20    ],
    colors        : {
      plate        : '#222',
      majorTicks  : '#f5f5f5',
      minorTicks  : '#ddd',
25    title        : '#fff',
      units        : '#ccc',
      numbers     : '#eee',
      needle      : { start : 'rgba(240, 128, 128, 1)', end :
                    'rgba(255, 160, 122, .9)' }
    }
30  });
  gauge.setValue( 0 );
  gauge.draw();
}
```

```
35 //Curva de Potencia
    function curve(points){ //función que convierte un
        //vector unidimensional en bidimensional para su
        //representación
        var x = new Array();
        for(i = 0; i<101; i++){
            x[i] = [i, points[i]];
40 return x;
    }

    var options = { //opciones de inicio del gráfico
        series: { shadowSize: 0, color:'DeepSkyBlue' },
45 yaxis: { min: 0, color:'white'},
        xaxis: { show: true,}
    };

    var x = curve(interpol(points)); //interpolación y
    //posterior conversión a 2dimensiones
50 plot = $.plot($("#chartdiv"), [x], options); //
    //dibujo de la curva de potencia

    //Entrenamiento
    function showReps() {
        setTimeout(function(){ //se aplica un delay de 0.5 seg
            //para que cargue correctamente
55 var xChartAx = new Array();
            for (var i=0; i<auxData.length; i++){
                xChartAx[i] = [i+1, i+1]; //eje x del gráfico
            };

60 var options2 = { //opciones de gráfico, en este caso
            //es de barras
            series: { shadowSize: 0, bars: { show: true,
                barWidth: 0.5, align:"center"}},
            yaxis: { tickFormatter: function(val, axis) {
                return val < axis.max ? val.toFixed(0) : "Power"
```

```
        ;}},  
        xaxis: { ticks: xChartAx},  
    };  
65 plot = $.plot($("#chartLol"), [auxData], options2); //  
    dibuja el gráfico  
    },500);  
}
```

Código D.6: Código correspondiente a los gráficos de la aplicación