

Proyecto Fin de Carrera: Ingeniería en Informática
Programación de aplicaciones para móviles utilizando la plataforma PhoneGap



Seguimiento Deportivo
Versión 1.0

Autor: Félix Chaves Gallego
Dirigido por Carlos Sánchez Rosa

Fecha: 11 de Junio 2012

Este PFC, la documentación y el código fuente de la aplicación que ha sido desarrollada queda licenciado bajo la licencia: **Reconocimiento-NoComercial-SinObraDerivada 3.0 España (CC BY-NC-ND 3.0)**.

Usted es libre de:

copiar, distribuir y comunicar públicamente la obra

Bajo las condiciones siguientes:



Reconocimiento — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).



No comercial — No puede utilizar esta obra para fines comerciales.



Sin obras derivadas — No se puede alterar, transformar o generar una obra derivada a partir de esta obra.

Ilustración A: Licencia

Más información en: <http://creativecommons.org/licenses/by-nc-nd/3.0/es/>

A mi abuela Ángeles, que el destino ha decidido que ya no esté con nosotros, DEP.

A mis padres Félix y Ángeles por inculcarme, apoyarme y ayudarme desde pequeño, para que tenga una carrera y con esta van dos.

A mi mujer Patricia, por ayudarme y apoyarme en este ciclo de mi vida, sin cuyo apoyo no hubiera sido posible.



1 Resumen

El presente Proyecto Final de Carrera tiene como objetivo la realización del diseño e implementación de un sistema de seguimiento deportivo para una organización tal que su fin sea hacer seguimiento a personas que deseen mejorar su calidad de vida, para ello un entrenador personal le pondrá un plan de entreno y una dieta equilibrada.

El sistema ha de permitir recoger información de dichos entrenos como ¿cuánto tiempo está caminando, corriendo o moviéndose en bicicleta?, ¿qué distancia ha recorrido?, ¿a qué velocidad media? y ¿cuándo se realiza dicho entreno?. De ésta forma, la organización sabrá perfectamente si la realización de un entreno ha sido beneficioso para la salud de la persona y dispondrá durante su desarrollo de los datos indicados anteriormente, previamente al comienzo del plan de entreno se talla, se pesa y se mide el diámetro de la cintura de cada persona. Una vez terminado el plan de entreno, se volverá a tomar esas mismas medidas, para observar la evolución.

Para alcanzar este objetivo, hemos considerado lo siguiente:

- Utilizar la plataforma "*PhoneGap*", con la que realizamos la aplicación para móviles, con la ventaja que es multiplataforma.
- Utilizamos la API "*Geolocation*", la cual se encarga de la implementación de geolocalización en PhoneGap.
- Utilizamos la API "*Storage*" para diseñar la base de datos (Diagrama E/R), hacer el script de creación de las tablas, índices, etc. necesarios e implementar los procedimientos almacenados que se requieren para que las aplicaciones los utilicen.
- Implementar un mecanismo de inicialización de la base de datos.

Esta memoria pretende detallar todas las fases por las que ha atravesado el proyecto, contando desde las circunstancias de su génesis y la metodología de trabajo empleada, pasando por el análisis de requerimientos y el diseño de la aplicación, hasta llegar a la implementación final. En el proceso, se intentará proporcionar una visión general de las tecnologías y conceptos empleados, así como de las expectativas iniciales y las metas finalmente alcanzadas.



Índice

1.1 Índice de Contenidos

1 Resumen.....	4
Índice.....	5
1.1 Índice de Contenidos.....	5
1.2 Índice de Figuras.....	6
1.3 Índice de Tablas.....	7
2 Introducción.....	8
2.1 Justificación del PFC y contexto en el que se desarrolla.....	9
2.2 Objetivos del PFC.....	9
2.3 Enfoque y método seguido.....	10
2.4 Planificación del proyecto.....	11
Planificación real.....	12
2.5 Productos obtenidos.....	13
2.6 Estructura de la memoria.....	13
3 Análisis.....	14
3.1 Requerimientos funcionales.....	14
3.1.1 Sistema principal.....	14
3.2 Requerimientos no funcionales.....	15
3.2.1 Requisitos de usuario.....	15
3.2.2 Requisitos tecnológicos.....	15
3.2.3 Requisitos de desarrollo y restricciones de diseño.....	16
3.2.4 Requisitos de rendimiento.....	16
3.3 Casos de uso y diagramas de secuencia.....	16
3.3.1 Descripción de los actores.....	16
3.3.2 Diagramas de casos de uso.....	17
3.3.3 Diagramas de secuencia.....	21
4 Diseño y Arquitectura.....	22
4.1 Modelo conceptual.....	22
4.2 Arquitectura del sistema.....	24
4.3 Modelos de clases del diseño.....	25
4.3.1 Diagrama de clases.....	25
4.4 Diseño de la Base de datos para la aplicación.....	27
4.4.1 Descripción de la tabla SESION.....	27
4.5 Diseño de la Interfaz gráfica.....	28
5 Implementación.....	31
5.1 Introducción.....	31
5.2 Script Cronómetro.....	32
5.3 Script Geolocalización (GPS).....	34
5.4 Script Cálculo de distancia.....	36
5.5 Script Paso de Variables entre formularios método GET.....	37
5.6 Script Gestión de la Base de Datos.....	38
5.7 Script Notificaciones.....	40
5.8 Pruebas.....	40
5.8.1 Plan de Pruebas.....	40
6 Conclusiones.....	41
7 Glosario.....	42
8 Bibliografía.....	44
9 Anexos.....	46
9.1 Manual de instalación del Entorno de desarrollo.....	46
9.2 Manual de usuario de Seguimiento Deportivo.....	55



1.2 Índice de Figuras

Ilustración A: Licencia.....	2
Ilustración B: Planificación PFC.....	12
Ilustración C: Diagrama de casos de uso.....	17
Ilustración D: Diagrama de secuencia.....	21
Ilustración E: Modelo conceptual.....	22
Ilustración F: Modelo 3 capas.....	24
Ilustración G: Diagrama de clases.....	25
Ilustración H: Pantalla "Inicio".....	28
Ilustración I: Pantalla "Sesión".....	29
Ilustración J: Pantalla "Listado Sesiones".....	29
Ilustración K: Pantalla "Detalle Sesión".....	30
Ilustración L: JDK-6u30.....	46
Ilustración M: Eclipse Classic 3.7.2.....	47
Ilustración N: Android SDK.....	47
Ilustración O: Descarga de plataformas Android SDK.....	48
Ilustración P: Install New Software.....	48
Ilustración Q: Add.....	48
Ilustración R: URL Plugin.....	49
Ilustración S: Marcar check Developer Tools.....	49
Ilustración T: Instalador plugin de Eclipse.....	49
Ilustración U: Aceptar y Finalizar.....	50
Ilustración V: Reiniciar Eclipse.....	50
Ilustración W: Manager AVD.....	50
Ilustración X: New AVD.....	51
Ilustración Y: Creación de un AVD.....	51
Ilustración Z: Iniciar un AVD.....	52
Ilustración AA: PhoneGap.....	52
Ilustración BB: Install New Software.....	53
Ilustración CC: Add.....	53
Ilustración DD: URL Plugin.....	53
Ilustración EE: Marcar check MDS AppLaud.....	54
Ilustración FF: Instalador plugin de Eclipse.....	54
Ilustración GG: Reiniciar Eclipse.....	54
Ilustración HH: Pantalla Inicio.....	55
Ilustración II: Pantalla Sesión.....	56
Ilustración JJ: Sincronizando GPS.....	57
Ilustración KK: GPS Sincronizado.....	57
Ilustración LL: Sesión Iniciada.....	58
Ilustración MM: Sesión Pausada.....	58
Ilustración NN: Sesión Continuada.....	59
Ilustración OO: Sesión Finalizada.....	59
Ilustración PP: Listado de Sesiones.....	60
Ilustración QQ: Detalle Sesión.....	61
Ilustración RR: Mensaje de Confirmación de Borrado.....	61
Ilustración SS: Mensaje de Confirmación de Borrado de Todas las sesiones.....	62



1.3 Índice de Tablas

Tabla 1: Tabla SESION.....	27
Tabla 2: Tabla Descripción de la Sesión.....	27



2 Introducción

En estos últimos años la telefonía móvil ha evolucionado mucho. De los móviles pequeños y simples, con los que sólo podíamos hacer llamadas y enviar SMS, hemos pasado a aparatos complejos y con grandes pantallas táctiles, los cuales se pueden considerar ordenadores pequeños o de bolsillo. Estos aparatos tienen una gran capacidad de interacción e incluso ha tenido que crear sistemas operativos aptos para ellos. Uno de estos sistemas operativos es el llamado Android y es uno de los que más ha conseguido destacar en estos últimos años. Su evolución ha sido muy rápida y en muy poco tiempo se ha puesto líder en la lucha por ser el mejor sistema operativo.



2.1 *Justificación del PFC y contexto en el que se desarrolla*

La motivación principal en la realización de este PFC era diseñar y programar una aplicación útil que pudiera servir a otras personas y no algo que quedara en un cajón y que nadie utilizara jamás. Dando vueltas a esta idea se me ocurrió realizar una aplicación para registrar distancias recorridas en un cierto tiempo desde un móvil con GPS, que podría ser útil para cualquier persona, no solo para deportistas, sino personas que quieran mejorar su calidad de vida, que mejor manera que controlarlo y registrarlo a través del propio teléfono móvil que hoy día casi todo el mundo disponemos de uno con nuevas tecnologías.

Llevar un control exhaustivo, de la distancia recorrida, el tiempo que se emplea para ello, la velocidad media, la velocidad instantánea, la altitud, la fecha y hora a la que se lleva a cabo, resulta complicado.

Durante años, desarrollar un software que cumpliera esta tarea ha sido un objetivo personal, promovido por la afición al running, que por una u otra circunstancia, nunca ha podido convertirse en realidad hasta que este Proyecto ha propiciado la justificación perfecta.

Para ello usaremos el entorno PhoneGap, aportando como gran ventaja permitir desarrollar código que podrá ser ejecutado en cualquier plataforma, basándose en estándares HTML5 y CSS y JavaScript. Todo teléfono móvil o dispositivo que acepte un navegador puede ejecutar las aplicaciones hechas con PhoneGap.

PhoneGap es un estándar basado en código abierto que permite el desarrollo de aplicaciones para móviles compatibles con iPhone/iPad, Google Android, Palm, Symbian, BlackBerry, Windows Mobile y más.

2.2 *Objetivos del PFC*

El objetivo del presente Proyecto Fin de Carrera consiste en la realización del diseño e implementación de una aplicación de seguimiento deportivo para dispositivos móviles, que utiliza hardware específico del dispositivo, en nuestro caso el GPS.

Consciente de las restricciones de tiempo a que estará sometido, el proyecto se plantea la consecución de unos objetivos mínimos o principales, manteniendo siempre en mente la posibilidad de ampliaciones futuras de funcionalidades que incluso puedan extender el alcance del software, pudiendo abarcar otros ámbitos como el estadístico. Obviamente, estas posibles funcionalidades extendidas y ampliaciones se quedan fuera del alcance de este Proyecto.

Llegados a este punto los objetivos principales o mínimos que se han planteado a la hora de afrontar este proyecto han sido principalmente cuatro:

- Deberá de ser capaz de sincronizar el sistema GPS del dispositivo móvil con dicha aplicación.
- Deberá de ser capaz de medir, mediante el sistema GPS, la distancia recorrida, desde que se inicia la sesión deportiva hasta que se termina.



- Deberá de ser capaz de medir el tiempo desde que se inicia la sesión deportiva hasta que se termina.
- Deberá de ser capaz de calcular e indicar la velocidad media de la sesión deportiva.

Otros objetivos secundarios han sido:

El logro de estas funcionalidades básicas significará la realización satisfactoria de las expectativas del Proyecto. Adicionalmente y sujetos a las restricciones de tiempo, se plantean objetivos secundarios como por ejemplo:

- Ser capaz de registrar los datos de la sesión.
- Mostrar un resumen de la sesión.

De ésta forma, tendrá un seguimiento deportivo del usuario, observando su evolución, pudiendo en función de los resultados aplicar correcciones en la planificación de los entrenos para que sean más eficientes.

2.3 Enfoque y método seguido

La realización de este proyecto ha seguido la metodología del modelo clásico de desarrollo en cascada (Waterfall).

Las fases por las que se ha pasado son:

Planificación PFC (PAC1). Esta tarea corresponde al Plan de Trabajo. Siguiendo las metodologías de planificación aprendidas, se determinarán las tareas necesarias para alcanzar los objetivos del proyecto y se estimará el esfuerzo y los recursos necesarios para llevarlas a cabo.

Análisis y Diseño del Sistema (PAC2). Esta tarea comprende las siguientes actividades:

- Una revisión de los requerimientos del PFC, para evaluar si son suficientes para comenzar el diseño e identificar los puntos abiertos que requieran una clarificación.
- La elaboración de un diseño de la aplicación por medio de diagramas UML.
- Paralelamente a estas tareas de diseño, se llevará a cabo la instalación y configuración de las herramientas necesarias para el desarrollo del proyecto, de modo que esté disponible al inicio del desarrollo.
- Una vez finalizado el análisis, diseño y las pruebas de entorno, se realizará la documentación.
- Finalmente, se realizará la entrega prevista por la evaluación continua.

Desarrollo Aplicación Seguimiento Deportivo (PAC3). Esta tarea comprende las siguientes actividades:



- La construcción de las clases identificadas para la aplicación.
- La realización de las pruebas unitarias de las clases construidas.
- La realización de las pruebas integradas de funcionamiento de la aplicación una vez probadas las clases individuales.
- Una vez finalizado el desarrollo, se realizará una revisión del código desarrollado y de la documentación para verificar que son completos, homogéneos y consistentes.
- Finalmente, se realizará la entrega prevista por la evaluación continua.

Elaboración Memoria y Presentación PFC (PAC4). Esta tarea recoge la realización y entrega de los documentos finales de este Proyecto Final de Carrera.

2.4 Planificación del proyecto

La planificación original ha sufrido numerosos desvíos por diferentes factores: acontecimientos personales, problemas técnicos y de otros factores han hecho que no se pudiera ajustar a lo que se planteaba en el cronograma inicial. El diagrama siguiente muestra una aproximación bastante cercana a la evolución real del proyecto, aunque en algunos momentos las diferentes fases se han cruzado y intercalado debido a revisiones y replanteamientos no previstos.

En cualquier caso, la fase más larga de todo el proceso ha sido la de Desarrollo de la aplicación. Esta incluye las fases de codificación/aprendizaje y pruebas unitarias/integradas que se han necesitado para efectuarla.



Planificación real

Con la ayuda del MS-Project, la planificación real ha sido la siguiente:

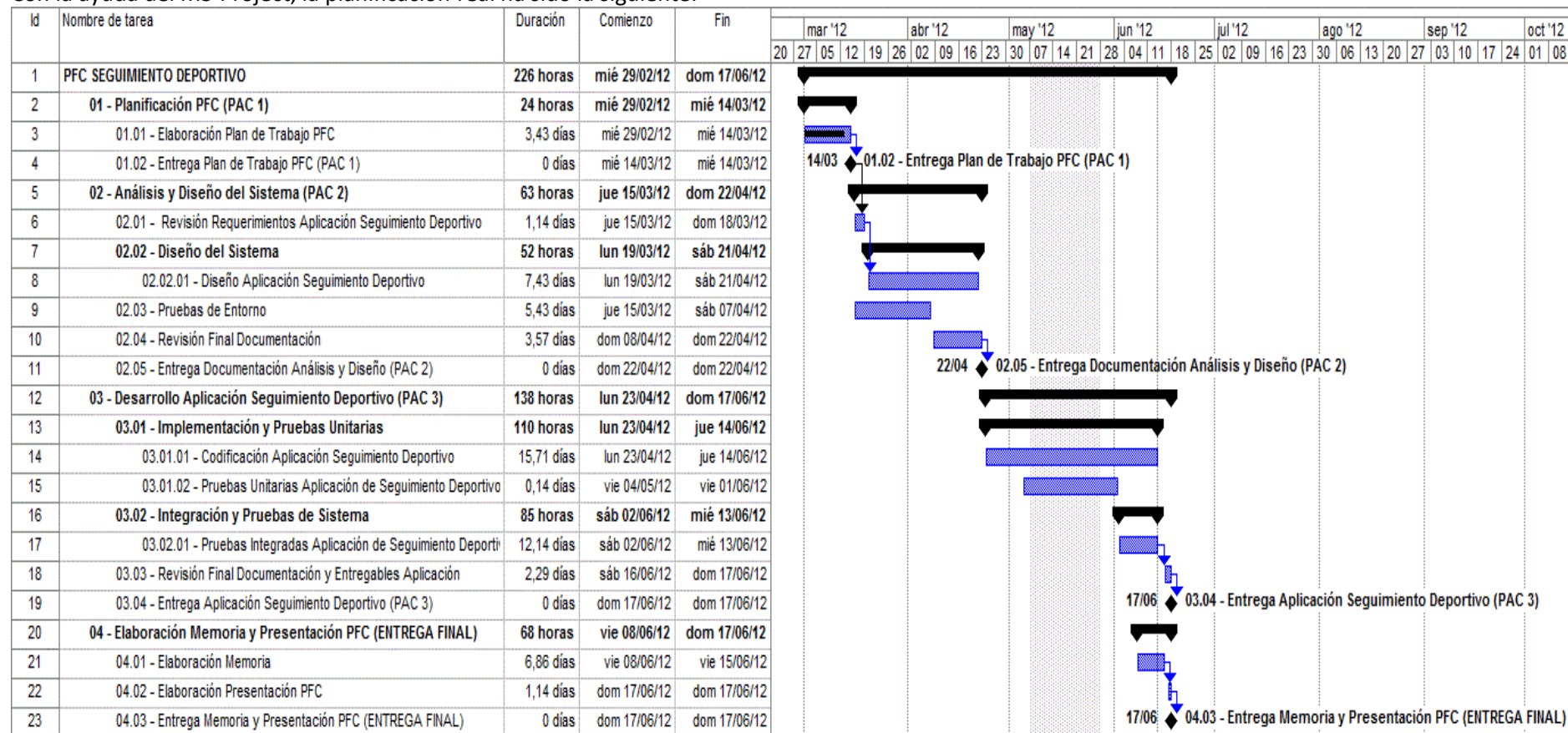


Ilustración B: Planificación PFC



2.5 *Productos obtenidos*

A lo largo de todo el proyecto se han generado los siguientes productos:

- El Plan de Trabajo, que recoge la planificación y estimación de las tareas necesarias para llevar a cabo los objetivos previstos.
- El Documento de Análisis y Diseño del Sistema, que recoge: los requerimientos, los casos de uso, la arquitectura del sistema, el diseño de la BD, etc.
- La Aplicación Seguimiento Deportivo, con:
 - Instalable de la aplicación.
 - Código fuente debidamente comentado.
 - Informe del desarrollo.
 - Manual de instalación, tanto del emulador, como de la aplicación.
- La presente Memoria, que es el documento que sintetizará el trabajo realizado y mostrará que se han alcanzado los objetivos propuestos. Incorporará toda la información relevante para comprender el problema planteado, la metodología utilizada para su resolución y detallará la solución elaborada.
- La Presentación, que resumirá de forma clara y concisa el trabajo realizado y los resultados obtenidos.

2.6 *Estructura de la memoria*

La memoria se divide en tres grandes bloques, correspondientes a las fases principales del desarrollo del proyecto:

- Análisis, donde se describen los requerimientos que se prevén para la aplicación, los casos de uso identificados y los recursos que se estimen necesarios para su implementación y desarrollo.
- Diseño, donde se explican las consideraciones de arquitectura que se han planteado, describiendo y explicando brevemente las tecnologías y patrones elegidos, se incluyen los diferentes diagramas para el desarrollo y se muestran imágenes seleccionadas del interfaz gráfica de interacción con el usuario.
- Implementación, donde se encuentra el guión de creación de la base de datos de la aplicación, un diagrama simplificado de la estructura del producto final, y una valoración de las funcionalidades disponibles y de las posibilidades de extensibilidad y ampliación futuras.

Finalmente, se añaden las conclusiones personales del autor, un glosario de términos, una bibliografía y los Anexos



3 Análisis

3.1 Requerimientos funcionales

3.1.1 Sistema principal

RF1. Al entrar en la aplicación, se mostrará una lista con dos opciones disponibles. La primera de ellas siempre será “*Nuevo entreno*” y la segunda de ellas, el “*Diario de entrenos*” (mostrando todas las sesiones registradas, ordenadas por fecha y hora, de forma descendente, primero las más recientes y la última la más antigua).

RF2. Si el usuario pulsa sobre un elemento de la lista, se abrirá una nueva pantalla en la que se accederá a la opción seleccionada.

RF3. En caso de pulsar la opción “*Nuevo entreno*”, no sólo se mostrará una pantalla en la que aparecerá: Ritmo medio (min/km), Ritmo (min/km), Altitud (m), Distancia (km) y Tiempo (hh:mm:ss), una opción de Iniciar (Inicia la sesión de entrenamiento) y otra de Atrás (Vuelve a la pantalla del menú principal), sino que todos los contadores de duración, distancia y ritmo serán inicializados a cero.

RF4. En caso de pulsar la opción “*Diario de entrenos*”, no sólo se mostrará una pantalla con la lista de sesiones realizadas ordenadas cronológicamente de más reciente a más antiguo, sino que permitirá: Ver la sesión abre una nueva pantalla en donde mostrará un resumen de ella, Borrar una sesión de un día y hora en concreto, Borrar todas las sesiones o volver a la lista de opciones disponibles mediante la opción de Atrás (Vuelve a la pantalla del menú principal).

RF5. En caso de haber Iniciado la sesión de entrenamiento en tiempo real se mostrará por pantalla los datos: Ritmo medio (min/km), Ritmo (min/km), Altitud (m), Distancia (km) y Tiempo (hh:mm:ss), actualizándose cada un segundo. Permitiendo en todo momento pausar la sesión, continuar la sesión, tras una pausa y parar o terminar la sesión (con lo que procede a archivarse en el “*Diario de entrenos*”).

Nuevos requerimientos no contemplados anteriormente en la PAC2, que por necesidad, se han añadido.

RF6. Se inicia la sincronización del GPS, en el momento que se está en la “pantalla Sesión”, mostrando por pantalla el mensaje “Sincronizando GPS ... Por favor Espere ...”, el usuario si no quiere esperar a que se sincronice es libre de pulsar el botón Iniciar para que comience el entreno, mientras tanto se sigue intentando sincronizar. Los únicos botones habilitados la primera vez que se entra en la pantalla “Sesión” son “Iniciar” y “Atrás”

RF7. Una vez se pulsa el botón “Iniciar” en la pantalla “Sesión”, emite un sonido para avisar de que ha comenzado el entreno, se cambia el botón “Iniciar” por “Fin” y se habilitan los 3 botones “Fin”, “Pausa” y “Atrás”.

RF8. Una vez sincronizado se muestra el mensaje: “GPS Sincronizado!”. Si el cronómetro esta parado por que no se haya iniciado aún o porque esté en pausa, solo se muestra las



variaciones de cambios de Ritmo (Velocidad instantánea) y Altitud, en caso contrario, si el cronómetro esta iniciado, se muestra todas las variaciones actualizándose cada segundo.

RF9. Una vez iniciado el entreno si se pulsa el botón de "Pausa", el móvil vibra y el botón "Pausa cambia a "Continuar"

RF10. Al pulsar el botón "Continuar" cambia a "Pausa".

RF11. Al pulsar el botón de "Fin" el móvil emite un sonido para avisar del fin del entreno y se deshabilitan los botones de "Fin" y "Pausa", quedando sólo habilitado el botón "Atrás"

RF12. El GPS siempre que haya cobertura y esté sincronizado estará activo, si en algún momento sufre un corte por que este en un sitio cerrado, intentará por todos los medios sincronizar hasta que se pulse el botón de Fin en cuyo caso se desactiva el GPS y se guarda la sesión de entreno.

RF13. Si el cronómetro está parado, ya sea por que no se ha iniciado el cronómetro o por que esté en modo pausa el GPS, aunque el GPS esté activo, en todo momento se controla para que no sume esas distancias aunque el usuario se esté desplazando. Las únicas variaciones de valores que está autorizado a mostrar en este caso sería la altitud y el Ritmo (velocidad instantánea), tal como se indica en la RF7.

3.2 Requerimientos no funcionales

3.2.1 Requisitos de usuario

RNF1. La aplicación ha de ser tan fácil e intuitiva de usar de modo que un usuario no necesite de un manual para manejarla.

3.2.2 Requisitos tecnológicos

RNF2. El usuario ha de disponer de un dispositivo móvil que disponga de GPS y que acepte un navegador. Hoy día los smartphones cumplen sobradamente estas características.



3.2.3 Requisitos de desarrollo y restricciones de diseño

RNF3. Los requisitos para el entorno de desarrollo de la aplicación son los siguientes:

- Descargar e instalar JDK 1.6.30 (32 bits).
- Descargar e instalar Eclipse Classic 3.7.2 (32 bits)
- Descargar e instalar Android SDK (Lo integramos con Eclipse mediante el Plugin ADT y dispone de un emulador de dispositivos móviles Android) y descargar las plataformas 2.1, 2.2 y 2.3.3, para hacer pruebas.
- Descargar e instalar Plugin ADT para Eclipse
- Descargar e instalar PhoneGap 1.7.0 (Lo integramos con Eclipse mediante el Plugin MDS AppLaud).
- Descargar el Plugin MDS AppLaud que integra PhoneGap con Eclipse

3.2.4 Requisitos de rendimiento

RNF4. La aplicación debe poder ejecutarse en cualquier dispositivo Smartphone no siendo necesario un procesador superior de 512Mhz ni más de 256Mb de RAM para su ejecución.

RNF5. En caso de producirse errores por problemas de sincronización con el GPS deberán mostrarse los errores correspondientes al usuario para que pueda tratar de subsanarlos. Tratando siempre de evitar el cierre abrupto de la aplicación en la medida de lo posible.

3.3 Casos de uso y diagramas de secuencia

En este apartado se describen los casos de uso que el sistema va a contemplar, así como los actores que van a interactuar con el sistema. En el caso particular de esta aplicación es interesante además mostrar los diagramas de secuencia de la misma para poder explicar detalladamente cada funcionalidad de la aplicación.

3.3.1 Descripción de los actores

- Usuario: persona, que no necesita registrarse y va a hacer uso de la aplicación.



3.3.2 Diagramas de casos de uso

A continuación veremos los casos de uso que pueden producirse en el sistema. En el diagrama de casos de uso, hemos incluido los casos de uso de la aplicación Seguimiento Deportivo.

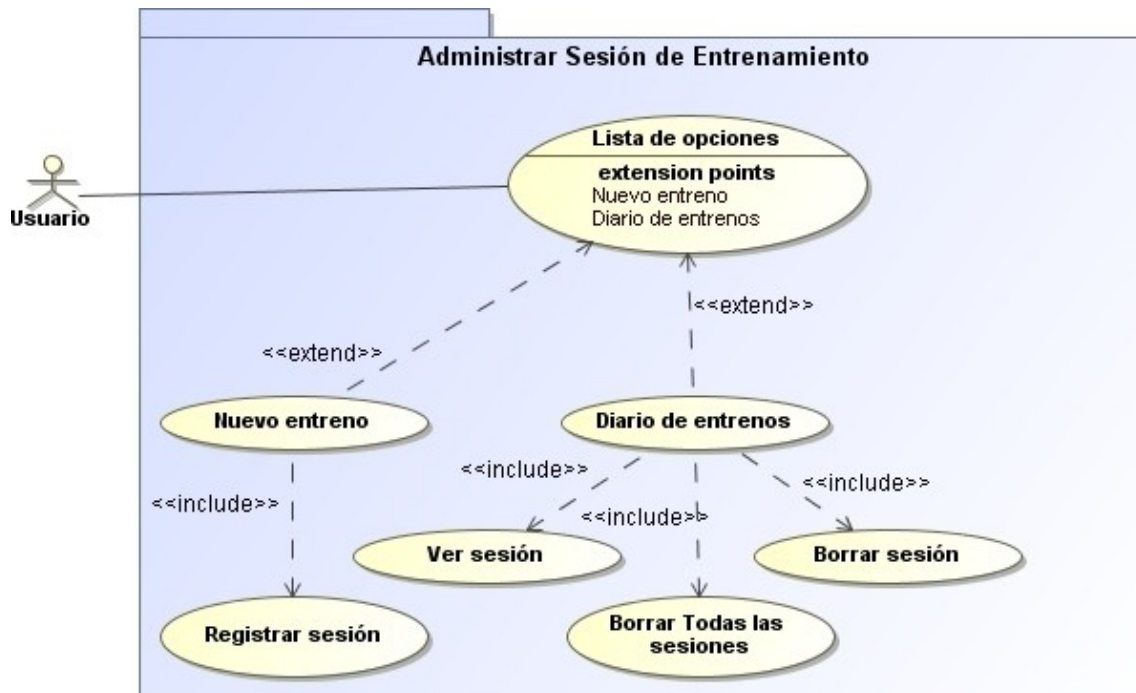


Ilustración C: Diagrama de casos de uso

Código: CU_1
Nombre: Lista de opciones
Descripción: Permite al usuario listar las opciones disponibles de la aplicación.
Actor: Usuario
Precondiciones: No aplicable
Postcondiciones: El usuario ha consultado la lista de opciones solicitada.
Flujo principal: <ol style="list-style-type: none">1) El usuario solicita la lista de opciones de la aplicación.2) El sistema muestra la lista de opciones en la pantalla.3) El usuario consulta la lista de opciones.4) El caso de uso finaliza.
Flujo alternativo:
Pantallas:



Código: CU_2
Nombre: Nuevo entreno
Descripción: Permite al usuario iniciar y registrar una sesión nueva de entrenos.
Actor: Usuario
Precondiciones: El usuario ha seleccionado la opción del listado “Nuevo entreno”
Postcondiciones: La sesión iniciada ha sido registrada en el sistema.
Flujo principal: <ol style="list-style-type: none">1) El sistema emite un sonido.2) Se muestra el mensaje: “Sincronizando GPS ... Por favor Espere ...”3) El usuario inicia una sesión nueva de entrenos y selecciona “Iniciar”.4) El sistema muestra la pantalla “Sesión”.5) El usuario termina la sesión y selecciona “Fin”.6) El sistema emite un sonido.7) El sistema registra la sesión (caso de uso incluido “Registrar sesión”).8) El caso de uso finaliza.
Flujo alternativo: <ol style="list-style-type: none">1a) Se muestra el mensaje: “GPS Sincronizado!”2a) El usuario solicita una pausa en la sesión y selecciona “Pausa”, el móvil vibra.<ol style="list-style-type: none">2a.1) El usuario seleccionar “Continuar”.<ol style="list-style-type: none">1a.1a) Ir al punto 3.2a.2) Ir al punto 4.2b) Se muestra el mensaje: “Pérdida de señal GPS!”
Pantallas: <ul style="list-style-type: none">• Sesión Datos mostrados: <ul style="list-style-type: none">- Ritmo medio (min/km)- Ritmo (min/km)- Altitud (m)- Distancia (km)- Duración (hh:mm:ss)

Código: CU_3
Nombre: Diario de entrenos
Descripción: Permite al usuario listar las sesiones registradas.
Actor: Usuario
Precondiciones: El usuario ha seleccionado la opción del listado “Diario de entrenos”
Postcondiciones: El usuario ha consultado la lista de sesiones de entrenos solicitada.
Flujo principal: <ol style="list-style-type: none">1) El usuario solicita la lista de sesiones.2) El sistema muestra la lista de sesiones en la pantalla.3) El usuario consulta la lista de sesiones.4) El caso de uso finaliza.
Flujo alternativo:
Pantallas: <ul style="list-style-type: none">• Listado Sesiones Datos mostrados: <ul style="list-style-type: none">- Fecha y hora de realización de la sesión.



- Distancia (km)
- Duración (hh:mm:ss)

Código: CU_4

Nombre: Registrar sesión

Descripción: Permite al usuario registrar una sesión nueva de entrenos.

Actor: Usuario

Precondiciones: El usuario ha finalizado la sesión iniciada.

Postcondiciones: La sesión ha sido registrada en el sistema.

Flujo principal:

- 1) El sistema registra la sesión (guardando la fecha y hora de realización de la sesión, el tiempo que ha transcurrido, la distancia recorrida y el ritmo medio).
- 2) El caso de uso finaliza.

Flujo alternativo:

Pantallas:

Código: CU_5

Nombre: Ver sesión

Descripción: Permite al usuario visualizar una sesión de entrenos.

Actor: Usuario

Precondiciones: El usuario ha seleccionado una sesión de la lista de sesiones mediante la opción del listado "Ver sesión"

Postcondiciones: La sesión ha sido mostrada por el sistema.

Flujo principal:

- 1) El sistema muestra los datos de la sesión seleccionada (la fecha y hora de realización de la sesión, el tiempo que ha transcurrido, la distancia recorrida y el ritmo medio), o lo que es lo mismo la pantalla "Detalle Sesión".
- 2) El caso de uso finaliza.

Flujo alternativo:

Pantallas:

- Detalle Sesión

Datos mostrados:

- Fecha y hora de realización de la sesión.
- Duración (hh:mm:ss:milesimas)
- Distancia (km)
- Ritmo medio (min/km)



Código: CU_6
Nombre: Borrar sesión
Descripción: Permite al usuario borrar una sesión de entrenos.
Actor: Usuario
Precondiciones: El usuario ha seleccionado una sesión de la lista de sesiones mediante la opción del listado "Borrar sesión"
Postcondiciones: La sesión ha sido borrada en el sistema.
Flujo principal: <ol style="list-style-type: none">1) El sistema pide confirmación para borrar la sesión.2) El sistema elimina los datos de la sesión seleccionada (la fecha y hora de realización de la sesión, el tiempo que ha transcurrido, la distancia recorrida y el ritmo medio).3) El caso de uso finaliza.
Flujo alternativo: <ol style="list-style-type: none">1a) El usuario cancela el borrado.1b) Ir al punto 3.
Pantallas:

Código: CU_7
Nombre: Borrar Todas las sesiones
Descripción: Permite al usuario borrar todas las sesiones de entrenos.
Actor: Usuario
Precondiciones: El usuario ha seleccionado la opción "Borrar Todas las sesiones"
Postcondiciones: Todas las sesiones han sido borrada en el sistema.
Flujo principal: <ol style="list-style-type: none">1) El sistema pide confirmación para borrar todas la sesiones.2) El sistema elimina la tabla SESION.3) El caso de uso finaliza.
Flujo alternativo: <ol style="list-style-type: none">1a) El usuario cancela el borrado.1b) Ir al punto 3.
Pantallas:



3.3.3 Diagramas de secuencia.

En este diagrama vemos la interacción del Usuario con la pantalla de la Lista de Opciones:

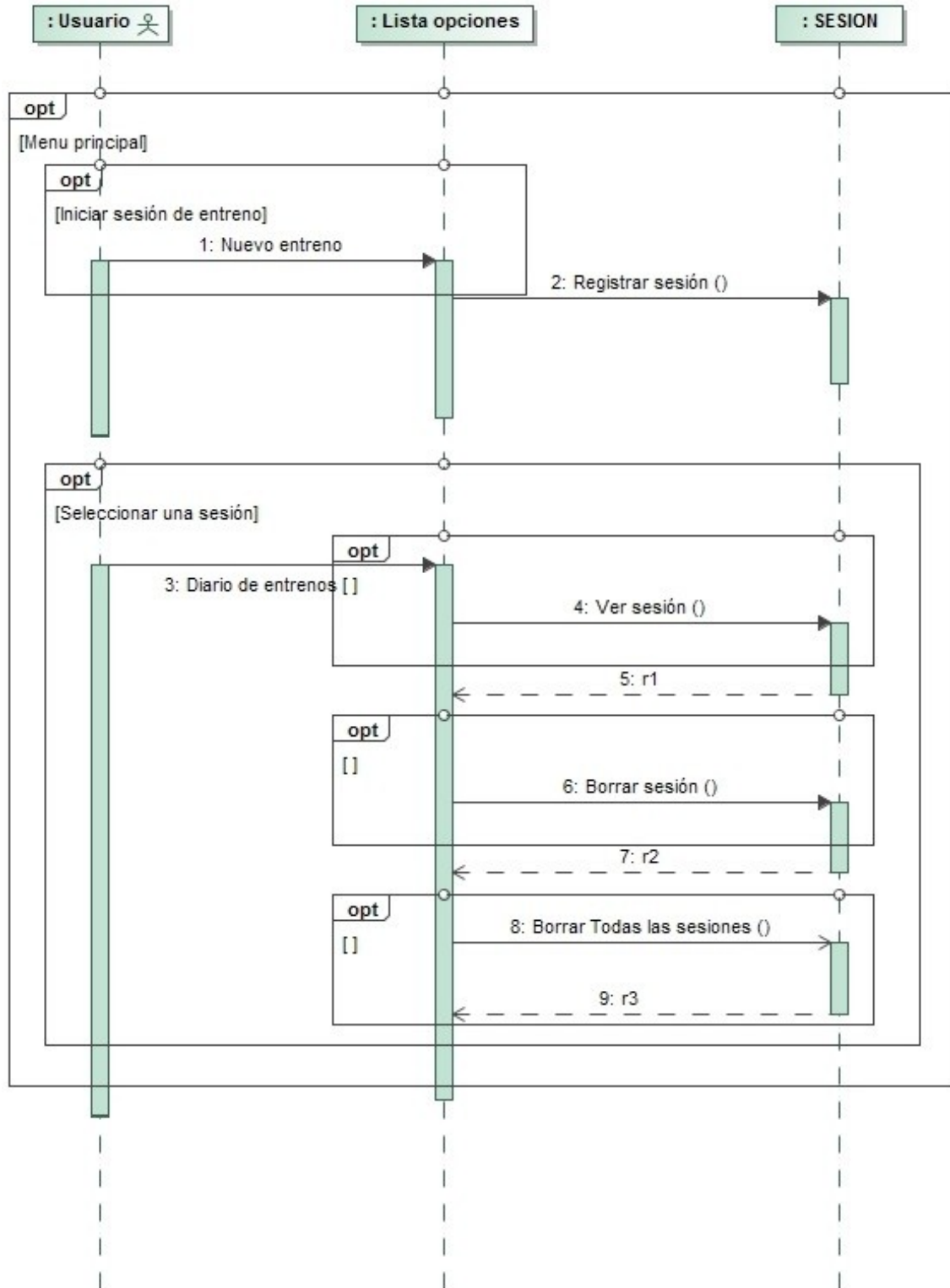


Ilustración D: Diagrama de secuencia



4 Diseño y Arquitectura

4.1 Modelo conceptual

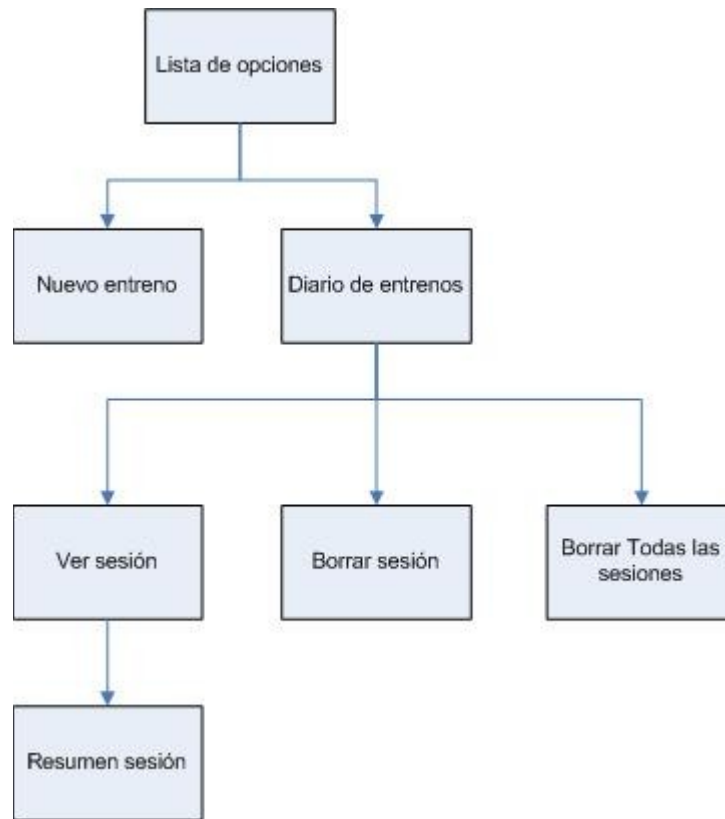


Ilustración E: Modelo conceptual

En el modelo conceptual podemos ver un resumen de todos los pasos por los que puede pasar la aplicación, como podemos observar la pantalla de Lista de opciones tiene 2 opciones.

En la opción "Nuevo entreno" dentro de la pantalla "Sesión" al pulsar la tecla "Iniciar" comenzará la sesión de entrenamiento y al pulsar la tecla "Atrás" la aplicación regresará a "Lista de opciones".

En la opción "Diario de entrenos", dentro de la pantalla "Listado Sesiones", se listarán todas las sesiones de entrenos ordenadas por fecha cronológicamente, es decir, de la más reciente a la más antigua. A parte para cada sesión de entreno se muestra "Ver sesión" y "Borrar sesión".

Si se pulsa "Ver sesión" llevará a la opción "Resumen sesión" que muestra la pantalla "Detalle sesión" mientras que si se pulsa "Borrar sesión" se pedirá la confirmación para borrar la sesión, por otro lado si se pulsa "Borrar Todas" pedirá la confirmación para borrar todas las sesiones de entrenos, en el caso de que se borre alguna sesión, se actualiza en pantalla sobre la marcha y si se pulsa la tecla "Atrás" la aplicación regresará a "Lista de opciones".



En la pantalla "Detalle sesión", se muestra a modo de resumen los datos de la sesión de entrenos seleccionada, por otro lado si se pulsa la tecla "Atrás" la aplicación regresará a la opción "Diario de entrenos".



4.2 Arquitectura del sistema

La arquitectura del sistema que vamos a utilizar es de tres capas: la capa de presentación, la capa de negocio y la capa de datos.

- La Capa de Presentación: Esta capa realiza la presentación al usuario, mostrando todos los componentes que comunican información y capturan la información del usuario. También es conocida como interfaz gráfica. Se comunica con la Capa de de Negocio.
- La Capa de de Negocio: Esta capa incorpora los componentes que implementan la lógica del negocio en el sistema. Incluye las entidades de negocio que representan los objetos específicos para el dominio del negocio y establecen todas las reglas del negocio. Se comunica con la Capa de Presentación facilitando los datos y recibiendo los datos introducidos por el usuario.
- La Capa de Datos: es la capa que guarda los datos. Desde esta capa se acceden a los datos que está formada por un gestor de base de datos que se encarga del almacenamiento de estos. Se comunica con la Capa de Negocio facilitando los datos y recibéndolos.

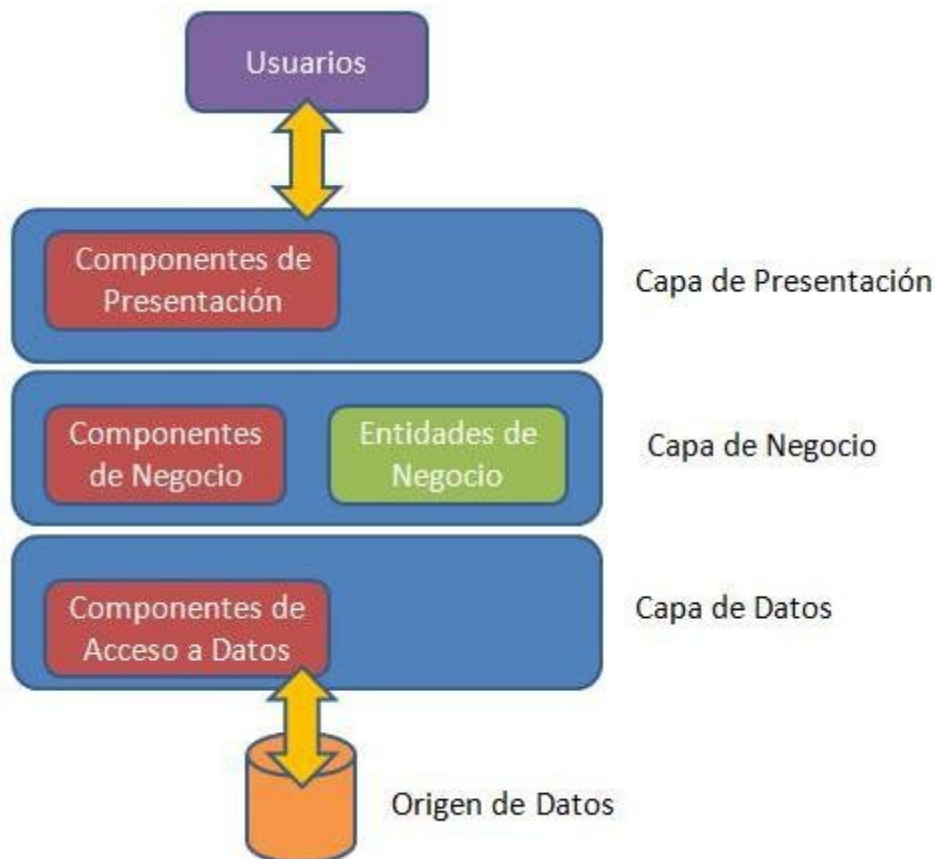


Ilustración F: Modelo 3 capas



4.3 Modelos de clases del diseño

4.3.1 Diagrama de clases



Ilustración G: Diagrama de clases



A continuación se indica las justificaciones de la creación de las clases descritas en el diagrama:

Clase Usuario

Descripción

La clase Usuario permite crear deportistas para que realicen sesiones de entrenamiento.

Asociación Usuario – Sesión

Permitimos que un usuario realice múltiples sesiones de entrenamiento.

Clase Sesión

Descripción

Una sesión de entrenamiento se caracteriza por que un deportista recorre una cierta distancia en un cierto tiempo.

Esta clase además dispone de otra información relevante para la sesión de entrenamiento como son la identificación de la sesión, fecha en que se realiza sesión, la hora en que se inicia la sesión, la duración, la distancia y el ritmo (atributo calculado a partir de la duración y la distancia de la sesión de entrenamiento).

Asociación Sesión – Usuario

Una sesión de entrenamiento necesita tener un usuario asociado.



4.4 Diseño de la Base de datos para la aplicación

SESION	
PK	<u>id</u>
	fecha hora duracion distancia ritmo

Tabla 1: Tabla SESION

Creamos una tabla de registro de Sesiones "tabla Sesión" ya que vamos a necesitar rescatar la sesión ya sea para listarlas, verlas o borrarlas y el identificador será "id".

No creamos tabla para usuario, ya que la gestión de las sesiones, las realizamos de forma genérica y sin vinculación al usuario de manera que con sólo entrar en la aplicación accedamos a todas las sesiones que se hayan registrado.

4.4.1 Descripción de la tabla SESION

Campo	Descripción
<i>id</i>	Contiene un identificador de la sesión, es un valor interno y automático del sistema.
<i>fecha</i>	Contiene la fecha (dd/mm/aa) en que se realiza sesión de entrenamiento.
<i>hora</i>	Contiene la hora (hh:mm) en que se realiza sesión de entrenamiento.
<i>duracion</i>	Contiene la duración (hh:mm:ss) de la sesión de entrenamiento.
<i>distancia</i>	Contiene la distancia (km) recorrida de la sesión de entrenamiento.
<i>ritmo</i>	Contiene el ritmo medio (mm:ss/km) de la sesión de entrenamiento.

Tabla 2: Tabla Descripción de la Sesión



4.5 *Diseño de la Interfaz gráfica*

A continuación mostramos las pantallas del interfaz y su utilidad:

- La pantalla de "Inicio" de la aplicación en donde tenemos 2 opciones: "Nuevo entreno" que nos lleva a la pantalla "Sesión" y "Diario de entrenos" que nos lleva a la pantalla de "Listado Sesiones".

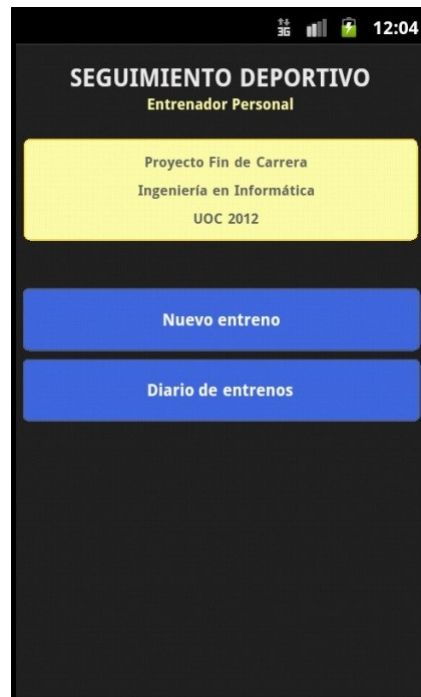


Ilustración H: Pantalla "Inicio"



- La pantalla de "Sesión" de la aplicación, donde se muestran todos los datos de la sesión de entreno en tiempo real, una vez pulsado el botón "Iniciar".



Ilustración I: Pantalla "Sesión"

- La pantalla de "Listado Sesiones", donde se muestra un Diario de entrenos.



Ilustración J: Pantalla "Listado Sesiones"



- La pantalla de "Detalle Sesión", donde se muestra un resumen de la sesión de entreno, seleccionada.



Ilustración K: Pantalla "Detalle Sesión"



5 Implementación

5.1 Introducción

El entorno de desarrollo utilizado ha sido Eclipse integrado con PhoneGap, mediante el plugin MDS AppLaud.

El plugin MDS AppLaud para Eclipse amplía las herramientas de desarrollo de Android y el framework PhoneGap proporcionando un IDE completo para el desarrollo de Android:

- Soporta JavaScript, HTML5 y CSS
- Escala sus proyectos de desarrollo web con el poder de Eclipse
- Construido sobre Eclipse JSDT (JavaScript Development Tools)
- Entorno de generación automática, objetivo fácil y despliegue del emulador
- Sistema híbrido de Java y desarrollo web en un sólo proyecto.

Al iniciar el proyecto se ha realizado la configuración de los permisos correspondientes a las API, en dos ficheros: AndroidManifest.xml y plugins.xml

Se ha implementado el menú principal (index.html), el formulario Sesión (Sesion.html), el formulario Listado de Sesiones (ListadoSesiones.html) y el formulario Resumen de Sesión (DetalleSesion.html), junto con la hoja de estilo (master.css)

Por otro lado se han desarrollado los Scripts Cronómetro, Geolocalización (GPS), Cálculo de Distancia, Gestión de la Base de Datos y Notificación (SeguimientoDeportivo.js), los cuales se proceden a explicar con más detalle.



5.2 Script Cronómetro

Mediante este script se obtiene el cronómetro de la sesión.

Según se solicite se inicia el cronómetro mediante el botón "Iniciar", se para provisionalmente mediante el botón "Pausa", se continua tras una pausa mediante el botón "Continuar" o se para definitivamente mediante el botón "Fin".

También se alternan los botones "Iniciar" por "Fin" y "Pausa" por "Continuar" y viceversa según se necesite.

```
/*  
*****  
*   CRONOMETRO   *  
*****  
*/  
  
/**  
* Detiene el Cronómetro y se guardan los datos  
* de la sesión de entreno en la Tabla SESION de la BD  
*/  
function DetenerCrono (){  
    clearTimeout(CronoID);  
    CronoActivo = false;  
    stop_watchlocation();  
    ValorCrono += ":" + msec;  
    document.crono.tiempo.value = ValorCrono;  
    document.crono.PausaContinua.disabled = true;  
    document.crono.Iniciar.disabled = true;  
    vibrate();  
    beep(2);  
    createDB();  
    insertDB();  
}  
  
/**  
* Pausa el Cronómetro  
*/  
function PausarCrono (){  
    // Medimos la distancia desde la última toma hasta el momento en que se pulsa el botón  
de Pausa  
    document.crono.PausaContinua.value = ' Continuar ';  
    document.crono.PausaContinua.onclick = ContinuarCrono;  
    // Vibración del teléfono móvil  
    vibrate();  
    clearTimeout(CronoID)  
    CronoActivo = false;;  
}  
  
/**  
* Continúa el Cronómetro  
*/  
function ContinuarCrono(){  
    document.crono.PausaContinua.value = ' Pausa ';  
    document.crono.PausaContinua.onclick = PausarCrono;  
    start = new Date()-diff;  
    start = new Date(start);  
    MostrarCrono();  
    tomaGPS = 1;  
    CronoActivo = true;  
}  
}
```




```
/**
 * Inicializa el Cronómetro
 */
function InicializarCrono () {
    //inicializa contadores globales
    decimas = 0;
    segundos = 0;
    minutos = 0;
    horas = 0;

    //pone a cero los marcadores
    document.crono.tiempo.value = '00:00:00';
}

/**
 * Muestra el Cronómetro
 */
function MostrarCrono () {
    end = new Date();
    diff = end - start;
    diff = new Date(diff);
    msec = diff.getMilliseconds();
    segundos = diff.getSeconds();
    minutos = diff.getMinutes();
    horas = diff.getHours()-1;
    if(msec < 10){
        msec = "00" +msec;
    }
    else if(msec < 100){
        msec = "0" +msec;
    }

    //configura la salida
    ValorCrono = (horas < 10) ? "0" + horas : horas;
    ValorCrono += (minutos < 10) ? ":0" + minutos : ":" + minutos;
    ValorCrono += (segundos < 10) ? ":0" + segundos : ":" + segundos;
    document.crono.tiempo.value = ValorCrono;
    Muestra_RMedio();

    //incrementa el crono cada 1 segundo
    CronoID = setTimeout("MostrarCrono()", 1000);
    return true;
}

/**
 * Inicia el Cronómetro
 */
function IniciarCrono () {
    FechaHoraActual();
    InicializarCrono();
    InicializarGPS();
    tomaGPS = 1;

    document.crono.PausaContinua.disabled = false;
    document.crono.Iniciar.value = ' Fin ';
    document.crono.Iniciar.onclick = DetenerCrono;
    start = new Date();
    MostrarCrono();
    CronoActivo = true;
    beep(1);
}
```



5.3 Script Geolocalización (GPS)

Se utiliza la API Geolocation.

Mediante este script se calcula las coordenadas GPS del móvil, velocidad instantánea y altitud.

Se informa al usuario cuando esté el GPS sincronizado o no.

La velocidad se obtiene en m/s pero mediante un algoritmo se convierte a min/km.

Cuando se obtienen dos coordenadas validas (se considera válida cuando el Crono está funcionando, en el script se identifica como **CronoActivo=true**) se llama a la función Calculatedistance y se le pasa ambas coordenadas para que calcule la distancia entre ambas.

En el siguiente Script por un lado la función *InicializarGPS()* se encargará de poner a cero, los valores necesarios y la función *SincronizarGPS()* se encargará tanto de sincronizar el GPS, como de calcular las coordenadas GPS del móvil, velocidad instantánea y altitud.

```
/*
 * Geolocalización (GPS)
 * utilizamos la api-geolocation
 */

/**
 * Inicializa contadores para el GPS
 */
function InicializarGPS() {
    //inicializa contadores globales
    distancia = 0.00;

    //pone a cero los marcadores
    document.sesion.Rmedio.value = '0:00';
    document.sesion.ritmo.value = '0:00';
    document.sesion.altitud.value = '0';
    document.sesion.distancia.value = '0.00';
}

/**
 * Sincroniza el GPS
 */
function SincronizarGPS(){
    InicializarGPS();
    initiate_watchlocation();
}

var wsuccess = function(position) {
    // Si viene de un corte de señal
    if (!GpsSincronizado){
        if (tomaGPS == 0){
            alert("GPS Sincronizado!");
        }
        else if (tomaGPS > 0){
            alert("Recuperada la se\u00f1al de GPS!");
        }
        tomaGPS=1;
        GpsSincronizado = true;
    }
    // Obtenemos la altitud, velocidad, latitud y longitud.
    velocidad = position.coords.speed;
    if (isNaN(velocidad) || (velocidad == 0)){
        Valorritmo = "0:00";
    }
    else{
        // convertimos m/s a min/km si la velocidad es distinto de cero y no es nula
    }
}
}
```



```
    if (velocidad > 0){
        velocidad = 1/(velocidad * 0.06);
    }
    // Configuro la salida
    var minxkm = truncate(velocidad);
    var decimales = Math.abs(velocidad - minxkm);
    var seg = decimales * 60;
    var segxkm = truncate(seg);

    Valorritmo = minxkm;
    Valorritmo += (segxkm < 10) ? ":0" + segxkm : ":" + segxkm;
}
document.sesion.altitud.value = position.coords.altitude;
document.sesion.ritmo.value = Valorritmo;
if (CronoActivo){
    if (tomaGPS > 1){
        distancia += calculateDistance(startPos.coords.latitude, startPos.coords.longitude,
position.coords.latitude, position.coords.longitude);
        if (isNaN(distancia) || (distancia == 0)){
            document.sesion.distancia.value = '0.00';
        }
        else{
            document.sesion.distancia.value = roundNumber(distancia,2);
        }
    }
    else{
        if (isNaN(distancia) || (distancia == 0)){
            document.sesion.distancia.value = '0.00';
        }
    }
    startPos = position;
    tomaGPS++;
}
else{
    tomaGPS=1;
}
};

/**
 * Error al iniciar el observador GPS
 */
var wfail = function(error) {
    switch(error.code)
    {
        case error.PERMISSION_DENIED:
            alert("ERROR: Usuario ha denegado el acceso al seguimiento de la posici\u00f3n");
            window.location.href = "index.html";
            break;
        case error.POSITION_UNAVAILABLE:
            alert("ERROR: Hay problemas para obtener la posici\u00f3n del dispositivo!");
            break;
        case error.TIMEOUT:
            console.log("ERROR: La aplicaci\u00f3n esta tratando de obtener la posici\u00f3n del dispositivo!");
            if (!GpsSincronizado){
                alert("Sincronizando GPS ... \nPor favor Espere ...");
            }
            else{
                alert("P\u00e9rdida de la se\u00f1al de GPS!");
                GpsSincronizado = false;
            }
            break;
        default:
            alert("ERROR: Problema desconocido!");
            window.location.href = "index.html";
            break;
    }
};
```



```
/**
 * Inicia el observador GPS
 */
function initiate_watchlocation() {
  if (watchID == null) {
    var options = { frequency: 3000, maximumAge: 5000, timeout: 5000, enableHighAccuracy:
true };
    watchID = navigator.geolocation.watchPosition(wsuccess, wfail, options);
  }
}

/**
 * Detiene el observador GPS
 */
function stop_watchlocation() {
  if (watchID != null) {
    navigator.geolocation.clearWatch(watchID);
    watchID = null;
  }
}
```

5.4 Script Cálculo de distancia

Una vez fijada la posición inicial, dentro del periodo de tiempo establecido se va consultando el GPS para calcular la distancia hasta la posición caracterizada por las coordenadas actuales. El cálculo se efectúa en radianes y el resultado se da en metros.

Para simplificar el paso de radianes a metros se ha considerado un valor medio de equivalencia grado/kilómetro que se ha fijado en 6371, que equivale al radio medio de la tierra, en km.

El script *calculateDistance()* realiza un algoritmo geométrico para determinar la distancia entre dos coordenadas. La implementación de Javascript es una adaptación de un guión proporcionado por [Moveable Type](#) bajo una licencia Creative Commons:

```
/**
 * Calcula la distancia Geodésica entre dos tramos
 */
function calculateDistance(lat1, lon1, lat2, lon2) {
  var R = 6371; // Radio de la tierra en km
  var dLat = (lat2 - lat1).toRad();
  var dLon = (lon2 - lon1).toRad();
  var a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +
    Math.cos(lat1.toRad()) * Math.cos(lat2.toRad()) *
    Math.sin(dLon / 2) * Math.sin(dLon / 2);
  var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
  var d = R * c;
  return d;
}
Number.prototype.toRad = function() {
  return this * Math.PI / 180;
}
```



5.5 Script Paso de Variables entre formularios método GET

Desde la pantalla de "Listado de Sesiones" si se pulsa en "VER SESION", se muestra en la pantalla "Detalle Sesión" un resumen de los datos de la sesión seleccionada, para recibir los datos desde DetalleSesion.html es necesario hacer un algoritmo para que puedan pasar las variables.

La llamada a DetalleSesion.html pasándole los valores desde SeguiendoDeportivo.js, en concreto desde la función *showRecords()* se ha planteado de la siguiente manera:

```
<a href="DetalleSesion.html?' + item['id'] + '&' + item['fecha'] + '&' + item['hora'] + '&' +  
item['duracion'] + '&' + item['distancia'] + '&' + item['ritmo'] + '"><font color="green">[VER  
SESI&Oacute;N]</a>
```

La recepción de los valores en DetalleSesion.html se ha planteado de la siguiente manera:

```
<script>  
  var arrValores = leerGET();  
  document.write('<h3>Fecha de inicio : ' + arrValores[1] + '</h3>');  
  document.write('<h3>Hora de inicio : ' + arrValores[2] + '</h3>');  
  document.write('<h3>Duraci&Aacute;n : ' + arrValores[3] + '</h3>');  
  document.write('<h3>Distancia : ' + arrValores[4] + ' km</h3>');  
  document.write('<h3>Ritmo medio : ' + arrValores[5] + ' min/km</h3>');  
</script>
```

El script *leerGET()* se ha planteado de la siguiente manera:

```
/**  
 * Leer valores por el método GET  
 */  
function leerGET(){  
  var cadGET = location.search.substr(1,location.search.length);  
  var arrGET = cadGET.split("&");  
  return arrGET;  
}
```



5.6 Script Gestión de la Base de Datos

Se utiliza la API Storage.

Mediante este script se crea/abre la BD, se guardan los datos de la sesión y posteriormente se pueden ver o borrar según se solicite.

```
/*
 * Gestión de la BD
 * utilizamos la api-storage
 */

/**
 * Intentando crear la Tabla SESION en la BD
 */
function populateDB(tx) {
    tx.executeSql('CREATE TABLE IF NOT EXISTS SESION (id INTEGER NOT NULL PRIMARY KEY
    AUTOINCREMENT, fecha, hora, duracion, distancia, ritmo)');
}

/**
 * Error al crear la Tabla Sesion en la BD
 */
function errorCB(err) {
    alert("Error al crear la Tabla");
    console.log("Error processing SQL: " + err.code);
}

/**
 * Creada la Tabla SESION en la BD
 */
function successCreateCB() {
    console.log("Success creating Database 1.0");
}

/**
 * Crea/Abre la BD
 */
function createDB(){
    if (!db) {
        db = window.openDatabase("Database", "1.0", "Seguimiento Deportivo", 5*1024*1024); //5MB
    }
    db.transaction(populateDB, errorCB, successCreateCB);
}

/**
 * Inserta un registro en la Tabla SESION, con los datos de la sesión de entreno.
 */
function insertDB(){
    var sql = "INSERT INTO SESION (fecha, hora, duracion, distancia, ritmo) values
    (?, ?, ?, ?, ?)";
    db.transaction(
        function(tx){
            tx.executeSql(sql, [fecha_in, hora_in, ValorCrono, roundNumber(distancia,2),
            ValorRmedio],
                function(tx, result){
                    console.log('Ejecuci\u00f3n correcta.\n' + result.rowsAffected + '
                    sesi\u00f3n afectada');
                },
                function(tx, error){
                    alert('Error: ' + error.message);
                }
            );
        }
    );
}
```



```
/**
 * Muestra todos los registros de la tabla SESION
 * ordenados descendientemente por fecha de registro, es decir,
 * el primero el más reciente el último el más antiguo.
 */
function showRecords() {
    if (!db) {
        db = window.openDatabase("Database", "1.0", "Seguimiento Deportivo", 5*1024*1024); //5MB
    }
    document.getElementById('db-results').innerHTML = '';
    db.transaction(function(tx) {
        tx.executeSql("SELECT * FROM SESION", [], function(tx, result) {
            for (var i = result.rows.length-1, item = null; i >= 0; i--) {
                item = result.rows.item(i);
                document.getElementById('db-results').innerHTML +=
                    '<li><span>' + item['fecha'] + ' ' + item['hora'] + ' ' + item['distancia'] +
                    'km ' + item['duracion'] + '<br></span>' + '<a href="DetalleSesion.html?' + item['id'] + '&' +
                    item['fecha'] + '&' + item['hora'] + '&' + item['distancia'] + '&' +
                    item['ritmo'] + '<\/a><\/font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;' + '<a
                    href="#" onclick="deleteRecord('+item['id']+')"><font color="red">[BORRAR
                    SESI&Oacute;N]<\/a><\/font><\/li>';
            }
        });
    });
}

/**
 * Actualiza registros de la Tabla SESION de la BD
 */
function updateRecord(id, textE1) {
    db.transaction(function(tx) {
        tx.executeSql("UPDATE SESION SET text = ? WHERE id = ?", [textE1.innerHTML, id], null,
    onError);
    });
}

/**
 * Borra un registro de la Tabla SESION de la BD
 */
function deleteRecord(id) {
    if (confirm('Se va a proceder a Borrar la Sesi&u00f3n de Entreno.&n\u000BFesta seguro?'))
    {
        db.transaction(function(tx) {
            tx.executeSql("DELETE FROM SESION WHERE id=?", [id],
                function(tx, result) { showRecords() },
                onError);
        });
    }
}

/**
 * Borra la Tabla SESION de la BD
 */
function dropTable() {
    if (confirm('Se va a proceder a BORRAR TODAS las sesiones de entreno.&n\u000BFesta seguro?'))
    {
        db.transaction(function(tx) {
            tx.executeSql("DROP TABLE IF EXISTS SESION", [],
                function(tx) { showRecords() },
                onError);
        });
    }
}

var log = document.getElementById('db-log');

function onError(tx, error) {
    console.log("Error processing SQL: " + error.code);
    console.log("Error processing SQL: " + error.message);
    log.innerHTML = '<p class="error">Error: ' + error.message + '<\/p>';
}
}
```



5.7 Script Notificaciones

Se utiliza la API Notification.

Mediante este script se emite un sonido o se hace vibrar el teléfono.

```
/*  
 * Gestión de Notificaciones *  
 * utilizamos la api-notification *  
 */  
  
/**  
 * Emite un Sonido  
 */  
function beep (sound) {  
    navigator.notification.beep(sound);  
};  
  
/**  
 * Vibra el dispositivo  
 */  
var vibrate = function() {  
    navigator.notification.vibrate(0);  
};
```

5.8 Pruebas

Evidentemente, se ha de probar el correcto funcionamiento de la aplicación y para ello se realizan pruebas parciales durante la fase de implementación y una vez terminado, se realiza un plan de pruebas para evaluar si el funcionamiento es el esperado.

5.8.1 Plan de Pruebas

Las pruebas se han realizado en un primer momento en el emulador (AVD) en sus versiones 2.1, 2.2 y 2.3.3, al que se accede desde el SDK, para cargar el programa, depurarlo, monitorizar la ejecución y simular los datos desde el dispositivo GPS y en un segundo momento en un móvil Samsung Galaxy Ace S5830 con Android 2.3.4.

Se han implementado todos los casos de uso con éxito, para las versiones anteriormente mencionadas.

No se han realizado pruebas exhaustivas con todas las versiones disponibles en el mercado, ya que el sistema objetivo ha sido el 2.3.* como máximo, pero se ha detectado un problema de compatibilidad con la versión 3.0, 3.1, 3.2 y 4.0.*, a la hora de pasar parámetros vía URL, reportado en el siguiente enlace: <http://code.google.com/p/android/issues/detail?id=17535> .

Cabe destacar la precisión del algoritmo *CalculateDistance()*, se han realizado varias pruebas de medición de hasta 35km, en automóvil, con una leve variación de entre 0% y 1%.



6 Conclusiones

La elaboración de este Proyecto de Fin de Carrera me ha permitido explorar más profundamente la base sobre la que éstas se han especificado, el alcance real de sus posibilidades y la diversidad de la oferta disponible de software que las implementa y soporta.

Por otro lado, me ha permitido vivir en primera persona el proceso completo de desarrollo de una aplicación para móvil a través de todas sus fases (planificación, análisis, diseño, implementación, ...), así como experimentar la necesidad de tomar decisiones cruciales para el proceso, como la elección de patrones de diseño adecuados o la elección e incorporación de productos que se ajusten a los requerimientos del software.

En definitiva y como conclusión final, me gustaría resaltar como principal fruto del PFC la visión global sobre programación de aplicaciones para móviles y su entorno de aplicación que me ha proporcionado este proyecto, de igual manera quiero destacar el enriquecimiento personal que la labor de investigación y búsqueda que ha llevado vinculado me ha supuesto, aportando cohesión y perspectiva a los conocimientos adquiridos durante estos años de estudios en Ingeniería Informática.



7 Glosario

AGPS: (*Assisted Global Positioning System*) Sistema de posicionamiento más común en dispositivos móviles, que además del GPS puede utilizar redes de datos o la propia de telefonía móvil.

Android: Es un sistema operativo para móviles, basado en una versión modificada del núcleo Linux.

Android SDK: (Android Software Development Kit) es el kit de desarrollo de software para Android.

API: (*Application Program Interface*) Es la interfaz que un sistema, librería o aplicación proporciona para permitir que otros programas hagan peticiones de servicio o intercambien datos con ellos.

APK: Un archivo con extensión “.apk” es un paquete para el sistema operativo Android. Este formato es una variante del formato JAR de Java y se usa para distribuir e instalar componentes empaquetados para la plataforma Android para smartphones y tablets.

AVD: (*Android Virtual Device*) Dispositivo Virtual de Android donde se pueden probar las aplicaciones gracias al runtime que incluye.

CSS: (Cascading Style Sheets) Lenguaje de hojas de estilo que permite describir el aspecto y el formato de un documento escrito en un lenguaje de marcas.

Eclipse: Es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido"

GPS: (*Global Positioning System*) Sistema de posicionamiento que, por medio de satélites, permite localizar las coordenadas terrestres y la altura sobre el nivel del mar donde se encuentra un receptor.

HTML5: (*HyperText Markup Language, versión 5*) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. Un lenguaje de marcas que permite incorporar elementos con video y fotografías de forma nativa.

IDE: (Integrated development environment) Entorno de desarrollo integrado. Es un programa informático compuesto por un conjunto de herramientas de programación.

PFC: Proyecto Fin de Carrera.

PhoneGap: Es una plataforma de código abierto para el desarrollo multiplataforma de aplicaciones nativas para móviles que aprovecha las tecnologías web HTML5, CSS3 y JavaScript.

Plugin: Es una aplicación que se relaciona con otra para aportarle un función nueva y generalmente muy específica.

Smartphone: (Teléfono inteligente) es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de computación y conectividad que un teléfono móvil convencional.



Waterfall: (Cascada) Uno de los modelos de desarrollo de software clásicos más comunes, donde las diferentes fases se suceden una tras otra (como una cascada): análisis de requerimientos, diseño, implementación, pruebas, integración y mantenimiento.



8 Bibliografía

La biografía más destacada que se ha utilizado es la siguiente:

Adobe Systems Inc. *PhoneGap* [en línea]. Disponible en Web: <<http://phonegap.com>> [Consulta: Febrero 2012].

Adobe Systems Inc. *Apache Cordova Documentation API* [en línea]. Disponible en Web: <<http://docs.phonegap.com/en/1.7.0/index.html>> [Consulta: Mayo - Junio 2012].

Adobe Systems Inc. *Apache Cordova Documentation API* [en línea]. Getting Started with Android. Disponible en Web: <http://docs.phonegap.com/en/1.7.0/guide_getting-started_android_index.md.html#Getting-Started-with-Android> [Consulta: Mayo - Junio 2012].

Adobe Systems Inc. *Apache Cordova Documentation API* [en línea]. Geolocation. Disponible en Web: <http://docs.phonegap.com/en/1.7.0/cordova_geolocation_geolocation.md.html#Geolocation> [Consulta: Mayo - Junio 2012].

Adobe Systems Inc. *Apache Cordova Documentation API* [en línea]. Notification. Disponible en Web: <http://docs.phonegap.com/en/1.7.0/cordova_notification_notification.md.html#Notification> [Consulta: Mayo - Junio 2012].

Adobe Systems Inc. *Apache Cordova Documentation API* [en línea]. Storage. Disponible en Web: <http://docs.phonegap.com/en/1.7.0/cordova_storage_storage.md.html#Storage> [Consulta: Mayo - Junio 2012].

Android Developer. *The official site for Android developers. Provides the Android SDK and documentation for app developers and designers.* [en línea]. Disponible en Web: <<http://developer.android.com/index.html>> [Consulta: Marzo-Abril 2012].

Carbonell, José M. *Almacenamiento con HTML5: Almacenamiento Sql* [en línea]. Disponible en Web: <<http://theproc.es/2011/4/21/18450/almacenamiento-con-html5--almacenamiento-sql>> [Consulta: Mayo 2012].

Clarísó, Robert. *Orientacions generals per la realització del treball / projecte final de carrera als Estudis d'Informàtica, Multimèdia i Telecomunicació de la UOC.* [en línea]. [Barcelona]: UOC, Marzo 2011. Disponible en Web: <<http://www.slideshare.net/rclariso/orientacions-generals-per-al-treball-final-a-la-uoc>> [Consulta: Marzo 2012].

Kinlan, Paul. *A simple todo list using HTML5 Webdatabases* [en línea]. Disponible en Web: <<http://www.html5rocks.com/en/tutorials/webdatabase/todo/>> [Consulta: Mayo 2012].

MDS. *AppLaud Eclipse Plugin* [en línea]. Disponible en Web: <<http://www.mobiledevelopersolutionidsns.com/home>> [Consulta: Abril 2012].



Pérez Navarro, A. *Treball final de carrera* [en línea]. Bataller Díaz, A; Beneito Montagut, R; Sáenz Higuera, N; Vidal Oltra, R. [Barcelona]: FUOC, 2008. Disponible en Web:
<http://materials.cv.uoc.edu/continguts/XW08_19018_00443/> [Consulta: Marzo 2012].

Rey, Javier. *Tabla de caracteres especiales UNICODE* [en línea]. Disponible en Web:
<<http://javirey.es/tabla-de-caracteres-especiales-unicode/>> [Consulta: Junio 2012]

Solís, Carlos. *PhoneGap - Revolución móvil* [en línea]. Disponible en Web:
<<http://revolucion.mobi/category/phonegap/>> [Consulta: Mayo 2012].

Veness, Chriss. *Calculate distance and bearing between two Latitude/Longitude points using Haversine formula in JavaScript* [en línea]. Disponible en Web:
<<http://www.movable-type.co.uk/scripts/latlong.html>> [Consulta: Mayo 2012].

Wikimedia Foundation. *Wikipedia, The Free Encyclopedia* [en línea]. Disponible en Web:
<<http://www.wikipedia.com>> [Consulta: Junio 2012]

W3C. *Geolocation API Specification* [en línea]. Last updated on May 10, 2012. Disponible en Web: <<http://www.w3.org/TR/geolocation-API/>> [Consulta: Mayo 2012].



9 Anexos

9.1 Manual de instalación del Entorno de desarrollo

El entorno de desarrollo está basado en Eclipse con funciones específicas de PhoneGap y un emulador de Android.

Mediante Eclipse podemos controlar la ejecución del programa que se efectúa en el emulador.

Android SDK dispone de un emulador de Android que facilita las pruebas de la aplicación. En cuanto al hardware podremos interactuar como si se trata de un aparato real generando eventos de dispositivos en conveniencia.

A continuación vamos a indicar las versiones recomendadas junto con su URL, haciendo CTRL+clic bien sobre la captura de pantalla o sobre el título de la ilustración.

- 1) Lo primero que debemos de descargar e instalar es Java Software Development Kit, recomendamos la versión **JDK-6u30**, que es la que se ha empleado en este Proyecto.

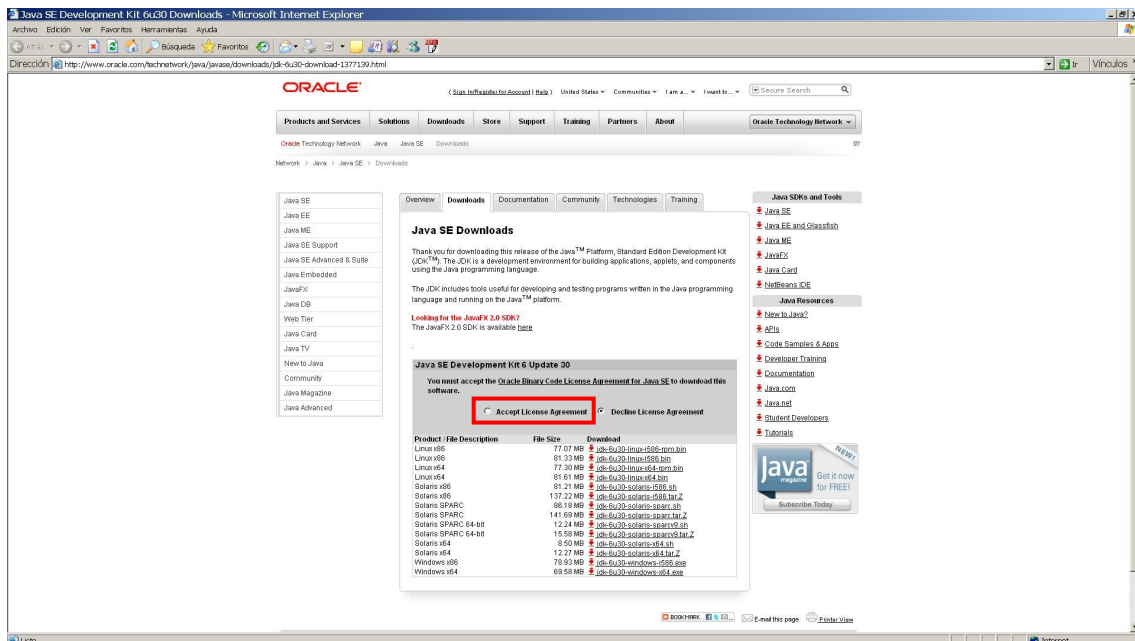


Ilustración L: [JDK-6u30](#)



- 2) A continuación, tenemos que descargar y descomprimir Eclipse Classic 3.7.2 buscando la versión correspondiente a nuestro SO (32/64 bits).

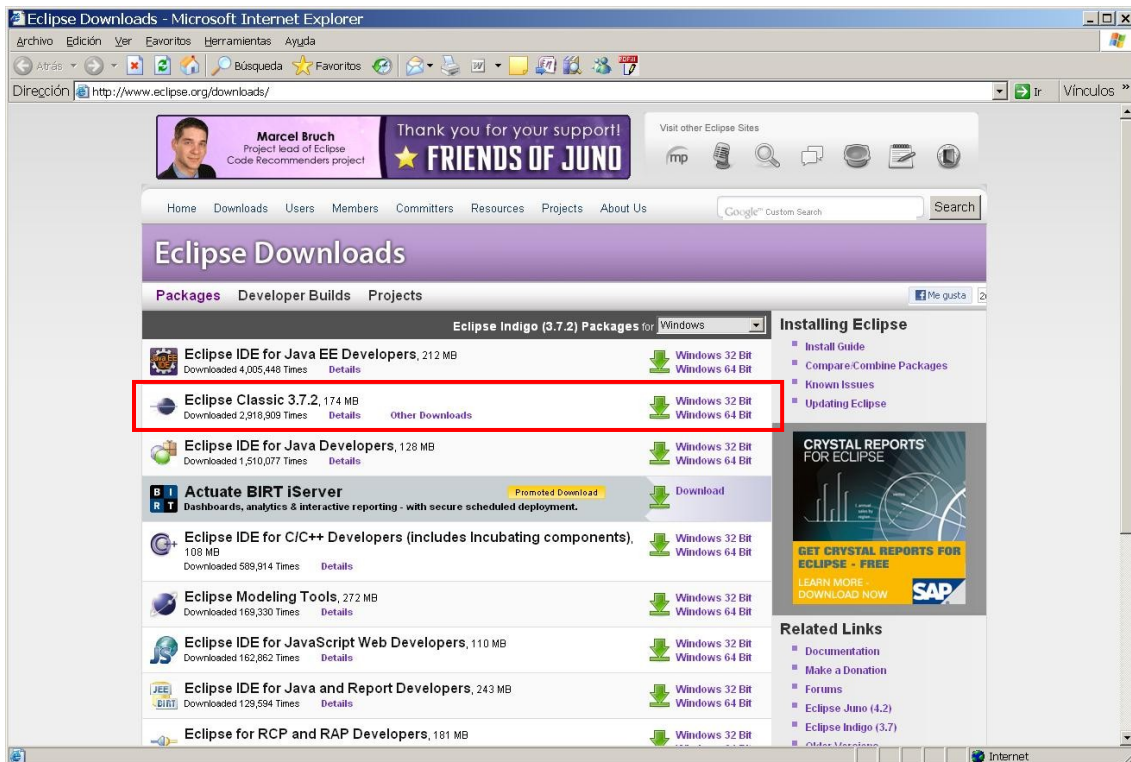


Ilustración M: [Eclipse Classic 3.7.2](#)

- 3) Posteriormente, descargamos e instalamos Android SDK.

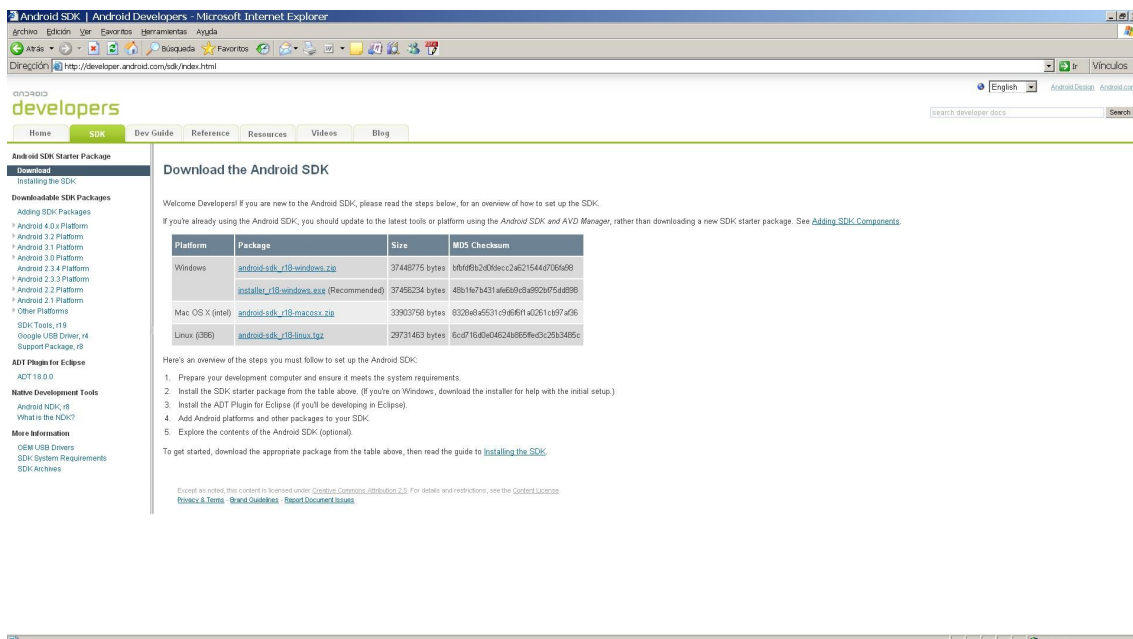


Ilustración N: [Android SDK](#)



- 4) Al terminar de instalarse tenemos que descargar las plataformas que deseamos manipular, es decir, sobre las cuales correrán nuestras aplicaciones. En nuestro caso, bajamos las versiones 2.1, la 2.2 y la 2.3.3, para hacer pruebas diversas.

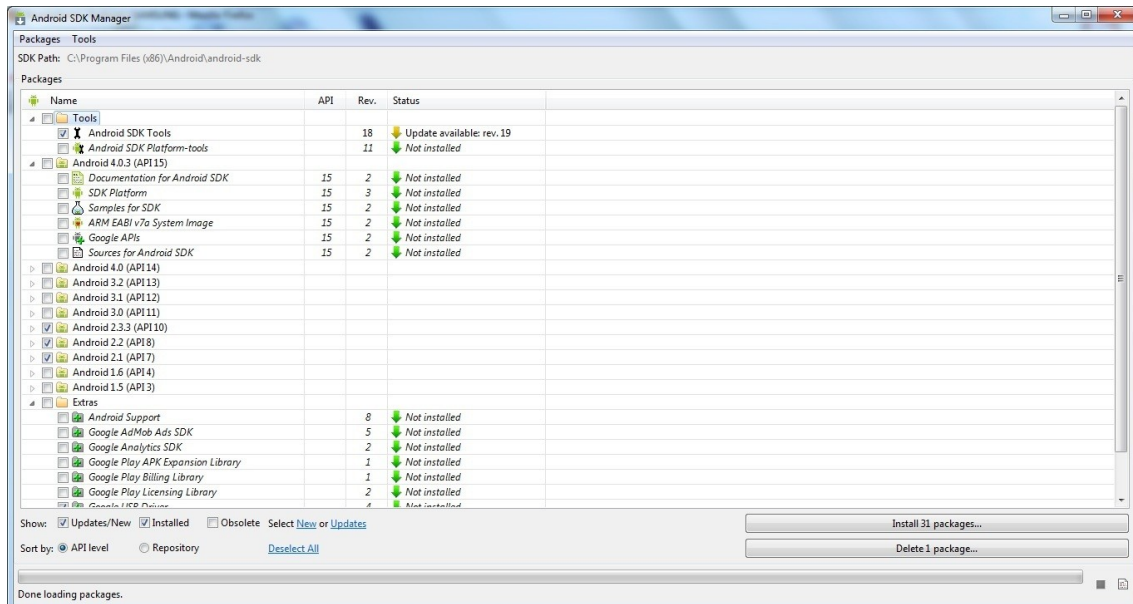


Ilustración O: Descarga de plataformas Android SDK

- 5) Instalamos el Android ADT plugin para Eclipse:

- a. En Eclipse, clic **Help** --> **Install New Software**

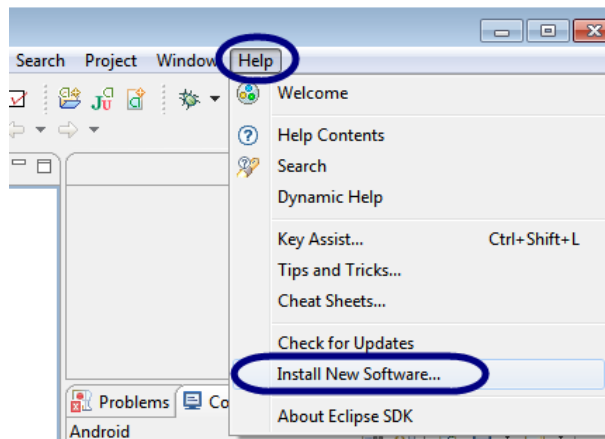


Ilustración P: Install New Software

- b. Clic **Add**.

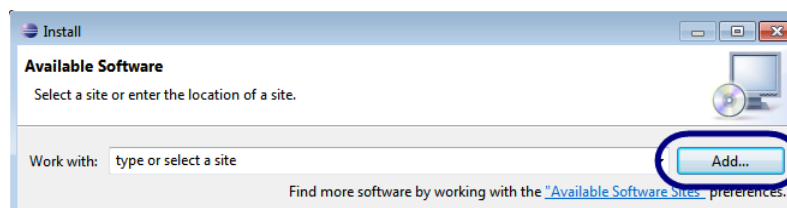


Ilustración Q: Add



- c. Introducimos ADT Plugin y <https://dl-ssl.google.com/android/eclipse>.
clic **OK**.

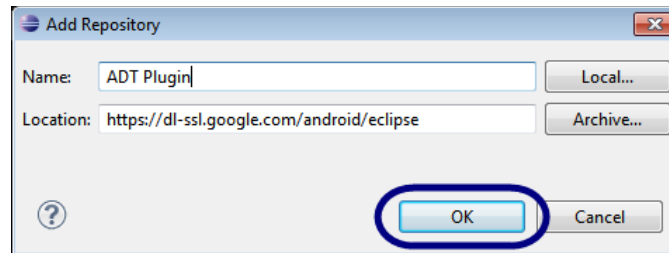


Ilustración R: URL Plugin

- d. Marcamos el check Developer Tools y clic **Next**.

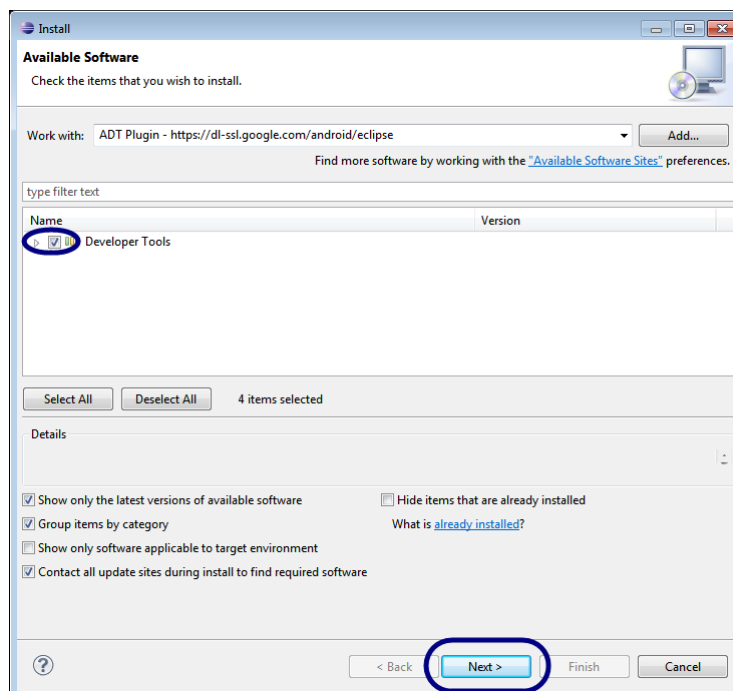


Ilustración S: Marcar check Developer Tools

- e. Clic **Next**.

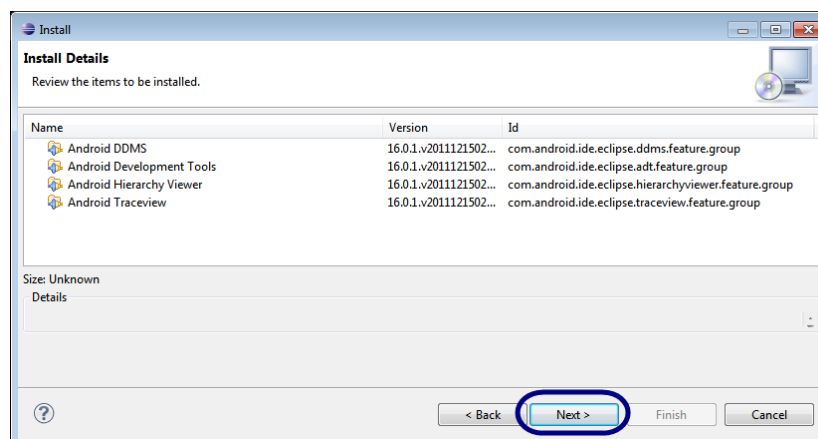


Ilustración T: Instalador plugin de Eclipse



- f. Aceptamos los términos y condiciones y clic **Finish**.

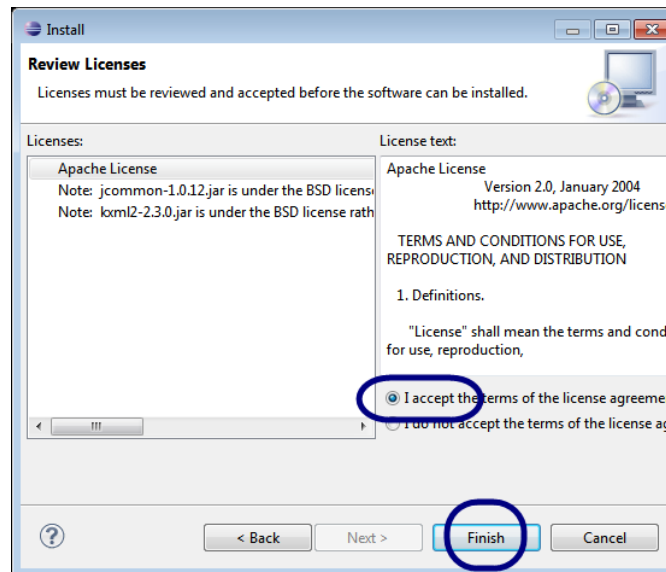


Ilustración U: Aceptar y Finalizar

- g. Cuando solicite reiniciar Eclipse SDK, clic **Restart Now**.

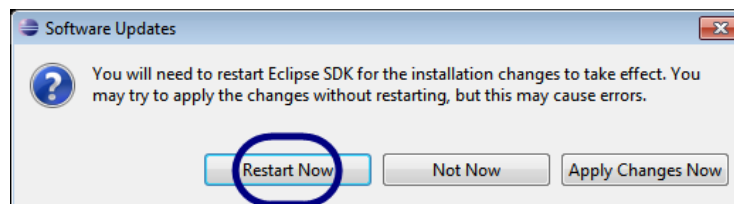


Ilustración V: Reiniciar Eclipse

- 6) Creamos AVD (Android Virtual Devices). Los AVD son las máquinas virtuales que nos permiten probar versiones específicas de Android así como configuraciones de hardware.

- a. Abrimos el SDK manager de Android buscamos la opción "Tools" y seleccionamos "manage AVDs..."

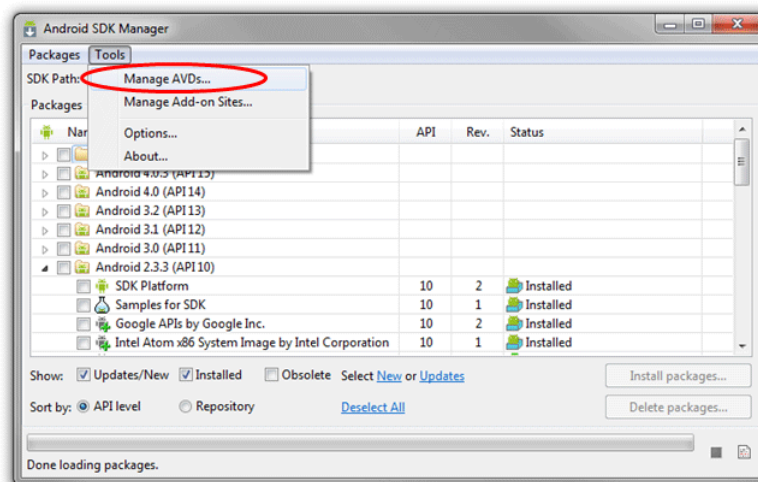


Ilustración W: Manager AVD



- b. El administrador de AVD's se mostrará a continuación. Para crear un nuevo AVD presionamos el botón "New"

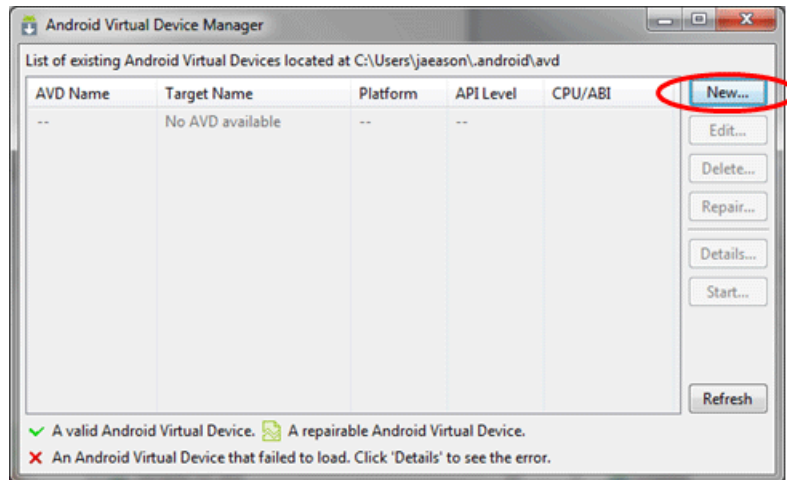


Ilustración X: New AVD

- c. Escribimos el nombre para el nuevo AVD y seleccionamos de la lista "target" la opción por ejemplo: "Android 2.3.3 - API Level 10. En "SD Card" indicamos que el tamaño (Size) es de 512Mb y en "Skin" indicamos en Built-in la opción por defecto: "Default WVGA800". Finalmente para guardar el nuevo dispositivo virtual pulsamos en "Create AVD".

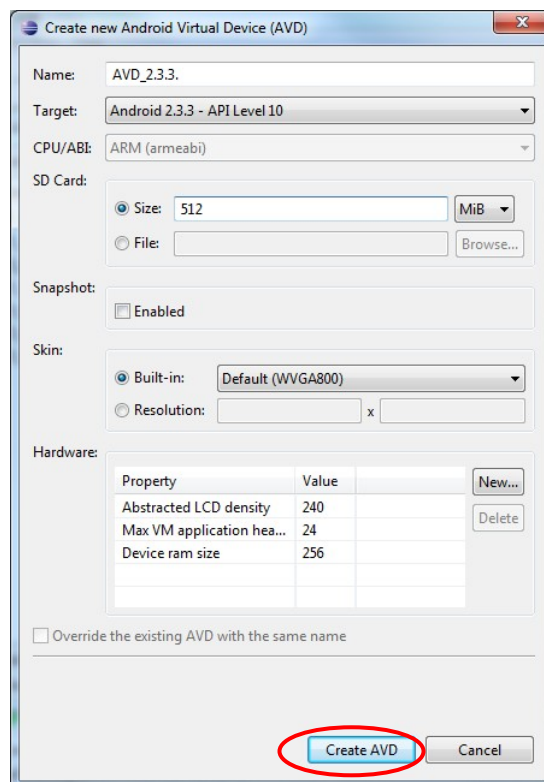


Ilustración Y: Creación de un AVD



- d. El nuevo dispositivo virtual AVD_2.3.3 ya está disponible para probar tus aplicaciones, sólo tienes que seleccionarlo y pulsar el botón "Start".

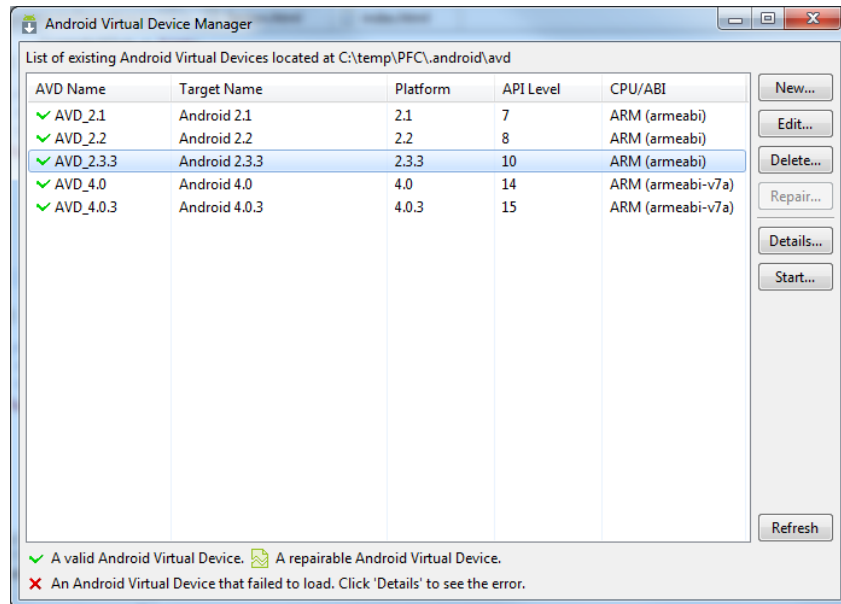


Ilustración Z: Iniciar un AVD

7) Descargar y descomprimir PhoneGap.



Ilustración AA: [PhoneGap](http://phonegap.com)



8) Instalamos el MDS AppLaud 1.2 para PhoneGap (Android plugin PhoneGap para Eclipse):

- a. En Eclipse, clic **Help** --> **Install New Software**

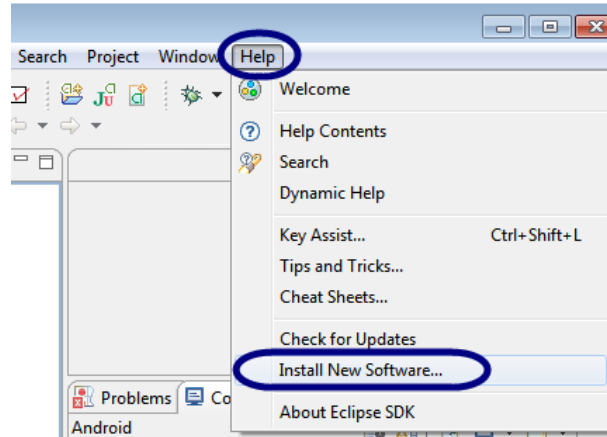


Ilustración BB: Install New Software

- b. Clic **Add**.

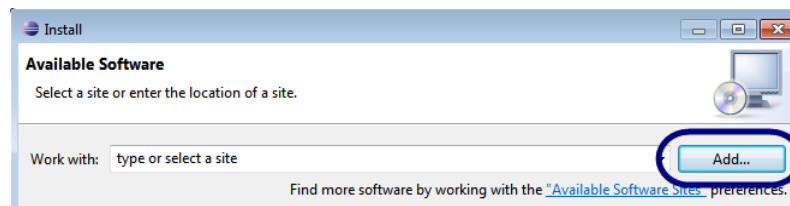


Ilustración CC: Add

- c. Introducimos en Name MDS AppLaud 1.2 y en Location: <http://svn.codespot.com/a/eclipselabs.org/mobile-web-development-with-phonegap/tags/r1.2/download> clic **OK**.

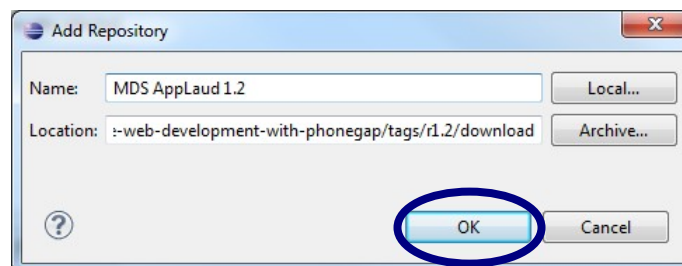


Ilustración DD: URL Plugin



d. Marcamos el check MDS AppLaud y clic **Next**.

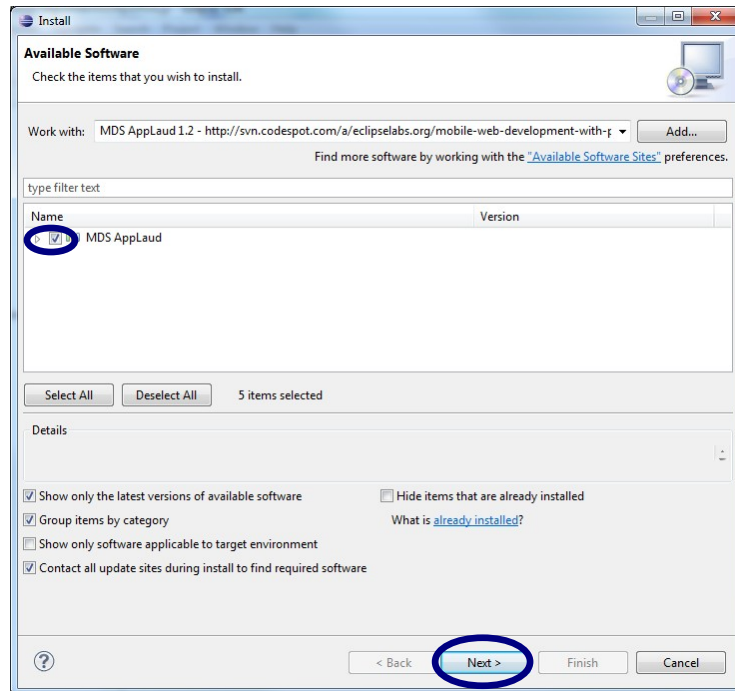


Ilustración EE: Marcar check MDS AppLaud

e. Clic **Next**.

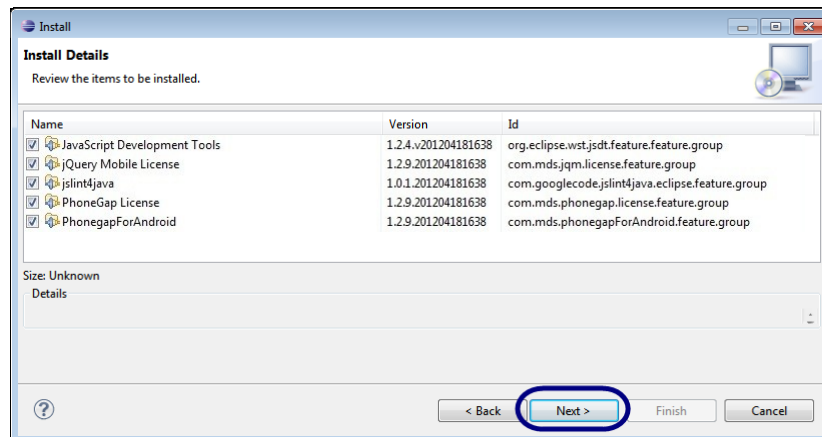


Ilustración FF: Instalador plugin de Eclipse

f. Cuando solicite reiniciar Eclipse SDK, clic **Restart Now**.

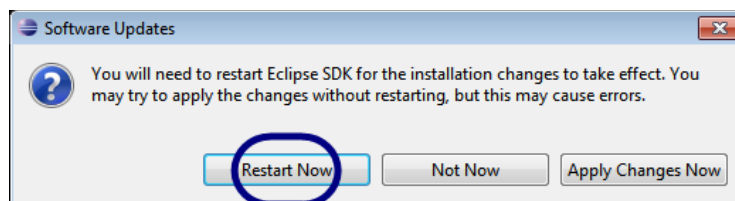


Ilustración GG: Reiniciar Eclipse



9.2 Manual de usuario de Seguimiento Deportivo

La página de Inicio es la primera página que se carga al iniciar la aplicación Seguimiento Deportivo. En la siguiente Figura se puede ver, esta página. Se dispone de un menú con dos opciones "Nuevo entreno" y "Diario de entrenos".

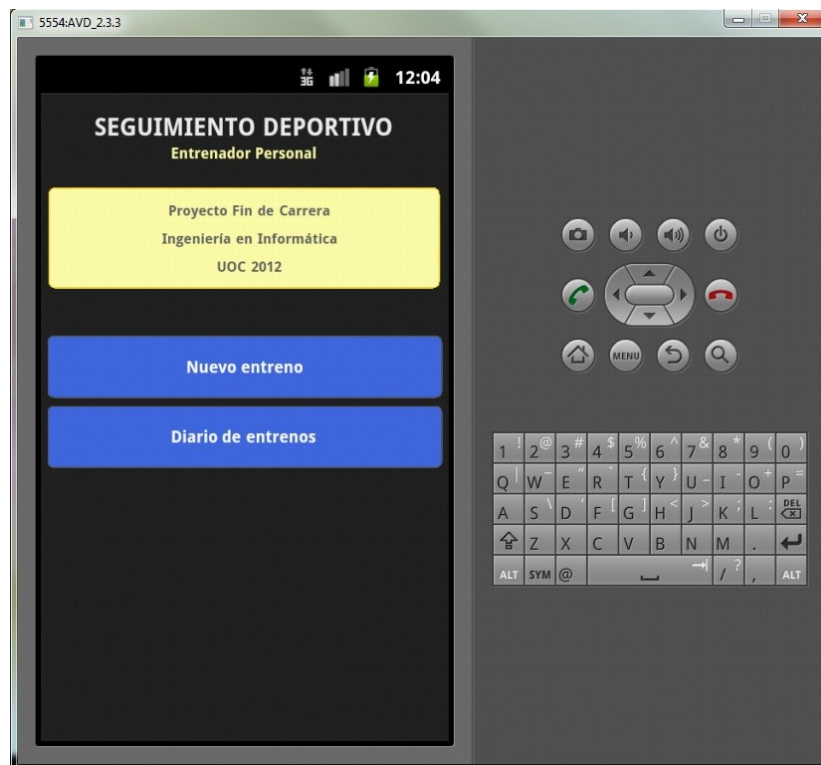


Ilustración HH: Pantalla Inicio



- 1) Si se selecciona **"Nuevo entreno"**, nos lleva a la pantalla Sesión de la aplicación, donde se muestran todos los datos de la sesión de entreno (Ritmo medio, Ritmo que corresponde al ritmo instantáneo, Altitud, Distancia y Tiempo) a parte se dispone de tres botones "Iniciar", "Pausa" y "Atrás".

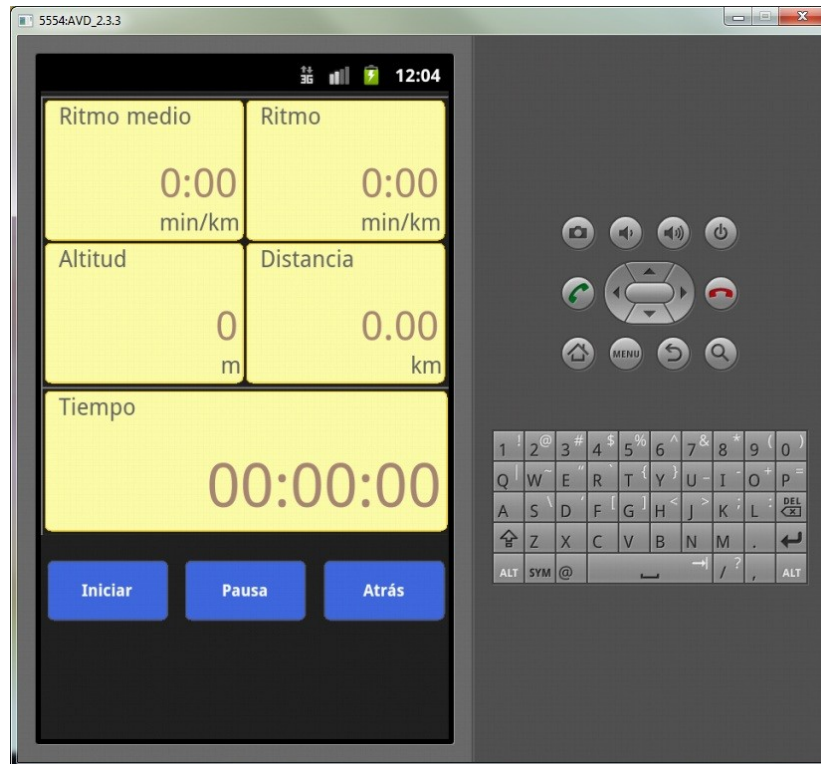


Ilustración II: Pantalla Sesión

Si se pulsa el botón "Pausa", no hace nada, ya que aún no se ha iniciado el entreno.



Lo primero que intentará el programa independientemente de que se pulse el botón "Iniciar" es sincronizar el GPS, avisando mediante el mensaje siguiente.

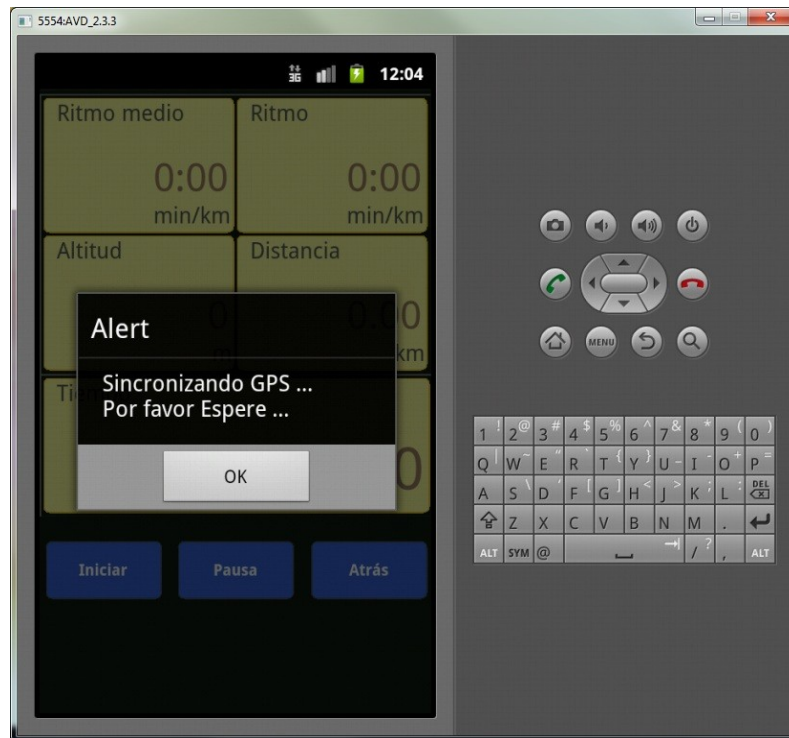


Ilustración JJ: Sincronizando GPS

Una vez esté sincronizado el GPS se avisará con el correspondiente mensaje.

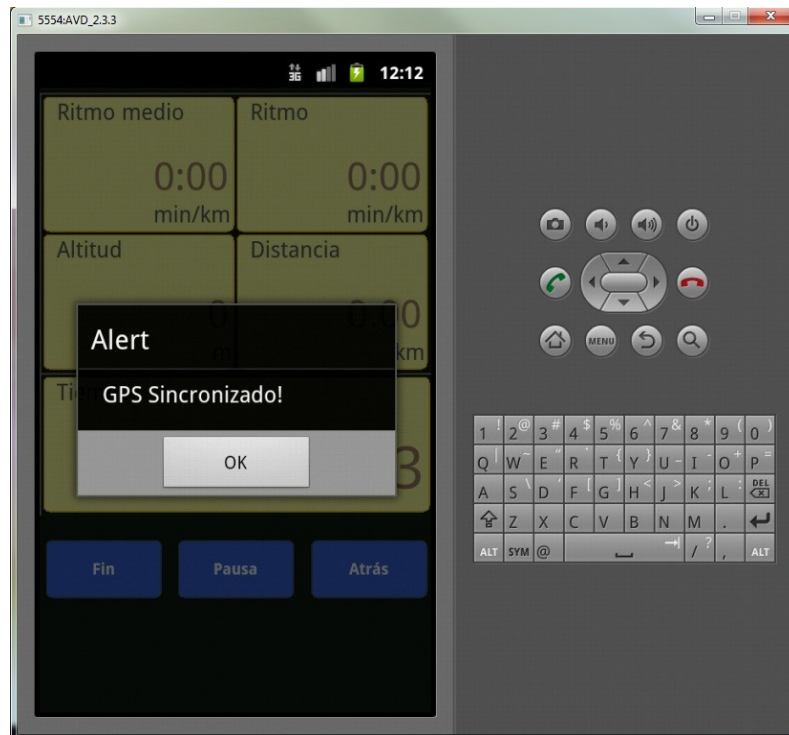


Ilustración KK: GPS Sincronizado



Si se pulsa el botón "Iniciar" comienza el entreno, en la pantalla empiezan a actualizarse los valores, que estén sufriendo cambios y se cambia el botón "Iniciar" por "Fin". Si no se ha esperado a que el GPS haya sincronizado, sólo cambiara el Tiempo, como el caso de la figura siguiente.

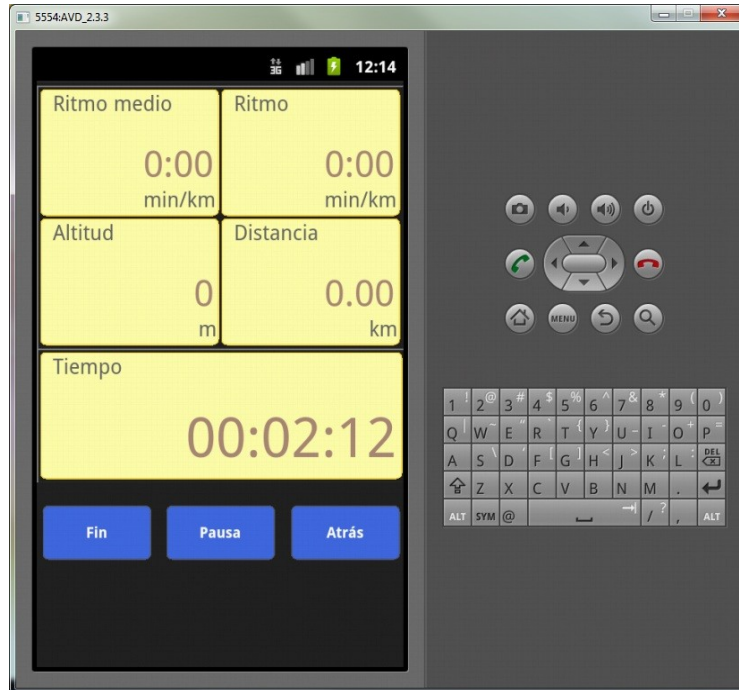


Ilustración LL: Sesión Iniciada

Si se pulsa el botón "Pausa" se mantiene pausado el entreno y se cambia el botón "Pausa" por "Continuar".

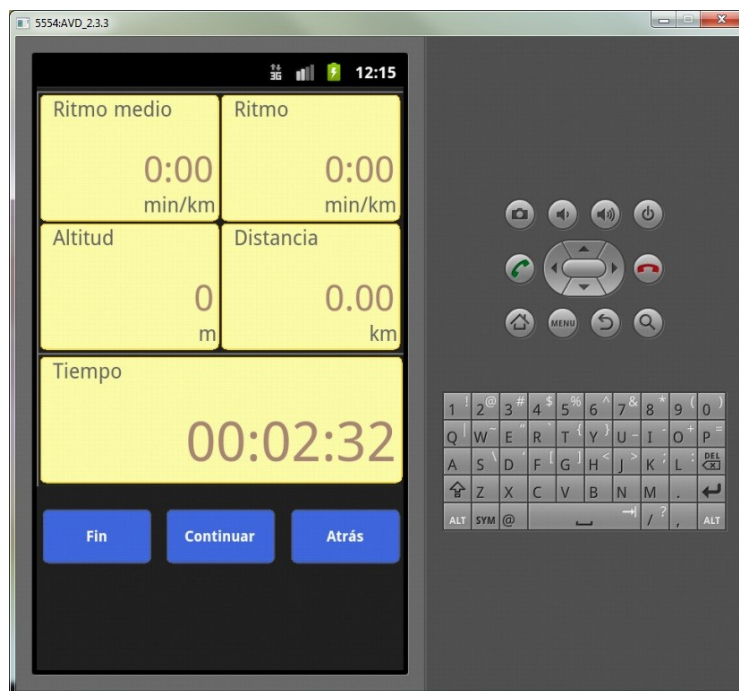


Ilustración MM: Sesión Pausada



Si se pulsa el botón "Continuar" se reanuda el entreno y se cambia el botón "Continuar" por "Pausa". Y así tantas veces se quiera.

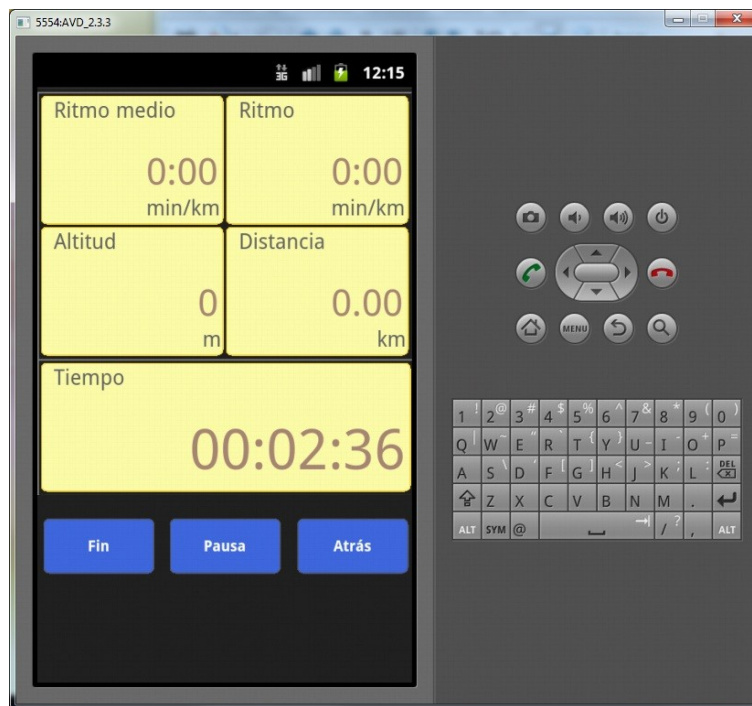


Ilustración NN: Sesión Continuada

Si se pulsa el botón "Fin" termina la sesión de entreno, guardándose los datos en la BD. Los botones "Fin" y "Pausa" o "Continuar", depende de ¿en qué situación estén?, se desactivan, sólo queda activo el botón "Atrás", que si se pulsa llevará de nuevo a la pantalla de Inicio.

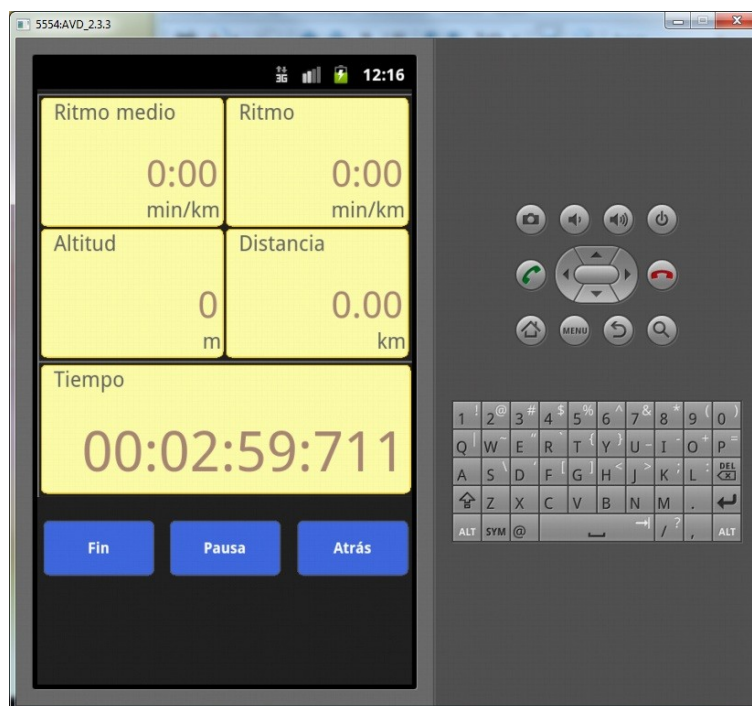


Ilustración OO: Sesión Finalizada



- 2) Si se selecciona "**Diario de entrenos**" nos lleva a la pantalla de "Listado Sesiones", donde se muestra un Diario de entrenos ordenados por fecha de los más recientes a los más antiguos. Por cada sesión de Entreno, se tienen las opciones "VER SESIÓN" y "BORRAR SESIÓN". A parte se dispone de los botones "Borrar Todo" y "Atrás". Si se pulsa el botón Atrás nos llevara a la pantalla de Inicio.



Ilustración PP: Listado de Sesiones



Si se pulsa "VER SESIÓN" por ejemplo de la primera sesión, nos lleva a la pantalla "Detalle Sesión", donde muestra un resumen de la sesión de entreno seleccionada. El botón Atrás si se pulsa nos llevará a la pantalla anterior "Listado de Sesiones".

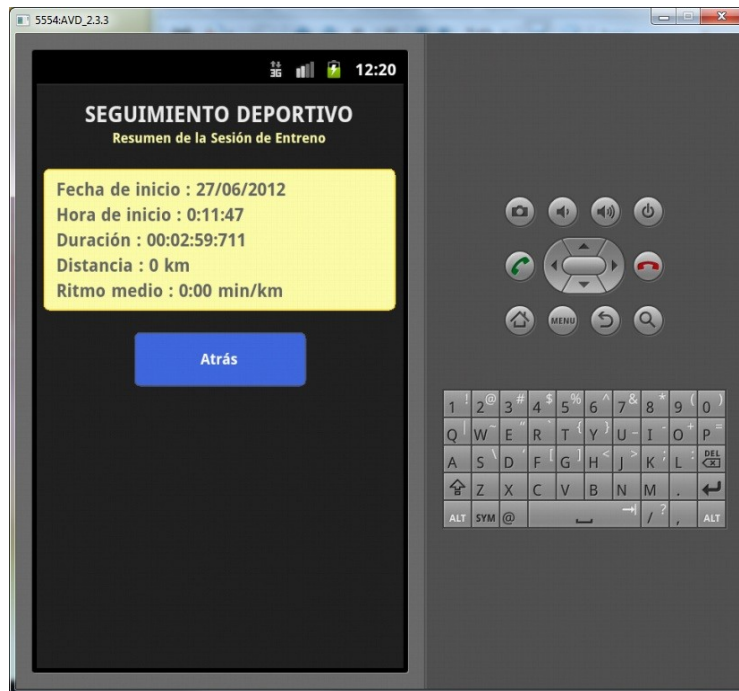


Ilustración QQ: Detalle Sesión

Si se pulsa "BORRAR SESIÓN" de una sesión, se muestra un mensaje de confirmación de Borrado. Sólo la borrará si se pulsa en OK.

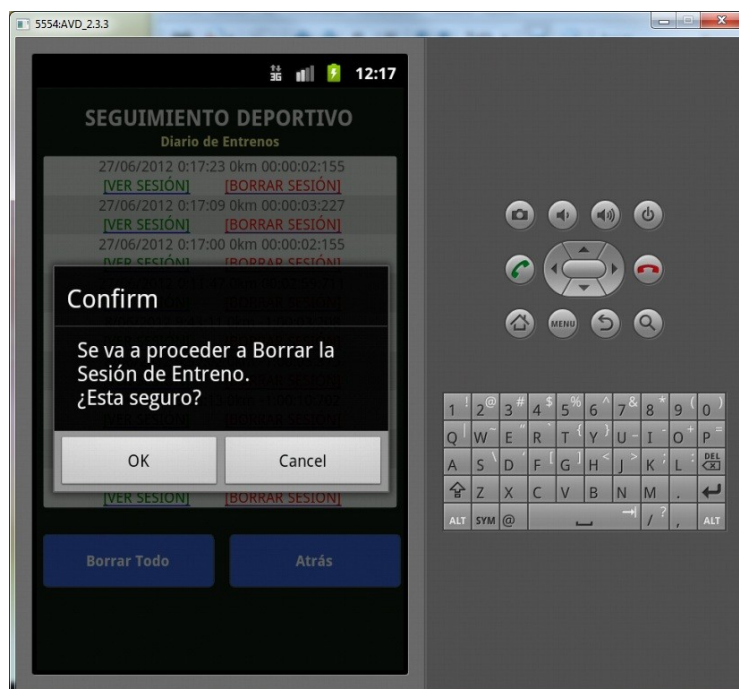


Ilustración RR: Mensaje de Confirmación de Borrado



Si se pulsa "Borrar Todo", se muestra un mensaje de confirmación de Borrado. Sólo se borrará si se pulsa en OK.

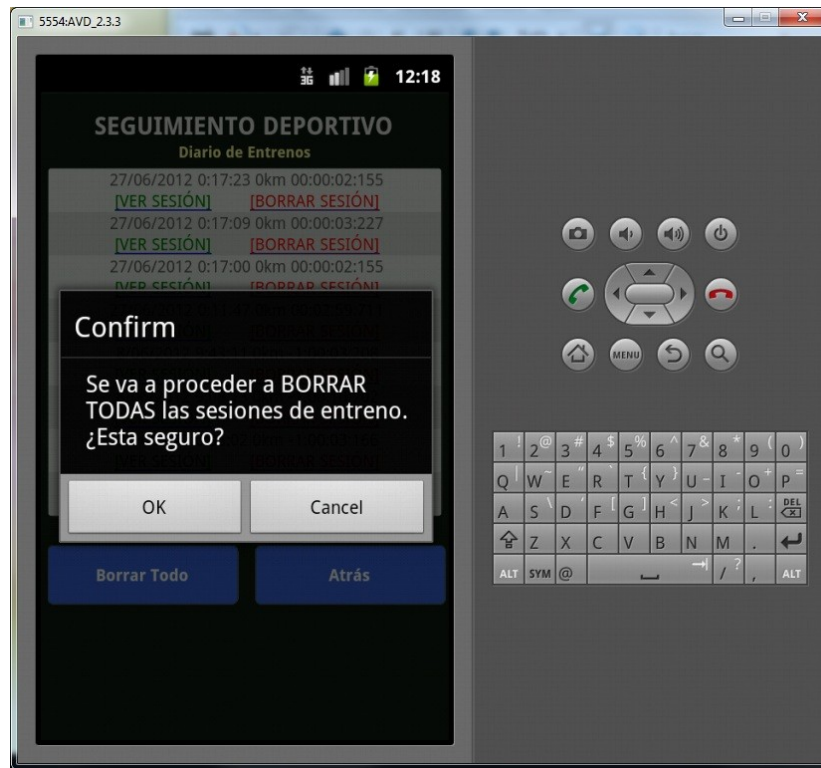


Ilustración SS: Mensaje de Confirmación de Borrado de Todas las sesiones