



UNIVERSIDAD DE GRANADA

ESCUELA TÉCNICA SUPERIOR DE  
INGENIERÍAS INFORMÁTICA Y  
TELECOMUNICACIONES

DEPARTAMENTO DE LENGUAJES Y SISTEMAS  
INFORMATICOS

MASTER UNIVERSITARIO EN DESARROLLO DE SOFTWARE

**SISTEMA DE ENTRENAMIENTO ASISTIDO  
PARA CORREDORES DE LARGA DISTANCIA  
EN DISPOSITIVOS MÓVILES**

SAMUEL SANDRU

DIRECCIÓN ACADÉMICA: MIGUEL J. HORNOS BARRANCO

JULIO 2011



*Agradecimientos*

*A mi familia,  
por su apoyo y ánimo durante el proyecto y no solo.*

*A mi tutor Miguel J. Hornos Barranco  
por su dedicación e interés demostrado.*

*A mis compañeros y amigos que me han aguantado y ayudado en muchos momentos.*



# Índice:

<b>1. INTRODUCCIÓN .....</b>	<b>11</b>
1.1. OBJETIVOS GENERALES .....	11
1.2. OBJETIVOS ESPECÍFICOS .....	11
<b>2. CONSIDERACIONES TÉCNICAS PARA CORREDORES DE LARGA DISTANCIA .....</b>	<b>15</b>
<b>3. COMPUTACIÓN UBICUA, MÓVIL Y SENSIBLE AL CONTEXTO .....</b>	<b>25</b>
<b>4. REDES DE TELEFONÍA MÓVIL: CONCEPTO, TÉRMINOS Y GENERACIONES.....</b>	<b>31</b>
4.1. TELEFONÍA MÓVIL DE PRIMERA GENERACIÓN O 1G.....	32
4.2. TELEFONÍA MÓVIL DE SEGUNDA GENERACIÓN O 2G.....	33
4.2.1. <i>Servicio general de paquetes vía radio y EDGE o 2.5G</i> .....	34
4.3. TELEFONÍA MÓVIL DE TERCERA GENERACIÓN O 3G .....	36
4.3.1. <i>CDMA2000 y UMTS</i> .....	36
4.3.2. <i>IP móvil</i> .....	38
4.3.3. <i>Avances hacia la tecnología de cuarta generación</i> .....	39
4.4. TELEFONÍA MÓVIL DE CUARTA GENERACIÓN O 4G .....	39
<b>5. EVOLUCIÓN DEL SOFTWARE PARA DISPOSITIVOS MÓVILES .....</b>	<b>41</b>
<b>6. MÉTODOS DE LOCALIZACIÓN EN DISPOSITIVOS MÓVILES.....</b>	<b>49</b>
6.1. LOCALIZACIÓN POR ID DE CELDA, TDOA, TOA, AOA.....	51
6.2. GPS .....	52
6.3. AGPS O GPS ASISTIDO .....	54
6.4. POSICIONAMIENTO WI-FI .....	55
<b>7. ARQUITECTURA ORIENTADA A SERVICIOS .....</b>	<b>57</b>
7.1. SOAP.....	59
7.2. REST.....	60
7.3. ARQUITECTURA CLIENTE-SERVIDOR EN DISPOSITIVOS MÓVILES .....	61
<b>8. MODELO DE USUARIO.....</b>	<b>67</b>
<b>9. TRABAJOS RELACIONADOS .....</b>	<b>71</b>
9.1. TRABAJOS INVESTIGADORES EN EL ÁREA.....	71
9.2. APLICACIONES COMERCIALES .....	72
9.3. CONCLUSIONES .....	77
<b>10. MI PROPUESTA .....</b>	<b>79</b>
10.1. DESCRIPCIÓN DE LA PROPUESTA .....	79
10.1.1. <i>Módulo de planificaciones</i> .....	79
10.1.2. <i>Módulo de recomendación de trayectos</i> .....	81
10.1.3. <i>Otras consideraciones</i> .....	83

10.2. MÉTODOS DE LOCALIZACIÓN .....	84
10.3. MODELO DE USUARIO .....	87
10.4. DEFINICIÓN DE LA ARQUITECTURA .....	89
<b>11. PROCESO DE DESARROLLO DE SOFTWARE.....</b>	<b>93</b>
11.1. INGENIERÍA DE REQUISITOS.....	95
11.1.1. <i>Estudio de viabilidad</i> .....	95
11.1.2. <i>Análisis de requisitos</i> .....	96
11.1.3. <i>Requisitos no funcionales</i> .....	99
11.1.4. <i>Definición de requisitos funcionales</i> .....	99
11.1.5. <i>Modelo de casos de uso</i> .....	101
11.1.6. <i>Modelo de dominio</i> .....	102
11.2. DISEÑO DEL SISTEMA.....	103
11.2.1. <i>Modelo de la interfaz de usuario</i> .....	103
11.2.2. <i>Modelo de diseño de la lógica de negocio</i> .....	106
11.2.3. <i>Modelo de diseño de los datos</i> .....	112
11.3. IMPLEMENTACIÓN .....	114
11.3.1. <i>Principales tecnologías utilizadas</i> .....	114
11.3.2. <i>Otras tecnologías utilizadas</i> .....	114
11.4. PRUEBAS Y DESPLIEGUE .....	115
<b>12. CONCLUSIONES Y TRABAJOS FUTUROS.....</b>	<b>117</b>
12.1. CONCLUSIONES .....	117
12.2. TRABAJO FUTURO .....	119
<b>13. BIBLIOGRAFÍA .....</b>	<b>121</b>
<b>ANEXO 1. ESPECIFICACIÓN CASOS DE USO .....</b>	<b>1216</b>

## Índice de figuras:

FIGURA 2.1. TIPO DE ZAPATILLAS ADECUADAS	16
FIGURA 3.1. CONTEXTO PRIMARIO Y SUS INTERDEPENDENCIAS	27
FIGURA 4.1. CELDAS AGRUPADAS EN 2 CLÚSTERES DE 7	32
FIGURA 4.2. ARQUITECTURA GPRS	35
FIGURA 5.1. EVOLUCIÓN DE LOS SERVICIOS Y APLICACIONES MÓVILES	42
FIGURA 5.2. ESTRUCTURA JAVA ME	43
FIGURA 5.3. ARQUITECTURA DEL SO ANDROID	45
FIGURA 5.4. CUOTA DE MERCADO EN SMARTPHONES SEGÚN SO	46
FIGURA 5.5. APLICACIÓN WEB MÓVIL	457
FIGURA 5.6. APLICACIÓN NATIVA	46
FIGURA 6.1. ÁNGULO DE LLEGADA DE UNA SEÑAL A UN VECTOR DE SENSORES	49
FIGURA 6.2. FUNCIONAMIENTO DE TOA CON 3 BASES	50
FIGURA 6.3. FUNCIONAMIENTO DE LA TECNOLOGÍA TDOA	51
FIGURA 7.1. ARQUITECTURA CLIENTE-SERVIDOR DE 3 CAPAS Y SERVIDOR DE 2 NIVELES	63
FIGURA 9.1. CAPTURAS DE PANTALLA DE LA APLICACIÓN RUNKEEPER	73
FIGURA 9.2. CAPTURAS DE PANTALLA DE LA APLICACIÓN MAPMYRUN	75
FIGURA 9.3. CAPTURAS DE PANTALLA DE LA APLICACIÓN SMART COACH	76
FIGURA 10.1. RELACIÓN DE CONSUMO POR SENSOR	86
FIGURA 10.2. CONSUMO DEL SENSOR GPS	86
FIGURA 10.3. ARQUITECTURA DEL SISTEMA	91
FIGURA 11.1. DIAGRAMA DE CASOS DE USO	101
FIGURA 11.2. MODELO DE DOMINIO	102
FIGURA 11.3. BOCETO PARA LA PANTALLA INICIAL DEL PLANIFICADOR	103
FIGURA 11.4. BOCETO PARA LA PANTALLA INTERMEDIA Y FINAL DEL PLANIFICADOR	103
FIGURA 11.5. BOCETO PARA LAS PANTALLAS DE SESIONES Y DE RECOMENDACIÓN DE TRAYECTOS	10345
FIGURA 11.6. BOCETOS PARA LAS PANTALLAS DE PARÁMETROS DE LA SESIÓN Y DE TRAYECTO ACTUAL	105
FIGURA 11.7. BOCETO PARA LA PANTALLA DE GUARDAR EL TRAYECTO ACTUAL	105
FIGURA 11.8. DIAGRAMA DE FLUJOS DE DATOS GENERAL	106
FIGURA 11.9. DIAGRAMA DE FLUJO DE DATOS PARA EL PROCESO VER PLANIFICACIÓN	107
FIGURA 11.10. DIAGRAMA DE FLUJOS DE DATOS PARA EL PROCESO CREAR PLANIFICACIÓN	108
FIGURA 11.11. DIAGRAMA DE FLUJOS DE DATOS PARA EL PROCESO VER TRAYECTOS	109
FIGURA 11.12. DIAGRAMA DE FLUJOS DE DATOS PARA EL PROCESO VER TRAYECTOS RECOMENDADOS	109
FIGURA 11.13. DIAGRAMA DE FLUJOS DE DATOS PARA EL PROCESO COMENZAR SESIÓN	110
FIGURA 11.14. MODELO DE ENTIDADES DE NEGOCIO	111
FIGURA 11.15. COMPONENTES DE ACCESO A DATOS	112
FIGURA 11.16 VISTA CONCEPTUAL DEL MODELO DE DATOS	113
FIGURA 11.17. VISTA FÍSICA DEL MODELO DE DATOS	113





## Índice de tablas:

TABLA 2.1. RELACIÓN ENTRE VDOT Y TIEMPO DE CARRERA (PARTE I) .....	18
TABLA 2.2. RELACIÓN ENTRE VDOT Y TIEMPO DE CARRERA (PARTE II) .....	19
TABLA 2.3. RELACIÓN ENTRE VDOT Y RITMOS DE SESIÓN (PARTE I).....	20
TABLA 2.4. RELACIÓN ENTRE VDOT Y RITMO DE SESIÓN (PARTE II) .....	21
TABLA 2.5. EJEMPLO DE PLANIFICACIÓN PARA UN MARATÓN.....	24
TABLA 6.1. PRECISIÓN DE LOS DISTINTOS MÉTODOS DE LOCALIZACIÓN .....	56
TABLA 7.1. SOBRES SOAP, PETICIÓN (IZQUIERDA) Y RESPUESTA (DERECHA) .....	59
TABLA 10.1. RANGO SEMANAL Y FRECUENCIA.....	80



# 1. Introducción

Para muchas personas hoy en día el ejercicio físico es una parte vital de su salud mental y física. Hay un cierto deporte que cada vez más acapara más adeptos, correr. Existen personas que realizan este deporte por mantenerse en forma, para otros está llegando a ser un hobby, mientras que para algunos puede ser una obsesión.

Las últimas dos categorías de corredores son las que suelen estar más dispuestas a gastar parte de su dinero en equipamiento deportivo y distintas herramientas que hagan su actividad más agradable. Algunas de las herramientas que los deportistas usan son muy caras, como los cronómetros o los monitores cardíacos, que pueden costar varios cientos de euros. Algunas personas piensan que este gasto puede ser reducido con la creciente entrada de dispositivos móviles, especialmente los *smartphones*, en nuestras vidas, creando aplicaciones que sustituyan y mejoren las herramientas que actualmente se utilizan.

Éste es el entorno en el cual nos situamos a la hora de realizar una propuesta que ayude a los corredores de larga distancia a entrenar para una determinada carrera. El interés por desarrollar un sistema de estas características viene motivado además por el intento de evitar las típicas lesiones tan comunes en este deporte. La propuesta intentará guiarnos de forma correcta durante la preparación de una carrera de larga distancia, como es un maratón.

## 1.1. Objetivos generales

El objetivo del trabajo consiste en analizar, diseñar e implementar un sistema de entrenamiento inteligente que ayude al usuario ofreciendo consejos de entrenamiento, recomiende trayectos y realice un seguimiento durante la preparación de una carrera de larga distancia, en nuestro caso, un maratón.

## 1.2. Objetivos específicos

Una de las principales características de este sistema será su capacidad para ofrecer una planificación correcta en función de ciertos parámetros, como tiempo de la última sesión de correr, kilometraje semanal y fecha del maratón que se piensa correr. La

planificación será capaz de asesorarnos sobre los parámetros que hay que seguir en cada sesión, para así poder encontrar la intensidad adecuada, intentando de esta forma evitar lesiones y hacer el esfuerzo más llevadero.

Aparte de la planificación, el sistema nos ofrecerá una recomendación de trayectos mediante búsqueda automática teniendo en cuenta los parámetros de la sesión actual. Para ello, nos ofrecerá una interfaz inteligente, automática e intuitiva que, en función de nuestra situación geográfica, nos propondrá las rutas que mejor se amolden a nuestras necesidades. Las rutas se crean a medida que los usuarios utilizan la aplicación y tendrá un carácter colaborativo.

Una vez comenzada la sesión, habrá un módulo encargado de ofrecernos varios parámetros de crucial importancia a la hora del ejercicio, como son el ritmo o la distancia. También se ofrece la posibilidad de realizar sesiones libres, fuera de la planificación, para que los corredores esporádicos también puedan hacer uso de la aplicación que se presenta.

En base a lo anteriormente descrito, se pueden resumir los objetivos específicos (OE) de nuestro trabajo en los siguientes:

- OE-1: Ofrecer una planificación personalizada para corredores de larga distancia.
- OE-2: Monitorizar los parámetros de sesión.
- OE-3: Controlar el ritmo, con el propósito de evitar lesiones.
- OE-4: Almacenar los trayectos realizados por los usuarios, junto con sus parámetros.
- OE-5: Recomendar trayectos de forma sensible al contexto.

Estos objetivos deberán basarse en un previo estudio del estado del arte en las siguientes áreas, por lo que consideraremos los siguientes puntos como necesidades específicas (NE):

- NE-1: Entrenamiento para corredores de larga distancia.
- NE-2: Computación móvil y sensibilidad al contexto.
- NE-3: Análisis de la tecnología móvil actual y las restricciones que impone en el desarrollo software.
- NE-4: Métodos de localización en dispositivos móviles.

- NE-5: Arquitectura orientada a servicios.
- NE-6: Modelo de usuario.
- NE-7: Trabajos relacionados.

En base a estas ideas se ha hecho un estudio y análisis, tanto en el mundo de la investigación como en el mundo comercial, para observar las distintas aplicaciones propuestas relacionadas con el mundo del deporte. En la búsqueda se han observado las funcionalidades que estas aplicaciones ofrecen y se ha concluido que muchos de los objetivos que nosotros tenemos no se cumplen en su totalidad o de forma parcial. Debido al vacío existente en este campo, este trabajo intenta proponer un sistema innovador, profesional y de real utilidad que tenga en cuenta las necesidades de los corredores de larga distancia.

El resto de la memoria se estructurará en dos partes o bloques. La primera parte analizará el estado del arte de los conceptos y tecnologías que intervienen en nuestra propuesta, y está formada por los siguientes capítulos:

- *Capítulo 2. Consideraciones técnicas para corredores de larga distancia.*  
Proporciona al lector una base sobre el entrenamiento de los corredores de larga distancia.
- *Capítulo 3. Computación ubicua, móvil y sensible al contexto.*  
Ofrece una breve descripción de los aspectos de la Computación ubicua, móvil y sensible al contexto que influyen en nuestro trabajo.
- *Capítulo 4. Redes de telefonía móvil: concepto, términos y generaciones.*  
Presenta las distintas generaciones de tecnología móvil, así como su evolución y características más importantes.
- *Capítulo 5. Evolución del software para dispositivos móviles.*  
Resume la evolución del software para dispositivos móviles y su estrecha relación con la evolución en la tecnología móvil.
- *Capítulo 6. Métodos de localización en dispositivos móviles.*  
Presenta los distintos métodos de localización disponibles para dispositivos móviles.
- *Capítulo 7. Arquitectura orientada a servicios.*  
Introduce los principios básicos de la arquitectura orientada a servicios, la arquitectura cliente-servidor y las distintas tecnologías que se pueden utilizar.
- *Capítulo 8. Modelo de usuario.*  
Ofrece una definición de los modelos de usuario y una descripción de los tipos de modelos más utilizados.

- *Capítulo 9. Trabajos relacionados.*  
Analiza los trabajos de investigación y propuestas comerciales relacionados con nuestro proyecto.

En la segunda parte se realizará un análisis de los conceptos vistos en el bloque anterior, se hará la propuesta del sistema y se describirá el proceso de desarrollo software seguido. Esta parte está formada por los siguientes capítulos:

- *Capítulo 10. Mi propuesta.*  
Se argumentan las decisiones tomadas para la propuesta, explicando las elecciones realizadas con respecto a las distintas opciones vistas en los capítulos anteriores para cada cuestión en particular.
- *Capítulo 11. Proceso de desarrollo de software.*  
Se presenta el proceso de desarrollo software seguido para el desarrollo del prototipo que se ha implementado.

## 2. Consideraciones técnicas para corredores de larga distancia

En este capítulo se realizará un repaso de los puntos más importantes a tener en cuenta en el deporte de correr para el caso de los corredores de larga distancia. Correr es el mejor deporte que una persona puede ejercer desde el punto de vista cardiovascular, y dentro de este deporte, ya sea por demostrar, por ambición u otros motivos, el maratón ocupa uno de los lugares más importantes.

Correr un maratón significa correr una distancia de 42.195 metros y, aunque es una distancia larga, hay muchas personas hoy en día que deciden completar este reto.

La encargada de imponer las reglas en el atletismo es la Asociación Internacional de Federaciones Atléticas (*IAAF – International Association of Athletics Federations*), que también se ocupa de la parte estadística que guarda los records. El actual récord del mundo en una carrera de maratón lo sustenta Patrick Makau Musyoki, de Kenia, con un tiempo de 2:03:38 horas, obtenido en el maratón de Berlín del 25 de Septiembre 2011 [1].

El récord anterior supone correr durante más de 2 horas a una velocidad de aproximadamente 21 km/h. Incluso para los atletas no profesionales, terminar un maratón en 3:30:00 h es una gran hazaña, dado que tendrán que correr durante 3 horas y 30 minutos a un ritmo medio de 12 km/h. ¿Cómo es posible realizar esto? El trabajo continuo y controlado es la respuesta. El entrenamiento en una carrera de esta amplitud es enorme, y por lo tanto requiere de una preparación avanzada, exacta y de larga duración.

El mayor desafío al que nos enfrentamos cuando decidimos correr un maratón es el modo de hacerlo, ya que en principio parece prácticamente imposible. Aparte de una condición física decente y unas zapatillas específicas diseñadas para este deporte, es absolutamente esencial el entrenamiento continuo.

En cuanto a las zapatillas, éstas se eligen en función de la huella que deja el pie. Así, para obtener nuestra huella, se puede realizar el test de los pies mojados [2], que realmente nos ayuda a darnos cuenta de la altura de nuestro arco plantar. Este test consiste en mojarse los pies, después de lo cual hay que colocarlos sobre un cartón de color marrón, teniendo una postura relajada y los pies abiertos, de forma que la línea imaginaria que une los pies y los hombros forme una recta perpendicular a la superficie del cartón. Después de 1 minuto, la huella que han dejado nuestros pies en el cartón nos ayudará a elegir las zapatillas adecuadas, tal y como se muestra en la Figura 2.1. Aunque este análisis tan simple

es suficiente en muchos de los casos, a veces es mejor acudir a un podólogo para que nos realice un análisis dinámico completo de nuestra forma de correr.



Figura 2.1. Tipo de zapatillas adecuadas<sup>1</sup>

En cuanto al segundo problema, el entrenamiento continuo, llega un momento en el que se tienen que correr decenas de kilómetros a la semana durante varios meses, este gran esfuerzo puede llevar muy fácilmente a lesiones, y cuando se está entrenando para un maratón, una lesión significa que todo el esfuerzo anterior y las ganas de superarse han sido en vano, ya que a este ritmo una lesión es sinónimo de retirada.

Las lesiones más comunes en este tipo de deporte son las relacionadas con la rodilla y los ligamentos que rodean el tobillo. A continuación vamos a ofrecer una visión general de las lesiones más comunes y los motivos que las provocan:

- *Síndrome de la cintilla iliotibial*  
Consiste en la inflamación de la cintilla iliotibial por el continuo rozamiento contra el epicóndilo lateral femoral. Puede ocurrir por alguno de los siguientes factores:
  - Distinta longitud de las piernas
  - Pies planos o zapatillas no adecuadas
  - Aumentos bruscos en el entrenamiento

<sup>1</sup> Fuente: [70]



- Correr sobre superficies duras o inclinadas

Para más información sobre este tipo de lesión se puede consultar [3].

- *Rodilla del corredor o síndrome del dolor patelofemoral*

Ocurre cuando la rótula irrita los ligamentos que hay alrededor del surco femoropatelar, sobre el cual descansa. Como motivos de esta lesión se pueden destacar:

- El sobreuso
- Mala alineación de la rótula
- Anomalías en los pies o calzado no adecuado

Se puede consultar más información sobre este tipo de lesión en [4].

- *Fascitis plantar*

Es la inflamación del tejido conectivo grueso en la zona del talón o zona media del pie. Como motivos principales de esta lesión se pueden considerar:

- Anomalías en los pies o calzado no adecuado
- Pronación excesiva

Se puede consultar más información sobre este tipo de lesión en [5].

En todas estas lesiones, como en muchas otras, las palabras claves son anomalías en los pies o el sobreuso/incremento brusco, y es que la mayoría de las lesiones en el caso de los corredores tienen 3 causas principales:

- Demasiada distancia.
- Demasiado pronto.
- Demasiado rápido.

Debido al incumplimiento de estas reglas esenciales, las lesiones ocurren. El sistema que se propone viene a solventar estos problemas, basándose para ello en la investigación del doctor Jack Daniels sobre la fórmula ideal de correr, investigación que está comprendida en varios de sus libros, de los cuales se puede destacar [6] y [7].

En su estudio, este autor intenta determinar la forma en la que se relaciona la cantidad inhalada de oxígeno ( $Vo_{2Max}$ ) con la velocidad y duración de las sesiones de correr. Después de medir esta cantidad de oxígeno inhalado a miles de atletas de distinto nivel y su rendimiento en carrera, el Dr. Daniels fue capaz de crear las siguientes tablas:

Tabla 2.1, Tabla 2.2, Tabla 2.3 y Tabla 2.4, que relacionan el ritmo en varios tipos de carrera con el nivel físico de cada atleta.

La relación entre  $Vo_{2Max}$  y los ritmos de sesión está intermediada por la constante VDOT, cuyos valores se pueden observar en la primera y última columna de las tablas.

A partir de la

Tabla 2.1 y Tabla 2.2 se puede averiguar la constante VDOT, mirando la columna correspondiente al tipo de carrera correspondiente: 1500 metros, 1 milla, 3000 metros, 2 millas, 5000 metros, 10000 metros, 15000 metros, media maratón o maratón. Por ejemplo, un atleta que recientemente ha terminado una carrera de media maratón en un tiempo de 1 hora y 40 minutos, tendría que buscar en la

Tabla 2.1, en la columna correspondiente a media maratón, el valor más cercano a ese tiempo. En nuestro caso, el tiempo más cercano en la tabla es de 1 hora 40 minutos y 20 segundos, que corresponde a una constante VDOT de 45, como se puede observar en la primera o en la última columna de dicha fila. Así pues, nuestro atleta tendría una constante VDOT de 45.

VDOT	1500	Mile	3000	2-mile	5000	10,000	15,000	1/2 Mara	Marathon	VDOT
30	8:30	9:11	17:56	19:19	30:40	63:46	98:14	2:21:04	4:49:17	30
32	8:02	8:41	16:59	18:18	29:05	60:26	93:07	2:13:49	4:34:59	32
34	7:37	8:14	16:09	17:24	27:39	57:26	88:30	2:07:16	4:22:03	34
36	7:14	7:49	15:23	16:34	26:22	54:44	84:20	2:01:19	4:10:19	36
38	6:54	7:27	14:41	15:49	25:12	52:17	80:33	1:55:55	3:59:35	38
40	6:35	7:07	14:03	15:08	24:08	50:03	77:06	1:50:59	3:49:45	40
42	6:19	6:49	13:28	14:31	23:09	48:01	73:56	1:46:27	3:40:43	42
44	6:03	6:32	12:55	13:56	22:15	46:09	71:02	1:42:17	3:32:23	44
45	5:56	6:25	12:40	13:40	21:50	45:16	69:40	1:40:20	3:28:26	45
46	5:49	6:17	12:26	13:25	21:25	44:25	68:22	1:38:27	3:24:39	46
47	5:42	6:10	12:12	13:10	21:02	43:36	67:06	1:36:38	3:21:00	47
48	5:36	6:03	11:58	12:55	20:39	42:50	65:53	1:34:53	3:17:29	48
49	5:30	5:56	11:45	12:41	20:18	42:04	64:44	1:33:12	3:14:06	49
50	5:24	5:50	11:33	12:28	19:57	41:21	63:36	1:31:35	3:10:49	50
51	5:18	5:44	11:21	12:15	19:36	40:39	62:31	1:30:02	3:07:39	51
52	5:13	5:38	11:09	12:02	19:17	39:59	61:29	1:28:31	3:04:36	52
53	5:07	5:32	10:58	11:50	18:58	39:20	60:28	1:27:04	3:01:39	53
54	5:02	5:27	10:47	11:39	18:40	38:42	59:30	1:25:40	2:58:47	54
55	4:57	5:21	10:37	11:28	18:22	38:06	58:33	1:24:18	2:56:01	55
56	4:53	5:16	10:27	11:17	18:05	37:31	57:39	1:23:00	2:53:20	56
57	4:48	5:11	10:17	11:06	17:49	36:57	56:46	1:21:43	2:50:45	57
58	4:44	5:06	10:08	10:56	17:33	36:24	55:55	1:20:30	2:48:14	58
59	4:39	5:02	9:58	10:46	17:17	35:52	55:06	1:19:18	2:45:47	59
60	4:35	4:57	9:50	10:37	17:03	35:22	54:18	1:18:09	2:43:25	60

Tabla 2.1. Relación entre VDOT y tiempo de carrera (parte I)

VDOT	1500	Mile	3000	2-mile	5000	10,000	15,000	1/2 Mara	Marathon	VDOT
61	4:31	4:53	9:41	10:27	16:48	34:52	53:32	1:17:02	2:41:08	61
62	4:27	4:49	9:33	10:18	16:34	34:23	52:47	1:15:57	2:38:54	62
63	4:24	4:45	9:25	10:10	16:20	33:55	52:03	1:14:54	2:36:44	63
64	4:20	4:41	9:17	10:01	16:07	33:28	51:21	1:13:53	2:34:38	64
65	4:16	4:37	9:09	9:53	15:54	33:01	50:40	1:12:53	2:32:35	65
66	4:13	4:33	9:02	9:45	15:42	32:35	50:00	1:11:56	2:30:36	66
67	4:10	4:30	8:55	9:37	15:29	32:11	49:22	1:11:00	2:28:40	67
68	4:06	4:26	8:48	9:30	15:18	31:46	48:44	1:10:05	2:26:47	68
69	4:03	4:23	8:41	9:23	15:06	31:23	48:08	1:09:12	2:24:57	69
70	4:00	4:19	8:34	9:16	14:55	31:00	47:32	1:08:21	2:23:10	70
71	3:57	4:16	8:28	9:09	14:44	30:38	46:58	1:07:31	2:21:26	71
72	3:54	4:13	8:22	9:02	14:33	30:16	46:24	1:06:42	2:19:44	72
73	3:52	4:10	8:16	8:55	14:23	29:55	45:51	1:05:54	2:18:05	73
74	3:49	4:07	8:10	8:49	14:13	29:34	45:19	1:05:08	2:16:29	74
75	3:46	4:04	8:04	8:43	14:03	29:14	44:48	1:04:23	2:14:55	75
76	3:44	4:02	7:58	8:37	13:54	28:55	44:18	1:03:39	2:13:23	76
77	3:41+	3:58+	7:53	8:31	13:44	28:36	43:49	1:02:56	2:11:54	77
78	3:38.8	3:56.2	7:48	8:25	13:35	28:17	43:20	1:02:15	2:10:27	78
79	3:36.5	3:53.7	7:43	8:20	13:26	27:59	42:52	1:01:34	2:09:02	79
80	3:34.2	3:51.2	7:37.5	8:14.2	13:17.8	27:41.2	42:25	1:00:54	2:07:38	80
81	3:31.9	3:48.7	7:32.5	8:08.9	13:09.3	27:24	41:58	1:00:15	2:06:17	81
82	3:29.7	3:46.4	7:27.8	8:03.7	13:01.1	27:07	41:32	59:38	2:04:57	82
83	3:27.6	3:44.1	7:23.1	7:58.7	12:53.0	26:51	41:06	59:01	2:03:40	83
84	3:25.5	3:41.8	7:18.5	7:53.7	12:45.2	26:34	40:42	58:25	2:02:24	84
85	3:23.5	3:39.6	7:14.1	7:48.9	12:37.4	26:19	40:17	57:50	2:01:10	85

Tabla 2.2. Relación entre VDOT y tiempo de carrera (parte II)

VDOT	E(Easy)/L(Long)		MP	T(Threshold Pace)			I (Interval Pace)				R (Rep Pace)		
	per Km	Per mile		400	1000	Mile	400	1000	1200	Mile	200	400	800
30	7:37	12:16	11:02	2:33	6:24	10:18	2:22				67	2:16	
32	7:16	11:41	10:29	2:26	6:05	9:47	2:14				63	2:08	
34	6:56	11:09	10:00	2:19	5:48	9:20	2:08				60	2:02	
36	6:38	10:40	9:33	2:13	5:33	8:55	2:02	5:07			57	1:55	
38	6:22	10:14	9:08	2:07	5:19	8:33	1:56	4:54			54	1:50	
40	6:07	9:50	8:46	2:02	5:06	8:12	1:52	4:42			52	1:46	
42	5:53	9:28	8:25	1:57	4:54	7:52	1:48	4:31			50	1:42	
44	5:40	9:07	8:06	1:53	4:43	7:33	1:44	4:21			48	98	
45	5:34	8:58	7:57	1:51	4:38	7:25	1:42	4:16			47	96	
46	5:28	8:48	7:48	1:49	4:33	7:17	1:40	4:12	5:00		46	94	
47	5:23	8:39	7:40	1:47	4:29	7:10	98	4:07	4:54		45	92	
48	5:17	8:31	7:32	1:45	4:24	7:02	96	4:03	4:49		44	90	
49	5:12	8:22	7:24	1:43	4:20	6:55	95	3:59	4:45		44	89	
50	5:07	8:14	7:17	1:42	4:15	6:51	93	3:55	4:41		43	87	
51	5:02	8:07	7:09	1:40	4:11	6:44	92	3:51	4:36		42	86	
52	4:58	7:59	7:02	98	4:07	6:38	91	3:48	4:33		42	85	
53	4:53	7:52	6:56	97	4:04	6:32	90	3:44	4:29		41	84	
54	4:49	7:45	6:49	95	4:00	6:26	88	3:41	4:25		40	82	
55	4:45	7:38	6:43	94	3:56	6:20	87	3:37	4:21		40	81	
56	4:40	7:31	6:37	93	3:53	6:15	86	3:34	4:18		39	80	
57	4:36	7:25	6:31	91	3:50	6:09	85	3:31	4:15		39	79	
58	4:33	7:19	6:25	90	3:45	6:04	83	3:28	4:10		38	77	
59	4:29	7:13	6:19	89	3:43	5:59	82	3:25	4:07		37	76	
60	4:25	7:07	6:14	88	3:40	5:54	81	3:23	4:03		37	75	2:30

Tabla 2.3. Relación entre VDot y ritmos de sesión (parte I)

VDOT	E(Easy)/L(Long)		MP	T(Threshold Pace)			I (Interval Pace)				R (Rép Pace)		
	per Km	Per mile		400	1000	Mile	400	1000	1200	Mile	200	400	800
61	4:22	7:01	6:09	86	3:37	5:50	80	3:20	4:00		36	74	2:28
62	4:18	6:56	6:04	85	3:34	5:45	79	3:17	3:57		36	73	2:26
63	4:15	6:50	5:59	84	3:32	5:41	78	3:15	3:54		35	72	2:24
64	4:12	6:45	5:54	83	3:29	5:36	77	3:12	3:51		35	71	2:22
65	4:09	6:40	5:49	82	3:26	5:32	76	3:10	3:48		34	70	2:20
66	4:05	6:53	5:45	81	3:24	5:28	75	3:08	3:45	5:00	34	69	2:18
67	4:02	6:30	5:40	80	3:21	5:24	74	3:05	3:42	4:57	33	68	2:16
68	4:00	6:26	5:36	79	3:19	5:20	73	3:03	3:39	4:53	33	67	2:14
69	3:57	6:21	5:32	78	3:16	5:16	72	3:01	3:36	4:50	32	62	2:12
70	3:54	6:17	5:28	77	3:14	5:13	71	2:59	3:34	4:46	32	65	2:10
71	3:51	6:12	5:24	76	3:12	5:09	70	2:57	3:31	4:43	31	64	2:08
72	3:49	6:08	5:20	76	3:10	5:05	69	2:55	3:29	4:40	31	63	2:06
73	3:46	6:04	5:16	75	3:08	5:02	69	2:53	3:27	4:37	31	62	2:05
74	3:44	6:00	5:12	74	3:06	4:59	68	2:51	3:25	4:34	30	62	2:04
75	3:41	5:56	5:09	74	3:04	4:56	67	2:49	3:22	4:31	30	61	2:03
76	3:39	5:52	5:05	73	3:02	4:52	66	2:48	3:20	4:28	29	60	2:02
77	3:36	5:48	5:01	72	3:00	4:49	65	2:46	3:18	4:25	29	59	2:00
78	3:34	5:45	4:58	71	2:58	4:46	65	2:44	3:16	4:23	29	59	1:59
79	3:32	5:41	4:55	70	2:56	4:43	64	2:42	3:14	4:20	28	58	1:58
80	3:30	5:38	4:52	70	2:54	4:41	64	2:41	3:12	4:17	28	58	1:56
81	3:28	5:34	4:49	69	2:53	4:38	63	2:39	3:10	4:15	28	57	1:55
82	3:26	5:31	4:46	68	2:51	4:35	62	2:38	3:08	4:12	27	56	1:54
83	3:24	5:28	4:43	68	2:49	4:32	62	2:36	3:07	4:10	27	56	1:53
84	3:22	5:25	4:40	67	2:48	4:30	61	2:35	3:05	4:08	27	55	1:52
85	3:20	5:21	4:37	66	2:46	4:27	61	2:33	3:03	4:05	27	55	1:51

Tabla 2.4. Relación entre VDOT y ritmo de sesión (parte II)

Utilizando la Tabla 2.3 o la Tabla 2.4, y la constante VDOT anteriormente averiguada a partir de la

Tabla 2.1/Tabla 2.2, se puede averiguar el ritmo al que se tiene que entrenar en los distintos tipos de sesiones. Aunque no vamos a entrar en detalles con respecto a las ventajas que ofrece cada tipo de sesión, sí que nos gustaría enumerarlas:

- Sesión larga (E/L).
- Sesión de maratón (MP).
- Sesiones tempo (*Threshold*).
- Sesión de intervalos (*Interval*).
- Sesión de repeticiones (*Rep*).

Se puede observar que los valores de las distintas columnas en las tablas anteriores mantienen cierta proporción, es por ello por lo que el autor Jack Daniels incide en que hay ciertas directrices que gobiernan el entrenamiento de un atleta, y éstas son:

- Las sesiones de velocidad (*Interval* y *Rep*) se realizan al 100% del Vo2Max.
- Las sesiones de ritmo alto (*Threshold*) se realizan aproximadamente al 88% del Vo2Max.
- Las sesiones de ritmo bajo (E/L) se realizan aproximadamente al 70% del Vo2Max.

Aparte de estos nuevos conceptos e índices relacionados con el tiempo y ritmo por sesión, en [6] se proponen unas pautas generales que deberían gobernar la planificación del entrenamiento de los corredores de larga distancia, algunas de estas reglas son:

- Las sesiones deben ser progresivas, incrementando su distancia como mucho en un 10% de una sesión con respecto a la anterior.
- El 70-80% del entrenamiento se debe realizar en sesiones largas.
- Dejar suficiente tiempo de recuperación entre sesiones.
- Incluir sesiones de entrenamiento combinado (natación o ciclismo, por ejemplo) en las sesiones de descanso.
- Bajar la intensidad de entrenamiento en las sesiones próximas a carreras en las que se compite, y también después de un cierto número de incrementos, normalmente 4.
- Mezclar sesiones aeróbicas con sesiones anaeróbicas.

Teniendo en cuenta lo anterior, en la creación de una planificación para una semana, las sesiones se distribuirían de la siguiente forma:

- El 70-80% del entrenamiento se tiene que realizar a ritmo relajado, sesiones largas con un ritmo correspondiente al 70% del Vo2Max.

- El 20-30% restante serán sesiones cortas de intervalos y repeticiones a ritmos de 88% y 100% del Vo2Max.

En cuanto a la distancia que hay que recorrer en cada sesión, se pueden aplicar los mismos principios vistos con anterioridad. Si se toma la distancia de una de las sesiones más recientes (o una media de ellas) como distancia base y se tiene en cuenta que la distancia de la sesión a calcular no se puede incrementar en más de un 10%, ya se puede obtener la siguiente sesión; lo único que queda es repetir el paso las veces que se desee y ya se tendrá la planificación.

¿Cuántas veces? Obviamente no se puede entrenar durante 8 meses incrementando un 10% cada sesión, si esto fuera posible se llegaría a correr ultramaraton<sup>1</sup> en las sesiones largas de entrenamiento para un maratón, lo cual no tiene sentido. En este aspecto, la mayoría de entrenadores coinciden en que el entrenamiento estándar para un maratón se desarrolla durante 4 meses aproximadamente [8], [9]. La

Tabla 2.5 presenta un ejemplo de planificación que dura 16 semanas.

Para interpretar la

Tabla 2.5, pueden ser necesarias las siguientes aclaraciones:

- DS/EX – Descanso (DS) o Entrenamiento combinado (EX), se considera entrenamiento combinado a realizar actividades diferentes al correr, ya sea hacer bicicleta, natación o sesión de gimnasio.
- L-5km@5.34 – Sesión larga de 5 kilómetros a un ritmo de 5.34 minutos por kilómetro.
- T-8km;3km@4.38 – Sesión tempo de 8 km, 3 km a un ritmo de 4.38 minutos por kilómetro, resto a ritmo normal.
- S-8km;2x1600@7.25 – Sesión de velocidad de 8 km, 2 sesiones de 1600 metros, a un ritmo de 7.25 min/km cada sesión, resto a ritmo normal.

Alguno de nosotros podría observar, por ejemplo, que entre la semana 4 y 5 existe un incremento de más de 10%, parámetro que dijimos no era recomendable superar. Si no damos cuenta, en la semana 4 las distancias decrecen en relación con la anterior, es pues una semana de descanso o recuperación, es por ello que en la siguiente semana el ritmo crece en más de un 10%. Es como si el ritmo se hubiera incrementado también durante la semana 4, aunque realmente haya sido una semana de descanso.

En este capítulo se ha ofrecido una introducción a las carreras de larga distancia, los conceptos relacionados y algún estudio de investigación realizado sobre cómo se tiene que entrenar para una carrera de este tipo. Consideramos que es un capítulo que sirve como

---

<sup>1</sup> Ultra maratón: Distancia mayor que un maratón



base para el resto de la memoria, ya que a veces se harán referencias a términos técnicos explicados aquí.

<b>Semana</b>	<b>Lunes</b>	<b>Martes</b>	<b>Miércoles</b>	<b>Jueves</b>	<b>Viernes</b>	<b>Sábado</b>	<b>Domingo</b>
1	DS/EX	DS/EX	DS/EX	L-5km@5.34	DS/EX	DS/EX	L-5km@5.34
2	DS/EX	DS/EX	DS/EX	L-5km@5.34	DS/EX	DS/EX	L-6km@5.34
3	DS/EX	DS/EX	DS/EX	L-6km@5.34	DS/EX	DS/EX	L-7km@5.34
4	DS/EX	DS/EX	DS/EX	L-5km@5.34	DS/EX	DS/EX	L-6km@5.34
5	DS/EX	DS/EX	DS/EX	L-8km@5.34	DS/EX	DS/EX	L-8km@5.34
6	DS/EX	DS/EX	DS/EX	S-8km;2x1600@7.25	DS/EX	DS/EX	L-10km@5.34
7	DS/EX	DS/EX	DS/EX	T-8km;3km@4.38	DS/EX	DS/EX	L-11km@5.34
8	DS/EX	DS/EX	DS/EX	L-8km@5.34	DS/EX	DS/EX	L-8km@5.34
9	DS/EX	DS/EX	DS/EX	T-8km;3km@4.38	DS/EX	DS/EX	L-13km@5.34
10	DS/EX	DS/EX	DS/EX	S-8km;2x1600@7.25	DS/EX	DS/EX	L-14km@5.34
11	DS/EX	DS/EX	DS/EX	T-7km;3km@4.38	DS/EX	DS/EX	L-16km@5.34
12	DS/EX	DS/EX	DS/EX	L-9km@5.34	DS/EX	DS/EX	L-10km@5.34
13	DS/EX	DS/EX	DS/EX	T-6km;3km@4.38	DS/EX	DS/EX	L-19km@5.34
14	DS/EX	DS/EX	DS/EX	S-7km;2x1600@7.25	DS/EX	DS/EX	L-23km@5.34
15	DS/EX	DS/EX	DS/EX	T-6km;3km@4.38	DS/EX	DS/EX	L-12km@5.34
16	3km@5.34	DS/EX	DS/EX	S-5km;1x1600@7.25	3km@5.34	DS/EX	Maratón

Tabla 2.5. Ejemplo de planificación para un maratón



### 3. Computación ubicua, móvil y sensible al contexto

En este capítulo se hará una breve descripción de la Computación Ubicua, y se expondrán los principios y características por los que se rige, así como los conceptos estrechamente relacionados con nuestros objetivos, siendo algunos de ellos la movilidad, la colaboración y la sensibilidad al contexto. También se explicará en qué consiste la Computación Móvil y los retos y limitaciones que impone.

En el campo de la Inteligencia Ambiental (IA a partir de ahora), la Computación Ubicua (CU) es la rama que, según Mark Wiser [10], representa “una nueva forma de la interacción persona ordenador”. Esta nueva forma de ver la interacción se basa en que la potencia de cálculo se duplica cada cierto tiempo. Los procesadores de los que se dispone aumentan cada día, en número y capacidad, distribuyéndose en el ambiente, de ahí la característica de ubicuidad. Algunos de los principios en los que se basa la CU (aparte de la ubicuidad) son:

- *Inteligencia de los objetos*: gracias a los dispositivos electrónicos incrustados en ellos, los objetos interactúan con el ambiente e interactúan entre ellos.
- *Movilidad*: los objetos inteligentes se desplazan con el individuo, trasladando su potencia de cálculo allí donde estén.
- *Colaboración*: los dispositivos colaboran entre ellos de forma transparente al usuario.
- *Invisibilidad o transparencia*: se refiere a que el modo en el que actuamos con los dispositivos inteligentes no es explícito, como en el caso de la interacción con el PC, sino natural y poco intrusivo, gracias a que los dispositivos ubicuos están embebidos en el medio y nos facilitan la vida de forma inteligente y transparente al usuario.
- *Sensibilidad al contexto*: los sensores captan información del mundo que nos rodea, en base a la cual se puede inferir el contexto del usuario, de modo que se puede adaptar la aplicación a dicho contexto.

Estos principios determinan una serie de características propias de la Computación Ubicua, de las cuales destacamos:

- *Reactividad*: el sistema o aplicación reacciona cuando ocurren determinados eventos.
- *Capacidad adaptativa*: la aplicación se adapta a las necesidades y preferencias del usuario.
- *Proactividad*: la aplicación se anticipa a nuestros deseos y preferencias.

En el caso de nuestra aplicación, son de especial interés los aspectos relacionados con la movilidad, colaboración y sensibilidad al contexto. En cuanto a la *movilidad*, en nuestra aplicación habrá un módulo encargado de realizar el seguimiento de la ruta que realizamos, por lo que la característica de movilidad está implícita, y es en ella en lo que se basa todo el sistema. El segundo aspecto interesante es la *colaboración*, obviamente en este tipo de aplicaciones es más enriquecedor que en lugar de que cada usuario realice sus propios trayectos y que sea él el único beneficiario, que los trayectos realizados por cada usuario puedan ser compartidos con el resto de usuarios de la aplicación. La *sensibilidad al contexto* es igual de importante que los anteriores factores y está estrechamente ligada a la movilidad y posición del dispositivo, ya que los trayectos ofrecidos por la aplicación tendrán en cuenta el contexto. A continuación se verá con más exactitud en qué consiste el contexto y qué aspectos pueden resultar útiles para nuestra aplicación.

Estrechamente ligada a la Computación Ubicua, se encuentra la Computación Sensible al Contexto y la Computación Móvil. Vamos a ver una definición más amplia del concepto contexto. *Contexto*, según el diccionario, se puede entender como una serie de circunstancias que caracterizan un evento. Según Dey et al. [11], el contexto representa “cualquier información que pueda ser usada para caracterizar una situación o entidad. Una entidad es una persona, un objeto o un sitio, que se considera relevante en la interacción entre el usuario y la aplicación, incluido el usuario y la aplicación”. El contexto se puede clasificar de la siguiente forma:

- *Contexto primario*: Inferido directamente de los datos obtenidos por los sensores. A su vez este se compone de 4 subcontextos o categorías (ver Figura 3.1):
  - *Temporal*: un ejemplo de aplicación que hace uso de este tipo de contexto sería [12] donde se estudia la posibilidad de clasificación de colecciones de fotos en función del tiempo de la toma, y del contenido de estas fotos.

- *Ubicación o localización:* se refiere al contexto proporcionado por la posición del usuario en el entorno considerado.
- *Identidad:* este contexto se encuentra estrechamente ligado con el contexto personal del individuo y puede tener como características el sexo, la edad o preferencias del usuario. Una aplicación que tiene en cuenta esto es [13], donde aparte de la ubicación del usuario se tiene en cuenta su edad y su sueldo para recomendarle restaurantes posiblemente más acordes con sus gustos y posibilidades económicas.
- *Actividad:* el contexto de actividad tiene en cuenta el proceso que el usuario está llevando a cabo cuando utiliza el terminal inteligente. Puede considerarse aquí una aplicación que aumenta el brillo de la pantalla cuando detecta que el usuario está tecleando un mensaje en su dispositivo, y lo disminuye cuando el usuario está escuchando música.

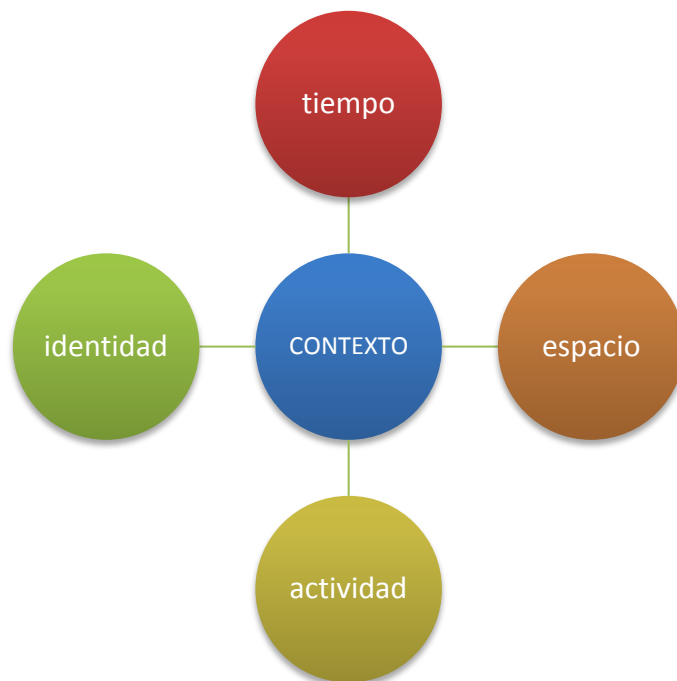


Figura 3.1. Contexto primario y sus interdependencias

- *Contexto secundario:* Se calcula a partir de información de contexto primario (refinada por combinación, deducción o filtrado) para derivar información de contexto de alto nivel, siendo más apropiada para el procesamiento precisado por cualquier sistema, pero generalmente menos precisa. En este tipo de contexto se pueden distinguir las siguientes subcategorías: personal,

técnico, espacial, social y físico. Como se puede ver, es un contexto más específico que el primario.

La Computación Móvil (CM) es una rama de la Interacción Persona-Ordenador que está relacionada con la Computación Ubicua, presentando alguna diferencia con respecto a ésta, y es que la CM no implica necesariamente distribución en el cálculo o colaboración entre los dispositivos, aunque también tienen características comunes, como la ubicuidad (que viene implícita por la movilidad), la capacidad adaptativa o la sensibilidad al contexto. Una de las dificultades de la CM viene justamente por esta última característica, y es que la sensibilidad al contexto requiere mucho esfuerzo, teniendo en cuenta que en este caso el contexto cambia muchas veces, especialmente el contexto espacial.

La movilidad propia de estos dispositivos se puede entender de tres formas:

- *Movilidad en la comunicación*: inherente a este concepto están los protocolos de comunicación, formatos de datos y distintas tecnologías que se emplean.
- *Movilidad del hardware*: este concepto está estrechamente ligado a la movilidad en la comunicación y al cambio contextual.
- *Movilidad del software*: representan las características y funcionalidades de estos terminales, es lo que los hace únicos y adaptables a cada usuario.

Más sobre la movilidad de los dispositivos móviles y cómo afecta a la comunicación y cambio contextual se podrá ver en el capítulo 4.

De los principales retos y limitaciones que se encuentran en el campo de la computación móvil, se pueden resaltar los siguientes:

- *Ancho banda limitado*: los distintos protocolos mediante los cuales los dispositivos móviles obtienen datos ofrecen velocidades reducidas debido al ancho de banda limitado del que disponen. Aunque durante los últimos años se han hecho grandes avances, en realidad la velocidad sigue estando muy por debajo de la que puede ofrecer una infraestructura de red local.
- *Seguridad inalámbrica*: es bien conocido que las comunicaciones inalámbricas no son tan seguras como las comunicaciones por cable, debido al poco control físico que se puede realizar. Cuando tratamos con dispositivos móviles, muy vinculados a la identidad y personalidad del usuario, hay que tener especial cuidado en la manipulación de la comunicación.
- *Autonomía de batería limitada*: por muy simple que parezca, con la cantidad de sensores y aplicaciones de los últimos *smartphones*, la batería (que es uno

de sus componentes básicos) está sufriendo un consumo considerable. Se espera que un buen software tenga en cuenta esta limitación.

- *Interfaz de interacción:* en un espacio tan limitado, las pantallas y otros dispositivos de entrada/salida (como los teclados) tendrán su tamaño correspondiente. Muchas veces el reto es incorporar formas alternativas de interacción, como entradas por voz.

Teniendo en cuenta todos los aspectos innovadores que caracterizan la Computación Ubicua y sus ramas, de las cuales destacamos la Computación Móvil, con sus limitaciones, impuestas por la misma movilidad, nos damos cuenta que el desarrollo de software en este ámbito no es trivial, siendo más bien un campo nuevo que supone métodos y aspectos innovadores, de un gran auge, con perspectivas de futuro.





## 4. Redes de telefonía móvil: concepto, términos y generaciones

Muchos de nosotros hemos escuchado alguna vez hablar de redes celulares o incluso, en algunas zonas, la palabra “celular” usándose como sinónimo de terminal móvil, ¿pero qué es realmente un “celular”? ¿Tiene algo que ver con una célula? En [14] se ofrece una revisión suficientemente amplia en base a la cual se intentará contestar a estas preguntas. Así pues, durante este capítulo se realizará un repaso de las distintas tecnologías y estándares que los terminales móviles usan. Esto servirá como base para entender cómo las tecnologías de geolocalización y la movilidad hardware funcionan, ayudándonos también a entender mejor las limitaciones de ancho de banda y seguridad que tienen.

Los sistemas de comunicación antiguos, que permitían utilizar los móviles, eran analógicos, es decir, utilizaban directamente la propagación de las ondas sin modificarlas (modularlas) utilizando dimensiones discretas (tiempo, amplitud) que es lo que hacen los sistemas digitales. Estos sistemas eran muy limitados en cuanto al número de usuarios a los que podía abastecer, ya que si cada usuario utiliza un canal de voz de 25Hz, en el ancho de banda de 1800-1900MHz, se podría dar servicio a solo 4000 usuarios. En las zonas urbanas este número se sobrepasa rápidamente, por lo cual el uso de una nueva tecnología era necesaria.

Los nuevos sistemas se basan en dividir el espacio de cobertura en zonas, circulares o hexagonales, si se prefiere. La verdad es que estas zonas no tienen una forma muy delimitada, ya que dependen del terreno y otros obstáculos. El diámetro de estas zonas puede variar en función de la densidad de la población, e implícitamente de la cantidad de dispositivos móviles que haya en dicha zona. En zonas urbanas pueden ser de 1km, mientras que en zonas rurales pueden llegar a ser mucho más grandes. Estos sistemas colocan en cada parte hexagonal una o varias antenas que emitirán ondas a una cierta frecuencia, una vez que la onda sobrepasa los límites del hexágono, la señal pierde intensidad y se hace inusable. Para evitar posibles solapamientos, se utilizan frecuencias distintas en hexágonos adyacentes y se agrupan estos hexágonos para aislar más las celdas que utilizan las mismas frecuencias. Estos hexágonos se llaman celdas. Cuando estas celdas se agrupan en grupos, formados por 7 o más celdas, a la agrupación se le denomina *clúster*. Un ejemplo de clúster se puede ver en la Figura 4.1.

En cuanto a las antenas, muchas veces se utilizan más de una antena por celda, por ejemplo 3, proporcionando cada una de ellas una cobertura de 120°. Este aspecto puede resultar interesante cuando se hable de métodos de localización en dispositivos móviles.

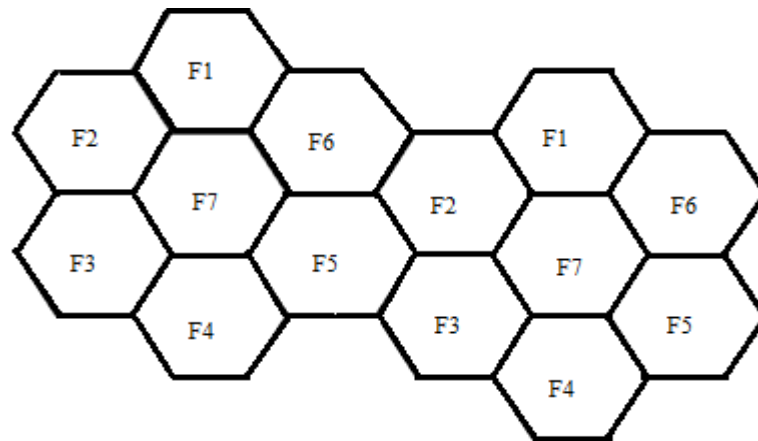


Figura 4.1 Celdas agrupadas en 2 clústeres de 7

En cuanto a la infraestructura de estas redes, comentar que, aparte de la *antena conectada a su estación base*, que es componente electrónico encargado de recibir y enviar datos, hay más componentes que forman este tipo de redes, como el *controlador de la estación base*, que se encarga de conectar a muchas estaciones base y realizar la comunicación entre ellas, y el *centro móvil de conmutación*, que es un componente primordial de las redes de telefonía, ya que provee el enlace entre las distintas estaciones base. Aunque nuestros terminales móviles se comuniquen directamente con las estaciones bases, ¿qué ocurre cuando nos comunicamos con una persona que no depende de la misma estación base? Es necesario pues un enlace entre las distintas estaciones. Este centro es el encargado de mantener una base de datos con las localizaciones más recientes de los dispositivos móviles, haciendo esto por motivos de seguridad o necesidad en la comunicación terminal móvil-estación base, y de facilitar la autenticación de un terminal en la red. En cuanto a la autenticación, simplemente decir que considera como identidad el número del teléfono móvil de cada usuario, y que es un proceso obligatorio antes de que nuestro terminal esté disponible. También es a nivel de este centro donde se realiza el contacto entre la red móvil y la red telefónica básica.

#### 4.1. Telefonía móvil de primera generación o 1G

Como ya se ha dicho, los primeros sistemas de comunicación móvil eran sistemas analógicos que permitían comunicarse a usuarios que usaban dispositivos móviles que podían pesar varios kilogramos. En función de la localización geográfica, existían varios sistemas analógicos, como la Telefonía Móvil Nórdica (NMT – *Nordisk Mobil Telefoni*), que fue el primer sistema completamente automático de telefonía móvil que empezó a ofrecer sus servicios en los Países Nórdicos el 1 de Octubre de 1981, según el tele-museo noruego.

Otro sistema fue el Sistema Avanzado de Telefonía Móvil (AMPS – *Advanced Mobile Phone Service*), que daba servicio a dispositivos en los Estados Unidos, Israel y Australia. Por último, el Sistema de Comunicación de Acceso Total (TACS – *Total Access Communication System*) se creó por parte de Motorola y funcionaba en el Reino Unido y otros países europeos. Estos sistemas analógicos ya no se utilizan hoy en día, o se utilizan en zonas extremadamente limitadas, debido a las ventajas que ofrecen los sistemas digitales.

La infraestructura de un sistema analógico sigue el modelo descrito anteriormente. Los terminales móviles se comunican con la estación base, utilizando 2 tipos de canales, canales de voz y canales de control; cada canal está dividido en 2, utilizándose cada uno para enviar y recibir datos respectivamente. Obviamente, los canales de voz se utilizan para el envío de voz entre la estación base y el terminal móvil, mientras que los canales de control se utilizan para llamar al terminal móvil y otras tareas subyacentes, como autenticación y control. Un papel especialmente importante lo desempeña el canal de control en el sentido de envío desde la estación base, ya que este canal transmite continuamente y actúa como balizamiento para los nuevos terminales que acceden a la red.

Las estaciones base están conectadas entre sí mediante un controlador de estación base y un centro móvil de conmutación, que es el encargado de conectar a muchos controladores de estación base. La red de telefonía móvil está formada por varios centros móviles de conmutación y cada estación base está a cargo de 20 o 30 canales de voz.

## **4.2. Telefonía móvil de segunda generación o 2G**

El estándar GSM, o estándar de segunda generación, obtiene su nombre original del francés *Groupe Speciale Mobile*, nombre que se modificó a posteriori para mostrar la naturaleza global de este sistema, así se originó el Sistema Global para las Comunicaciones Móviles.

El estándar está definido por el Instituto Europeo de Estándares de Telecomunicación (ETSI – *European Telecommunications Standards Institute*) y se creó fundamentalmente con dos objetivos principales en mente:

- (1) Crear un sistema que dé soporte a la movilidad de multitud de usuarios que viajaban entre diferentes países, intercambiando implícitamente entre distintos sistemas de telecomunicación de primera generación, como NMT, AMPS y TACS, que no son compatibles entre sí.
- (2) Fortalecer la seguridad en este tipo de sistemas.

Bajo la directiva de ETSI, el estándar GSM ha conseguido de sobra sus objetivos y hasta el año 2006 ha subido de forma continua el número de usuarios abonados, cuya cifra

ronda los 3 millares de usuarios actualmente. Últimamente, las riendas del estándar están en manos del Proyecto Asociación de Tercera Generación (3GPP – *Third Generation Partnership Project*), que es también el encargado del estándar de tercera generación, que se verá a continuación.

La infraestructura del estándar es parecida a la del estándar de primera generación, y está formada por estaciones base, controladores de estaciones base y centros de conmutación móvil, del cual dependen los componentes encargados de autenticación y registro.

Como novedad, el estándar GSM incluye varios identificadores para proveer movilidad y seguridad, muchos de estos identificadores se guardan en la tarjeta del módulo de identificación del suscriptor (SIM – *Subscription Identification Module*).

Otra facilidad del estándar es la inclusión de la *técnica de salto de frecuencia*, que realiza saltos de frecuencia en el envío de los datos, de modo que un dispositivo permanecerá sobre una frecuencia lo suficiente como para mandar un paquete de datos, después del cual saltará a la siguiente frecuencia, determinada de forma pseudo-aleatoria. Esto incrementa el nivel de seguridad de forma obvia, haciendo que otro dispositivo no pueda escuchar sobre una frecuencia la conversación de otro.

Lo que se conoce hoy en día como Servicio de Mensajes Cortos (SMS – *Short Message Service*) también fue una nueva propuesta introducida por el estándar GSM. El servicio es capaz de enviar texto de hasta 160 caracteres alfanuméricos, utilizando el método de almacenar y enviar del que disponen los centros de mensajería. Es importante decir que este servicio no hace uso de los canales de voz, sino de los de control, con lo cual hace posible la comunicación verbal y por medio de SMS simultánea.

Aparte del estándar europeo de segunda generación GSM, en Estados Unidos, a mano de la empresa Qualcomm, se desarrolló otro estándar, el denominado IS-95 o CDMA (*Code Division Multiple Access*) One. La infraestructura de red es igual a la del estándar GSM, utilizando estaciones base, controladores de estaciones base y centros de conmutación móvil. La arquitectura de la comunicación, al igual que GSM, está basada en 3 capas, capa física, capa de control de acceso al medio y capa de enlace, pero la estructura de todas las capas es distinta. La velocidad máxima obtenible en condiciones ideales por el estándar IS-95 es de 115.2 Kbps, mientras que la del estándar GSM es de 14.4 Kbps.

#### **4.2.1. Servicio general de paquetes vía radio y EDGE o 2.5G**

Con la evolución de los terminales móviles, cada vez se hizo más evidente la necesidad de desarrollo de una estructura para el tráfico de datos, así es como apareció el Servicio General de Paquetes vía Radio (GPRS – *General Packet Radio Service*). Se considera como evolución por parte de algunos, pero es más bien una extensión al estándar GSM, que

permite alcanzar velocidades de transferencia de hasta 144 Kbps en bajada y 48 Kbps en subida, una mejora considerable si se tiene en cuenta que GSM, por sí solo, únicamente puede llegar a 14.4 Kbps.

La infraestructura de GPRS, que se puede observar en la Figura 4.2, hace uso de la ya existente para el estándar GSM en el tramo comprendido entre la estación base (BTS) y el controlador de la estación base (BSC), pero desde el controlador, los datos son enviados a los Nodos de Servicio GPRS (SGSN – *Serving GPRS Support Node*), que está encargado de la autenticación, y de dar acceso a los terminales a la red de datos móviles (red GPRS), de verificar la calidad del servicio para cada terminal móvil. A partir del SGSN los datos viajan a la Puerta de Enlace de los Nodos de Servicio GPRS (GGSN – *Gateway GPRS Support Node*), que es la encargada de dar acceso a una red móvil (red GPRS) hacia Internet u otra red corporativa. La forma en la que los datos circulan por la red móvil utiliza una arquitectura por capas, que difiere en función de si el paquete se encuentra entre el terminal y la estación base, en la red móvil, o entre el GGSN y el exterior.

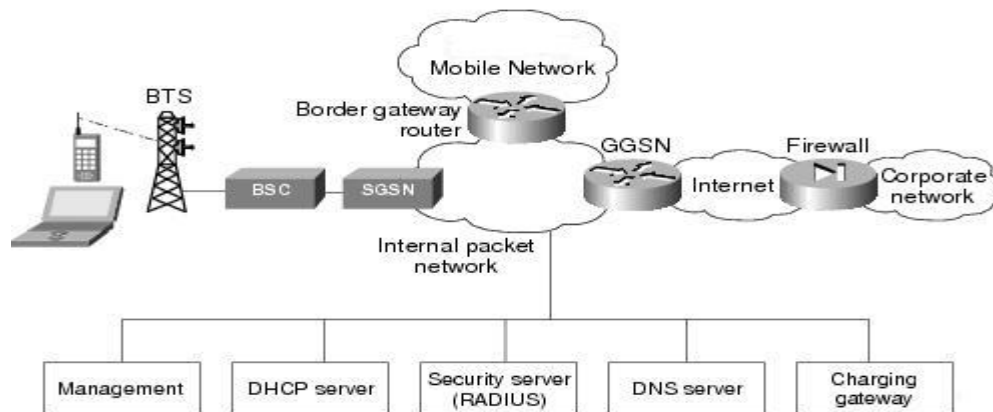


Figura 4.2. Arquitectura GPRS<sup>1</sup>

Los dispositivos móviles que hacen uso del servicio GPRS se pueden clasificar en tres clases en función de la habilidad de usar simultáneamente esta tecnología con el servicio GSM:

- *Clase A*: Terminales capaces de hacer uso de los dos servicios de forma simultánea.
- *Clase B*: Terminales capaces de estar conectados a los dos servicios de forma simultánea, pero hacer uso de uno solo en un momento dado. El intercambio del servicio se realiza de forma automática. La mayoría de terminales móviles están incluidos en esta categoría.

<sup>1</sup> Fuente: [67]

- *Clase C*: Terminales capaces de estar conectados únicamente a un servicio en un instante de tiempo. El intercambio del servicio se hace de forma manual.

EDGE, acrónimo proveniente del inglés, significa Tasas Mejoradas para la Evolución GSM (EDGE – *Enhanced Data Rates for GSM Evolution*) y representa una evolución de GPRS, la diferencia proviene de la modificación en la modulación. EDGE utiliza una mejor modulación, mejorando la carga de datos por unidad de tiempo, es decir, consigue transmitir más datos en el mismo intervalo de tiempo. La infraestructura que utiliza es casi idéntica a la del sistema GPRS, aunque requiere el acoplamiento al nivel de estación base, de unos módulos capaces de enviar y recibir datos con la modulación adecuada.

En condiciones ideales EDGE puede alcanzar una velocidad de bajada de 384 Kbps y de subida de 128 Kbps.

### **4.3. Telefonía móvil de tercera generación o 3G**

#### **4.3.1. CDMA2000 y UMTS**

CDMA2000 es realmente una familia de estándares de telecomunicación de tercera generación, decimos familia porque está compuesto por una serie de variantes, como 1xEV, 1xEV-DO, 1xEV-DV, donde EV significa evolución, DO significa Solo Datos (*Data Only*), mientras que DV significa Datos y Voz (*Data Voice*).

Especificar que esta familia de estándares utiliza la tecnología de acceso múltiple por división de código, al igual que su hermana menor CDMA One, de la que realmente es una evolución.

La primera versión de CDMA2000, la 1xEV, es considerada por algunos como tecnología 2.5G, por el hecho de que no consigue la velocidad necesaria para encuadrarse como tecnología 3G, aunque sí cumple con las directivas de diseño. La principal mejora de esta tecnología frente a su antecesora, la CDMA One, es la incorporación de una señal piloto que permite el uso de una modulación mejor, lo que incrementa el tráfico entre la estación base y el terminal móvil. También se han realizado algunos cambios a nivel de capa de enlace para permitir la incorporación del control de la calidad de servicio. Por último, decir que el tráfico se realiza sobre 2 canales físicos (envío, recepción) de 1.25 Mhz cada uno y que esta tecnología permite alcanzar velocidades de como máximo 384 Kbps.

Las siguientes dos implementaciones de este estándar, EV-DO y EV-DV, básicamente cambian de nuevo la modulación que utilizaba 1xEV y añaden más canales. Mediante estas mejoras el estándar EV-DO consigue unas velocidades de 3.8 Mbps en bajada y 1.8 Mbps en subida. Si se habla de EV-DV, éste supera en muy poco las velocidades de EV-DO, por lo que

las operadoras móviles prefieren seguir utilizando EV-DO hasta que la nueva implementación de CDMA 3x entre en funcionamiento. El distintivo de 3x es porque utilizará 3 veces el ancho de banda que utiliza actualmente, es decir  $3 \times 1.25$  Mhz.

El estándar CDMA2000, se utiliza actualmente en Estados Unidos y en muchos países de Asia, como China, India o Corea. Su mayor competidor es el estándar UMTS.

El sistema universal de telecomunicaciones móviles (UMTS – *Universal Mobile Telecommunication System*) ha sido desarrollado por 3GPP debido a los requerimientos de la Unión Internacional de Telecomunicación (ITU – *International Telecommunication Union*) para dar soporte a la tecnología 3G en Europa.

La arquitectura de este nuevo estándar de tercera generación está compuesta por las siguientes tres partes:

- *Núcleo de red*: Una de las piezas claves de esta parte es el centro móvil de conmutación, que es común tanto al estándar GSM de segunda generación como al estándar UMTS de tercera generación, aunque realmente es una entidad de segunda generación, ya que funciona conmutando circuitos, mientras que el SGSN que vimos en el apartado anterior pasa a la tecnología de conmutación de paquetes, básicamente mucho más adecuada para el tráfico de datos, ya que la conmutación de paquetes no requiere una conexión dedicada entre los dos extremos. El núcleo de red se encarga de la conmutación, del control de los servicios y de dar acceso a las redes exteriores a la UMTS.
- *Red de acceso radio*: Es la encargada de realizar la comunicación entre el usuario y el núcleo de red, y está compuesta por varios nodos B y un controlador de red radio (RNC – *Radio Network Controller*). Si tuviéramos que hacer una comparación entre este estándar y el de segunda generación GSM, el RNC se correspondería al BSC y los nodos B se corresponderían a las estaciones bases. Aparte de estos dos nuevos componentes, todos los demás, como GGSN, SGSN, el centro de autenticación y los distintos registros utilizados para movilidad e identificación, se han incorporado de los estándares anteriores 2G y 2.5G.
- *Equipamiento del usuario*: Básicamente es el terminal móvil junto a la tarjeta SIM.

La principal ventaja de la tecnología 3G frente a la tecnología 2G es el aumento considerable de la velocidad de bajada, de 115.2 Kbps a 7.2 Mbps, lo que ha permitido cierta evolución de los terminales móviles, de simples dispositivos para la comunicación a una herramienta compleja considerada por muchos como un centro multimedia móvil. La velocidad de subida sigue siendo de 128 Kbps.

### 4.3.2. IP móvil

Como se ha visto anteriormente, los estándares 3G empleados ofrecen velocidades decentes de varios Mbps, que seguramente aumentarán con la introducción de nuevas tecnologías. Estas velocidades hacen posible que un nodo móvil, ya sea un portátil o un terminal móvil pueda conectarse a Internet como si de una red local se tratase, aparece aun así el problema de la movilidad hardware. Cuando un terminal móvil o portátil, nodo móvil en adelante se conectada a una red local, se le asigna una dirección IP, la cual se utiliza para la comunicación del dispositivo en Internet, ¿qué ocurre cuando el nodo sale del alcance de la red y entra en otra? Probablemente la conexión desaparecerá y se formará otra con la nueva red, esto implica que cualquier conexión en marcha en el nodo móvil se interrumpirá. Aunque esto no representa un problema en el caso de las redes locales, ya que pocos usuarios cambian de red local en movimiento, sí representa un desafío en el caso de los nodos móviles.

La solución a este problema se llama IP móvil y consiste en lo siguiente: cada nodo, está identificado por dos direcciones IP, una permanente asociada a la estación base donde el nodo permanece de forma habitual, llamémosla *IP propia*, y una *IP temporal* que depende de la posición actual del nodo, esta IP existe siempre que el nodo móvil intente conectarse a la red desde una localización distinta de la que alcanza su estación base habitual.

Aparte de los conceptos de IP propia e IP temporal, asociadas a cada subred existen dos entidades que se encargan de guardar información sobre los nodos:

- *Agente propio*: guarda información sobre los nodos cuya dirección IP propia se encuentra en la red del agente propio actual.
- *Agente remoto*: guarda información sobre los nodos cuya dirección IP propia no se encuentra en la red del agente remoto actual, es decir, nodos en movilidad.

Cuando un nodo quiera ponerse en contacto con nuestro nodo móvil, enviará paquetes de datos cuya dirección IP de destino es la IP propia del nodo móvil, estos paquetes llegarán primero al agente propio, que ya sabe que nuestro nodo móvil no se encuentra en su área, porque el agente remoto se puso en contacto con él. El agente propio, utilizando un túnel contra el agente remoto, encapsulará el paquete con la dirección IP temporal del nuestro nodo móvil y lo enviará al agente remoto, el cual enviará el paquete a nuestro nodo móvil analizando para ello la cabecera de dicho paquete. Para la comunicación en sentido contrario se utiliza el mismo túnel en sentido inverso. Utilizando este mecanismo el terminal móvil utiliza la misma dirección IP permitiendo a las aplicaciones móviles mantenerse en funcionamiento.

IP móvil es un estándar IP desarrollado por el Grupo Especial sobre Ingeniería en Internet (IETF – *Internet Engineering Task Force*) en colaboración con el Consorcio World



Wide Web (W3C – *World Wide Web Consortium*) y la Organización Internacional de Estandarización (ISO – *International Standardization Organization*). La versión actualmente en uso es la *Mobile IPv4* previéndose para el futuro la incorporación de *Mobile IPv6*.

### **4.3.3. Avances hacia la tecnología de cuarta generación**

3GPP la asociación que se encarga de la especificación del estándar de tercera generación está trabajando para poder hacer público el estándar para la cuarta generación en telecomunicaciones para telefonía móvil. Mientras eso no sea posible se estará haciendo uso de tecnologías 3.5G, de las que se hará un breve resumen a continuación, según [15]:

HSPA (*High-Speed Packet Access*) es una optimización del estándar UMTS de tercera generación que ofrece velocidades de hasta 14 Mbps en sentido de bajada y 384 Kbps en subida. Esto se consigue mediante la introducción de varios canales de comunicación más y la modificación en la modulación adaptativa en función de las condiciones radio existentes. Con la introducción de esta tecnología, algunas empresas están ofreciendo módems USB para el acceso a Internet, estos módems pueden llegar a la velocidad de bajada descrita anteriormente pero únicamente en condiciones ideales.

HSUPA (*High-Speed Uplink Packet Access*) es también una optimización del UMTS que ofrece velocidades de subida de hasta 5.6 Mbps.

HSPA+ (*High-Speed Packet Access Plus*) permite velocidades de subida de 11Mbps y de bajada de 42Mbps.

DC-HSPA (*Dual Cell HSPA*) permite velocidades de subida de 24 Mbps y de bajada de 42Mbps.

LTE (*Long Term Evolution*) su versión final data de Diciembre 2009 y se basa en la necesidad de seguir ofreciendo la misma competitividad del estándar UMTS de tercera generación. Esta tecnología ofrece tiempos de latencia muy bajos de hasta 10 ms, es compatible con las versiones anteriores de UMTS y el ancho de banda por canal físico es variable en el rango 1.4 – 20 MHz. Permite velocidades de hasta 300 Mbps en bajada y 75 Mbps en subida [16].

## **4.4. Telefonía móvil de cuarta generación o 4G**

La Unión Internacional de Telecomunicaciones (ITU - *International Telecommunication Union*) acuñó el término de IMTA (*Improved Mobile Telecommunication Advanced*) para designar un nuevo estándar que vaya más allá del ya existente 3G. Este nuevo estándar debería ser capaz de:

- Ofrecer funcionamiento alrededor del mundo y posibilidad de “roaming” a nivel global.
- Ofrecer compatibilidad con los estándares anteriores.
- Posibilidad de interconexión con redes de acceso radio.
- Velocidades punta de acceso de hasta 1 Gbps en bajada.
- Que sea una red de conmutación de paquetes implementada sobre IP.

La nueva versión de LTE fue bautizada LTE Advanced, ésta no es una nueva tecnología, más bien una mejora de la anterior LTE de tercera generación. Esta tecnología, que se está sometiendo actualmente a un análisis a fondo por parte de la Unión Internacional de Telecomunicación, cumple e incluso sobrepasa algunos de los criterios impuestos por esta organización, como por ejemplo, la capacidad de alcanzar 1 Gbps en velocidad de bajada y 500 Mbps en velocidad de subida.

WiMAX es otra tecnología que intenta ser candidata para cumplir los requisitos impuestos por la ITU. Esta tecnología, que está desarrollada por IEEE (*Institute of Electrical and Electronics Engineers*) en el estándar 802.16e, ofrece una velocidad de 128 Mbps en sentido de bajada y 56 en sentido de subida, utilizando como tecnología radio una variante de multiplexación por división de frecuencia.

Como el rendimiento anterior no era suficiente para ser considerado una tecnología 4G, el IEEE mejoró dicha tecnología en un nuevo estándar, el 802.16m comúnmente conocido como WiMax2. Este estándar sí que cumple con los requisitos IMT Advanced [17] pudiendo así ser considerado un estándar de cuarta generación. Esta tecnología especifica sistemas capaces de soportar usuarios con movilidad alta o baja, distintas velocidades en función del entorno del usuario, aplicaciones multimedia de alta calidad y otros aspectos importantes relacionados con la calidad de servicio.

Aunque su implementación no se ha llevado a cabo actualmente, se prevé que las primeras teleoperadoras ofrecerán esta tecnología a los usuarios a finales del año 2012. Con esta tecnología, se puede decir que se lleva a cabo una homogeneización de las tecnologías de telecomunicación, lo que afectará a la arquitectura de los dispositivos móviles y permitirá una mayor ubicuidad de los servicios y aplicaciones.

## 5. Evolución del software para dispositivos móviles

Al igual que las tecnologías móviles, el software ha ido evolucionando condicionado por las características de los terminales móviles y las tecnologías de comunicación vistas en el capítulo 4. En este capítulo se hará un repaso de los distintos tipos de aplicaciones, la relación que tienen con las tecnologías móviles existentes y las distintas tecnologías software que se pueden utilizar para la creación de estas aplicaciones.

El primer terminal móvil fue el Motorola Dynatac 8000x, que vio la luz en 1983 y medía 33 cm de longitud, 4.5 cm de ancho y 9 cm de largo, con un peso aproximado de 790 gramos. Este terminal ofrecía solamente servicios de voz, no disponía de pantalla, y tal y como se ha visto en el capítulo 4. las aplicaciones y el servicio de texto SMS, no se desarrollaron hasta la implementación del estándar GSM, que fue 9 años después.

Los PDA (*Personal Digital Assistant*) se introdujeron a partir de 1992, al principio solamente funcionaban en modo fuera de línea, después fueron poco a poco adquiriendo conectividad a Internet, utilizando conexión por línea conmutada, redes móviles y más tarde Wi-Fi. En la misma época se hicieron hueco en el mercado los teléfonos móviles, tal y como se conocen hoy en día, con su pantalla que permitía al usuario hacer uso de nuevo software desarrollado especialmente para estos dispositivos. El desarrollo de aplicaciones para estas primeras generaciones de teléfonos móviles se hacía únicamente por parte de las empresas productoras de estos terminales, como Apple, Windows, Nokia, Sony o Motorola y dentro de estas por un grupo muy limitado de desarrolladores. En este punto las aplicaciones que accedían a Internet utilizaban el protocolo WAP (*Wireless Access Protocol*) [18] que englobaba una serie de tecnologías como WML (*Wireless Markup Language*), WSP (*Wireless Session Protocol*) o WTLS (*Wireless Transport Layer Security*) que permitían que los dispositivos inalámbricos se pudieran utilizar para acceder a correo electrónico, noticias y otros servicios en Internet. Así pues, las aplicaciones móviles en los años 90 ya daban acceso a correo electrónico, y poco a poco las aplicaciones de mensajería instantánea o IM (*Instant Messaging*) fueron también ocupando su sitio en la vida de los usuarios.

En la década de los 2000, otros tipos de aplicaciones se fueron introduciendo en los dispositivos móviles que ha permitido ahorrar costes, especialmente desde el punto de vista de los usuarios de los terminales móviles, son las aplicaciones de voz sobre IP o VoIP. Obsérvese que la introducción de este tipo de aplicaciones no es totalmente fortuita, sino que coincide con la introducción de la tecnología móvil de tercera generación, que permite velocidades más altas, tal y como se ha visto en el capítulo 4. Otro tipo de aplicaciones que

se introducen en el mercado son las de envío de contenido multimedia, mediante las cuales se permite a los usuarios compartir imágenes, sonidos o videos. El desarrollo de software sigue siendo una tarea dirigida por las grandes corporaciones que comercializan estos terminales, y el usuario final no tiene acceso a desarrollar software para su propio dispositivo, ya que las plataformas utilizadas son muy privativas. Un resumen del avance en el campo de las aplicaciones software para dispositivos móviles en relación con la tecnología que ha hecho posible estos avances (véase cap. 4) se puede ver en la Figura 5.1<sup>1</sup>.

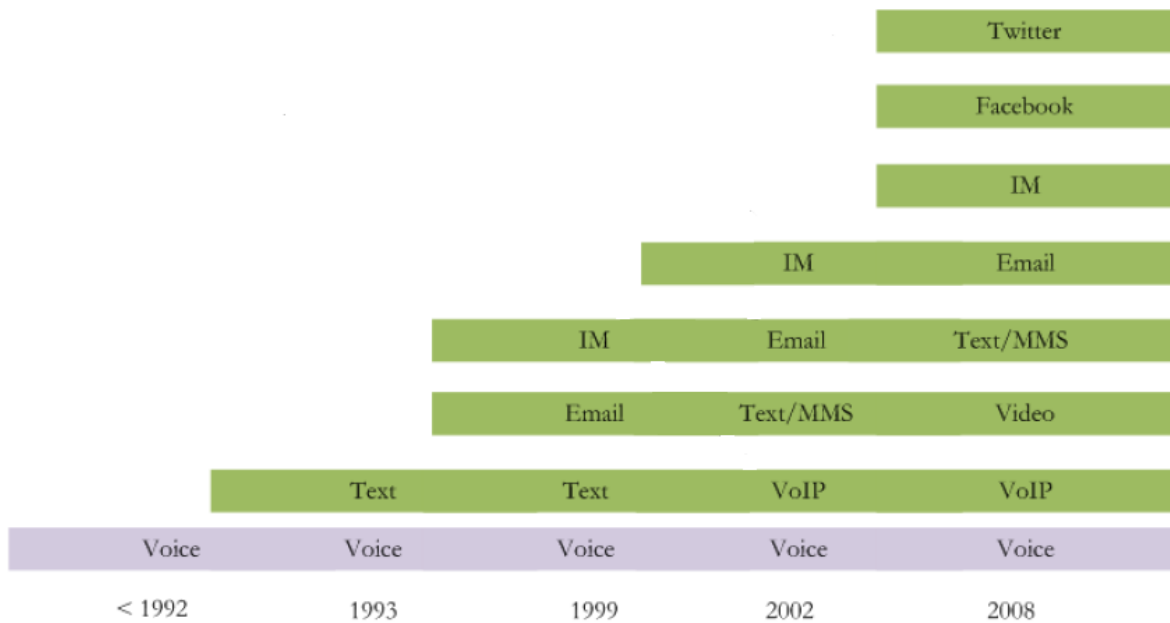


Figura 5.1 Evolución de los servicios y aplicaciones móviles

El desarrollo de otros dispositivos portátiles, como cámaras, consolas de video, y reproductores mp3, entre otros, ha hecho que los usuarios pidan cada vez más tener un dispositivo capaz de realizar todas estas tareas, además de funcionar como teléfono móvil. El precio de la memoria se está abaratando cada vez más, las baterías están mejorando su tiempo de vida, mientras que a nivel de software cada vez más terminales móviles introducen sistemas operativos muy conocidos basados en Windows o Linux. Éstas son algunas de las razones más importantes para el desarrollo de los nuevos terminales móviles, llamados *smartphones*, o terminales inteligentes.

La competencia cada vez más dura entre las distintas compañías del sector, así como las ideas y requerimientos en auge por parte de los usuarios, hacen que estas empresas cambien sus políticas proteccionistas con respecto a las plataformas de desarrollo software permitiendo de esta forma al usuario final idear y crear su propio software en función de sus necesidades. Poco a poco se hacen públicas plataformas de desarrollo para dispositivos móviles de las cuales destacamos, Java ME [19] , BREW [20] , iOS SDK [21] o Android SDK

<sup>1</sup> Fuente: [71]

[22]. A continuación se ofrece un repaso de estas plataformas de desarrollo, así como de las principales ventajas que ofrecen.

*Java ME (Java Micro Edition)* es una tecnología que se creó para dar soporte al desarrollo de aplicaciones para dispositivos con recursos limitados. Según Oracle [23], en el mercado existen 3 mil millones de dispositivos que pueden hacer uso de esta tecnología, decir también que los dispositivos más nuevos, que tienen sistema operativo Android, iOS, Windows Phone o BlackBerry, no utilizan esta tecnología, ya que cada uno de ellos dispone de su propia plataforma. Java ME está basado sobre los siguientes 3 conceptos:

- *Configuración*: no es más que un set básico de librerías y una máquina virtual, que pueden ser usadas por un amplio número de dispositivos. Existen dos tipos de configuración dentro de Java ME: la Configuración para Dispositivos de Conexión Limitada (CLDC – *Connected Limited Device Configuration*) y la Configuración para Dispositivos Conectados (CDC – *Connected Device Configuration*).
- *Perfil*: es un grupo de librerías soportadas por un número más reducido de dispositivos. Se puede considerar un perfil como un subconjunto de una configuración.
- *Paquete opcional*: son librerías específicas a una tecnología en particular.

Como resumen de la estructura descrita con anterioridad presentamos la Figura 5.2<sup>1</sup>.

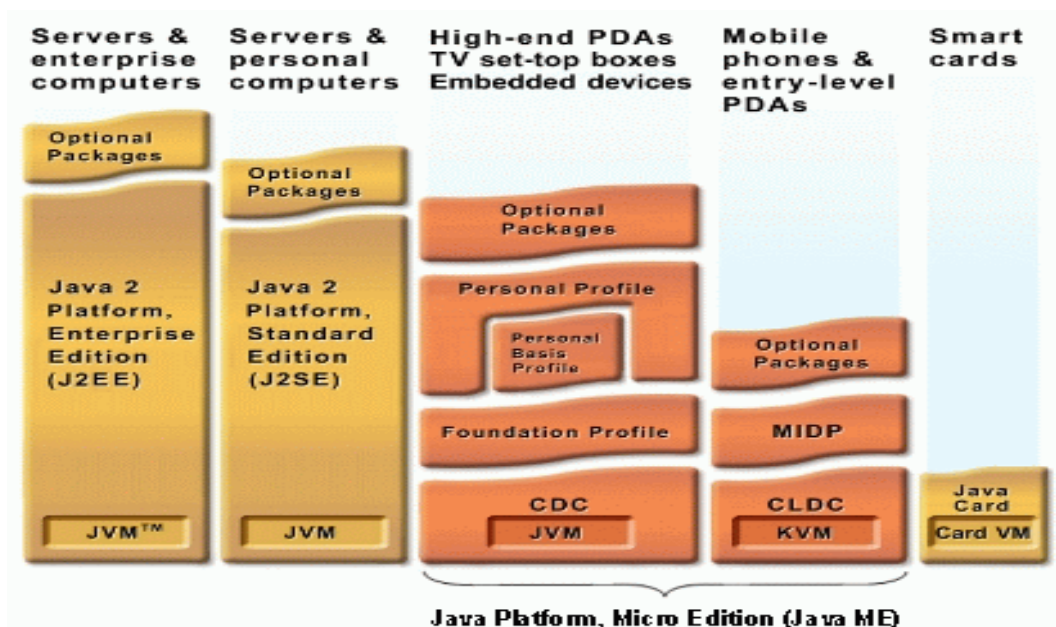


Figura 5.2 Estructura Java ME

<sup>1</sup> Fuente: [19]

*BREW (Binary Runtime Environment for Wireless)* es una plataforma de desarrollo de aplicaciones creada por Qualcomm que permite a los usuarios desarrollar aplicaciones en C/C++ o Java haciendo uso del kit de desarrollo que esta empresa ofrece. La ejecución de las aplicaciones no se realiza sobre una máquina virtual, sino que hace uso de un pseudo sistema operativo de tiempo real, lo que ofrece sus beneficios a la hora de priorizar procesos. Cabe destacar que la plataforma se utiliza para crear software para dispositivos móviles clásicos, y no tanto para *smartphones*; de hecho, el primer y único *smartphone* que permite ejecutar aplicaciones BREW es el HTC Freestyle, que se lanzó en el año 2011.

*iOS SDK (i Operating System Software Development Kit)* es un kit de desarrollo de la empresa Apple para los dispositivos que comercializa y que hacen uso del sistema operativo iOS, es decir, *iPhone*, *iPad* y *iPod Touch*. El kit de desarrollo está compuesto por lo siguiente:

- Las librerías necesarias que forman las distintas interfaces de programación de aplicaciones.
- Un entorno de desarrollo integrado llamado xCode donde se implementa, depura, verifica y ejecuta el código.
- *iOS Simulator*, que es una aplicación *MAC OS X* que nos permite simular la ejecución de la aplicación.
- *iOS Developer Library* compuesta por la documentación y referencias que ayudan a los programadores durante el proceso de desarrollo.

Para desarrollar aplicaciones haciendo uso de esta plataforma es necesario disponer de un ordenador que corra el sistema operativo *MAC OS X* pudiendo instalar de esta forma las herramientas necesarias para el proceso de desarrollo. Decir también que el lenguaje de programación en el cual se realiza la implementación es *Objective C*, un lenguaje que apareció en el año 1983, es orientado a objetos y tiene influencias de *C* y *Smalltalk*.

*Android SDK (Android Software Development Kit)* es el kit de desarrollo de la empresa Google para los dispositivos que hacen uso de su sistema operativo Android. En la Figura 5.3<sup>1</sup> se puede observar la arquitectura del sistema operativo Android y los niveles o capas que tiene. Tal y como se puede observar, la capa de aplicaciones se apoya en las distintas librerías que forman el marco de trabajo de las aplicaciones. A su vez, las librerías anteriormente mencionadas se basan en otras de más bajo nivel, véase el nivel *Libraries*. Las aplicaciones creadas se ejecutan en una máquina virtual Java, denominada *Dalvik*, que a su vez se basa en un núcleo Linux y en los controladores correspondientes al hardware del dispositivo.

---

<sup>1</sup> Fuente: [22]

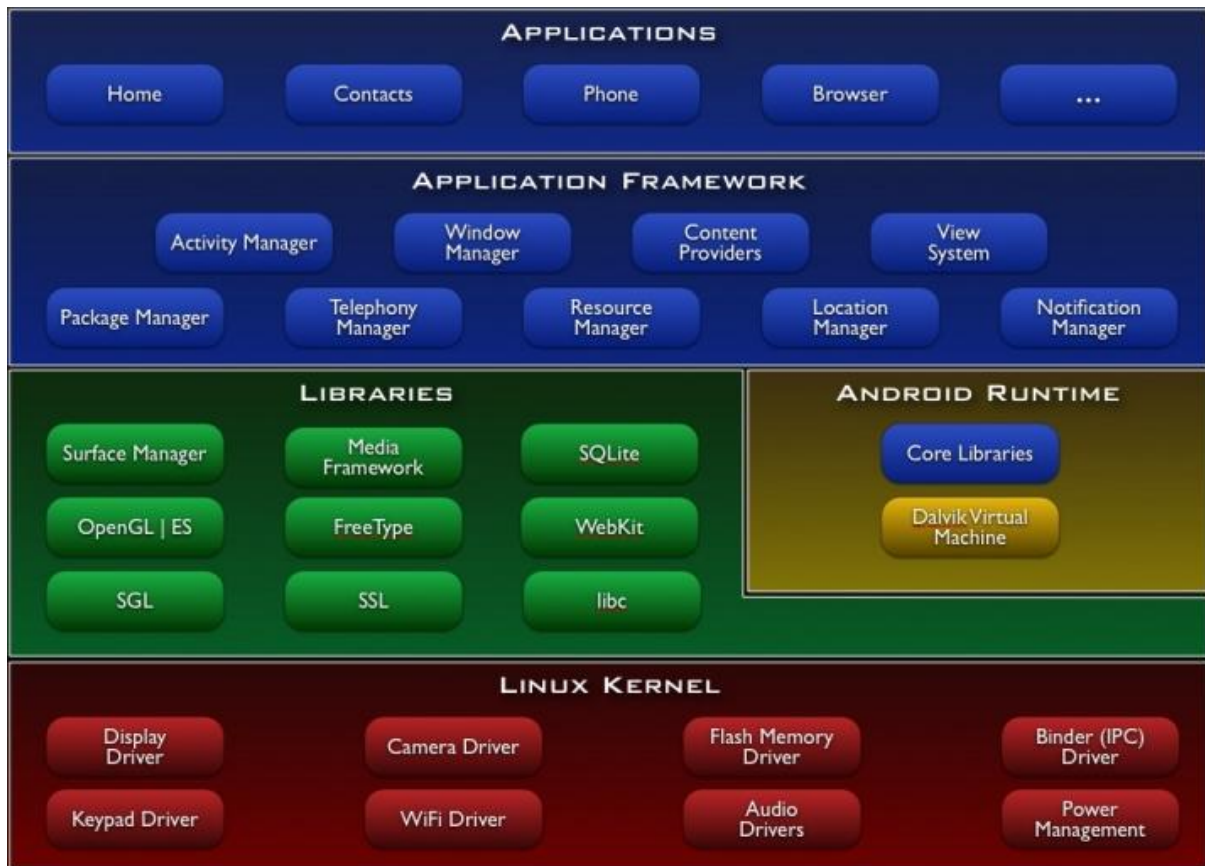


Figura 5.3 Arquitectura del SO Android

El lenguaje de programación que se utiliza para crear aplicaciones para Android es Java. El kit de desarrollo está compuesto por una serie de herramientas, de las cuales destacamos el emulador y el depurador. Aparte, existen las herramientas correspondientes a la plataforma, que se actualizan con cada nueva versión del sistema operativo, y de las cuales destacamos el Puente de Depurado Android (ADB – *Android Debug Bridge*) que nos facilita el control del emulador durante la ejecución.

Vistas las plataformas de desarrollo más interesantes en el mundo móvil, en la Figura 5.4<sup>1</sup> se presenta un estudio de mercado, que pone de manifiesto la posición de las distintas compañías en este ámbito de importante auge hoy en día.

<sup>1</sup> Fuente: [71]

<b>Operating System</b>	<b>4Q11 Units</b>	<b>4Q11 Market Share (%)</b>	<b>4Q10 Units</b>	<b>4Q10 Market Share (%)</b>
Android	75,906.1	50.9	30,801.2	30.5
iOS	35,456.0	23.8	16,011.1	15.8
Symbian	17,458.4	11.7	32,642.1	32.3
Research In Motion	13,184.5	8.8	14,762.0	14.6
Bada	3,111.3	2.1	2,026.8	2.0
Microsoft	2,759.0	1.9	3,419.3	3.4
Others	1,166.5	0.8	1,487.9	1.5
<b>Total</b>	<b>149,041.8</b>	<b>100.0</b>	<b>101,150.3</b>	<b>100.0</b>

Figura 5.4 Cuota de mercado en smartphones según SO

Tal y como se ha visto con anterioridad, a estas alturas ya existen varias plataformas para el desarrollo de software para dispositivos móviles, y esto es debido principalmente a una política más abierta hacia el usuario y empresas no directamente relacionadas con la fabricación de terminales, lo que hace que aparezcan o se desarrollen más ampliamente nuevos tipos de aplicaciones como:

- *Sitios web móviles*: son sitios web especialmente desarrollados para dispositivos móviles, no se tienen que confundir con la visualización de un sitio web normal en un navegador móvil.
- *Widgets web móviles*: relacionados con los sitios web móviles, los widget web son partes de código HTML ejecutado de una manera peculiar por el usuario final, decimos peculiar porque se necesita de algún otro software sobre el cual ejecutarse, normalmente una plataforma widget compatible.
- *Aplicaciones web móviles*: son aplicaciones que proveen una interfaz y experiencia parecida a las aplicaciones nativas pero que están construidas utilizando lenguajes como HTML, CSS y JavaScript propios de la web. Un ejemplo de tal aplicación se puede observar en la Figura 5.5.
- *Aplicaciones nativas y juegos*: son aplicaciones que se crean y compilan para cada plataforma móvil en particular, por ello, el nombre más adecuado sería aplicaciones de plataforma. Hacen uso más extensivamente de los sensores y características de cada terminal y plataforma en particular. Un ejemplo de este tipo de aplicación se puede ver en la Figura 5.6.

Gracias a los avances realizados en la tecnología y a la multitud de sensores que llevan incorporados los smartphones actualmente, tales como termómetros, acelerómetros,



sensores de proximidad o cámaras fotográficas cada vez las aplicaciones desarrolladas para las plataformas móviles son más lucrativas. Entre los varios mercados existentes se calcula que el valor intercambiado se aproxima a los quince mil millones de dólares para el año 2011 y esta cifra podrá rondar los cincuenta y ocho mil millones de dólares para el año 2014 según la revista Gartner [24].



Figura 5.5. Aplicación web móvil



Figura 5.6. Aplicación nativa

En este capítulo se ha repasado brevemente la historia del software para dispositivos móviles y se han visto varios conceptos del entorno actual en el mundo del smartphone y de las aplicaciones que lo rodean. En los capítulos venideros se hará un análisis del estado del arte de las tecnologías que influyen en el cumplimiento de nuestros objetivos.



## 6. Métodos de localización en dispositivos móviles

Durante los últimos años se han incrementado las aplicaciones sensibles al contexto, y una parte considerable del contexto se refiere a la localización del usuario, gracias a algún dispositivo que lleve consigo.

A continuación se verá una breve descripción de las distintas tecnologías en base a las cuales los sistemas de localización calculan la posición de un dispositivo móvil. Para la descripción de estos métodos nos basaremos en lo indicado en [25].

*Ángulo de Llegada o AOA (Angle of Arrival)*: Su funcionamiento básico se puede llevar a cabo usando un vector de sensores, separados por fracciones de longitud de onda. Imaginémonos un vector de sensores compuesto por dos sensores,  $s1$ ,  $s2$ , separados por una distancia,  $d$ . Tal y como se puede observar en la Figura 6.1 para llegar al sensor  $s2$  la onda (rojo) viaja  $d \cdot \sin(\theta)$  más que para llegar al sensor  $s1$ . Usando trigonometría simple se puede averiguar el ángulo proveniente de la señal; si por ejemplo la onda llega simultáneamente a  $s1$ ,  $s2$  se puede concluir que llega con un ángulo  $\theta$  de  $0^\circ$  o  $180^\circ$ , por lo que el ángulo proveniente de la señal es  $90^\circ - \theta$ .

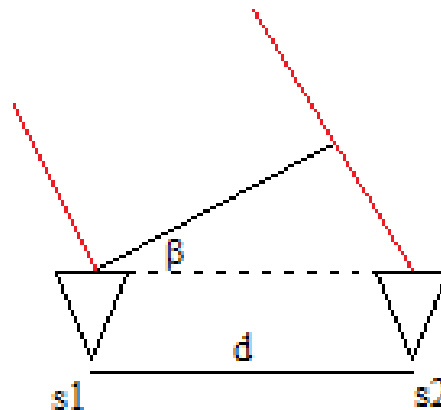


Figura 6.1. Ángulo de Llegada de una señal a un vector de sensores

En consecuencia, en un plano bidimensional, con dos vectores de sensores se puede averiguar el origen de la onda mediante la intersección de las dos trayectorias, y por lo tanto, identificar el posicionamiento del objeto emisor.

*Tiempo de Llegada (TOA – Time of Arrival):* A veces llamado *tiempo de vuelo*, se basa en el cálculo del tiempo de llegada de una señal desde el emisor hasta el receptor. Si se conoce la velocidad de propagación en el medio, entonces se puede determinar la distancia a la que el emisor está con respecto al tiempo de vuelo. Para llevar a cabo esta medida se necesita de una línea de visión directa entre el emisor y el receptor, es decir que no haya obstáculos. Decir también, que por motivos obvios, esta medida necesita de sincronización entre el emisor de la onda y el sensor receptor. La traza de posibles posiciones será un círculo alrededor del receptor. Cuando se combinan más de tres receptores en una misma zona, se puede calcular la posición del origen de la señal, como la intersección de los tres círculos.

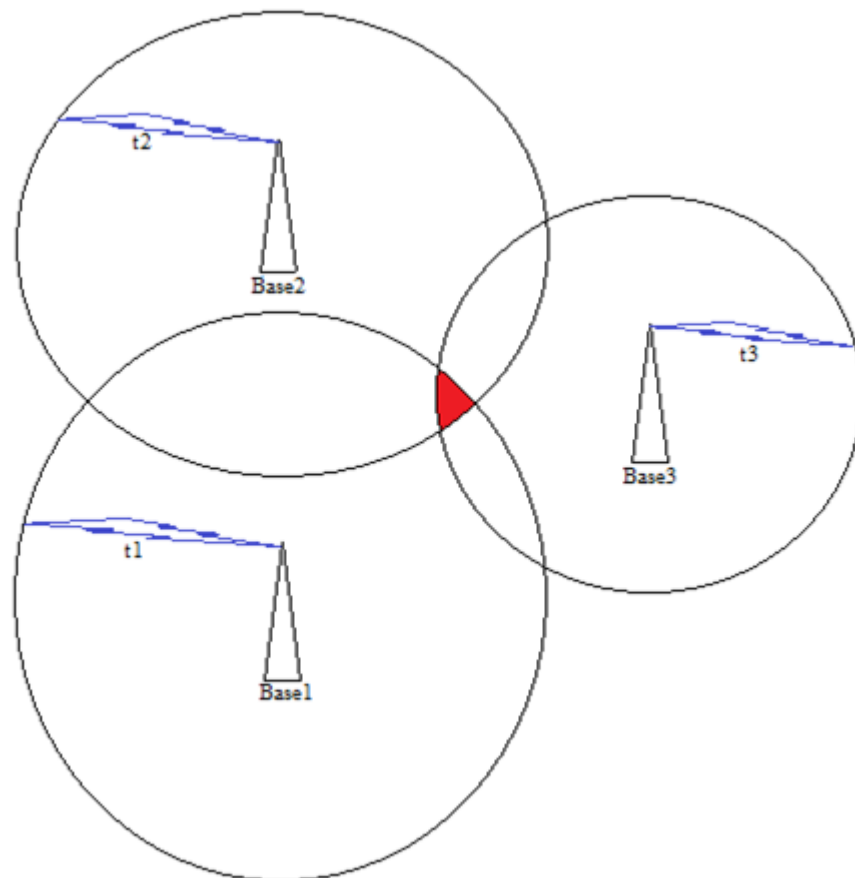


Figura 6.2. Funcionamiento de TOA con 3 bases

*Diferencia de tiempo de llegada (TDOA – Time Difference of Arrival):* Para evitar la necesidad de sincronización entre el emisor y el receptor, se utilizan dos o más receptores sincronizados. El emisor emite una onda que tarda un tiempo  $t_1$  para llegar al sensor  $s_1$  en función de la distancia, si no se sabe el tiempo del emisor, es imposible averiguar  $t_1$ , y por lo tanto, hacer una traza de posibles posiciones. Aun así, es posible averiguar la diferencia de tiempo que la onda tarda en llegar a  $s_1$  y  $s_2$ , utilizando esta se puede trazar una hipérbola de posibles posiciones del origen de la señal. El funcionamiento de esta tecnología se puede observar en la Figura 6.3, donde se supone que cada base recibe la señal emitida por un

dispositivo móvil, unos nanosegundos antes o después que sus estaciones bases vecinas, por lo cual se encuentra unos metros más cerca o más lejos respectivamente del emisor.

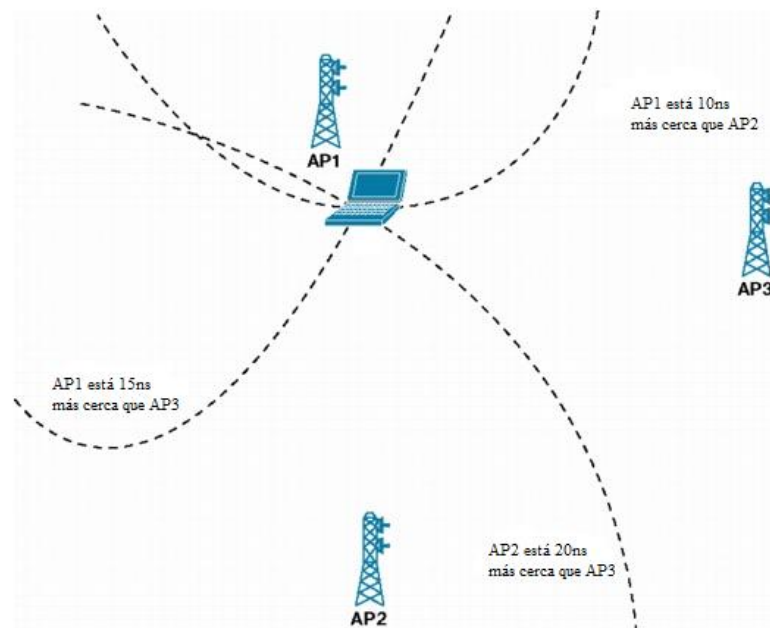


Figura 6.3. Funcionamiento de la tecnología TDOA<sup>1</sup>

Por otra parte, Zimpekis et al. [26] divide los métodos de localización en dos tipos:

- *Métodos propios.*
- *Métodos remotos.*

Los dispositivos que utilizan *métodos propios* son aquellos que utilizan señales transmitidas por antenas u otras emisoras para calcular su propia posición. A diferencia de estos, los dispositivos móviles que utilizan *métodos remotos* son posicionados utilizando varios receptores que miden intensidades en las señales que estos dispositivos móviles transmiten.

## 6.1. Localización por ID de celda, TDOA, TOA, AOA

Con la aparición de la primera generación de telefonía móvil surgió el problema de la localización del usuario; para ello los sistemas de primera generación hacían uso de *multilateración*, utilizando alguno de los métodos de funcionamiento remoto descritos anteriormente.

---

<sup>1</sup> Fuente: [68]

Aparte de estos métodos también se utilizaba el método de *localización por ID de celda*. Este método, que es muy básico, se basa en que una vez que un terminal está conectado a la red móvil, la red tiene que conocer su posición para poder facilitarle la comunicación. Como se dijo en el primer capítulo, muchas de las celdas están divididas en sectores utilizando una o más antenas, por lo que con este método no solo se puede identificar la celda en que se encuentra un dispositivo, sino que también el sector. Aunque sea posible identificar la celda y el sector esta tecnología no es de las más precisas, ya que las celdas que funcionan en un ancho de banda bajo (900 MHz, por ejemplo) tienden a ser más extensas que las que funcionan en bandas altas. El terreno y la densidad de la población también son factores que delimitan la extensión de una celda. En las zonas urbanas altamente pobladas una celda puede tener una extensión de 1 km o menos, no es el caso de las zonas rurales, donde una celda puede llegar a cubrir un área de 35 km. Supongamos una celda que cubre un área de  $5 \text{ km}^2$  utilizando 3 antenas, cada una dando servicio a  $120^\circ$  del área total, en este caso la precisión de este método es de  $\frac{5 \text{ km}}{3} = 1.66(6) \text{ km}^2$ .

Con la introducción del estándar de segunda generación GSM, se introdujo también una tecnología nueva en la localización de dispositivos móviles, el GPS o más propiamente dicho GPS Asistido.

## 6.2. GPS

El Sistema de Posicionamiento Global (GPS - Global Positioning System) [27] es una tecnología que utiliza un método de localización propio. Fue creada por el departamento de defensa de los Estados Unidos en 1973 para que funcionara en el exterior e incorpora ideas de algunos sistemas de localización previos. El GPS basa en la comunicación de al menos 4 satélites activos (41 en total hoy<sup>1</sup>) con receptores GPS en la Tierra.

La potencia de la señal que los satélites transmiten apenas llega a 27 vatios (W); para tener una referencia de lo que esto significa tener en cuenta que las bombillas que iluminan nuestras casas pueden utilizar perfectamente 100 W. Teniendo en cuenta que los satélites están a una distancia de aproximadamente 20200 km cuando la señal llega a los dispositivos receptores tiene aproximadamente una potencia de 100 attoW donde atto significa  $10^{-18}$  W. Es por ello por lo que cualquier obstáculo, ya sea muro, árboles hasta el techo de un coche puede hacer que está señal no sea óptima y por lo tanto el GPS no pueda ser utilizado. Decir también que todos estos satélites son de procedencia estadounidense, aunque también hay otros sistemas de posicionamiento, como GLONASS, que es de Rusia, o Galileo, de la Unión Europea.

---

<sup>1</sup> 20 Diciembre 2011

Hay tres tipos de datos que los satélites transmiten:

- Tiempo de transmisión del mensaje.
- Coordenadas orbitales precisas, también conocidos como *datos efemérides*.
- Estado general del satélite y coordenadas genéricas para los demás satélites, esto se conoce como *almanaque*.

Cada satélite emite sus datos efemérides cada 30 segundos, estos datos son válidos durante 4 horas desde el tiempo de transmisión del mensaje. Al igual pasa con el almanaque, pero éste se transmite cada 12.5 minutos y es válido durante 180 días.

Estos datos se transmiten a una velocidad de 50 bps y cada satélite los transmite sobre 2 frecuencias L1 (1575.42 MHz) y L2 (1227.60 MHz). Para el uso civil se tiene en cuenta únicamente los datos mandados por L1 mientras que para el uso militar se tiene en cuenta los dos. En el caso del GPS militar se realiza un posicionamiento más exacto mediante una corrección ionosférica, debido a que la ionosfera tiene un efecto negativo sobre los datos transmitidos.

En la utilización de esta tecnología es muy importante el término “tiempo de primera fijación”, este término se refiere al tiempo transcurrido desde que se enciende el receptor hasta que se puede calcular nuestra posición. Este tiempo es en la mayoría de los casos de 1 minuto, dado que el almanaque tiene una validez de 6 meses y no necesita ser descargado cada vez, lo único que se tendrá que descargar son los datos efemérides. El proceso para obtener la primera fijación pasa por varias fases:

- Cálculo del rango desde el receptor a los distintos satélites.  
Cuando se habla de rango, nos referimos a distancia, en tiempo, entre el reloj de nuestro dispositivo receptor y el reloj del satélite. Los dos relojes tienen errores que se tienen que corregir durante el proceso de localización, mientras que el error del tiempo en los satélites es conocido de antemano no ocurre lo mismo con el reloj del receptor, y éste tiempo se tiene que calcular durante el proceso de la primera fijación.
- Cálculo de la posición de los satélites, utilizando los datos efemérides (órbita y reloj).  
Los datos efemérides, en su sentido más estricto son una tabla de datos con coordenadas solares. En nuestro contexto, se entiende por datos efemérides los parámetros que determinan la órbita que se deduce de estos números más la diferencia entre los dos relojes.
- Cálculo de nuestra posición en función de los parámetros anteriores

Un receptor GPS puede calcular el rango de los satélites en varios milisegundos, pero el tiempo en el que recibe los datos efemérides y su decodificación es bastante más lenta, sobre todo en las ciudades, donde la comunicación con el satélite se puede ver interrumpida por edificios, árboles etc. Como los datos efemérides se transmiten una vez cada 30 segundos, si seríamos capaces de obtener estos datos inmediatamente, nuestra búsqueda se puede acortar al no tener que esperar los datos del satélite.

La tecnología AGPS viene en ayuda para resolver los problemas de tiempo en obtención de los datos efemérides o problemas de baja señal.

### **6.3. AGPS o GPS Asistido**

Como se ha visto, un sistema GPS necesita varios parámetros para calcular la posición, pero antes de poder comunicarse con los satélites, este receptor necesita saber la frecuencia y el código de retardo con la que ponerse en contacto con estos. Aunque todos los satélites transmiten los datos a la misma frecuencia absoluta de 1575,42 MHz, los datos no se perciben igual en la Tierra debido al efecto Doppler.

El efecto Doppler explica el cambio en la frecuencia relativa cuando un objeto se aleja o acerca de la fuente de frecuencia absoluta. Éste es el efecto por el cual cuando escuchamos el sonido de la sirena de un coche acercándose, lo escuchamos más fuerte y más agudo que cuando se aleja.

Un receptor, si de antemano no conoce esta frecuencia relativa, modificada por el efecto Doppler, empezará a buscar por todas las frecuencias posibles dentro de un rango, y aunque encuentre alguna, no podrá comunicarse con el satélite hasta conocer el código de retardo, habiendo así un espacio bidimensional donde buscar.

AGPS significa GPS Asistido [28]. Los datos de asistencia nos ayudan a restringir este espacio al mínimo posible y hacer la búsqueda mucho más eficiente, estos datos pueden consistir en:

- Almanaque y datos efemérides.
- Tiempo preciso, que evita necesidades de sincronización.

Como se ha visto antes, un dispositivo receptor que dispone del almanaque y de los datos efemérides, y que tiene que calcular una posición por primera vez, puede tardar aproximadamente 1 minuto. Ahora bien, un receptor que ya ha computado anteriormente una posición tiene una idea de varios de estos parámetros, aunque éstos no estén lo más actualizados posible por lo que el receptor puede tardar 30 segundos en encontrar la nueva posición. Como se puede observar, el tiempo se ha reducido a la mitad, siendo esto además



lo más común. El mejor caso es si un receptor GPS ha sido utilizado, apagado y encendido otra vez minutos después; en este caso, los datos efemérides ya se conocen, al igual que la diferencia temporal entre receptor y satélite, por lo que el tiempo de búsqueda de la frecuencia también disminuye sustancialmente, pudiendo el receptor llegar a calcular la nueva posición en aproximadamente 1 segundo.

En los dispositivos móviles, los datos de asistencia provienen de un servidor de localización de la red GSM, o posteriormente UMTS. Este servidor dispone de servicios que están continuamente actualizando sus datos con respecto a los satélites disponibles en el área geográfica y además las redes móviles a las que pertenece pueden transmitir datos hacia los terminales móviles a velocidades mucho más elevadas que los satélites, como se ha visto en el capítulo 4, las evoluciones de GSM pueden llegar a 384 Kbps y las de UMTS a 300 Mbps haciendo que los dispositivos AGPS integrados en los smartphones sean viables.

Por último, decir también que estas tecnologías están disponibles para el 100% de la población global, sin coste, o mejor dicho, los costes van a cargo de los contribuyentes estadounidenses, y que su precisión puede llegar a ser de hasta 1 metro. No vamos a entrar en ejemplos de aplicaciones útiles que hacen uso de esta tecnología, hoy en día ubicua.

## 6.4. Posicionamiento Wi-Fi

Es un método de posicionamiento creado para funcionar en el interior (de edificios), aunque también funciona en el exterior en áreas urbanas, haciendo uso de un método de localización basado en el Indicador de la Potencia de Señal Recibida (RSSI - *Received Signal Strength Indicator*) y de las direcciones MAC únicas que emiten los puntos de acceso de nuestras redes Wi-Fi.

Este método de posicionamiento es posible gracias a una idea pionera en el campo que tuvo la empresa Skyhook Wireless [29], y que posteriormente ha sido copiada por empresas como Google o Apple. Esta idea consiste en mantener una base de datos extensa de todos los puntos de acceso y su localización, haciéndose esta recopilación de forma transparente al usuario (un coche patrullando las calles y recolectando información) o de forma explícita mediante software, preguntándole al usuario si permite que estas empresas tengan acceso a su localización, para posteriormente transmitir esta información a una BD.

Cuando un usuario de esta tecnología desea conocer su posición, hace una consulta a la BD, transmitiendo como parámetro varios indicadores relacionados con las redes Wi-Fi cercanas y la potencia de la señal de cada uno, devolviéndole el servidor las coordenadas del dispositivo. Decir que este sistema está disponible en EEUU, algunos países de la EU o Asia. Su uso es recomendable en el interior o en zonas densamente pobladas, debido

obviamente a la mayor probabilidad de encontrar redes Wi-Fi catalogadas en la BD. Un mapa con las zonas en las que ofrece cobertura se puede ver en [30], también decir que la empresa pionera Skyhook sostiene que el servicio ofrece una precisión que puede rondar los 20 o 30 metros, precisión decente teniendo en cuenta que el mismo servicio de Google puede rondar varios cientos de metros.

Skyhook permite hacer uso de su base de datos mediante una API y un Kit de Desarrollo de Software, unos de los clientes más grandes de esta empresa ha sido, por ejemplo Apple, pero según Steve Jobs dejó de serlo en Abril de 2010. Algunas aplicaciones que se han desarrollado utilizando esta tecnología se pueden encontrar en la página oficial de la empresa [29]. SpotRank, por ejemplo, es una aplicación que estima la densidad de la población, en función del número de puntos de acceso disponibles; obviamente ésta aplicación únicamente tiene sentido en áreas donde está disponible esta tecnología.

Ahora que se ha hecho una introducción sobre las tecnologías disponibles para la localización de dispositivos móviles en el exterior, la Tabla 6.1 muestra un resumen de la precisión de los distintos métodos vistos.

<b>Método</b>	<b>Precisión</b>
Identificador de celda	10m-35km
Ángulo de llegada	50m-150m
Diferencia de tiempo de llegada	50m-150m
Posicionamiento Wi-Fi	20m-500m
GPS asistido	1m-10m

Tabla 6.1. Precisión de los distintos métodos de localización<sup>1</sup>

---

<sup>1</sup> Fuente: Adaptación de la Tabla 1 de [69]

# 7. Arquitectura orientada a servicios

En el presente capítulo se hará una introducción a la arquitectura orientada a servicios, su funcionamiento y las distintas tecnologías que se pueden utilizar en la creación de dichos servicios. También se ofrece una vista de la arquitectura cliente-servidor en dispositivos móviles, las diferencias que ésta tiene con respecto a la arquitectura cliente-servidor clásica y cómo estas diferencias pueden influir en el desarrollo de software para este tipo de dispositivos.

El principal problema del que surge este tipo de arquitectura es el continuo aumento de la información, la imposibilidad de las aplicaciones de ofrecer esta información de forma sencilla, útil y coherente. El continuo aumento de sistemas provoca cierta incompatibilidad a la hora de intercambiar esta información entre ellos, debido a la diferencia en la tecnología que usa cada uno. Para solucionar este problema se habían propuesto dos soluciones:

- (1) Un intercambio manual de la información, donde la transferencia la realiza una persona o grupo de personas, con el riesgo que el factor humano supone para este tipo de operaciones.
- (2) La utilización de una interfaz de intercambio propia a los dos sistemas. Esta solución no es la más eficiente ya que su mantenimiento no es trivial en el caso de que alguno de los sistemas modifique su funcionamiento.

La arquitectura orientada a servicios (SOA - *Service Oriented Architecture*) [31] se crea para ofrecer un método de integración de los distintos sistemas y aplicaciones heterogéneas. Es un tipo de arquitectura software que supone la distribución de la funcionalidad de una aplicación en unidades más pequeñas, bien definidas y débilmente acopladas, denominadas *servicios*. La forma más común de implementar este tipo de arquitectura es utilizar *servicios web*, que hacen uso de estándares de transporte de datos, y utilizan codificación y protocolos de intercambio de información que permiten la intercomunicación entre sistemas distintos.

En cuanto a los servicios y sus respectivos clientes, éstos se comunican utilizando datos en una forma estructurada y bien definida. Algunas de las estructuras que se utilizan en el intercambio de los datos pueden hacer uso del Lenguaje de Marcado Extensible (XML – *Extensible Markup Language*) [32] mientras que otras estructuras hacen uso de la Notación de Objetos JavaScript (JSON – *JavaScript Object Notation*) [33].

XML es un lenguaje de marcado creado por El consorcio World Wide Web (W3C - *World Wide Web Consortium*) que define varias reglas para codificar mensajes, de tal forma que sean legibles tanto por computadores como por personas humanas, un breve ejemplo de lo que puede ser un mensaje XML se puede observar a continuación:

```
<persona>
  <nombre>Samuel</nombre>
  <apellido>Sandru</apellido>
  <edad>23</edad>
</persona>
```

JSON es un formato basado en texto plano, que sirve para representar estructuras de datos denominadas objetos. Se trata de un formato independiente del lenguaje de programación. Un pequeño ejemplo de JSON, que representa información sobre una persona se puede observar a continuación:

```
{
  "nombre": "Samuel",
  "apellido": "Sandru",
  "edad": "23"
}
```

La gran facilidad con la cual se pueden *reutilizar* estos servicios en distintas aplicaciones es una de las características más importante que define esta arquitectura, pero además hay otras características como:

- **Abstracción:** se refiere al hecho de que la lógica de las funcionalidades que el servicio ofrece es invisible de cara al usuario.
- **Débil acoplamiento:** es la característica de los servicios de no mantener un vínculo entre ellos o entre ellos y sus clientes, realizándose la comunicación por paso de mensajes.
- **Flexibilidad y mantenimiento:** los servicios siendo independientes de la estructura en la que se integran permiten una mayor flexibilidad en el diseño.

Entre las tecnologías más conocidas que se basan en esta arquitectura son el Protocolo de Acceso Simple de los Objetos (SOAP – *Simple Object Access Protocol*) [34] y el de Transferencia de Estado Representacional (REST – *REpresentational State Transfer*) [35]. A continuación vamos a ver cómo funcionan y algunas de las ventajas e inconvenientes que ofrecen.

## 7.1. SOAP

Este protocolo se basa en XML para definir los mensajes que el servicio y el cliente enviarán y utiliza por defecto el protocolo HTTP (*Hyper Text Transfer Protocol*) [36] para llevar a cabo la comunicación entre ambos. Los mensajes en este protocolo usan una forma predeterminada muy parecida a los sobres que se utilizan para la comunicación humana; de hecho se llaman así, *sobres*.

Los sobres identifican al mensaje, para que se sepa que es un mensaje SOAP, y especifican qué tipo de codificación se ha utilizado en el resto del mensaje, aparte contienen tres cosas más: La primera, la *cabecera*, aunque opcional esta parte del mensaje ofrece información útil para saber más información específica de la aplicación, puede estar relacionada con la autenticación. La segunda parte del sobre es el *cuerpo*, este cuerpo representa el mensaje que realmente necesitan el cliente y el servidor, de modo que todas las demás partes están allí simplemente para envolver y ofrecer una forma correcta de procesar este mensaje. La tercera y última parte del mensaje, aunque realmente es más bien una parte del cuerpo, es el *elemento de error*, que expresa posibles errores en los mensajes.

En la Tabla 7.1 se puede observar una petición SOAP y la respuesta correspondiente:

<pre>&lt;?xml version="1.0"?&gt; &lt;soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap- envelope" soap:encodingStyle="http://www.w3.org/2001/12 /soap-encoding"&gt;  &lt;soap:Body xmlns:m="http://www.example.org/stock"&gt;   &lt;m:GetStockPrice&gt;     &lt;m:StockName&gt;IBM&lt;/m:StockName&gt;   &lt;/m:GetStockPrice&gt; &lt;/soap:Body&gt;  &lt;/soap:Envelope&gt;</pre>	<pre>&lt;?xml version="1.0"?&gt; &lt;soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap- envelope" soap:encodingStyle="http://www.w3.org/2001/12 /soap-encoding"&gt;  &lt;soap:Body xmlns:m="http://www.example.org/stock"&gt;   &lt;m:GetStockPriceResponse&gt;     &lt;m:Price&gt;34.5&lt;/m:Price&gt;   &lt;/m:GetStockPriceResponse&gt; &lt;/soap:Body&gt;  &lt;/soap:Envelope&gt;</pre>
---	--

Tabla 7.1. Sobres SOAP, petición (izquierda) y respuesta (derecha)<sup>1</sup>

Después de esta breve explicación del protocolo vamos a ver algunas de las ventajas e inconvenientes que presenta este protocolo:

---

<sup>1</sup> Fuente: W3Schools

- El protocolo de transporte puede ser cambiado. Así, SOAP también es capaz de utilizar el protocolo SMTP (*Simple Mail Transfer Protocol*), por ejemplo, en lugar de HTTP.
- Utiliza el Lenguaje de Descripción de Servicios Web (WSDL – *Web Service Description Language*) [37] lo que permite describir la funcionalidad que ofrece el servicio.
- No resulta problemático en el caso de tener que enviar el mensaje SOAP por varios nodos, ya que al utilizar HTTP como protocolo de transporte, el mensaje no representa un problema para los cortafuegos.
- Porque hace uso de XML en el paso de los mensajes, se puede considerar como un protocolo más pesado y lento que sus competidores, como REST, que se verá a continuación.

## 7.2. REST

Es un protocolo más reciente que su compañero SOAP, y tiene un modo de funcionamiento parecido al de la WWW en cuanto que asocia a cada recurso un localizador de recursos uniforme o URL (*Uniform Resource Locator*).

REST considera que la web ha disfrutado de un gran éxito y escalabilidad gracias a ciertos principios que se intentan incorporar en el protocolo. Algunos de estos principios son:

- *Operaciones bien definidas*: la web permite un conjunto restringido de operaciones de las cuales se pueden sobresaltar *Get*, *Post*, *Delete* y *Put*. Estas operaciones se aplican a todos los recursos de información.
- *Identificador universal*: cada recurso de información se puede alcanzar utilizando una única URL.
- *Protocolo cliente-servidor sin estado*: cada mensaje contiene toda la información necesaria para su comprensión, no es necesario manejar estados para realizar la comunicación.

Así pues, el protocolo REST utiliza parte de la funcionalidad del protocolo HTTP para modificar, obtener y borrar ciertos objetos identificados por una URL. Esto parece una ventaja en cuanto a la simplicidad, y se basa en la filosofía de que si existen cosas ya

inventadas, por qué no utilizarlas en lugar de inventar otras parecidas. Un ejemplo de esta simplicidad se da en el caso de querer obtener un objeto, siendo suficiente hacer una petición HTTP GET a su URL, para borrarlo, HTTP DELETE, y para modificarlo, HTTP POST.

Al igual que su hermano SOAP, REST puede hacer uso de XML para llevar a cabo el intercambio de información, pero aparte permite otras tecnologías, como JSON, que no siguen un protocolo de creación de mensajes, como el basado en sobres de SOAP. A nivel de implementación existe otra diferencia conceptual, y es que mientras en una aplicación que utiliza el protocolo SOAP se hacía hincapié en la variedad de los atributos de los objetos en el servidor, véase la multitud de operaciones de tipo “get” o “set” para una sola entidad, en el caso de REST, se hace más hincapié en la variedad de los objetos.

A continuación vamos a ver algunas de las ventajas e inconvenientes de este protocolo:

- *Simplicidad y rapidez:* al no utilizar el tipo de mensaje que utiliza SOAP, con su extensa cabecera, REST se suele ver como una opción más rápida y menos pesada, muy válido para aplicaciones que disponen de poco ancho de banda y necesitan pocos datos.
- No existe un *estándar* para la descripción del servicio, como WSDL para SOAP.
- No cubre todas las necesidades en los clientes, como las *transacciones* o la *coordinación*.

### **7.3. Arquitectura cliente-servidor en dispositivos móviles**

Tal y como se ha visto hasta el momento, la arquitectura orientada a servicios se basa en la existencia de proveedores de servicios y clientes de estos servicios, por lo que desde otro punto de vista esta arquitectura podría llamarse arquitectura cliente-servidor.

La arquitectura cliente-servidor es una arquitectura que se utiliza en las aplicaciones distribuidas, que distinguen entre consumidores de recursos, *clientes*, y proveedores de recursos, *servidores*. Aunque los procesos correspondientes a las dos partes, cliente y servidor, se ejecuten en la misma máquina física, la arquitectura sigue siendo una distribuida, debido a que estos procesos se ejecutan en entornos software distinto. Aparte de esta posibilidad del cliente y el servidor ejecutándose en la misma máquina física, hay otra posibilidad (de hecho, más utilizada) y es que una máquina física actúe como servidor y

una o más aplicaciones accedan a ella; de esta forma, se ve claramente el rol del servidor como proveedor de servicios.

Aunque las posibilidades de dividir los programas en partes que se ejecuten en máquinas cliente o servidor son múltiples, hay varios patrones que se han ido formando durante el tiempo. A continuación vamos a ver alguna de las características en las que se basan estos patrones.

- *Independencia de la tecnología*: hoy en día lo ideal sería desarrollar aplicaciones que se adapten a la multitud de máquinas existentes, es decir que sean independientes de plataforma. Muchas veces esto es difícil, sin embargo hay aplicaciones que consideran ciertos dispositivos como favoritos y aparte se pueden adaptar a una serie heterogénea de dispositivos.
- *Escalabilidad*: es un requisito que siempre hay que tener en cuenta para ofrecer al usuario aplicaciones capaces de altas prestaciones, dado que en las aplicaciones que se comercializan siempre puede haber un incremento en el número de usuarios que las utilizarán, ésto hay que pallearlo haciendo una *escalabilidad horizontal*, añadiendo más máquinas proveedoras de servicios, y una *escalabilidad vertical*, instalando nuevas máquinas capaces de rendimientos más altos. Los dos pasos anteriores se pueden hacer con más o menos facilidad en función de si se ha tenido en cuenta este posible factor durante el diseño de nuestro programa.
- *Requisitos*: esta parte es la base de dicha arquitectura que es responsable de cómo se estructuran las funcionalidades de una aplicación. También los requisitos son parte fundamental de cuestiones que tienen que ver con la usabilidad o la viabilidad de un proyecto.

Para garantizar la reutilización, escalabilidad y flexibilidad en una aplicación software, esta se suele diseñar y crear para funcionar por capas. Esta división en la aplicación se puede asociar a los distintos módulos que componen la funcionalidad de una aplicación y que se ejecutan en la misma máquina física. Un ejemplo de varios módulos que se podrían considerar serían los siguientes:

- *Módulo de interfaz de usuario o capa de presentación*: presenta al usuario la interfaz de la aplicación, y le permite interactuar con esta.
- *Módulo de procesamiento o capa de negocio*: encargado de recoger solicitudes del usuario y devolver respuestas.



- *Módulo de gestión de datos o capa de datos*: se ocupa de la comunicación con la base de datos.

Cada una de estas capas tiene distintas funciones bien delimitadas, y lo que se consigue con esta delimitación es que si en un momento dado, por causas determinadas, la aplicación necesita un cambio, el mantenimiento se efectúe de forma fácil únicamente en la capa correspondiente. Según [38], en la aplicación cliente se suelen utilizar de 0 a 3 capas; el caso más especial, el de 0 capas, se puede observar en el caso de los navegadores web, como aplicación cliente que depende íntegramente del servidor para ofrecer sus funciones. Este tipo de clientes que utilizan 0 capas, o clientes que se utilizan dependiendo casi íntegramente del servidor se llaman *clientes ligeros*. La otra opción son los clientes que pueden ofrecer sus funcionalidades limitadas sin conectarse a un servidor, pero que ofrecen todas las funcionalidades cuando están en comunicación con el proveedor de datos/servicios. Así que se puede distinguir entre clientes que dependen en gran parte del servidor (en general clientes de 0 o 1 capa) y clientes que no dependen tanto del mismo (clientes de 1 a 3 capas) de modo que la nomenclatura que se utiliza para unos y otros es la de *cliente ligero* o *cliente pesado*.

Se ha hablado de división por capas en la parte del cliente, pero esta división también se puede realizar en la parte del servidor, aunque por definición, en este caso no se puede hablar de un servidor de 0 capas. Un servidor dividirá su código en función de las capas, pero aparte, aquí se tiene un concepto nuevo el de *división por niveles*. Un nivel no es más que una capa que se ejecuta en una máquina física distinta. ¿Por qué es esto necesario? Porque la división por niveles es la que nos ofrece en gran parte la escalabilidad y seguridad en la parte del servidor. La seguridad se puede reforzar colocando cortafuegos entre los distintos niveles en la parte del proveedor de servicios, creando lo que se conoce como *zona desmilitarizada*, que se encargue de ofrecer servicios a los clientes externos, pero no ofrezca acceso desde los clientes a los siguientes niveles del servidor. En la Figura 7.1 se puede observar un ejemplo de arquitectura cliente-servidor estructurada por capas y niveles.

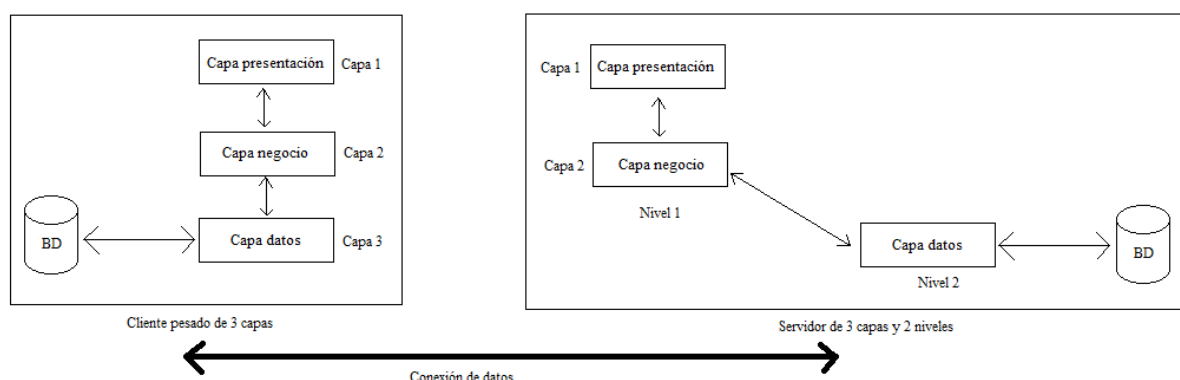


Figura 7.1. Arquitectura cliente-servidor de 3 capas y servidor de 2 niveles

A continuación se tratará la arquitectura cliente-servidor, pero desde una perspectiva más orientada a los dispositivos móviles. Hay varias formas mediante las cuales los dispositivos móviles se pueden comunicar, ya sea con alguna red, con el servidor o entre ellos. En [38] se distinguen tres formas de comunicación:

- *Siempre conectado*: ésta es la forma por defecto, PDA, Tablet PC, y especialmente dispositivos móviles, como los teléfonos móviles, se han diseñado para una comunicación continua con la red, ésta siendo la forma deseable y la que ofrece más beneficios en cuanto a rapidez y sincronización.
- *Parcialmente conectado*: hay casos en los cuales la conectividad se ve interrumpida durante periodos de tiempo. Por ejemplo, un agente de policía podrá acceder a su servidor central utilizando su PDA, para descargar información sobre una infracción o persona, después se desconectará y hará uso de los datos en función de las circunstancias, en modo desconectado. Este caso es el más interesante desde el punto de vista del desarrollo de aplicaciones para dispositivos móviles, porque implica cuestiones como, gestión de información en el dispositivo móvil local, sincronización, etc.
- *Nunca conectado*: el caso menos común es el de ciertos dispositivos, como algunas videoconsolas, que no requieren de una comunicación con el servidor para ofrecer sus servicios.

Durante la enumeración anterior se ha hablado del concepto clave de la sincronización. Éste concepto afecta al modo en el cual los datos se actualizan en los dispositivos móviles, hay varias formas de hacerlo:

- *Sincronización continua*: se realiza en los dispositivos de comunicación “siempre conectado”, donde los datos siempre estarán actualizados. Como desventajas, esto puede implicar un gasto más amplio del ancho de banda, ya limitado de por sí en los dispositivos móviles. Esta sincronización se puede realizar de dos formas:
  - *síncrona*: se envía una petición de intercambio de datos, seguida de los datos, el servidor confirma el intercambio después que el último bit se haya intercambiado.
  - *asíncrona*: se envía una petición de intercambio de datos, seguida de los datos, el servidor confirma el intercambio después de cada paquete de datos.
- *Sincronización en almacenamiento y reenvío*: cuando la comunicación no es posible en modo “siempre conectado”, resulta interesante guardar la información hasta que ésta se pueda enviar al servidor. También es

interesante usar esta técnica cuando los datos a transmitir no dependen del tiempo, son incompletos o no suponen una urgencia de envío. Esta técnica puede implicar cuestiones técnicas, como el mantenimiento de la integridad de los datos, o barajar las distintas formas de comunicación que ofrecen los dispositivos móviles, como GSM, UMTS, Wi-Fi, Bluetooth, infrarrojos etc.

Después de haber visto las posibles divisiones por capas y/o niveles que se pueden realizar en la parte del cliente o del servidor en una aplicación que emplea esta arquitectura, hay que hacer hincapié en que los dispositivos móviles de hoy en día, no disponen de un ancho de banda muy elevado.

Tal y como habíamos visto en el capítulo 4, se están haciendo esfuerzos para aumentar las velocidades que son capaces de alcanzar los dispositivos móviles. Gracias a este aumento, es posible desarrollar aplicaciones que se basan en transferencia de voz sobre IP, videoconferencia o transferencia de datos de cierto volumen en las aplicaciones que hacen uso de una arquitectura cliente-servidor.

Aunque la mejora en la velocidad en las redes móviles es innegable, hay que tener en cuenta que las redes locales de ordenadores de hoy en día pueden llegar a velocidades de 100 Mbits (*Fast Ethernet*) y 1000 Mbits (*Gigabit Ethernet*), con lo cual en una red móvil se pueden tener perfectamente velocidades 10 veces menores en cuanto a la transferencia de los datos, esto supone un tiempo de espera 10 veces mayor, lo que también se podría traducir en un decremento importante en la usabilidad que el usuario le puede sacar al programa en dichos entornos. Hasta la implementación de los estándares de cuarta generación seguirá habiendo esta diferencia entre las redes móviles y las redes locales. Resulta imprescindible pues un estudio previo de la arquitectura en capas que se utilice. Como diseñar los módulos de tal forma que la comunicación con el proveedor de servicios sea la correcta, ofrezca un buen tiempo de respuesta y evite la saturación tanto en el cliente como en el servidor. Por poner un ejemplo, una aplicación que utiliza una sincronización continua y síncrona, puede ser adecuada si los datos son vitales, si el entorno permite una conectividad continua y si los datos no tienen un tamaño considerable. Decimos esto último por la característica de sincronización de este método, ya que hasta que se reciba la respuesta del servidor puede pasar un tiempo considerable para el usuario del dispositivo móvil, si éste no puede usar la aplicación porque ésta se está comunicando con el servidor, lo más seguro que al usuario no le satisfaga nuestra elección. Resulta por lo tanto decisivo el análisis de todos estos factores que influyen en la arquitectura del sistema basado en dispositivos móviles que funcionan en entornos limitados.



## 8. Modelo de usuario

En el campo de la Inteligencia Ambiental, en lo que a la interacción persona-ordenador se refiere, uno de los problemas más importantes cuando nos enfrentamos a la tarea de diseño de un sistema es cómo diseñarlo para miles de personas y al mismo tiempo que parezca que está diseñado para cada usuario específicamente. Una de las posibles respuestas a este problema puede ser la creación del modelo de usuario.

Un modelo de usuario representa una colección de preferencias y atributos específicos de un usuario particular y es la base de los sistemas adaptativos en el campo de la interacción persona-ordenador. Estos modelos de usuario son la interpretación que el sistema tiene de los usuarios. Esto es distinto de los modelos mentales que el usuario tiene, ya que éstos representan la idea que los usuarios tienen de los sistemas y sus funcionalidades. El gran objetivo de la obtención de este modelo es que el sistema sea capaz de ofrecer la información adecuada, en el momento adecuado y en la forma adecuada [39].

En el párrafo anterior se ha utilizado el concepto de sistema adaptativo, hay que hacer especial hincapié en que este término no es lo mismo que sistema adaptable. Que un sistema sea *adaptable* indica la posibilidad de este que éste pueda ser cambiado, adaptado, pero es el usuario de forma explícita el que realiza este cambio, por ejemplo mediante la edición de sus preferencias. *Adaptativo* significa que aparte de que el sistema pueda ser cambiado para las necesidades de cada usuario, esta adaptación se realiza de forma implícita y transparente al usuario, utilizando el modelo de usuario.

Los atributos que se pueden incluir en un modelo de usuario pueden variar, desde datos de carácter personal a datos inferidos por el sistema informático durante su uso. En la práctica hay distintos tipos de modelos de usuario, cada tipo determinando qué tipo de información se recoge, cómo y cuándo. A continuación se presentará un breve resumen de estos tipos:

- *Modelo estático*: este modelo es de los más simples, ya que no cambia, una vez que se crea. La información se recopilada no cambia, esto no significa que el sistema no sea adaptativo, nos damos cuenta de que cada modelo es para cada usuario, por lo que la unicidad desde el punto de vista del usuario se consigue, aunque el modelo no cambie.
- *Modelo dinámico*: la diferencia entre el modelo dinámico y el estático, como se puede intuir, es que los atributos que lo definen son cambiables en el

tiempo. El cambio lo puede realizar el usuario o se puede realizar de forma transparente al usuario, mediante un algoritmo de aprendizaje que funciona en nuestro sistema observando las preferencias e interacciones entre el usuario y el sistema.

- *Modelo basado en estereotipos*: este modelo se distingue más de los anteriores, ya que basa su funcionamiento en la clasificación de los usuarios, en base a su comportamiento, en grupos o estereotipos. El sistema usa esta clase de modelo suponiendo que la mayoría de usuarios pertenecientes a un grupo actúan de forma similar, de modo que el sistema es capaz de adelantarse a las preferencias del usuario, una vez que dispone de los suficientes datos para ubicar al usuario en el grupo adecuado.

En realidad, un sistema suele hacer uso de modelos que combinan las ventajas de cada tipo de modelo, en función de un previo estudio del problema y de la viabilidad de uso de este modelo de usuario, que en este caso se llama *híbrido*.

De acuerdo con Addie Johnson y Niels Taatgen en [40], un buen modelo de usuario tiene que cumplir las siguientes características:

- *Tener el número mínimo de parámetros libres*: muchos sistemas que hacen uso de un modelo de usuario tienen parámetros libres, a los que el diseñador del sistema asigna valores arbitrarios. Ésta es una desventaja ya que se le permite al diseñador y no al usuario crear su modelo.
- *Predecir comportamiento*: un buen modelo debería ser capaz, aparte de monitorizar el comportamiento del usuario, predecirlo. La predicción se hace en base a los datos recolectados.

Como se ha indicado, hay modelos de usuario que se adaptan, cambian con el tiempo; para hacer esto, una parte esencial es la recolección de la información para su posterior análisis. A continuación vamos a describir los métodos más utilizados para recopilar estos datos en los sistemas cognitivos que hacen uso de un modelo de usuario hoy en día:

- *Pedir atributos explícitamente*: se refiere a pedir datos durante las primeras veces que un usuario interactúa con el sistema, una forma puede ser la de pedir datos cuando el usuario se registre en el sistema.
- *Deducir atributos en base a la observación*: se refiere a no pedirle al usuario directamente los datos, sino inferir la información en base a la interacción

que éste tiene con el sistema; se entiende por interacción el modo en el que resuelve un problema, sus intereses, etc.

De forma normal, se utiliza una combinación de las dos, es decir, un sistema híbrido de recopilación de información.

En cuanto a la aplicación de los modelos de usuario, cabe decir que su implementación se realiza de forma más extensa en el ámbito web, debido al carácter ubicuo que este medio tiene, donde existe una gran variedad de usuarios cuyas experiencias se ven mejoradas notablemente debido al uso de este método. A continuación se verán varios ejemplos:

*Sistemas de recomendación:* Son sistemas que ofrecen ítems u objetos que al usuario le pueden resultar de interés, la elegibilidad de estos objetos se realiza en función de relaciones entre objetos recién manipulados por el usuario. Ejemplos se pueden encontrar en portales de compra online, como Amazon. Otro ejemplo cotidiano es el recomendador de posibles amigos que tiene Tuenti, gracias al cual el sistema nos ofrece posibles amigos de nuestros amigos que posiblemente conozcamos.

*Sistemas hipermedia adaptativos:* Son sistemas que ofrecen contenido web en base a las características y preferencias de los usuarios, son sistemas que tienden a ofrecer información más relevante para el usuario que utiliza el sistema. Un ejemplo de este tipo de sistemas y cómo funciona puede ser [41], en el que se expresa un posible funcionamiento de los motores de búsqueda que tienen en cuenta un modelo de usuario, mejorando de esta forma la efectividad de la búsqueda, entendiendo por efectividad la relación entre los resultados de la búsqueda y las necesidades del usuario.

*Sistemas de tutorización inteligentes:* En este tipo de sistemas se intenta ayudar al alumno mediante la creación de un modelo de usuario que contenga información sobre sus habilidades y necesidades. De esta forma el sistema se adapta al usuario, ofreciéndole tipos de ejercicios oportunos. El modo de funcionamiento de este tipo de sistema de aprendizaje adaptativo se distingue del de un sistema de aprendizaje normal por el hecho de que ofrece al usuario solamente la información que sea acorde con su modelo de usuario, disminuyendo de esta forma la sobreinformación y desorientación del usuario.





# 9. Trabajos relacionados

Después de haber visto el estado del arte en las distintas áreas que pueden tener influencia en nuestra aplicación, en el transcurso de este capítulo se intentará observar cuáles son las aplicaciones más parecidas a la que nosotros proponemos y qué aspectos novedosos proponen.

El capítulo se va a dividir en dos subpartes, la primera relacionada con el trabajo de investigación que se ha realizado en el área y la segunda relacionada con las aplicaciones disponibles en el mercado.

## 9.1. Trabajos investigadores en el área

Se ha hecho un análisis de las aplicaciones presentes en el área de investigación relacionada con el deporte. Algunas de las aplicaciones más interesantes propuestas por los diversos científicos se presentan a continuación:

*Fun and Sports: Enhancing the Home Fitness Experience* [42]. Este trabajo de investigación propone un entrenador virtual para realizar ejercicio en interior con una bicicleta estática. Los investigadores se propusieron hacer que un ejercicio a veces tan aburrido se transforme en una actividad interesante. Para tal fin han propuesto el uso de un entrenador virtual que ofrezca retroalimentación de varios parámetros como ritmo cardiaco, cadencia, para hacer la actividad más motivadora y atractiva para el usuario. El estudio concluye que el entorno creado por el uso del entrenador virtual tiene un efecto positivo sobre el interés y satisfacción con el que el usuario realiza dichas actividades físicas. Parece ser que el uso de un entrenador virtual en el deporte resulta en una experiencia positiva para el usuario final. Aunque nuestra propuesta se aplicará a un deporte distinto, es de suponer que tendrá la misma influencia positiva.

*Digital Fitness Connector: Smart Wearable System* [43]. El conector de fitness digital (DFC – *Digital Fitness Connector*) se centra en la conectividad de los terminales móviles inteligentes con los distintos dispositivos auxiliares que se pueden utilizar en la actividad física (acelerómetros, monitores de ritmo cardíaco, podómetros etc.). El dispositivo electrónico creado, es capaz de interconectar multitud de sensores y permite el funcionamiento en modo sincronizado o no sincronizado. El modo sincronizado entra en funcionamiento cuando el dispositivo se conecta con un smartphone; en este caso, el DFC tiene el funcionamiento de un buffer, almacena los datos recibidos de los distintos sensores

y los envía mediante Bluetooth a la aplicación del smartphone para ser interpretados. El modo no sincronizado supone que el DFC es capaz de almacenar toda la información de forma local, y en el momento de la conexión con un smartphone u ordenador, transferir dicha información para su posterior interpretación.

*Bringing Mobile Guides and Fitness Activities Together: A solution based on an embodied virtual trainer* [44]. Este estudio aprovecha la posibilidad que los nuevos smartphones ofrecen para crear un entrenador virtual personal para su uso en el exterior. El entrenador utiliza GPS para determinar la localización del usuario en el exterior, y unos mapas precargados en base a los cuales ofrece instrucciones vía voz. También dispone de un entrenador visual en el caso que queramos algún tipo de ejercicio en el exterior, pudiendo éste ser utilizado para enseñarnos la forma correcta de realizarlo. Se ha constatado que el efecto del entrenador sobre la motivación de los usuarios a la hora de realizar ejercicios físicos es muy beneficioso. Esta propuesta ofrece aspectos innovadores, como es el uso del GPS para determinar la posición de un usuario o el uso de la capacidad de los smartphones en la síntesis del habla. También aparece el concepto de mapa, y el seguimiento de trayectos.

Los trabajos investigadores a día de hoy parecen centrarse en la creación de entrenadores virtuales y con la interconectividad de sensores para fomentar la práctica del ejercicio físico. Se advierte una falta de investigación en el desarrollo de aplicaciones para dispositivos móviles en deportes concretos, como por ejemplo, los relacionados con correr. Esto puede ser debido también a que los *smartphones* y los sensores que incorporan son conceptos bastante nuevos y pueden parecer triviales a simple vista, para llamar la atención de los investigadores.

## 9.2. Aplicaciones comerciales

Afortunadamente, en el área de las aplicaciones comerciales, y a diferencia del área de investigación, existen opciones reales más parecidas a la que proponemos nosotros. Las diferentes compañías que están desarrollando este tipo de software parecen conocer y ser capaces de llegar con facilidad al usuario final. A continuación vamos a ver alguno de los software disponible para corredores y analizar las cosas que pueden tener en común con nuestra propuesta.

*RunKeeper* [45]. Es una aplicación disponible para Android y iPhone que permite realizar un seguimiento de las rutas en distintas actividades como correr, andar, hacer senderismo, etc. Además, RunKeeper utiliza el GPS para ofrecernos distintos parametros de sesión, como distancia, tiempo, ritmo, calorías quemadas y tiene la posibilidad de trabajar con un monitor de ritmo cardíaco específico en función del sistema operativo utilizado. En el

caso de iPhone es necesaria la utilización de un hardware específico una “llave” de la empresa Wahoo, que viene junto con el monitor cardíaco por el precio de 120\$.

En la Figura 9.1 se pueden observar algunas de las funcionalidades clave de la aplicación.

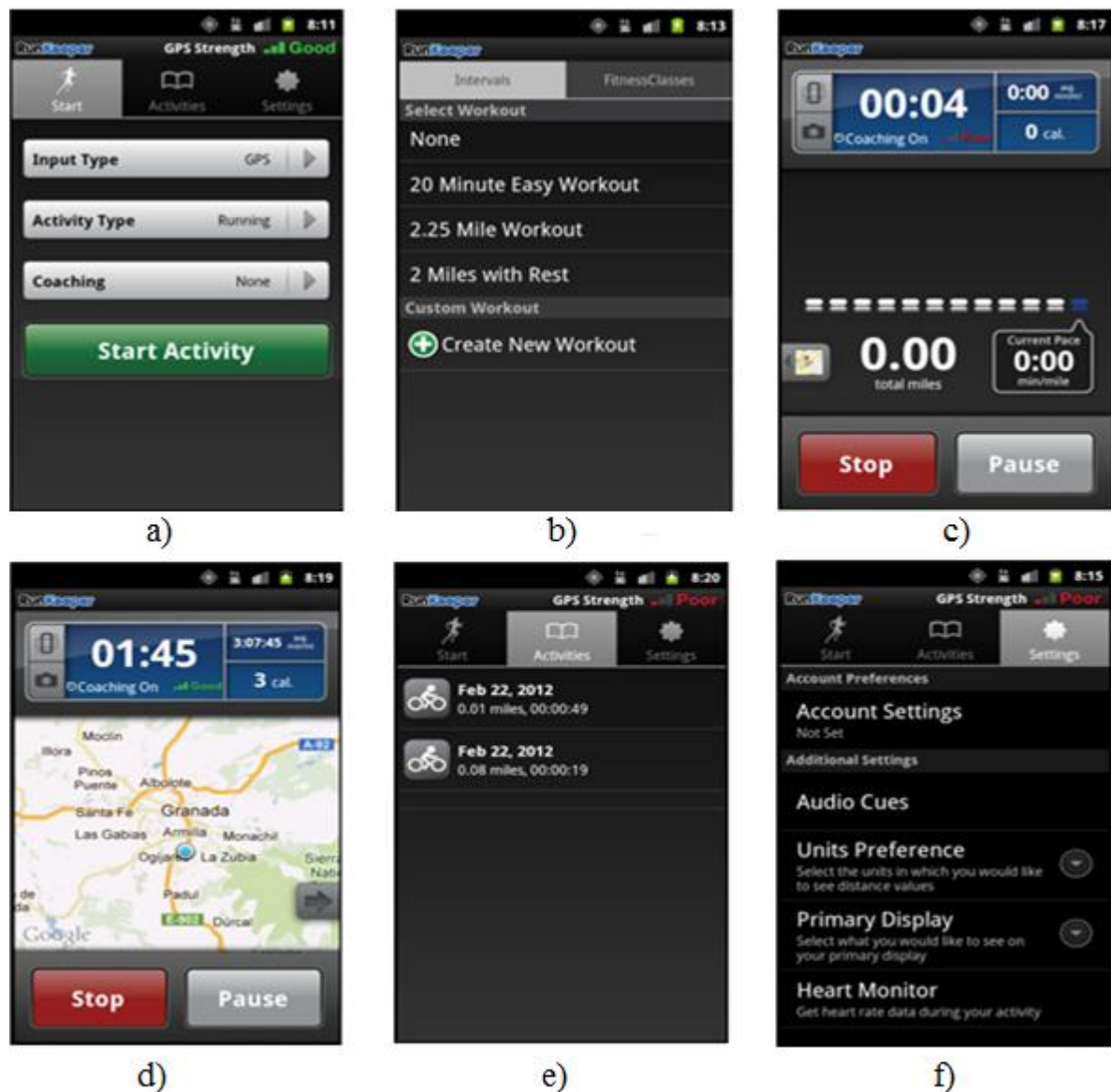


Figura 9.1. Capturas de pantalla de la aplicación RunKeeper

En realidad, la aplicación es bastante sencilla y fácil de utilizar. Una vez que se elige el tipo de actividad a realizar, correr, en nuestro caso (a), permite elegir si se prefiere realizar un plan de la sesión, permitiéndose aquí elegir la distancia, el ritmo deseado y los intervalos de descanso (b). Una vez iniciada la sesión, se muestran los parámetros de la misma (c) y un indicador hacia otra pantalla que nos permite ver en tiempo real la trayectoria que estamos realizando, utilizando Google Maps (d). Al terminar la sesión, se puede decidir si guardarla o no; en caso afirmativo, un resumen de la sesión se guardará en la pestaña actividades (e).

Las imágenes que vienen a continuación intentan ir conforme a los pasos realizados en la descripción. Por último, desde la pestaña “Settings” (f) se permite añadir hitos de sonido que se escucharan cuando se llegue a cierta distancia, por ejemplo, así como configurar el monitor de ritmo cardíaco y cambiar las unidades de preferencia.

Aspectos interesantes de RunKeeper son el uso de GPS para ofrecer datos interesantes sobre la sesión actual, así como la posibilidad de crear un plan de sesión, aunque esto no tenga en cuenta tu estado físico actual y no sea parte de una larga planificación para un fin concreto.

*MapMyRun* [46]. Es una aplicación bastante completa incluida en la plataforma desarrollada por la empresa *MapMyFitness*. A diferencia de *RunKeeper*, está diseñada especialmente para correr y no sirve para otros deportes. *MapMyFitness* ofrece otras aplicaciones parecidas para equitación, esquí, andar o senderismo. La aplicación utiliza GPS para ofrecer información en tiempo real de la distancia, velocidad, ritmo y calorías, además existe la posibilidad de sincronización de los resultados con una cuenta, anteriormente creada.

En la Figura 9.2 se puede observar la funcionalidad básica de la aplicación. Al instalar el programa nos encontramos una pantalla que nos ofrece múltiples posibilidades, tales como realizar una sesión, guardarla, ver nuestro perfil, ver rutas realizadas, nutrición, acceder al perfil o amigos y la típica opción de cambiar la configuración (a). Aparte de las características que ofrecía RunKeeper, como utilización del GPS para mostrar los parámetros de la sesión (b) y calcular la ruta (c), así como guardar los datos de nuestra sesión (d) en una cuenta online, MapMyRun ofrece consejo y asesoramiento alimenticio mediante la opción de nutrición (e) y nos da la posibilidad de socializarnos, ya que a diferencia de RunKeeper la aplicación puede estar sincronizada con redes sociales como Facebook o Twitter (f).

Lo que llama la atención es que en la opción de rutas, donde deberían ser visibles algunas de las rutas ya realizadas o alguna recomendación de rutas, no aparece nada. Decir también que el portal MapMyRun.com se ha hecho famoso principalmente por esta característica, ofrecer a los usuarios crear sus propios trayectos sobre Google Maps y mantener un registro con sus sesiones. Esto se hacía en modo post-sesión, cuando el usuario llegaba a casa después de correr. Personalmente me pareció una buena idea y no entiendo por qué no se utilizó la misma línea en la aplicación móvil. Otra cosa que permiten las dos aplicaciones, es la posibilidad de realizar fotos durante la sesión.

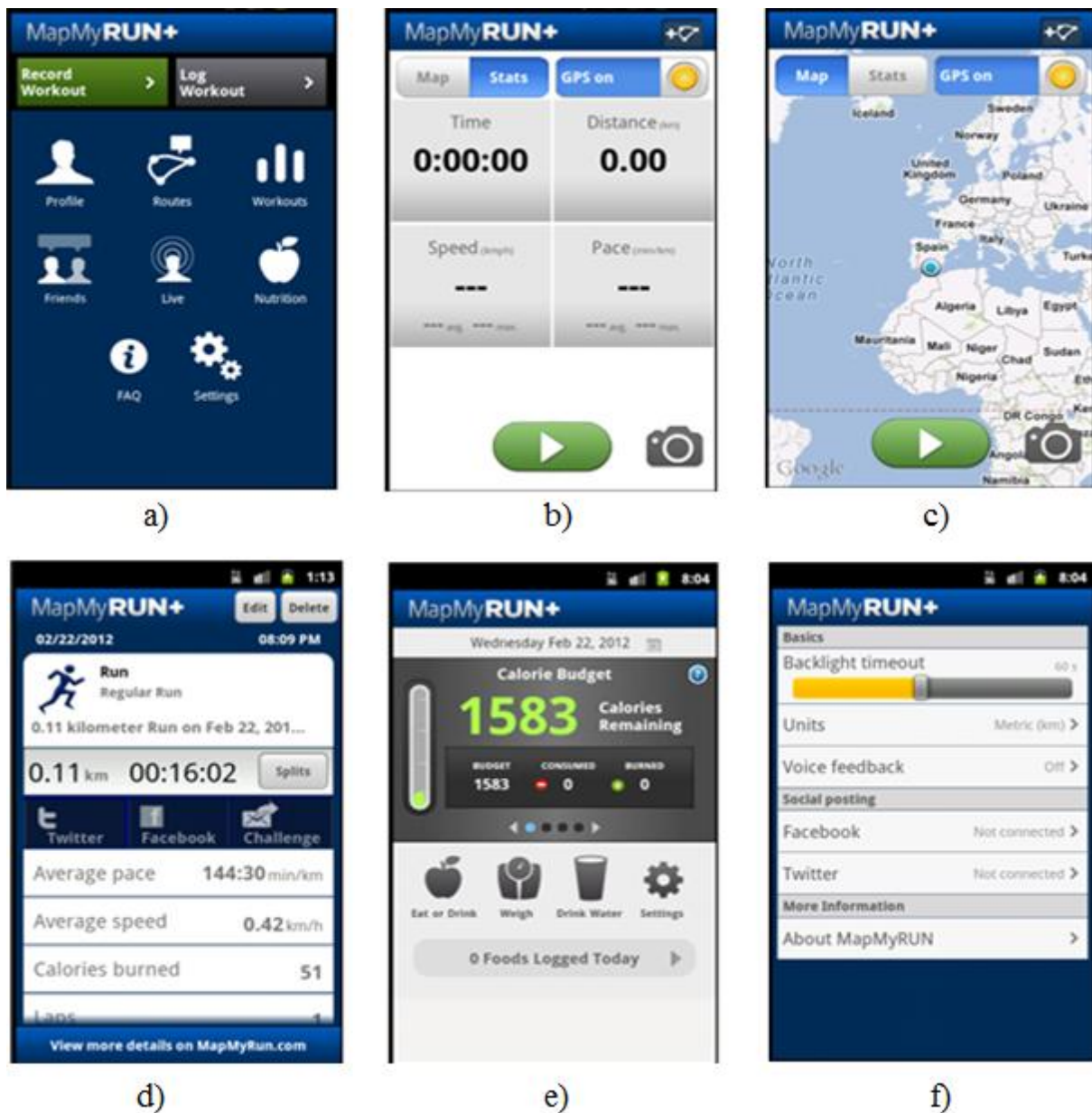


Figura 9.2. Capturas de pantalla de la aplicación MapMyRun

*Runner's World Smart Coach* [47]. Tal y como se esperaba, los profesionales de *Runner's World* [48] no han tardado mucho en sacar al mercado la idea original que tuvieron con la creación de planificaciones automáticas para corredores. Así pues, a finales de Agosto 2011 veía la luz la aplicación para dispositivos móviles de la conocida revista. Esta aplicación, según el mercado Android donde se comercializa, aunque gratis, no tiene mucho éxito, habiendo obtenido una puntuación de solo 2.8/5, que se puede considerar como muy baja para una aplicación desarrollada por profesionales. El fallo puede estar en el nombre que se le ha dado a la aplicación, entrenador inteligente, ya que me imagino que muchas personas cuando se interesan por una aplicación parecida, les interesan algún tipo de ayuda, como la retroalimentación de parámetros de sesión o facilidades similares a las que pueden recibir de las aplicaciones anteriormente mencionadas. Al no ser así, los usuarios huyen de esta

aplicación considerándola de poco uso. En la Figura 9.3 se puede observar la funcionalidad más importante de la aplicación. En la primera captura de pantalla, de izquierda a derecha, se pueden elegir los parámetros de una carrera reciente en base a los cuales se generará una planificación, la cual se puede observar en las dos capturas siguientes.

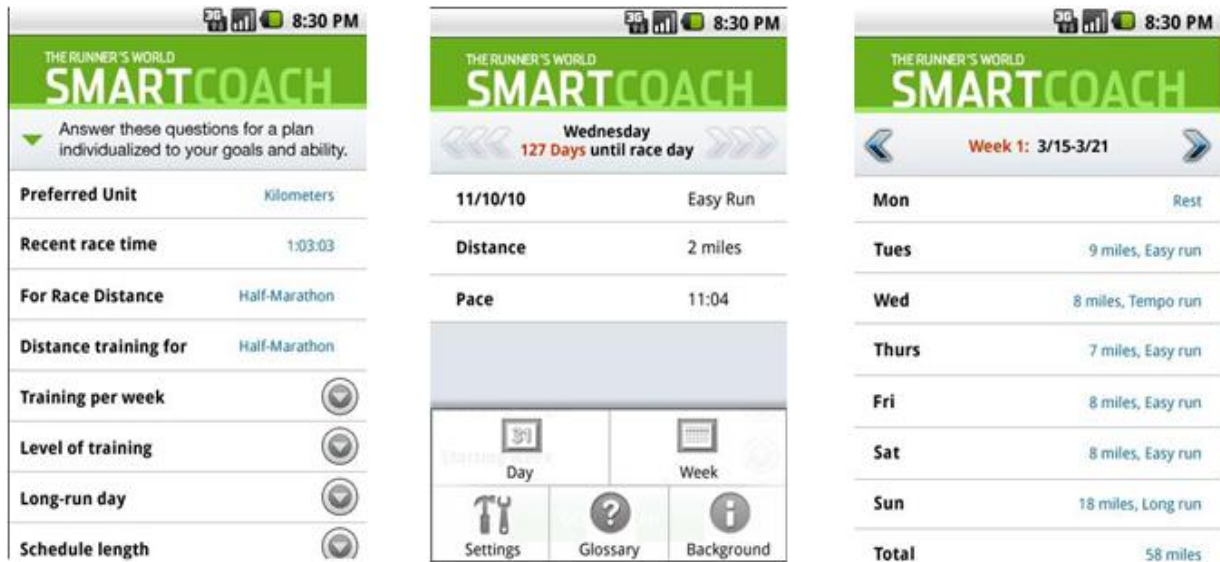


Figura 9.3. Capturas de pantalla de la aplicación Smart Coach

### 9.3. Conclusiones

Habiendo analizado las aplicaciones e investigaciones existentes más relacionadas con nuestra propuesta, se puede decir que, a grandes rasgos, los trabajos de investigación se centran en ayudar, o mejor dicho, motivar a que la gente practique deporte. Sin embargo, muy pocos trabajos existen sobre cómo la tecnología puede ayudar a los que ya están motivados o sobre cómo puede la tecnología mejorar el estilo de entrenamiento de los deportistas. En el caso del atletismo, personalmente no encontré ningún artículo que haga referencia a la introducción de mejoras en el modo de entrenamiento de los atletas y que esté relacionado con la informática. Es por eso por lo que en este trabajo se propone realizar un análisis de los principales factores que intervienen en esta rama del deporte, aplicando además la tecnología más reciente para el desarrollo de nuestra propuesta.

En cuanto a las aplicaciones para dispositivos móviles que se pueden encontrar en el mercado, se han citado algunas de las que nos parecieron más interesantes; aparte de éstas, existen muchas más, cada una con sus facilidades y restricciones. En cuanto a las restricciones, existen aplicaciones que son de pago, o no se pueden utilizar en ciertos países, siendo éste el ejemplo de *Runner's World Smart Coach* [47]. En cuanto a las funcionalidades, aparte de las ya vistas, existen aplicaciones que permiten escuchar música mientras realizamos el entrenamiento, o como se ha visto anteriormente, aplicaciones que nos permiten realizar fotografías del trayecto realizado. Cada vez que un nuevo sensor se incorpora en un dispositivo móvil, son muchas las aplicaciones que intentan incorporar su funcionalidad sin un previo estudio de si realmente su inclusión conlleva un beneficio real para el usuario. En los capítulos venideros se verán cuáles de las facilidades que ofrecen tienen sentido en el sistema que nosotros proponemos y cuáles no. Una cosa es cierta, en el mercado hay múltiples opciones para elegir entrenadores virtuales para el mundo del deporte. Con tal cantidad de aplicaciones, resulta extraño que no exista ninguna aplicación especialmente dedicada al mundo de los atletas.

Debido al vacío existente en este campo, nuestro trabajo intenta proponer un sistema innovador, profesional y de real utilidad que tenga en cuenta las necesidades de los corredores de larga distancia. En los capítulos que vienen a continuación se va a realizar un análisis exhaustivo de los aspectos que hasta ahora se han visto y se intentarán tomar las decisiones más adecuadas para el desarrollo de nuestra aplicación.





# 10. Mi propuesta

Hasta el momento se ha visto el estado del arte en los distintos subsistemas que podrían influir en la estructura y desarrollo de nuestro sistema. A continuación se pasará a analizar los distintos aspectos a los que se refieren cada uno de los capítulos anteriores y a tomar decisiones en función de los beneficios y los riesgos que suponen, teniendo en cuenta el contexto de nuestra aplicación.

## 10.1. Descripción de la propuesta

Tal y como se ha visto en el primer capítulo, el sistema de entrenamiento estará básicamente formado por un módulo de planificación y otro de recomendación de trayectos. Aparte de éstos, habrá módulos auxiliares encargados del seguimiento de trayectos, y de ofrecer al usuario parámetros de sesión en tiempo real así como un módulo de control de ritmo, cuyo propósito básico es evitar lesiones.

### 10.1.1. Módulo de planificaciones

El módulo de creación de planificaciones puede ser considerado la parte central del sistema, ya que en función de este módulo se irán recomendando los trayectos y se realizará el seguimiento del rendimiento del usuario en función de los parámetros de sesión.

En el primer capítulo se han visto algunos de los principios que se tenían que respetar en la creación de las planificaciones, de los cuales los más importantes eran:

- Sesiones progresivas, con un incremento de como mucho el 10%
- El 70-80% del entrenamiento se debe realizar en sesiones largas, el 20-30% restante en sesiones rápidas.
- Dejar suficientemente tiempo de recuperación entre sesiones.
- Incluir sesiones de entrenamiento combinado en las sesiones de descanso.
- Bajar la intensidad de entrenamiento en las sesiones próximas a carreras en las que se compite y también después de un cierto número de incrementos, normalmente 4.
- Mezclar sesiones aeróbicas con sesiones anaeróbicas, muy relacionado con el segundo punto.

El primer punto es de vital importancia, las sesiones tienen que ser progresivas e incrementar como mucho en un 10%. Hay que resolver el problema del umbral mínimo, es decir, desde donde comenzamos el incremento, así como cuánto es lo que corre cada usuario, que obviamente serán parámetros que el usuario tendrá que introducir. Nosotros vamos a optar por crear ciertos rangos, ya que esto es más fácil de apreciar por el usuario que saber exactamente la distancia que corre durante una semana; también es ésta la solución que se propone en SmartCoach [47]. Así pues, en la Tabla 10.1 se definen los siguientes rangos semanales y la frecuencia en días/semana de las veces que el atleta entrena.

<b>Sistema métrico</b>	<b>Sistema imperial</b>	<b>Frecuencia</b>
10-18 km	6-11 millas	2 días/semana
18-26 km	11-16 millas	2 días/semana
26-34 km	16-21 millas	3 días/semana
34-42 km	21-26 millas	3 días/semana
42-50 km	26-31 millas	3 días/semana
50-58 km	31-36 millas	4 días/semana
58-66 km	36-41 millas	4 días/semana
66-74 km	41-46 millas	5 días/semana
74-82 km	46-51 millas	5 días/semana
82-90 km	51-56 millas	6 días/semana
90-98 km	56-61 millas	6 días/semana

Tabla 10.1. Rango semanal y frecuencia

Para el cálculo de la frecuencia se ha tenido en cuenta el segundo punto, un usuario tiene que entrenar el 70-80% del tiempo en sesiones largas. Autores de trabajos científicos en este campo, como el Dr. Jack Daniels, y otros entrenadores coinciden en que este entrenamiento de sesiones largas se tiene que hacer realmente en una sesión. Si el lector piensa un poco, esto tiene sentido, ya que se trata de entrenar para un maratón, por lo que es necesario acostumbrar al atleta a sesiones largas. Así pues cojamos el ejemplo de una persona que realiza 15 kilómetros semanales, primer rango en la Tabla 10.1, serían  $70\% \cdot 15 = 10$  km la sesión larga, por lo que habría que repartir el resto (5 km) entre el número de días de la semana; como resulta completamente viable que se realice una sola sesión de velocidad de 5 km, la frecuencia sería de 2 días/semana.

Aplicando la misma regla anterior otra persona que corre 30 kilómetros semanales, la sesión larga sería de  $70\% \cdot 30 = 21$  km, y los restantes 9 km para repartir en sesiones rápidas. Una sesión de 21 km es una sesión más que larga para un atleta que entrena para un maratón. No es realista comenzar la planificación de un maratón con una sesión de media maratón, ya que durante una planificación de 16 semanas a un incremento de 10% se realizarán varios maratones entrenando para un maratón, lo cual no tiene sentido. Por otra parte, los 9 km son muchos para una sesión rápida, y pocos para dos sesiones, así pues lo más sensato sería realizar dos sesiones largas y una corta. Es por ello que resulta imprescindible incorporar un día más a nuestro entrenamiento. También hay que tener en cuenta que estamos entrenando para una carrera de larga distancia, por lo que necesitamos una sesión larga, por eso las dos sesiones largas no pueden ser iguales, ya que entonces ambas serían demasiado cortas. Una solución como la siguiente parece más sensata: 9 km para una sesión a la que denominaremos ligera en lugar de larga, ya que se realiza al mismo ritmo que la larga pero sin tener tanta distancia, 8 km de sesión rápida y 13 km de sesión larga. Si nos damos cuenta, los porcentajes han variado ligeramente en la solución propuesta:  $9+13 = 22$  km en sesiones largas, lo que supone un 73.3% del total, a diferencia del ejemplo anterior, que era de un 70%. Esto pasa en el caso de todos los rangos, y es algo normal; por ello el primer principio indica un rango de porcentajes (70-80%) y no un valor absoluto (70%). Debido a esta variabilidad, se producen muchas excepciones o mejor dicho variaciones en el caso de los distintos rangos, sobre todo en los rangos finales, donde tanto la distancia como los días de entrenamiento aumentan. Es por ello por lo que puede resultar interesante la creación de planificaciones de forma estática, para cada uno de los rangos, 11 planificaciones, a las cuales se les asignará de forma dinámica los ritmos de sesión correspondientes, de acuerdo con las tablas VDOT vistas en el primer capítulo.

En cuanto a los ritmos, se ha visto que dependen del nivel físico en el que se encuentra el atleta; por eso, este módulo necesita que se le introduzcan como parámetros adicionales tanto el tiempo de la carrera para la que y el de la última carrera realizada por el usuario. Utilizando estos datos se puede encontrar el VDOT correspondiente.

### **10.1.2. Módulo de recomendación de trayectos**

Aparte de la distancia y el ritmo recomendado para cada sesión, se pueden recomendar también los trayectos a seguir. Por ello nos vamos a basar en alguno de los métodos vistos en el capítulo de localización, aunque será en el capítulo venidero donde se decidirá cuál de estos métodos se utilizará.

Se pueden guardar en un servidor (para su posterior uso) todos los trayectos que los usuarios consideren interesantes o de utilidad para los demás usuarios. Por tanto, una vez que un usuario realice un trayecto, siempre podrá decidir si guardarlo o no en el servidor, para compartirlo con el resto de usuarios. La recomendación se hará en función de la proximidad del trayecto a la posición actual del usuario en cuestión, aparte de este aspecto

habrá otros como la distancia el trayecto o el desnivel acumulado. Se recomendarán trayectos con una distancia superior a la que se tiene que realizar, aunque este requisito no es completamente necesario, ya que el usuario siempre puede volver sobre el mismo trayecto. No obstante, pueden existir sendas, caminos o aceras de sentido único lo suficientemente frecuentadas por otros corredores como para que correr en sentido contrario no sea aconsejable. Claro está que si un usuario considera factible ir y volver sobre un mismo trayecto, éste se puede guardar sin impedimento alguno en su doble sentido de circulación.

En cuanto al desnivel, hay muchas aproximaciones que se pueden tomar. La aplicación RunStar [49] es también bastante conocida en el mundo de los amantes del correr, no se ha incluido en el apartado de trabajos relacionados debido al hecho de que su funcionalidad es muy limitada en la versión gratuita. Para los dispuestos a pagar, la aplicación ofrece cosas como realimentación audio, reproducción de música o mapa de altitud. En cuanto al mapa de altitud, un usuario la puede apreciar de modo positivo si el desnivel acumulado es negativo, es decir, si bajamos, y negativamente si el desnivel acumulado es positivo, es decir, subimos. Para un usuario no experimentado, ésta es la primera impresión, la facilidad del trayecto. Pero existen más cosas relacionadas con la bajada o subida, aparte de la facilidad del trayecto.

Gottschall et al. [50], en un estudio publicado en 2005, hacen un análisis de lo que supone correr cuesta arriba, correr sobre superficies llanas o correr cuesta abajo. El resultado puede ser sorprendente para los que no están necesariamente involucrados en el tema del atletismo. Las fuerzas de impacto cuando se corre sobre una superficie descendente inclinada a 9° incrementan un 54% comparado con las correspondientes a una superficie llana. La conclusión a la que llega el estudio es que: “correr sobre superficies descendentes aumenta substancialmente la probabilidad de lesión por sobreuso”. Está más que claro, correr cuesta abajo no es bueno, aunque pueda parecerlo y aunque realmente sea más fácil que hacerlo en llano o cuesta arriba.

Volviendo a nuestra aplicación, necesitamos una forma de advertir al usuario del peligro que correr cuesta abajo supone para su salud. En vez de mostrar un mapa de altitud, tal y como RunStar hace, lo que haremos será advertir al usuario cuando se encuentre en esta posición problemática, ya que al contrario el atleta podría mal interpretar el mapa de altitud. El atleta debe saber controlar, acortando su zancada e implícitamente disminuyendo su velocidad durante los descensos, para evitar lesiones.

Además, el cálculo del desnivel se puede utilizar para recomendar trayectos con un desnivel bajo para las sesiones rápidas. Si un usuario tiene que realizar su sesión rápida en un trayecto descendente, esto tendrá un efecto maligno sobre su salud, tal y como se ha visto anteriormente, al igual que realizarla sobre un trayecto ascendente, ya que el ritmo al cual tendrá que realizarla no se corresponderá con la realidad, y su ritmo cardíaco subirá en

exceso, lo cual también es maligno para su salud. Así pues es bastante recomendable realizar las sesiones rápidas en trayectos sobre superficies lo más llanas posibles.

### 10.1.3. Otras consideraciones

Para realizar el seguimiento de un trayecto, también resulta necesaria la utilización de una tecnología de localización precisa. Este seguimiento es necesario tanto para los que deciden utilizar la planificación como para los que deciden no utilizarla. Esta parte de seguimiento de trayectos es también la parte estrechamente ligada con los parámetros de carrera, como el ritmo o el tiempo. En cuanto al ritmo, es necesario realizar advertencias cuando este se encuentre por debajo o por encima del valor deseado. Las advertencias se realizarán mediante voz, se considera un umbral de 15 segundos<sup>1</sup> de margen en los dos casos, más lento o más rápido, antes de realizar una advertencia. Es aquí también donde se realiza el control del desnivel ya que está estrechamente ligado al ritmo, así pues en caso de descenso con un ángulo superior a 3 grados<sup>2</sup>, no se advertirá una bajada del ritmo, dado que se considerará como precaución por parte del corredor, aunque sí se advertirá una subida de ritmo, ya que ésta puede provocarle lesiones. También se podrá considerar una bajada inferior, por ejemplo, de 2° o inferior, pero habrá que ser precavido, ya que en la vida real casi cualquier superficie está ligeramente inclinada. Hay que encontrar un valor adecuado para evitar falsos positivos y cansar al usuario con advertencias.

Aparte de los corredores que eligen seguir la planificación, es necesario dejar al usuario utilizar la aplicación en modo de sesión libre también. En este caso, ya que no se dispone de ritmo aconsejado, se utilizará una media del ritmo acumulado para realizar advertencias en los descensos.

En el caso de la posibilidad de realizar fotos utilizando el dispositivo móvil, se ha decidido no incorporar a la aplicación características que no tengan que ver con la actividad de entrenamiento. Muchas de las aplicaciones también disponen de la posibilidad de escuchar música mientras se realiza la sesión de correr, esto puede ser bueno en cuanto a que la música es uno de los mejores estímulos conocidos, pero aquí también existe una desventaja. Cuando se corre largas distancias, la forma, es decir la técnica, es una parte importante, y escuchar música mientras se realiza el ejercicio físico puede hacer que el atleta se olvide de la técnica, lo que podría provocarle lesiones. Además durante las carreras pueden ocurrir accidentes, por lo que prestar atención es un requisito imprescindible. En el año 2007, en EEUU se dictó la norma 144.3 por parte de USA Track & Field [51], institución encargada de la organización de las carreras, la cual prohibía la utilización de dispositivos reproductores de música, luego se enmendó dicha norma dejando la potestad en mano de los directores de carrera. En Europa y demás continentes se aplica el mismo principio y la

---

<sup>1</sup> Es un valor propuesto y no se basa en estudio alguno.

<sup>2</sup> 3° supone un 18% de incremento del impacto, comparado con el 54% de incremento del impacto correspondiente a 9°, según el mismo estudio.

normativa 144.2b de IAAF [52], por la cual está prohibido la utilización de dispositivos electrónicos durante la carrera, para evitar que personas que están físicamente extenuadas sigan compitiendo basándose en el estímulo musical, y de esta forma, pongan en peligro su salud.

Por último, una característica importante que llama la atención es la realimentación vía sonido que ofrecen aplicaciones como RunStar y RunKeeper durante la sesión. Esta característica la consideramos importante, ya que permite al usuario centrarse en la forma de correr y no en el propio dispositivo móvil, siendo una característica que contribuye al principio de transparencia visto en el segundo capítulo. Lo que aquí se propone es utilizar 2 tipos de pitidos para guiar al usuario:

- Pitidos largos, para avisar al usuario de que su ritmo es lento y debe acelerarlo.
- Pitidos cortos, para avisar al usuario de que su ritmo es rápido y debe bajarlo.

Estas advertencias sirven tanto para avisar al usuario cuando se desvía del ritmo al cual tendría que realizar la sesión, como para evitar lesiones en el caso de descensos, tal y como se ha visto anteriormente. Opcionalmente, se puede utilizar un pitido corto al final de la sesión, cuando el usuario haya completado la distancia que debía recorrer.

Nos hemos decantado por la utilización de los pitidos en lugar de utilizar voz porque al realizar la sesión, el usuario tendría que tener el dispositivo móvil a una distancia aceptable del oído para escuchar las instrucciones, debiendo prestar más atención para discernir la voz, que para discernir la frecuencia del pitido e interpretarlo. Tal y como se ha visto antes, no es recomendable que el usuario desvíe la atención de la técnica de correr, en este caso para prestar atención a lo sintetizado por el dispositivo. Además, no siempre es viable o cómodo para el usuario tener el dispositivo sobre el brazo y así facilitar la escucha de las instrucciones, siendo mucho mejor dejar al usuario la libertad de llevarlo donde más cómodo le resulte.

## **10.2. Métodos de localización**

En el capítulo 6, se han visto las diferencias entre los distintos sistemas de localización, así como los beneficios y desventajas de cada uno. A continuación se intentará averiguar qué sistema sería el idóneo para nuestro proyecto. Hay que tener en cuenta que un atleta que tiene que realizar una sesión de 10-15 km no es muy probable que la realice dentro del núcleo de la ciudad, esto es debido al alto tráfico y a la cantidad de obstáculos (personas, semáforos, animales, etc.) que hay. Lo más habitual es que un atleta empiece el calentamiento, fase previa de la sesión, cerca del lugar donde vive, y se desplace hacia lugares más periféricos, más abiertos, que no interfieran con su sesión.

El método de localización ofrecido por la red móvil, aunque es el que menos recursos consume, es también el que menos precisión ofrece, véase Tabla 6.1. En su mejor caso, tiene una precisión de 50 m, mientras que en el peor caso, se puede llegar a varios kilómetros. Esto no es aceptable si se quiere trazar la ruta que sigue un corredor. Además, debido a la imprecisión en el cálculo de la posición, resulta prácticamente imposible calcular la velocidad a la que corre un atleta (que lleva un dispositivo móvil con nuestra aplicación) a partir de la fórmula:

$$v = \frac{\Delta d}{\Delta t}$$

Habría que encontrar otra forma de calcular la velocidad, una posible solución podría ser la de utilizar acelerómetros para el cálculo de la velocidad. Propuestas en este sentido se pueden encontrar en [53] y [54], aunque las soluciones planteadas suponen la utilización de acelerómetros en lugares precisos y fijos, no intercambiables, no siendo éste el caso de nuestro terminal móvil. En general el atleta cuando va a correr introduce el móvil en el bolsillo, diferentes bolsillos cada vez por qué no, sin importarle que la aceleración sufra variaciones importantes debido al movimiento dentro del bolsillo o la posición en la cual se encuentra. Debido a estas razones, se puede descartar este método de localización en el caso de nuestra aplicación.

El segundo método de localización, por posicionamiento Wi-Fi, presenta una precisión más adecuada para nuestro uso, pero sigue sin ser la solución definitiva, especialmente en las zonas periféricas de las ciudades, donde las redes Wi-Fi pueden escasear. En estos casos de zonas limítrofes, la precisión tiende a rondar el límite máximo y se da el problema de imposibilidad de cálculo de la velocidad, debido a la falta de precisión en la localización. Las soluciones son parecidas a las expuestas anteriormente, utilización de hardware auxiliar, con el coste económico y de computación que ello supone.

Por último, el método que parece más adecuado para nuestro sistema, el basado en GPS, esto es debido a que dispone de una gran precisión, lo cual nos permite realizar el seguimiento correcto de la trayectoria del atleta y también utilizar las distintas posiciones en el cálculo de la velocidad con la que éste se desplaza.

En campo abierto, la precisión del GPS puede llegar a 1 metro, si no hay interferencias ni obstáculos. Su uso parece perfecto en estas circunstancias, aun así hay que tener en cuenta una cosa de vital importancia, la batería del dispositivo móvil tiene una duración limitada cuando se utiliza el GPS, ¿será suficiente la batería de un dispositivo móvil para implementar una plataforma de corredores de larga distancia? Abdesslem et al. han realizado un estudio [55] para averiguar cuáles son los sensores que consumen más batería en un teléfono Nokia N95 con una batería de 1200mA; las conclusiones se presentan en la Figura 10.1.

<i>Sensor</i>	<i>Approximate battery life (hrs)</i>	<i>Average power consumption (mW)</i>
Video camera	3.5	1258
IEEE 802.11	6.7	661
GPS (outdoors)	7.1	623
GPS (indoors)	11.6	383
Microphone	13.6	329
Bluetooth	21.0	211
Accelerometer	45.9	96
All sensors turned off	170.6	26

Figura 10.1. Relación de consumo por sensor

Se puede observar que el GPS es uno de los sensores que más batería consume, manteniéndose encendido el teléfono durante aproximadamente 7 horas con el uso del GPS. Este dato, que puede parecer bastante malo a simple vista, se puede considerar como moderado, ya que en el caso mencionado arriba el GPS se ha ejecutado forzosamente utilizando un script, es decir, que ha tenido que determinar nuevas posiciones lo más rápido posible, siendo esto lo que más batería gasta, tal y como se puede observar en la Figura 10.2.

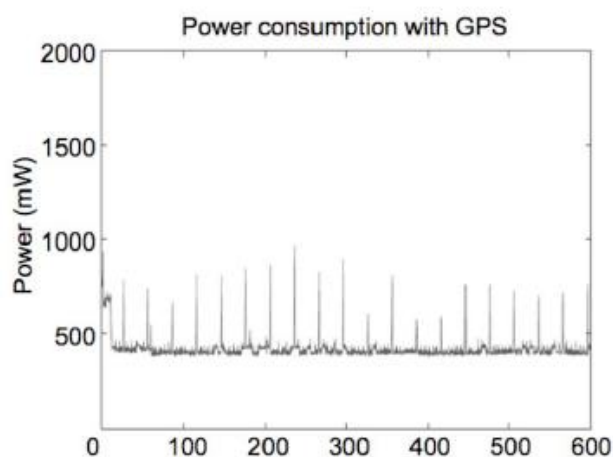


Figura 10.2. Consumo del sensor GPS<sup>1</sup>

Los picos, que pueden llegar hasta 1000mW, son los momentos en los que el GPS realiza una fijación. Disminuyendo el número de fijaciones, mejora el tiempo de vida de la batería. ¿Es esta una real solución? Podría serlo, ya que una persona que tiene que realizar una sesión de varios kilómetros suele elegir avenidas, campos o parques largos, no suele realizar carreras en zigzag, sino que suele correr en línea recta durante todo el tiempo que pueda, en éste caso, la pérdida de precisión en un trayecto no disminuye de forma importante. Otro beneficio sería que el usuario durante su sesión no suele utilizar otras características del terminal móvil, por lo que el tiempo de vida de la batería no disminuye gracias a ello. También hay inconvenientes si se disminuye mucho la frecuencia de las fijaciones, ya que no será posible ofrecerle al usuario un ritmo suficientemente actualizado.

<sup>1</sup> Fuente: [56]



Como un posible remedio a este problema, se puede realizar una planificación dinámica de las fijaciones, la cual consistiría en realizar fijaciones con una frecuencia más elevada al principio de la sesión, y una vez que el usuario se haya dado cuenta del ritmo adecuado para la sesión actual, ir bajando la frecuencia de fijación para evitar un gasto excesivo de la batería. Esta aproximación es también la que más se acerca a la realidad, indiferentemente del sistema utilizado, dado que un usuario suele fijarse al principio en el ritmo que tiene que mantener, y una vez determinado, no suele mantener el enfoque en el dispositivo medidor, salvo de vez en cuando para confirmar que sigue el ritmo planificado.

Por último, otras opciones para el ahorro de energía pasa por implementar alguna solución derivada de [56], como por ejemplo, la predicción lineal simple, que consiste en averiguar cuándo el usuario utiliza un camino recto y disminuir la frecuencia de las fijaciones en este caso. Otra opción interesante para algunas aplicaciones es la propuesta en el sistema denominado "*Senseless*" [55], que hace uso de los sensores más económicos energéticamente para predecir el movimiento, es decir, cuándo una persona corre, cuándo se ha parado o cuándo se ha sentado. Es una opción muy interesante para muchas aplicaciones, ya que obviamente cuando una persona está sentada, los demás sensores de movimiento pueden entrar en suspensión, ahorrando energía. Para nuestra aplicación, sin embargo, no resulta muy interesante, ya que un usuario cuando comienza su sesión suele realizarla completa; en el caso del descanso, este suele ser muy reducido como para que una solución de este tipo resulte rentable.

### **10.3. Modelo de usuario**

Para que el sistema sea adaptativo y adaptable, tal y como se ha visto en el capítulo 8, es recomendable la creación de un modelo de usuario. Estos modelos de usuario pueden ser: estáticos, dinámicos o basados en estereotipos. En cuanto al modelo de usuario basado en estereotipos, que básicamente se traduce en una agrupación de los usuarios según ciertas características, éste no es totalmente viable en el caso de los dispositivos móviles. Decimos que no es totalmente viable porque, para realizar una agrupación de los usuarios, tendríamos que tener en un mismo sitio sus perfiles, sus modelos de usuario. Nuestro sistema será un sistema distribuido, por el requisito de que todos los usuarios puedan acceder a los mapas que los demás usuarios graban. Una posibilidad es tener los modelos de usuarios en el servidor también, pero con los requisitos actuales del sistema, los beneficios que esto ofrece no son importantes. En la mayoría de los casos se opta por utilizar un modelo de usuario híbrido que incorpore ventajas de los distintos posibles modelos.

Nuestro sistema hará uso de un modelo de usuario dinámico, de modo que el usuario pueda cambiar sus preferencias en cualquier momento. Para averiguar los atributos

que se incorporarán en nuestros modelos hay que analizar nuestra propuesta. La aplicación está dirigida a usuarios que quieren entrenar para realizar una carrera de larga distancia, en nuestro caso, un maratón. Lo correcto es ofrecer a estos usuarios una interfaz personalizada, y habría que empezar por el idioma. La plataforma que se ha elegido para el desarrollo de la aplicación es Android, plataforma que ofrece la posibilidad automática de cambio de idioma mediante el uso de ciertos parámetros en función del idioma seleccionado. Para ser más concretos, en una aplicación se declaran ciertos valores, en distintos idiomas, para los textos que aparecerán en la aplicación y Android seleccionará los valores correspondientes al idioma del sistema. Es una ayuda importante, pero no suficiente, ya que un usuario no puede cambiar el idioma de su aplicación sin cambiar el idioma de todo el sistema Android. Resulta pues interesante realizar este cambio de forma manual dentro de la aplicación; por ello es necesario incluir el idioma preferente en el modelo de usuario.

Aparte del idioma, el usuario siempre tendrá presentes los valores de ritmo, distancia y tiempo. En función de la localización, existen dos sistemas métricos básicos:

- Sistema *internacional* o sistema métrico decimal.
- Sistema *imperial*, básicamente utilizado en EEUU.

En el sistema internacional, la unidad para la distancia es el metro y sus múltiplos, siendo el kilómetro, que consta de 1000 metros, el que nos interesa a nosotros. La unidad para el tiempo es el segundo y sus múltiplos, el minuto que consta de 60 segundos y la hora que consta de 60 minutos. En nuestra aplicación las unidades correspondientes al sistema métrico se expresarán de la forma antes mencionada.

En el sistema imperial, la unidad para la distancia es el pie, que consta de 0.3048 metros, y sus múltiplos, siendo en nuestro caso la milla, que consta de 1609.344 metros, la más importante. La unidad de medida para el tiempo es la misma que en el sistema internacional. Este sistema se utiliza básicamente en los Estados Unidos, pero en el Reino Unido se sigue utilizando combinado con el sistema internacional. Australia y Sudáfrica se adaptaron al sistema métrico bastante tarde, en la década de los 80, por eso la gente puede sentirse más cerca del estándar imperial que del métrico. Resulta pues imprescindible la incorporación del atributo del sistema métrico en el modelo de usuario.

También es importante especificar que el sistema de medición utilizado, en nuestra aplicación, no está ligado al idioma, precisamente para dejar flexibilidad a los usuarios de países como Reino Unido, Australia, Sudáfrica, que o utilizan los dos sistemas o se sienten más familiarizados con alguno de ellos.

Aparte de los atributos anteriores, cuando un usuario crea una planificación puede decidir si seguirla o utilizar la aplicación de forma libre. Si un usuario decide seguir una planificación sería interesante que nos anticipemos a sus deseos de forma transparente.

Cuando un usuario lanza la aplicación un día de entrenamiento, se tendrá que mostrar el plan de entrenamiento para el día en cuestión. Si un usuario lanza la aplicación un día de descanso o entrenamiento combinado, será la pantalla de sesión libre la que se mostrará al principio y no la planificación. Al igual que un usuario seguidor de la planificación en un día de descanso, a un usuario que no desea seguir una planificación se le mostrará la pantalla de sesión libre. Resulta pues interesante incorporar el atributo del tipo de usuario (que sigue la planificación o no) a nuestro modelo de usuario.

Respeto a cómo se creará el modelo de usuario, se dispone de dos opciones básicas, vistas en el capítulo 8. Pedir al usuario que introduzca los datos de forma explícita o deducir los datos en base a la observación. Para los primeros dos atributos, el idioma y el sistema de medición, se utilizará la segunda opción, ya que siempre es recomendable no interferir, a ser posible, con el usuario y realizar las cosas de forma transparente. Así pues, la primera vez que se lanza la aplicación se consultará el idioma del sistema, y en función de éste se inicializará nuestro modelo de usuario con los atributos correspondientes. Para el sistema métrico se hará algo parecido, en función del idioma se asigna un sistema métrico u otro, recordemos que el sistema imperial se utiliza en EEUU y otros países de habla inglesa, como Australia, Reino Unido o Sudáfrica. Una vez en la aplicación, el usuario podrá cambiar los parámetros a su gusto y de forma independiente. En el caso del seguimiento de la planificación, se pregunta de forma explícita al usuario sobre su deseo y se actualiza el modelo de usuario de forma acorde.

#### **10.4. Definición de la arquitectura**

En cuanto a la arquitectura de nuestro sistema, desde el punto de vista de la orientación a servicios, el carácter distribuido y colaborativo de nuestra aplicación hace que este tipo de arquitectura sea imprescindible. Los distintos dispositivos móviles, mediante nuestra aplicación, harán uso de un servicio implementado utilizando un servicio web. La utilización de la arquitectura orientada a servicios ofrece muy buenas perspectivas en caso de la extensión de la aplicación a distintas plataformas móviles, como podría ser el caso de iPhone. Nuestra decisión también ofrece un mayor grado de abstracción en el diseño de la aplicación, además de una flexibilidad y posibilidad de escalabilidad relativamente fácil.

El proveedor de servicios será capaz de gestionar las distintas operaciones relacionadas con la grabación y oferta de trayectos. Algunas de las funcionalidades principales encargadas a este servicio serán:

- Obtención de trayectos en función del contexto geográfico.
- Grabación de los trayectos para su posterior uso por parte de otros usuarios.

Aparte de estas funcionalidades orientadas al concepto de trayecto, también consideramos importante la recopilación de información sobre el uso que se le da al sistema. Algunos de estos datos estarán relacionados con la utilización de los distintos módulos, para averiguar si se hace un uso completo de la mayoría de sus funcionalidades.

En cuanto a las opciones de implementación existentes, se ha hecho hincapié en las dos más utilizadas, SOAP y REST. Nuestra opción es utilizar la tecnología REST, debido a las ventajas relacionadas con la simplicidad y rapidez así como al intercambio más ligero de mensajes. Como se puede ver en el capítulo 4, la velocidad actual a la que se permite el intercambio de datos en los dispositivos móviles es muy limitada, por lo que consideramos imprescindible la utilización de una tecnología que permita minimizar este tráfico, dando agilidad a la comunicación e implícitamente a la aplicación. En base a estas decisiones, se pretende crear una aplicación fiable y rápida que mejore la interacción del usuario con el dispositivo.

Desde el punto de vista de la arquitectura cliente-servidor, las otras funcionalidades explicadas en el apartado 10.1, serán propias de cada usuario y serán parte de la aplicación cliente. De esta forma, se creará un cliente relativamente pesado que permitirá su uso aunque no se disponga de conectividad con el proveedor de servicios.

Tanto el cliente como el servidor se basarán en una arquitectura de tres capas, capa de presentación, capa de negocio y capa de datos, lo que permitirá un mayor mantenimiento y flexibilidad. El servidor no se implementará utilizando varios niveles, dado que la información gestionada no es de carácter sensible, por lo que la característica de seguridad que ofrece esta división en niveles no es especialmente importante en nuestro caso.

Por último, los conceptos de conectividad y sincronización. Existen tres tipos de conectividad: siempre conectado, parcialmente conectado y nunca conectado, pudiéndose consultar los beneficios y características de cada uno en el apartado 7.3. El último tipo de conectividad se descarta, por definición de nuestra aplicación, dado que ésta requiere una comunicación con el servicio proveedor de trayectos. El modo por defecto de los dispositivos móviles es el de siempre conectado, ya que siempre ocurre un traspaso de información entre el dispositivo y la red móvil. Esta conectividad no es implícita en lo que tráfico de datos se refiere, pero la mayoría de los terminales a los que la aplicación se dirige disponen de facilidades de conexión continua de datos, garantizada por el protocolo IP móvil. La conectividad parcial en muchos de los casos implica problemas de sincronización de datos y de mantenimiento de la integridad de los mismos. Debido al requisito de que en el seguimiento de un trayecto es necesaria una conectividad continua para poder visualizar el mapa, es necesaria una conectividad continua para nuestra aplicación.

En lo que a sincronización se refiere, nuestra aplicación necesitará una conexión continua con el servidor, pero no necesariamente una sincronización continua. Recordemos

que la sincronización continua, que es propia del modo siempre conectado, supone un gasto de ancho de banda superior e implica cierta urgencia de los datos en relación con el tiempo. En nuestro caso, esta urgencia es inexistente además, es el usuario el que decide una grabación del trayecto a posteriori. Un usuario puede terminar su entrenamiento y decidir no grabar los datos utilizando el servicio, por lo cual la idea de una sincronización continua supone deshacer los cambios ya hechos y una pérdida de los datos enviados del 100%. Debido a estos motivos optamos por una sincronización en modo almacenamiento y reenvío, de manera que los datos se irán guardando en el dispositivo móvil hasta que el usuario decide su persistencia; estos datos, además, son incompletos y no son urgentes. Es obvio que este método permite ahorrar ancho de banda y ofrece al usuario más autonomía sobre lo que decide hacer en la aplicación.

Hasta el momento, se ha realizado una incursión en el estado del arte de los aspectos que intervienen en el sistema de entrenamiento que proponemos y se ha visto las posibilidades y opciones de las que se dispone. También en este capítulo se ha optado por algunas decisiones, en detrimento de otras, y se ha argumentado nuestras opciones. Después de haber visto los distintos módulos que forman nuestro sistema, y para concluir con este capítulo, nuestra propuesta se podría resumir en la Figura 10.3. A partir de aquí nuestro trabajo se centrará en el desarrollo software necesario para llevar a cabo un buen diseño, implementación y despliegue del sistema propuesto.

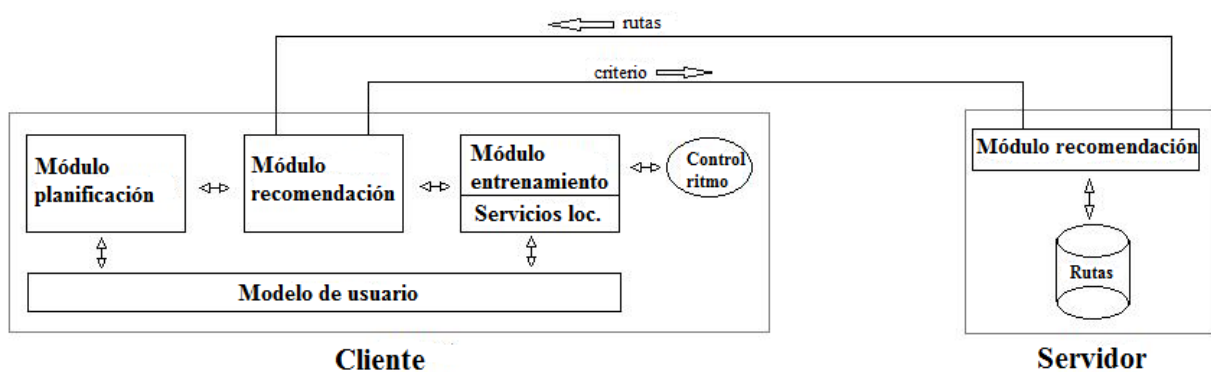


Figura 10.3. Arquitectura del sistema



# 11. Proceso de desarrollo de software

Se entiende por proceso de desarrollo de software la estructuración de las tareas que se siguen para crear un software. En la actualidad existen varias metodologías de procesos de desarrollo de software, que se pueden agrupar en las siguientes categorías:

- Codificar y corregir
- Modelos secuenciales
  - Modelo en cascada
  - Modelo de construcción de prototipos
  - Modelo de desarrollo rápido de aplicaciones
- Modelos evolutivos
  - Modelo de desarrollo incremental
  - Modelo en espiral
  - Modelo del proceso unificado
  - Modelo basado en metodologías ágiles

El modelo de codificar y corregir es un modelo muy básico, por lo que, debido a su simplicidad, es útil únicamente en casos de proyectos de pequeña envergadura. Este modelo está formado por tres subprocesos: posible especificación del sistema, codificación y corrección, entrega.

Los *modelos secuenciales* son modelos de procesos más avanzados, mejor estructurados pero que siguen presentando importantes desventajas.

En el caso del *modelo en cascada* este se basa en varios subprocesos, como el concepto de software, análisis de requisitos, diseño global, diseño detallado, codificación y depuración, y prueba del sistema. Tiene importantes desventajas como la linealidad, ya que hasta que no se termina una fase no se procede con la siguiente. No presenta el proceso real de desarrollo software, que no es lineal ya que siempre hay iteraciones.

Otro modelo secuencial es el *modelo de construcción de prototipos*, que consta de cuatro fases: concepto inicial, diseño e implementación del prototipo inicial, refinación del prototipo y entrega. Al igual que su antecesor, este modelo presenta una estructura lineal que no refleja el proceso de desarrollo de forma realista. La fase más importante, la de refinación del prototipo, es una fase muy larga y parecida a la fase de codificación y corrección del modelo de idéntico nombre.

Por último, el *modelo de desarrollo rápido de aplicaciones* es un método que se basa en el desarrollo paralelo de distintos módulos por parte de equipos diferentes. Esta

aproximación intenta utilizar el desarrollo basado en prototipos y métodos iterativos. Consta de cuatro fases: planificación de requerimientos, diseño basado en el usuario, fase de construcción y fase de traslado.

Los *métodos evolutivos* son los más utilizados y aceptados hoy en día, debido a la aproximación de la metodología al proceso natural de desarrollo, que consiste en repeticiones e incrementos.

El primer modelo propuesto aquí es el de *desarrollo incremental*, que es muy parecido al modelo en cascada, con la particularidad de que una vez llegado al final del proceso, se comienza de nuevo realizándose una refinación, lo que se conoce como iteración.

El *modelo en espiral*, es también un modelo iterativo, que se puede estructurar en cuatro fases: determinar objetivos, identificar y resolver riesgos, desarrollo y test, y planificación de la siguiente iteración. El desarrollo basado en este modelo se centra en realizar iteraciones circulares por las cuatro fases en las que subdivide el modelo.

El *modelo del proceso unificado* es un modelo dirigido por casos de uso, considerándolo muchos profesionales el modelo estándar, aunque esta distinción no esté fijada en sitio alguno. Es un modelo incremental e iterativo que consta de cuatro fases:

- Inicio: Se define el ámbito del proyecto y se desarrollan los casos de uso. En otras palabras, se realiza el modelado del sistema y se analizan los requisitos.
- Elaboración: Se realiza el plan del proyecto, estimaciones y diseño básico.
- Construcción: Se implementa el producto en base a iteraciones.
- Transición: Se realizan los test y se entrega el producto a los usuarios.

El *modelo basado en las metodologías ágiles* es un modelo relativamente nuevo, ya que en 2001 se firma entre varios desarrolladores software el manifiesto de las metodologías ágiles, que perseguían una forma más flexible de desarrollo software. Es un modelo basado en el desarrollo iterativo e incremental, donde las soluciones a los requisitos planteados se resuelven de manera colaborativa entre los distintos equipos desarrolladores. Algunas de las diferencias con respecto al proceso unificado, considerado por muchos como demasiado rígido, son:

- Las personas e iteraciones son más importantes que los procesos y herramientas.
- El software funcional es más importante que la documentación extensa.
- La colaboración con el usuario es más importante que los contratos.
- La capacidad de adaptación al cambio es más importante que el seguimiento de una planificación.



Es un método que tiene cosas en común con el modelo de desarrollo rápido de aplicaciones, ya que se basa en la colaboración entre los distintos equipos de desarrollo. También es uno de los métodos más utilizados hoy en día, debido a su mayor flexibilidad y colaboración.

Aunque con una estructura y *modus operandi* distintos, la mayoría de las metodologías se basan en la división del proceso de desarrollo en subprocesos que muchas veces son comunes o parecidos en diferentes metodologías. Los subprocesos más importantes se pueden considerar que son la ingeniería de requisitos, el diseño, la implementación, el periodo de prueba y el despliegue.

En este capítulo se intentará extender cada uno de los subprocesos anteriores, necesarios para llevar a cabo el desarrollo de un producto software final.

## **11.1. Ingeniería de requisitos**

Es el primer paso para el correcto desarrollo de un software, y comprende las fases relacionadas con la identificación, análisis y especificación de las distintas condiciones necesarias para el cumplimiento de los objetivos del sistema o aplicación a desarrollar.

### **11.1.1. Estudio de viabilidad**

Antes de comenzar la identificación y el análisis de los requisitos, es necesario crear un informe de viabilidad que determine, en nuestro caso, si el proyecto es posible con las restricciones hardware o software de hoy en día. Decimos en nuestro caso porque, aparte de las restricciones tecnológicas, este informe debería también tener puntos de vista relacionados con la rentabilidad comercial o su posible realización en el marco presupuestario. Ninguna perspectiva de este tipo procede considerar para nuestra aplicación ahora mismo, ya que por el momento no estamos interesados en su desarrollo para el ámbito comercial o empresarial.

Llegados a este punto deberíamos tener clara la propuesta hecha en el capítulo 10, el desarrollo de este tipo de sistema de entrenamiento supone el uso de nuevas tecnologías y conceptos no siempre fáciles de utilizar. La plataforma elegida, Android, hace necesaria la formación y adaptación a un nuevo modo de programación, la programación en entornos móviles, lo cual requiere un significativo esfuerzo temporal, más que intelectual, ya que como desarrollador nunca antes lo había utilizado. Aparte de la plataforma, el desarrollo de la aplicación supone el manejo de conceptos más o menos complejos, como sensibilidad al contexto, métodos de localización y una arquitectura cliente-servidor basada en capas y niveles. Los requisitos tecnológicos, en el caso de nuestro sistema de entrenamiento, se

pueden satisfacer utilizando la tecnología actual, aunque puede suponer bastante trabajo y recursos temporales.

Ya se ha comentado que la perspectiva de rentabilidad y el marco presupuestario son inexistentes en nuestro proyecto, aun así se puede tener en cuenta un punto de vista parecido, el esfuerzo temporal. Muchos de los conceptos vistos con anterioridad son novedosos desde un punto de vista personal, lo cual supone un esfuerzo considerable. Para llevar a cabo el desarrollo del proyecto en el marco temporal establecido es necesario un trabajo constante y una correcta planificación del esfuerzo dirigido al proyecto.

El sistema de entrenamiento que pretendemos desarrollar será de libre uso y distribución gratuita, a diferencia de algunas de las aplicaciones existentes orientadas al entrenamiento físico, vistas en el capítulo 9. El objetivo del sistema es ayudar a las personas que pretenden entrenar para terminar una carrera de larga distancia, más concretamente, un maratón. Tal y como se ha visto en el capítulo 9, existe cierta necesidad de desarrollo de una aplicación coherente para el entrenamiento de corredores de larga distancia que incorpore características útiles desde un punto de vista científico. Esta aplicación servirá para ofrecer una correcta guía de entrenamiento que pretende apoyar las decisiones del atleta e intentará evitar posibles lesiones.

En vista a las aportaciones y costes, temporales en nuestro caso, concluimos que el proyecto de desarrollo es viable siempre que se lleve a cabo de forma correcta en el límite temporal establecido.

### **11.1.2. Análisis de requisitos**

El análisis de requisitos es una parte fundamental de nuestro sistema, que ayuda al analista a entender el sistema, ya que realiza una identificación de los requisitos del mismo a partir de otros sistemas similares, de discusiones con usuarios o principios etnográficos.

En este subapartado se incluirá un análisis de los posibles actores que intervienen en nuestro sistema, los objetivos que tienen o funcionalidades que les ofrece el sistema y una clasificación de los requisitos. Se utilizarán casos de uso para la descripción de los requisitos funcionales del sistema.

En cuanto a los actores, éstos representan un conjunto coherente de roles que desempeñan los usuarios de los casos de uso al interactuar con el sistema. Estos roles pueden ser desempeñados por personas, por otros dispositivos y otros sistemas externos al nuestro. Los actores se pueden dividir en:

- *Actores primarios*: son los que requieren al sistema el cumplimiento de un objetivo.

- *Actores secundarios*: son utilizados por el sistema para satisfacer su objetivo.

En el caso de nuestra aplicación, se puede distinguir los siguientes actores:

- Actores que siguen una planificación, a partir de ahora llamados *seguidores*.
- Actores que solo utilizan de forma esporádica la aplicación, sin seguir ninguna planificación, a partir de ahora llamados *independientes*.

Los objetivos de los seguidores son la creación de una planificación y la posible modificación de esta cuando la planificación no esté de acuerdo a sus expectativas. Una vez creada la planificación, los seguidores la utilizarán en su entrenamiento, pudiendo en función de la fecha actual ver la sesión actual, anterior y siguiente. También se ofrece una vista general de la planificación, siendo esto especialmente útil a la hora de decidir si seguir o no la planificación. Aparte de la planificación, los seguidores pueden ver los trayectos recomendados para la sesión actual, así como solicitar más información sobre cada trayecto particular recomendado, para así poder decidir cuál de ellos elegir. También se permite ver todos los trayectos de una zona en particular, en lugar de solamente los recomendados. Durante la sesión, se permite la monitorización de los parámetros de sesión, como ritmo, tiempo, distancia y altitud, además de la monitorización del trayecto seguido, así como la activación o desactivación del módulo de control de ritmo. Una vez terminada la sesión, se le permite al seguidor guardar su trayecto en el servidor para que otros usuarios puedan acceder a él, antes de realizar la grabación el usuario podrá escribir un comentario sobre el trayecto.

Los objetivos del usuario independiente son parecidos a los del usuario seguidor, con algunas restricciones en el uso del módulo de planificación. Un usuario independiente podrá crear una planificación, pero tal y como se ha visto, por definición, no seguirá esta planificación. Los independientes no podrán acceder a algunas funcionalidades relacionadas con el módulo de recomendación de trayectos, ya que no existe una base sobre la que realizar la recomendación. Aun así, los independientes podrán ver todos los trayectos disponibles en su zona y también podrán realizar sesiones libres, donde dispondrán de las funcionalidades de monitorización de parámetros, trayecto o control de ritmo. Recordemos que en el capítulo 10 se había pensado en realizar una advertencia al usuario en el caso de las bajadas en función de su ritmo medio actual. También se le permite al usuario independiente guardar trayectos en el servidor y realizar comentarios sobre éstos.

Aparte de estos requisitos, tanto los usuarios independientes como los seguidores, podrán hacer uso de las funcionalidades relacionadas con el cambio de idioma o sistema de medición. Un usuario independiente se convierte en seguidor cuando decide seguir una planificación y un usuario seguidor se convierte en independiente cuando deja de seguirla.

Los objetivos de los usuarios nos sirven de apoyo en la creación de los requisitos del sistema. Los requisitos de un sistema son la descripción de los servicios proporcionados y sus restricciones operativas. Los requisitos se pueden dividir en varios grupos, pero los más importantes se considera que son, los requisitos funcionales y los no funcionales.

- *Requisitos funcionales*: Son la declaración de las funciones que tiene que proporcionar el sistema.
- *Requisitos no funcionales*: Son las restricciones que debe cumplir el sistema.

Los requisitos funcionales se enumeran a partir de los objetivos que tienen los distintos actores que interactúan con el sistema, y pueden cambiar durante el desarrollo de la aplicación. Los requisitos funcionales, agrupados en módulos, que nosotros identificamos son:

#### Módulo de planificación:

- Creación de la planificación
- Validar datos
- Modificación de la planificación
- Vista general de la planificación
- Seguir planificación
- Abandonar planificación
- Ver sesión actual
- Ver sesión anterior
- Ver sesión siguiente

#### Módulo de entrenamiento:

- Ver trayectos recomendados
- Ver todos los trayectos
- Ver detalles del trayecto
- Comenzar sesión
- Ver trayecto actual
- Ver parámetros de sesión
- Grabar trayecto
- Valorar trayecto
- Conmutar control de ritmo

#### Módulo de configuración:

- Cambiar idioma
- Cambiar sistema de medición

### 11.1.3. Requisitos no funcionales

Los requisitos no funcionales que tienen que ver con las restricciones que el sistema debe cumplir son los siguientes:

- El sistema de entrenamiento consta de una aplicación móvil sobre la plataforma Android y un servicio web implementado en JAVA, utilizando como referencia la Petición de Implementación Java 311 (JSR – Java Specification Request) [57], cuyo resultado es Jersey [58].
- El modelo de usuario y los datos referentes a cada usuario, como su planificación, se guardarán en el dispositivo móvil, en una base de datos SQLite [59].
- Los trayectos, con sus atributos correspondientes, se guardarán en el servidor, en una base de datos MySQL [60], haciendo uso de la API de Persistencia de Java [61].
- El sistema proveerá la seguridad necesaria en el almacenamiento y manipulación de los datos, según el marco legislativo vigente.
- El sistema proporcionará una interfaz sencilla y usable que no interferirá en la actividad del usuario.
- El dispositivo móvil necesitará conectividad continua a un servicio de datos en la red móvil.
- El sistema utilizará una arquitectura en capas que permitirá un fácil mantenimiento y escalabilidad.

### 11.1.4. Definición de requisitos funcionales

La fase de definición representa el proceso de descripción de los requisitos funcionales obtenidos en la fase de análisis. Para la definición se utilizará un lenguaje natural, y los requisitos funcionales se definirán desde la perspectiva del usuario, es decir, de forma más abstracta de la que se definirían los requisitos del sistema.

*Creación de la planificación:* A partir de ciertos datos, como el tiempo de la última carrera, el tipo de ésta carrera y la fecha del maratón en el que desea participar, creará una planificación inteligente basada en nuestro nivel físico.

*Validar datos:* Verifica que los parámetros introducidos por el usuario son correctos y corresponden a cierto rango, avisando al usuario en caso contrario.

*Vista general de la planificación:* Proporciona una vista completa de la planificación, con las fechas de entrenamiento, distancias y ritmos recomendados.

*Modificación de la planificación:* Si el usuario no está de acuerdo con la planificación creada, se le permite la modificación de los parámetros, para crear una planificación nueva.

*Seguir planificación:* En el caso de que el usuario esté de acuerdo con la planificación, ésta se puede seguir, de modo que el sistema tendrá más funcionalidad cuando se siga una planificación.

*Abandonar planificación:* Si decidimos que la planificación no es adecuada o no nos ayuda en nuestro entrenamiento, se puede abandonar una planificación.

*Ver sesión actual:* Se nos muestra la fecha, distancia y el ritmo para la sesión actual.

*Ver sesión anterior:* Permite ver los datos de la última sesión.

*Ver sesión siguiente:* Permite ver los datos de la sesión próxima.

*Ver trayectos recomendados:* En el caso de los usuarios seguidores permite ver los trayectos que el sistema considera oportunos para la sesión.

*Ver todos los trayectos:* Nos permite ver todos los trayectos disponibles en nuestra área.

*Ver detalles trayecto:* En el caso de querer obtener más datos sobre el trayecto, como distancia o comentarios, esta funcionalidad nos permite hacerlo.

*Comenzar sesión:* Esta funcionalidad representa el inicio de la actividad física, es el momento a partir del cual se comenzará la monitorización de los parámetros y del trayecto.

*Ver trayecto actual:* En caso de querer ver el trayecto ya realizado, ésta es la opción correspondiente.

*Ver parámetros de sesión:* La funcionalidad adecuada en caso de querer ver el ritmo, la distancia o el tiempo transcurrido.

*Grabar trayecto:* Una vez terminada la sesión, el usuario puede decidir si quiere grabar el trayecto actual o no. El trayecto se graba con sus datos inherentes: comentario, distancia, y mapa de altitud.

*Valorar trayecto:* Se permite realizar valoraciones sobre el trayecto, como añadir un comentario, por ejemplo.

*Conmutar control de ritmo:* Esta funcionalidad nos permite activar o desactivar el control de ritmo, cuya misión es evitar lesiones, asegurándose de que nuestro ritmo de carrera es el adecuado.

*Cambiar idioma:* Como su nombre indica, permite cambiar la interfaz del usuario en función del idioma elegido.

*Cambiar sistema de medición:* Permite el intercambio entre el sistema métrico y el imperial.

En el mundo de la empresa, la especificación de estos requisitos constituye la base a partir de la cual se negocia el contrato entre el cliente y el equipo de desarrollo, por lo que es conveniente especificarlos lo más detalladamente posible.

### 11.1.5. Modelo de casos de uso

Como resumen de la etapa de análisis y especificación de requisitos, la Figura 11.1 muestra el diagrama de casos de uso que representa toda la funcionalidad de nuestro sistema de entrenamiento. La especificación completa de los distintos elementos que aparecen en dicho diagrama se puede ver en el Anexo 1.

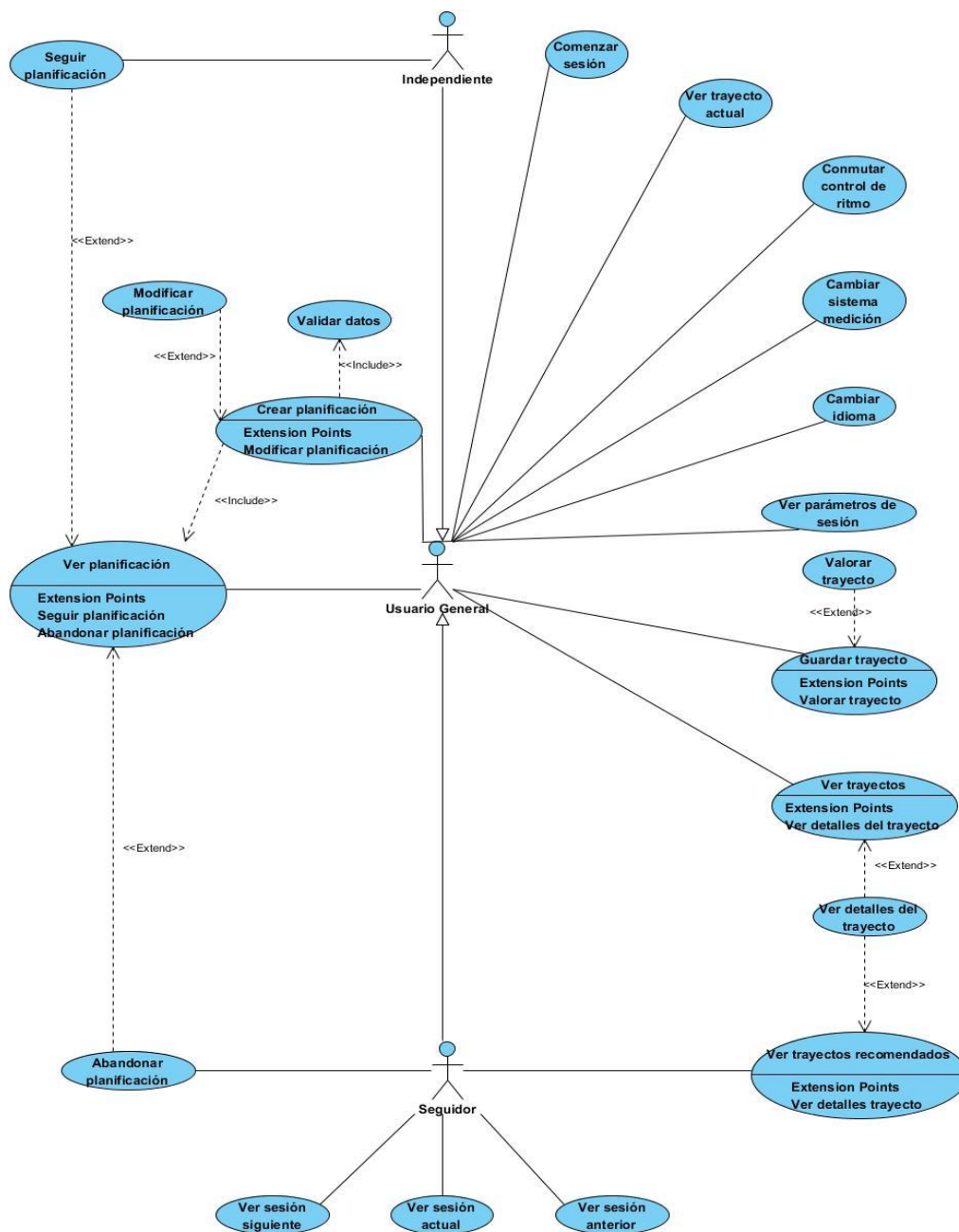


Figura 11.1. Diagrama de casos de uso

### 11.1.6. Modelo de dominio

Para dar por finalizado el proceso de análisis es necesario elaborar un modelo de dominio, que es una representación visual de las clases conceptuales existentes en el dominio de nuestro problema. En la creación del modelo de dominio se han tomado como entradas los casos de uso y la descripción del sistema para obtener un diagrama de clases simplificado de los objetos del mundo real existentes en nuestra aplicación.

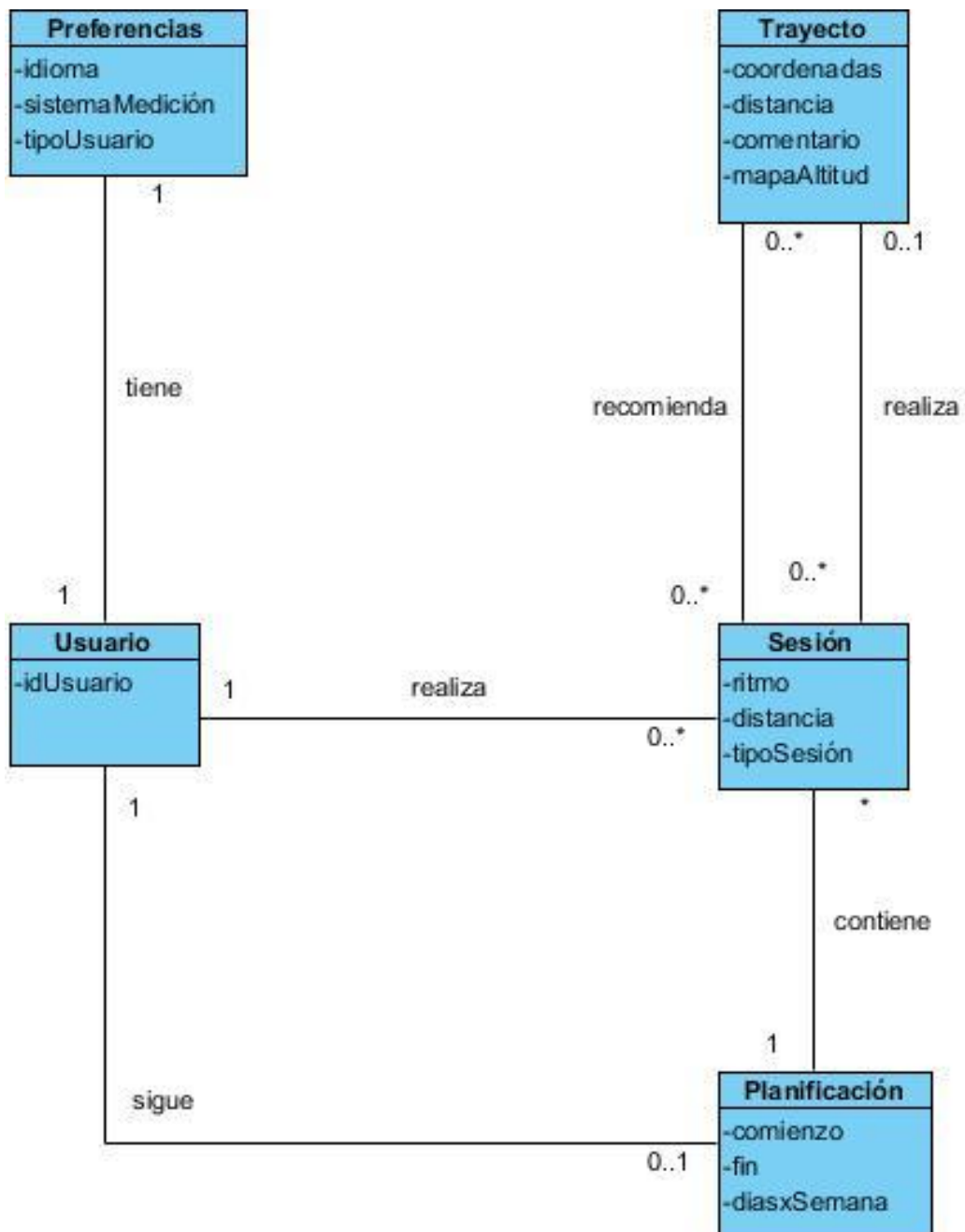


Figura 11.2. Modelo de dominio



## 11.2. Diseño del sistema

### 11.2.1. Modelo de la interfaz de usuario

Vista ya la ingeniería de requisitos, resulta necesaria la creación de un prototipo de interfaz de usuario, para poder mostrárselo al cliente o para darse cuenta de posibles necesidades o modificaciones de la aplicación, que puede resultar en la modificación de los requisitos y el diseño.

La técnica frecuentemente utilizada en la creación de un prototipo de interfaz de usuario son los *storyboards*. Los *storyboards* son una estructura organizada de imágenes, representando posibles pantallas de la aplicación, que ayudan a previsualizar el formato en el que los datos serán presentados al usuario.

A continuación vamos a presentar una serie de bocetos que representan los casos de uso más importantes de nuestra aplicación.

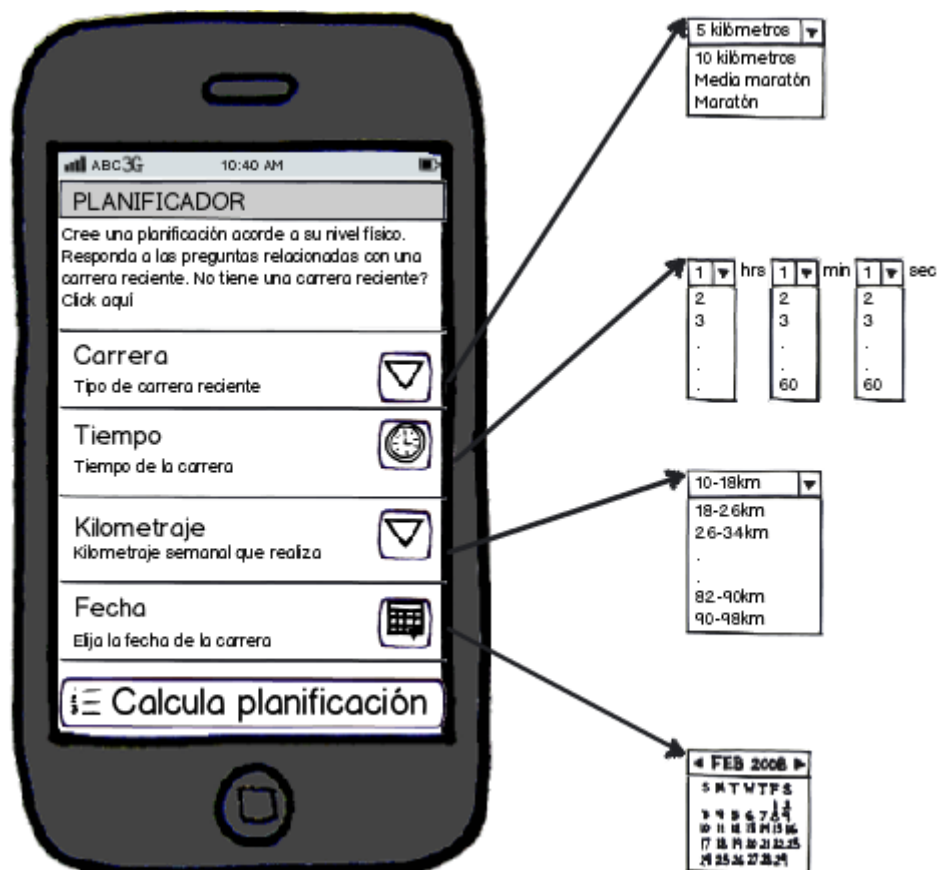


Figura 11.3. Boceto para la pantalla inicial del planificador

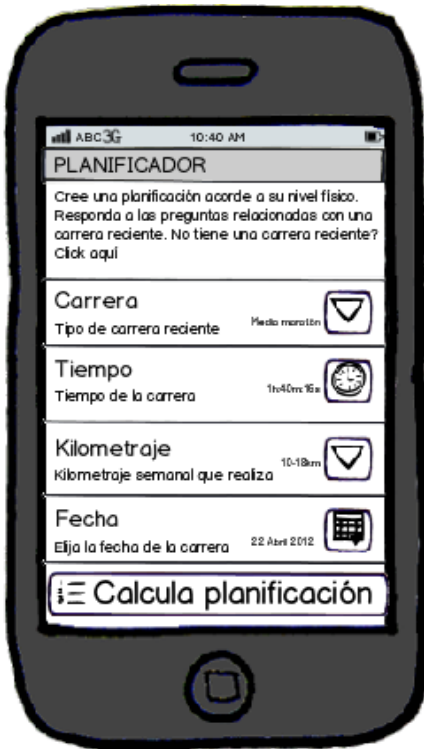


Figura 11.4. Bocetos para las pantallas intermedia y final del planificador



Figura 11.5. Bocetos para las pantallas de sesiones y de recomendación de trayectos



Figura 11.4. Bocetos para las pantallas de parámetros de la sesión y de trayecto actual



Figura 11.5. Boceto para la pantalla de guardar el trayecto actual

### 11.2.2. Modelo de diseño de la lógica de negocio

Un proceso de negocio es un grupo de tareas relacionadas que interactúan para producir un resultado determinado. De modo práctico, la lógica de negocio de una aplicación consiste en la funcionalidad a la que puede acceder la capa de presentación, es decir, los procesos que determinan el uso del sistema. Un diagrama de flujo de datos es un diagrama de modelización que nos muestra un sistema como una red de procesos conectados entre sí por flujos y almacenamientos de datos. Un diagrama de este tipo se utiliza para mostrar qué tipos de datos maneja el sistema, y cuáles son los procesos que manejan estos datos. Las siguientes figuras muestran los diagramas de este tipo correspondientes a nuestra aplicación.

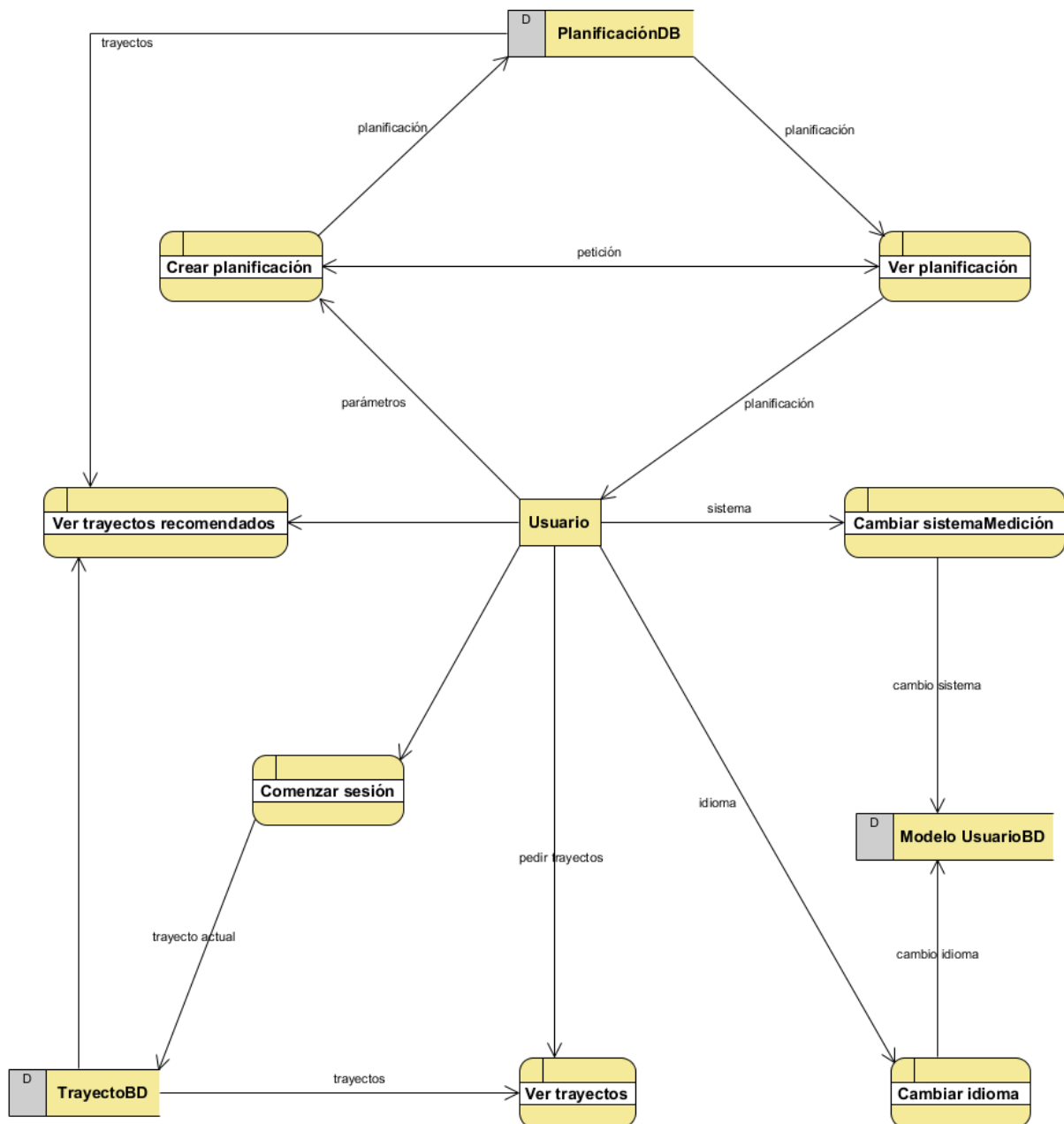


Figura 11.6. Diagrama de flujos de datos general

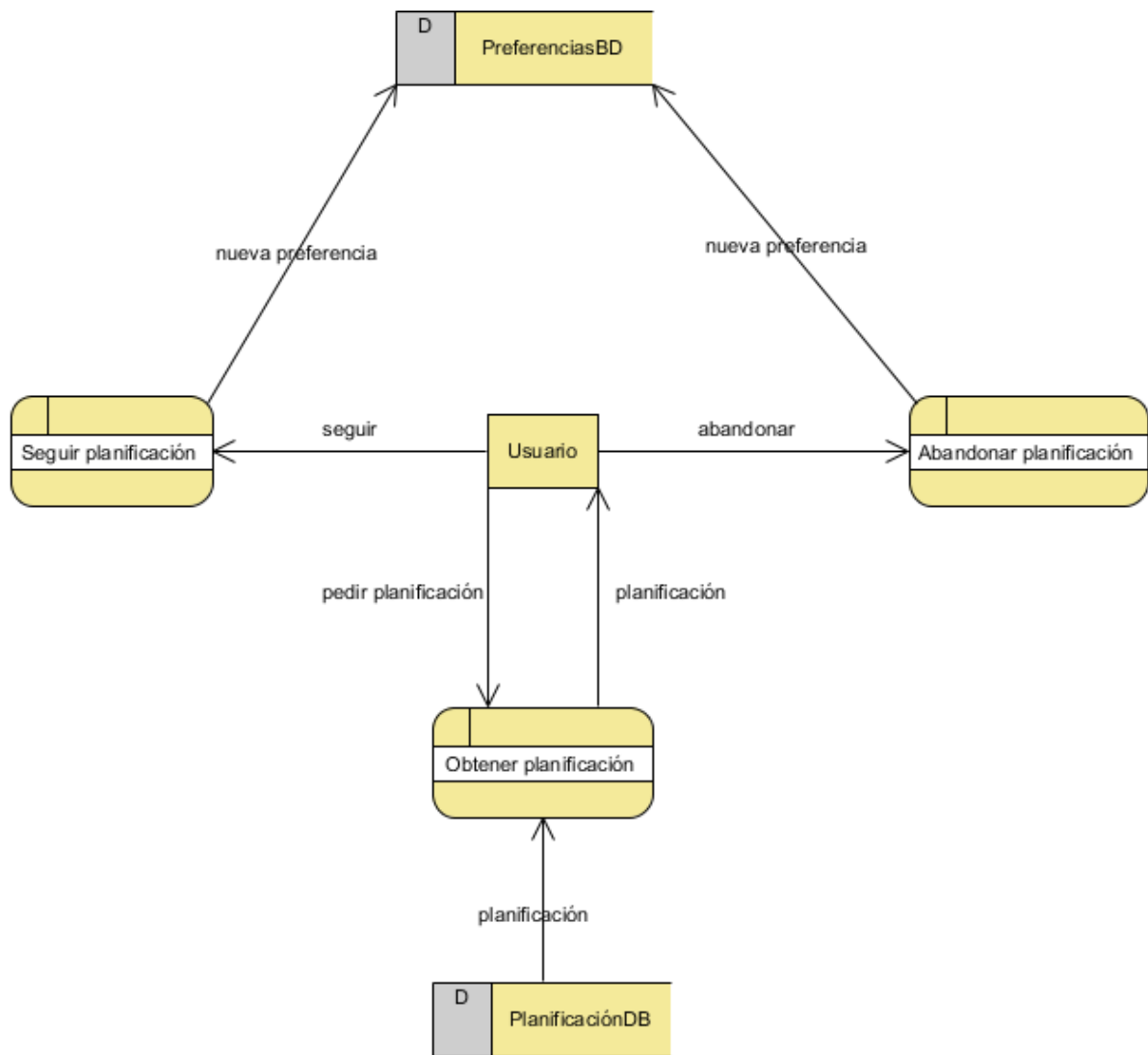


Figura 11.7. Diagrama de flujo de datos para el proceso Ver planificación



Figura 11.8. Diagrama de flujos de datos para el proceso Crear planificación

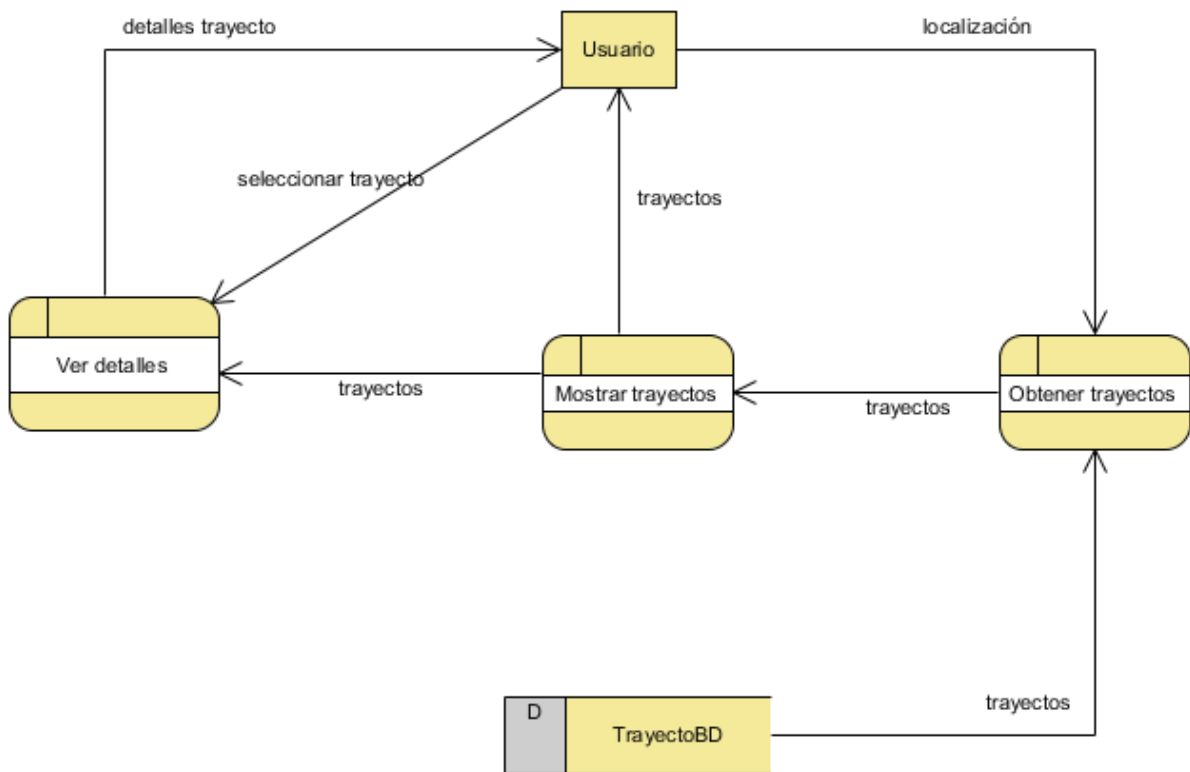


Figura 11.91. Diagrama de flujos de datos para el proceso Ver trayectos

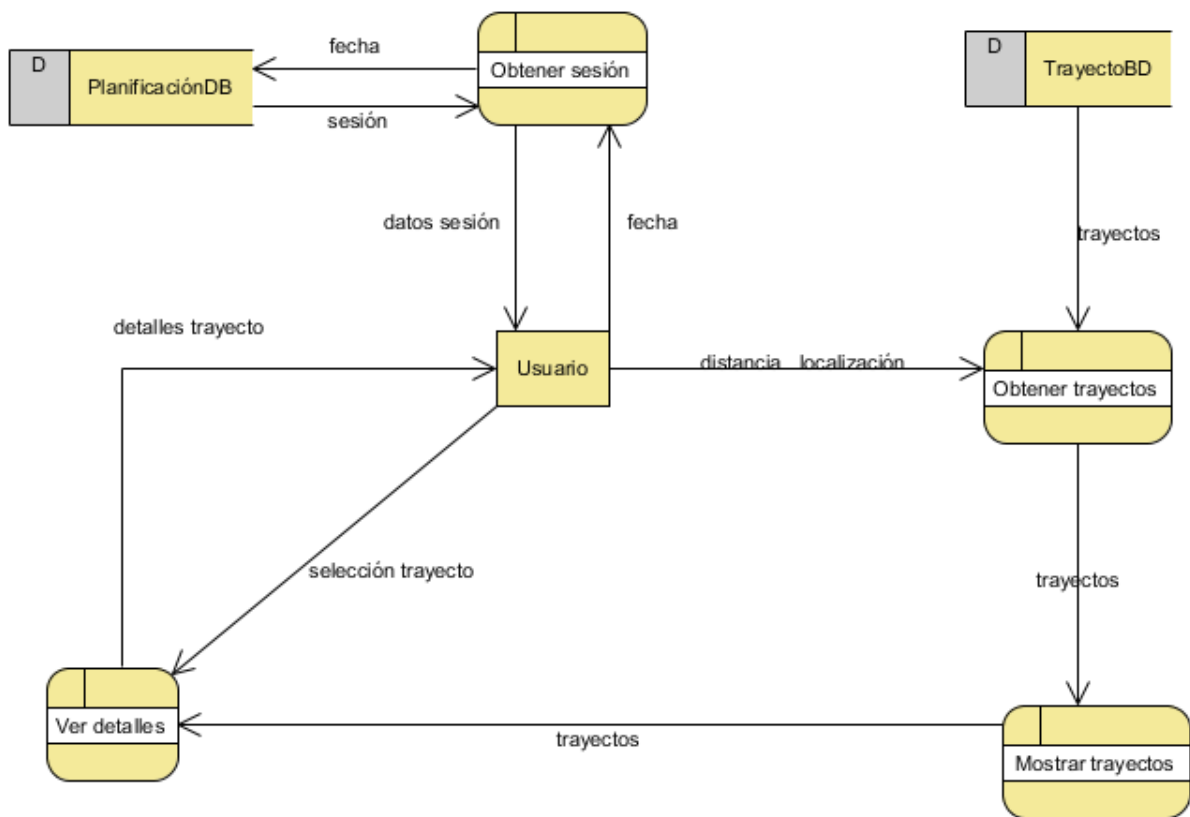


Figura 11.12. Diagrama de flujos de datos para el proceso Ver trayectos recomendados

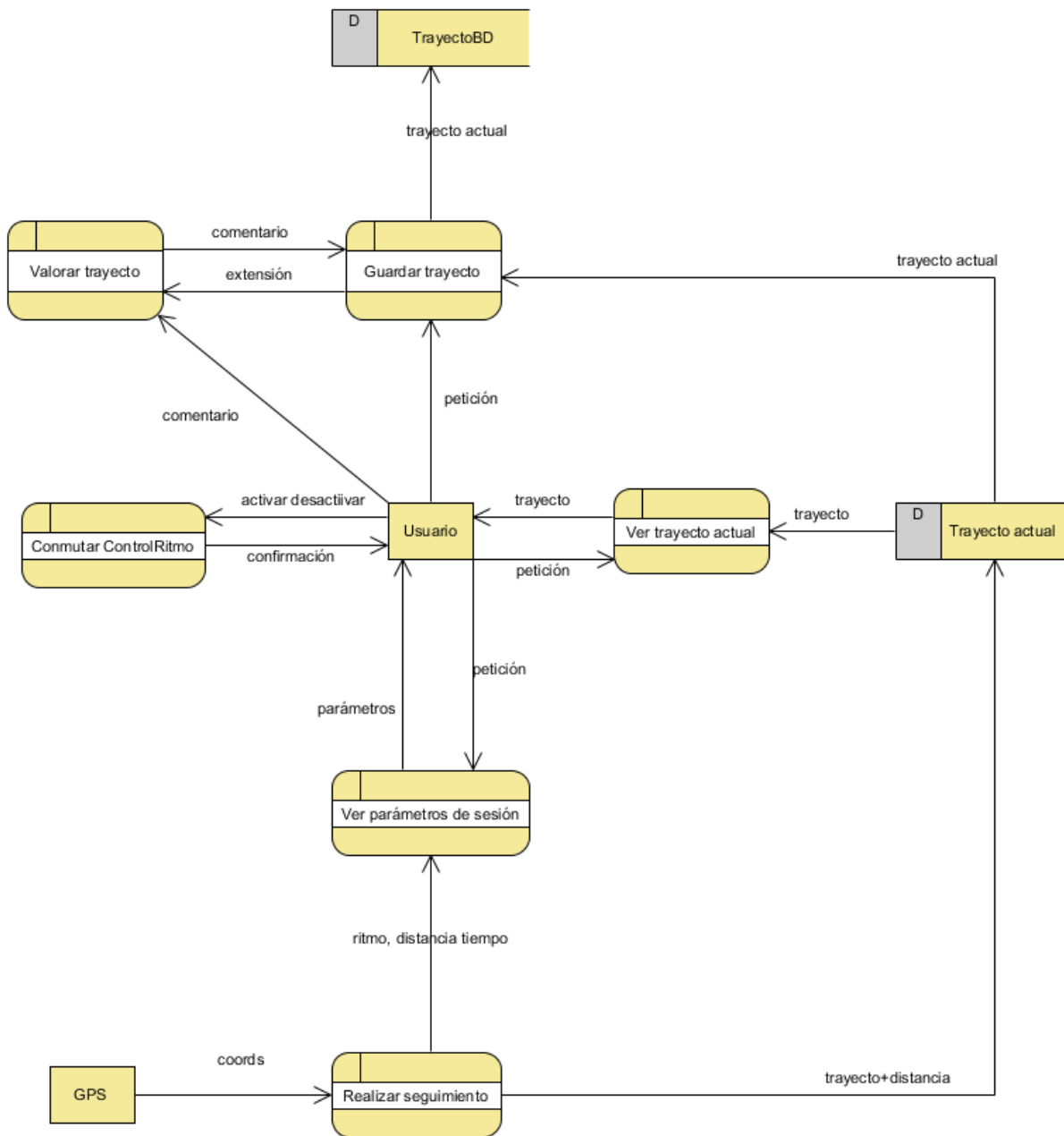


Figura 11.10. Diagrama de flujos de datos para el proceso Comenzar sesión



Una vez vistos los procesos que intervienen en la aplicación y los datos que se manejan, se procederá a describir más en detalle la estructura de los datos. Para la descripción de la estructura de datos se ha creado un modelo de entidades de negocio que puede verse en la Figura 11.14.

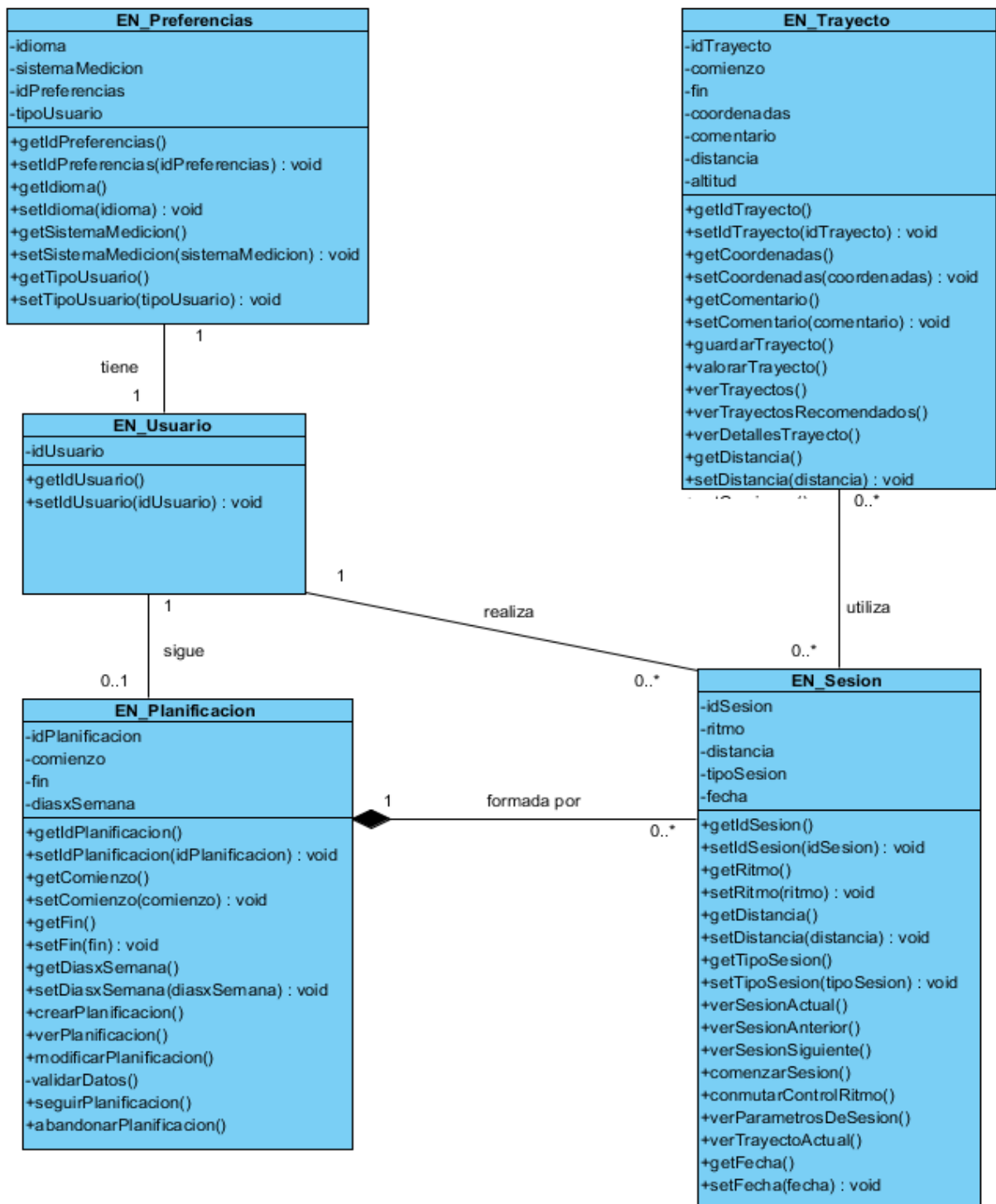


Figura 11.11. Modelo de entidades de negocio

### 11.2.3. Modelo de diseño de los datos

El modelo de diseño de datos pone de manifiesto la estructura que tomarán los datos en una aplicación. En nuestro caso se crearán dos diagramas que servirán para esta finalidad. Primero se creará un diagrama de clases que define una capa intermedia, que realiza la comunicación entre las entidades de negocio y la base de datos, esta capa intermedia se denomina *capa de componentes de acceso*, y se muestra en la Figura 11.15.

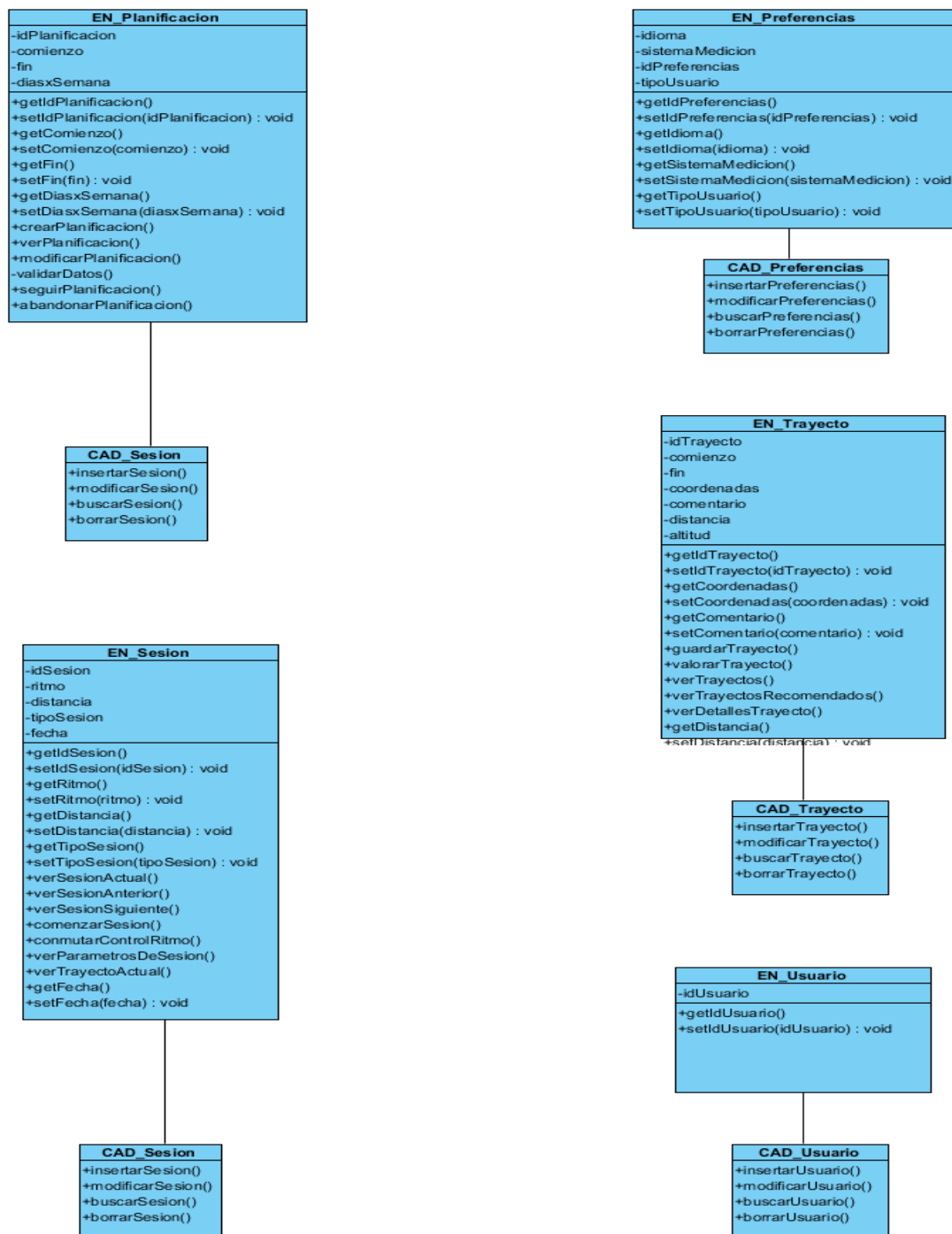


Figura 11.12. Componentes de acceso a datos

Por último, se creará la estructura de la bases de datos, utilizando el modelo entidad-relación. El esquema conceptual de la base de datos se puede ver en la Figura 11.16.

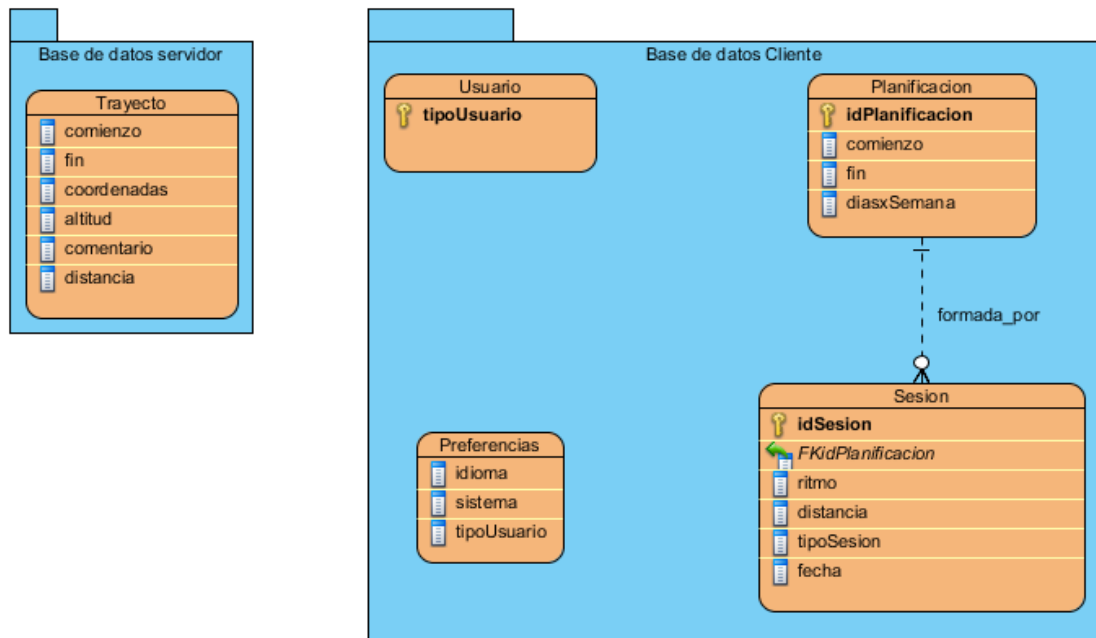


Figura 11.13 Vista conceptual del modelo de datos

El esquema físico de los datos es una derivación del modelo conceptual y es dependiente del Sistema de Gestión de Bases de Datos que se utilizará. El esquema físico de los datos se puede ver en la Figura 11.14.

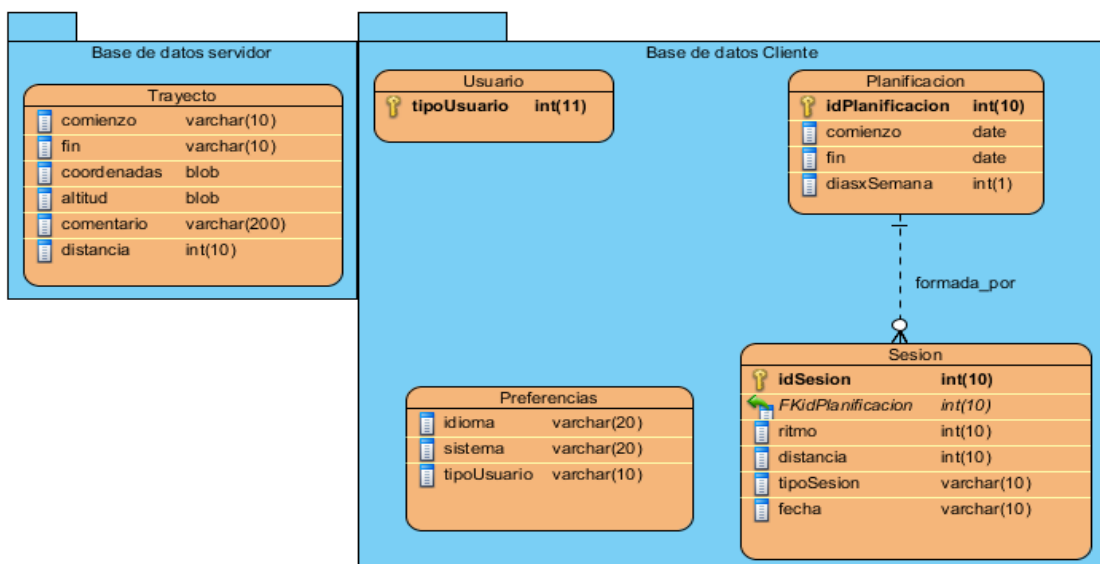


Figura 11.14. Vista física del modelo de datos

## 11.3. Implementación

### 11.3.1. Principales tecnologías utilizadas

Para la implementación de la aplicación se ha utilizado el entorno de desarrollo Eclipse [62] y el Kit de Desarrollo Software (SDK – Software Development Kit) de Android [22]. Para el desarrollo del servidor se ha utilizado Jersey sobre el servidor web Apache Tomcat [63]. Se ha implementado satisfactoriamente la totalidad de la funcionalidad de la aplicación creándose de esta forma un prototipo horizontal bastante avanzado. En lo que se refiere al servidor, su desarrollo también está lo suficientemente avanzado como para permitir realizar todos los objetivos propuestos en el capítulo 10.

### 11.3.2. Otras tecnologías utilizadas

Aparte de las tecnologías primarias anteriormente enumeradas, habría que mencionar también tecnologías auxiliares en las que nos hemos apoyado durante el desarrollo del sistema.

- **MySQL** [60]: Según la web de la propia empresa, el sistema de bases de datos relacional más utilizado en el mundo.
- **SQLite** [59]: Es un sistema de bases de datos relacional, bastante más limitado que el anterior, pero de mucho uso en dispositivos con capacidades limitadas, como los smartphones.
- **XML** [32]: Lenguaje de marcado.
- **JPA** [61]: Es una tecnología que permite el fácil mapeo a nivel de objeto relacional y se encarga de la persistencia de los objetos en la BD.
- **JPQL** [64]: Lenguaje para crear consultas cuando se utiliza JPA.
- **Google Maps** [65]: Servicio Web de mapas ofrecido por Google.
- **Android Patterns** [66]: Conjunto de patrones de diseño para la plataforma Android.

## 11.4. Pruebas y despliegue

En cuanto a las pruebas se refiere cabe decir que esta fase del desarrollo software es de las que más iteraciones sufre, sobre todo en el caso de desarrolladores no experimentados. En nuestro caso, se han realizado pruebas funcionales, que determinan la validez de los requisitos funcionales, al terminar cada uno de los módulos del sistema. Se han realizado también pruebas no funcionales, relacionadas con el mantenimiento y la migración, se intentó migrar el servidor web hacia el servicio de almacenamiento web que provee Google en la nube, siendo el resultado resultando completamente satisfactorio. No se han realizado pruebas no funcionales de escalabilidad o seguridad. Las pruebas realizadas han sido de carácter dinámico y manual, es decir, ejecutando el código de la aplicación sin hacer uso de ningún tipo de software de ejecución de pruebas. Algunas de las pruebas estáticas que habría que realizar son las revisiones o inspecciones.

Una vez realizadas las pruebas, encontrados y corregidos los errores, se procede al despliegue de la aplicación. Para el despliegue de la aplicación se ha utilizado el dispositivo móvil Samsung Galaxy S Plus, que tiene las siguientes características:

- Android OS 2.3
- Memoria 512 MB RAM
- Pantalla AMOLED capacitiva
- CPU 1.4 Ghz Scorpion
- GPU Adreno 205
- Conectividad GPRS, EDGE, HSDPA, HSUPA
- GPRS y sensores: acelerómetro, brújula y sensor de proximidad.

Para el despliegue del servidor se ha utilizado el equipo personal, con las siguientes características:

- Procesador Intel Core I5-2410M, 2.3Ghz y 3MB cache
- Pantalla 15.5 pulgadas
- Memoria RAM: 4GB
- Disco duro: 500GB, 5400 rpm
- Tarjeta gráfica: nVidia GT 540M, 1GB GDDR3
- Windows 7, 64 bits

Decir también que la conexión del servidor se ha realizado mediante una línea ADSL de 6MB/1MB de bajada, utilizándose un router Comtrend CT-536.



# 12. Conclusiones y trabajos futuros

## 12.1. Conclusiones

En este trabajo se ha propuesto un sistema de entrenamiento inteligente que ayude al usuario ofreciendo consejos de entrenamiento, recomendándole trayectos y realizando un seguimiento durante la preparación de una carrera de maratón. Las necesidades específicas previas relacionadas con el estudio y análisis del estado del arte en las distintas categorías o áreas que se habían previsto se han cumplido, tal y como se ha visto en los correspondiente capítulos y se indica a continuación:

- ✓ NE-1: Entrenamiento para corredores de larga distancia (Capítulo 2).
- ✓ NE-2: Computación Móvil y sensibilidad al contexto (Capítulo 3).
- ✓ NE-3: Análisis de la tecnología móvil actual y las restricciones que impone en el desarrollo software (Capítulo 4 y Capítulo 5).
- ✓ NE-4: Métodos de localización en dispositivos móviles (Capítulo 6).
- ✓ NE-5: Arquitectura orientada a servicios (Capítulo 7).
- ✓ NE-6: Modelo de usuario (Capítulo 8).
- ✓ NE-7: Trabajos relacionados (Capítulo 9).

Una vez que estas necesidades específicas previas se han cumplido, se ha podido diseñar un sistema que cumpla con los objetivos de desarrollo, el cumplimiento de dichos objetivos se puede observar en el capítulo que resume nuestra propuesta, el capítulo 10, donde se indica que se ha realizado lo siguiente, dando así cumplimiento a todos nuestros objetivos específicos:

- ✓ OE-1: Un módulo que ofrezca una planificación personalizada para corredores de larga distancia.
- ✓ OE-2: Un módulo de monitorización de parámetros de sesión.

- ✓ OE-3: Un módulo de control de ritmo que se acople al módulo de monitorización y cuyo propósito sea el de evitar lesiones.
- ✓ OE-4: Un servicio capaz de almacenar los trayectos realizados por los usuarios, junto con sus parámetros.
- ✓ OE-5: Un módulo de recomendación de trayectos sensible al contexto.

Así pues, un usuario puede hacer uso de nuestro sistema para crear una planificación basada en sus resultados recientes y utilizar esta planificación entrenando para una carrera de maratón. La carrera de maratón es la única soportada por el prototipo actual, aunque esta limitación es fácilmente ampliable a otras carreras de larga distancia, como las de media maratón.

Una vez que un usuario crea una planificación y decide seguirla, la funcionalidad de la aplicación a la que tiene acceso se amplía, pudiendo hacer uso del módulo de recomendación de trayectos. Este módulo realiza recomendaciones en base a la posición actual y a la distancia recomendada por el módulo de planificación. La recomendación también tiene en cuenta la altitud acumulada del trayecto, de tal forma que se evita recomendar trayectos con una pendiente negativa, ya que, tal y como hemos visto en el capítulo 10, puede tener efectos adversos.

El módulo de monitorización de parámetros es usable tanto por los usuarios seguidores como por los independientes. Es un requerimiento imprescindible de este tipo de sistemas, ya que es el encargado de ofrecer retroalimentación de los parámetros de la sesión de entrenamiento.

Una vez que el usuario termina su sesión de correr, ya sea el independiente o seguidor, se le permitirá guardar el trayecto, junto con los datos de distancia, altitud y un comentario que recoja su opinión en un servidor habilitado donde la información almacenada se utilizará para ser compartida, por parte del módulo de recomendaciones, con los demás usuarios.

El modelo de usuario dinámico subyacente a la aplicación permite mantener y modificar las preferencias del usuario. Entre las preferencias más importantes están las que permiten la internacionalización de la aplicación, haciendo posible cambiar el sistema de medición o el idioma. Actualmente la aplicación permite el intercambio entre el sistema imperial y el métrico, así como la posibilidad de utilizar la aplicación en Inglés o Castellano.

Indicar también que se ha realizado un artículo en base al trabajo presentado en esta memoria, artículo que ha sido aceptado para ser presentado en el XIII Congreso Internacional de Interacción Persona-Ordenador (Interacción 2012) y publicado en las actas del mismo.



Además, se ha pedido una beca FPU para realizar un doctorado en la Universidad de Granada, la memoria que se adjunta a la solicitud proviene de una idea que tiene como base la investigación realizada para el trabajo fin de master presente. La propuesta consiste en la creación de una metodología de desarrollo software para dispositivos móviles. Se ha pensado en dicho tema debido a que durante la realización del presente trabajo se observó la inexistencia de una metodología de este tipo.

## **12.2. Trabajo futuro**

Aunque se hayan cumplido todos los objetivos propuestos, siempre se pueden realizar mejoras y modificaciones al producto final. Algunas de las posibles mejoras que se pueden incorporar a nuestro sistema de entrenamiento son las siguientes:

- Será importante llevar a cabo una evaluación empírica del sistema de entrenamiento utilizando un grupo de atletas, que permitan determinar la idoneidad del sistema y/o posibles mejoras y extensiones, gracias a sus sugerencias.
- Realizar un módulo de seguimiento en tiempo real de la ruta seleccionada previamente, de esta forma el usuario se puede asegurar que está siguiendo la ruta seleccionada sin estar pendiente de la interfaz visual del terminal.
- Modificar el módulo de control de ritmo de tal forma que se tenga en cuenta las diferencias en humedad y temperatura en el ritmo de la sesión. Para llevar a cabo dicha modificación será necesario realizar un estudio de cómo se ve afectado el rendimiento de los atletas con el cambio en humedad y temperatura.
- Realizar un seguimiento de la evolución del atleta en lo que a sesiones de entrenamiento se refiere durante su vida deportiva utilizando nuestro sistema. Éste es un cambio profundo en la aplicación, que supone modificaciones en la arquitectura del sistema. Este cambio nos permitiría mantener un historial de cada atleta, obtener conocimiento en base a la información recopilada, analizar errores en el entrenamiento en el caso de que se produzcan las lesiones y hacer modificaciones y recomendaciones para poder evitarlas.
- Adaptar el sistema de puntuación de los trayectos de tal forma que el módulo sea más colaborativo. Sería interesante que cada usuario que termina un

trayecto sea capaz de dar una puntuación al trayecto, incorporándose esto al algoritmo de recomendación de trayectos actual.

- Llevar a producción el prototipo actual. Aunque dicho prototipo está bastante avanzado, tanto horizontalmente como verticalmente, no deja de ser un prototipo, por lo que se necesita un *refactoring* para que el sistema pueda hacerse público y ser utilizado por los usuarios finales.

# 13. Bibliografía

- [1]. IAAF, Records Statistics, <http://www.iaaf.org/statistics/index.html>.
- [2]. Runner's World. Take the Wet Test: Learn Your Foot Type. Runner's World. <http://www.runnersworld.com/article/0,7120,s6-240-319-326-7152-0,00.html>.
- [3]. American Academy of Family Physicians, Iliotibial Band Syndrome: A Common Source of Knee Pain, <http://www.aafp.org/afp/2005/0415/p1545.html>.
- [4]. WebMD, Runner's Knee. Better information better health, <http://www.webmd.com/pain-management/knee-pain/runners-knee>.
- [5]. WebMD, Plantar Fasciitis, <http://www.webmd.com/a-to-z-guides/plantar-fasciitis-topic-overview>.
- [6]. J. Daniels, Daniels' Running Formula, Versa Press, 2005.
- [7]. J. Gilbert, J. Daniels, Oxygen Power: Performance Tables for Distance Runners, 1979.
- [8]. H. Higdon, Marathon Training Guide - Introduction, <http://www.halhigdon.com/training/51135/Marathon-Training-Guide>.
- [9]. D. Kuehls, Four Months to a Four-hour Marathon, 2006.
- [10]. M. Weiser, The computer for the 21st century, ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 3, 1998.
- [11]. A. K. Dey, G. D. Abowd, A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, Journal in Human-Computer Interaction, Vol. 16, 2001.
- [12]. M. Boutell, J. Luo, Incorporating Temporal Context with Content for Classifying Image Collections, 17th International Conference on Pattern Recognition, 2004.
- [13]. B. H. Lee, H. N. Kim, J. G. Jung, G. S. Jo, Location-Based Service with Context Data for a Restaurant Recommendation, SpringerLink, págs. 430-438, Vol. 4080, 2006.
- [14]. I. Poole, Cellular Communications Explained From Basics to 3G, Elsevier Ltd, Cap. 11, 2005.
- [15]. 3GPP, HSPA, <http://www.3gpp.org/HSPA>.

- [16]. 3GPP LTE. [En línea] <http://www.3gpp.org/LTE>.
- [17]. IEEE, IEEE APPROVES 802.16m - ADVANCED MOBILE BROADBAND WIRELESS STANDARD, <http://standards.ieee.org/news/2011/80216m.html>.
- [18]. Wap Forum, WAP Forum Releases, <http://www.wapforum.org/what/technical.htm>.
- [19]. Oracle, Java ME and Java Card Technology, <http://www.oracle.com/technetwork/java/javame/index.html>.
- [20]. Qualcomm, The BREW Platform, <http://www.brewmp.com/brew-platform>.
- [21]. Apple Inc. iOS Dev Center. [En línea] 2008. <https://developer.apple.com/devcenter/ios/index.action>.
- [22]. Google Inc. Android SDK | Android Developers. [En línea] 2009. <http://developer.android.com/sdk/index.html>.
- [23]. Oracle Inc. Learn About Java Technology. [En línea] 2012. <http://www.java.com/en/about/>.
- [24]. Gartner, Gartner Says Worldwide Mobile Application Store Revenue Forecast to Surpass \$15 Billion in 2011, <http://www.gartner.com/it/page.jsp?id=1529214>.
- [25]. D. Muñoz, F. Bouchereau, C. Vargas, R. Enríquez-Caldera, Position Location Techniques and Applications, Elsevier, 2009.
- [26]. V. Zeimpekis, G. M. Giaglis, G. Lekakos, A taxonomy of indoor and outdoor positioning techniques for mobile location services, Athens University of Economics and Business, 2003.
- [27]. GPS.gov., What is GPS?, <http://www.gps.gov/systems/gps/>.
- [28]. F. v. Diggelen, A-GPS : Assisted GPS, GNSS, and SBAS, Artech House , 2009.
- [29]. SkyhookWireless, <http://www.skyhookwireless.com/>.
- [30]. Skyhook Wireless, Coverage Area Skyhook, <https://www.skyhookwireless.com/location-technology/coverage.php>
- [31]. M. Bell, Service Oriented Modeling: Service Analysis, Design and Architecture, John Wiley and Sons, 2008.
- [32]. T. Bray, J. Paoli, C. M. Sperberg-McQueen, Extensible Markup Language (XML) 1.0., <http://www.w3.org/TR/1998/REC-xml-19980210>.

[33]. D. Crockford, The application/json Media Type for JavaScript Object Notation (JSON), <http://www.ietf.org/rfc/rfc4627>.

[34]. D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, Dave Winer. Simple Object Access Protocol (SOAP) 1.1., <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>.

[35]. R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, Donald Breen School of Information and Computer Science, <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.

[36]. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. Hypertext Transfer Protocol -- HTTP/1.1., <http://www.ietf.org/rfc/rfc2616.txt>.

[37]. E. Christensen, F. Curbera, G. Meredith, S. Weerawarana. Web Services Description Language (WSDL) 1.1., <http://www.w3.org/TR/wsdl>.

[38]. H. Schneider, V. Lee, R. Schell, Introduction to Mobile Application Architectures. Prentice Hall, Cap 3, 2004..

[39]. G. Fischer, User Modeling in Human-Computer Interaction, Kluwer Academic Publishers Hingham, User Modeling and User-Adapted Interaction, Vol. 11, 2001.

[40]. A. Johnson, N. Taatgen, User Modeling, Handbook of human factors in Web design, Lawrence Erlbaum Associates, Cap 11, 2005.

[41]. P. Krömer, V. Snášel, J. Platoš, A. Abraham, Implicit User Modelling Using Hybrid Meta-heuristics, Eighth International Conference on Hybrid Intelligent System, 2008.

[42]. W. IJsselsteijn, Y. d. Kort, J. Westerink, M. d. Jager, R. Bonants, Fun and Sports: Enhancing the Home Fitness Experience, IFIP International Federation for Information Processing, 2004.

[43]. N. Gupta, S. Jilla, Digital Fitness Connector: Smart Wearable System, First International Conference on Informatics and Computational Intelligence, 2011.

[44]. F. Buttussi, L. Chittaro, D. Nadalutti, Bringing mobile guides and fitness activities together: a solution based on an embodied virtual trainer, Proceedings of the 8th conference on Human-computer interaction with mobile devices and services, 2006.

[45]. FitnessKeeper Inc., RunKeeper track, measure and improve your fitness, <http://runkeeper.com/>.

[46]. MapMyFitness Inc., MapMyRun, <http://www.mapmyrun.com/>.

[47]. Rodale Inc., Runner's World SmartCoach. <http://www.runnersworld.com>.













- [48]. Runner's World, <http://www.runnersworld.com>.
- [49]. Color Monkey, Runstar, <http://runstar.se/>.
- [50]. J. S. Gottschall, R. Kram, Ground reaction forces during downhill and uphill running, Elsevier, 2005.
- [51]. USA Track & Field, <http://www.usatf.org/Home.aspx>.
- [52]. International Association of Athletics Federations, Competition rules 2012-2013, [http://www.iaaf.org/mm/Document/06/28/26/62826\\_PDF\\_English.pdf](http://www.iaaf.org/mm/Document/06/28/26/62826_PDF_English.pdf).
- [53]. Y. Song, S. Shin, S. Kim, D. Lee, K. H. Lee, Speed Estimation From a Tri-axial Accelerometer Using Neural, 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Pag. 3224 - 3227, 2007.
- [54]. J. Neville, A. Wixted, D. Rowlands, D. James, Accelerometers: An Underutilized Resource in Sports Monitoring, Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Pag. 287 - 290, 2010.
- [55]. F. B. Abdesslem, A. Phillips, T. Anderson, Less is More: Energy-Efficient Mobile Sensing with SenseLess, Proceedings MobiHeld, 2009.
- [56]. I. Constandache, S. Gaonkar, M. Sayler, R. Roy Choudhury, EnLoc: Energy-Efficient Localization for Mobile Phones, IEEE Infocom, 2009.
- [57]. Oracle Inc, Java Community Process. Jax-RS. The Java Community Process(SM) Program - JSRs: Java Specification Requests - detail JSR# 311, <http://jcp.org/en/jsr/detail?id=311>.
- [58]. Java.net, Jersey, <http://jersey.java.net/>.
- [59]. D. R. Hipp, SQLite, <http://www.sqlite.org/>.
- [60]. MySQL AB. Mysql the world's most popular open source database, <http://www.mysql.com/>.
- [61]. Java Community Process, JPA. The Java Community Process(SM) Program - JSRs: Java Specification Requests - detail JSR# 317, <http://jcp.org/en/jsr/detail?id=317>.
- [62]. Eclipse Foundation, Eclipse - The Eclipse Foundation open source community website, <http://www.eclipse.org/>.
- [63]. Apache Software Foundation, Apache Tomcat - Welcome!, <http://tomcat.apache.org/>.

[64]. Oracle Inc., JPQL Language Reference, [http://docs.oracle.com/html/E24396\\_01/ejb3\\_langref.html](http://docs.oracle.com/html/E24396_01/ejb3_langref.html).












[65]. Google Inc., Google Maps, <http://maps.google.es/>.

[66]. Unitid, Android Patterns, <http://www.androidpatterns.com/>.

## Anexo 1. Especificación casos de uso

Nombre	Descripción
 Usuario General	<p>Este usuario representa una generalización de los dos tipos de usuarios existentes en el sistema: Independiente y Seguidor. Todos los casos de uso a los que él accede podrán ser accedidos por los dos tipos de usuario especializados.</p>
 Crear planificación	<p>A partir de los siguientes datos: tiempo de la última carrera, tipo de carrera y fecha del maratón y rango de kilómetros semanal, se creará una planificación inteligente basada en el nivel físico del usuario.</p>
 Ver planificación	<p>El sistema le muestra al usuario una vista general de la planificación creada, que contendrá las fechas de entrenamiento, las distancias y los ritmos propuestos.</p>
 Validar datos	<p>El sistema valida los parámetros introducidos. Se verifica que el tiempo para la carrera elegida no supere el tiempo mínimo (record actual) ni el máximo (límite en la tabla VDOT).</p>
 Guardar trayecto	<p>Esta funcionalidad permite guardar un trayecto para su posterior uso por parte de otros usuarios.</p>
 Ver trayectos	<p>Permite ver todos los trayectos correspondientes al área en el que se encuentra el usuario. También permite ver los detalles de cada trayecto.</p>
 Cambiar idioma	<p>Esta funcionalidad permite realizar el cambio en la interfaz de usuario en función del idioma seleccionado.</p>
 Cambiar sistema medición	<p>Esta funcionalidad permite seleccionar entre el sistema métrico y el imperial.</p>
 Conmutar control de ritmo	<p>Esta funcionalidad permite activar o desactivar la opción de seguridad que ofrece advertencias sobre el ritmo actual en relación con el recomendado.</p>
 Ver trayecto actual	<p>Esta funcionalidad realiza una traza del trayecto ya realizado en la sesión actual.</p>
 Comenzar sesión	<p>Esta funcionalidad debe coincidir con el inicio de la actividad física, es el momento a partir del cual se comenzará la monitorización de los parámetros y del trayecto.</p>
 Ver parámetros de sesión	<p>Esta funcionalidad le muestra al usuario los parámetros de la sesión en curso.</p>



 Independiente	El actor independiente es una especialización del Usuario General y se caracteriza por el hecho de que no sigue una planificación.
 Seguir planificación	Esta funcionalidad permite el seguimiento de la planificación creada, transformando un usuario independiente en uno seguidor.
 Seguidor	El actor seguidor es una especialización del Usuario General y se caracteriza por el hecho de que sigue una planificación.
 Abandonar planificación	Esta funcionalidad nos permite dejar de seguir una planificación, transformando un usuario seguidor en uno independiente.
 Ver sesión actual	Esta funcionalidad nos muestra la fecha de la sesión actual y los distintos parámetros de sesión, como distancia y ritmo recomendado.
 Ver sesión anterior	Permite ver la distancia recorrida y el tiempo transcurrido en la sesión anterior.
 Ver sesión siguiente	Permite ver la distancia a recorrer, el ritmo deseado y la fecha de la sesión siguiente.
 Ver trayectos recomendados	Permite obtener los trayectos recomendados relacionados con la sesión actual.
 Modificar planificación	Permite la modificación de los parámetros para la generación de una nueva planificación.
 Ver detalles trayecto	Permite obtener más datos sobre un determinado trayecto, como comentarios o distancia.
 Valorar trayecto	Permite realizar algún comentario sobre el trayecto acabado.



## Usuario General

Nombre	Valor
Visibilidad	public
Documentación	Este usuario representa una generalización de los dos tipos de usuarios existentes en el sistema: Independiente y Seguidor. Todos los casos de uso a los que él accede podrán ser accedidos por los dos tipos de usuario especializados.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 6:57:40 PM
Última modificación	Mar 7, 2012 7:45:14 PM
Modelo de negocio	falso
Actor ID	3

### Relaciones

Generalization	
A	Independiente
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 6:57:56 PM
Última modificación	Mar 7, 2012 7:25:14 PM
Sustituible	falso
Visibilidad	Sin especificar

Generalization	
A	Seguidor
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 6:58:26 PM
Última modificación	Mar 7, 2012 7:25:14 PM
Sustituible	falso
Visibilidad	Sin especificar



## Independiente

Nombre	Valor
Visibilidad	public
Documentación	El actor independiente es una especialización del Usuario General y se caracteriza por el hecho de que es el que no sigue una planificación.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:47:23 PM
Última modificación	Mar 7, 2012 7:45:14 PM
Modelo de negocio	falso
Actor ID	1



## Seguidor

Nombre	Valor
Visibilidad	public
Documentación	El actor seguidor es una especialización del Usuario General y se caracteriza por el hecho de que es el que sigue una planificación.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:47:41 PM
Última modificación	Mar 7, 2012 7:45:14 PM
Modelo de negocio	falso
Actor ID	2




## Crear planificación

Nombre	Valor
Estereotipos	Caso de uso
Documentación	A partir de los siguientes datos: tiempo de la última carrera, tipo de carrera y fecha del maratón y rango de kilómetros semanal, se creará una planificación inteligente basada en el nivel físico del usuario.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:51:32 PM
Última modificación	Jun 2, 2012 10:17:03 PM
Dificultad	Alta
Modelo de negocio	falso
Identificador CU	CU 1


### Relaciones

Include	
A	Ver planificación
Visibilidad	Sin especificar
Estereotipos	Include
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:56:01 PM
Última modificación	Mar 7, 2012 7:25:14 PM

Include	
A	Validar datos
Visibilidad	Sin especificar
Estereotipos	Include
Autor	Samuel Sandru
Fecha creación	Jun 2, 2012 10:11:10 PM
Última modificación	Jun 2, 2012 10:17:03 PM

Extend	
Desde	 Modificar planificación
Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 7:04:08 PM
Última modificación	Mar 7, 2012 7:25:14 PM

### Flujo de eventos

1. Usuario: Inicia el caso de uso
2. Sistema: Muestra el formulario con los datos a introducir: tiempo última carrera, tipo carrera, kilo metraje semanal y fecha maratón
3. Usuario: Introduce los datos
4. <b>if</b> ( (model element not found))
4.1. Sistema: Crea planificación
4.2. Sistema:  Ver planificación
<b>end if</b>

### Extension

4.a.
1. <b>while</b> !( (model element not found))
1.1. Sistema: Informar al usuario
1.2. Usuario: Modificar datos
<b>end while</b>
2. <b>jump to</b> <a href="#">4.1. Sistema: Crea planif...</a>

### Detalles

Nombre	Valor
Precondición	El usuario conoce los datos de la última carrera en la que participó, tiempo carrera y tipo carrera
Postcondición	Que se haya creado y mostrado la planificación
Autor	Samuel Sandru



## Ver planificación

Nombre	Valor
Estereotipos	Caso de uso
Documentación	El sistema le muestra al usuario una vista general de la planificación creada, que contendrá las fechas de entrenamiento, las distancias y ritmos propuestos.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:52:11 PM
Última modificación	Mar 7, 2012 7:15:14 PM
Dificultad	Alta
Modelo de negocio	falso
Identificador CU	CU 3

### Relaciones


Include	
Desde	Crear planificación
Visibilidad	Sin especificar
Estereotipos	Include
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:56:01 PM
Última modificación	Mar 7, 2012 7:25:14 PM

Extend	
Desde	Seguir planificación
Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Samuel Sandru
Create Date Time	Mar 7, 2012 1:05:38 PM
Last Modified	Jun 5, 2012 5:23:29 PM




Extend	
Desde	Abandonar planificación

Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Samuel Sandru
Create Date Time	Mar 7, 2012 7:06:36 PM
Last Modified	Mar 7, 2012 7:25:14 PM

### ***Flujo de eventos***

1.Sistema:  Crear planificación o Usuario: Inician el caso de uso
2.Sistema: Obtiene la planificación
3.Sistema: Muestra la planificación en una tabla de N(días) x M(semanas)

### **Extension**

3.a. Usuario:  Seguir planificación
3.b. Usuario:  Abandonar planificación
3.c. Usuario:  Modificar planificación

### ***Detalles***

<b>Nombre</b>	<b>Valor</b>
Precondición	Los parámetros de creación de la planificación son correctos, la planificación se ha creado.
Autor	Samuel Sandru



## Validar datos

Nombre	Valor
Estereotipos	Caso de uso
Documentación	El sistema valida los parámetros introducidos. Se verifica que el tiempo para la carrera elegida no supere el tiempo mínimo (record actual) ni el máximo (límite en la tabla VDOT).
Autor	Samuel Sandru
Fecha creación	Jun 2, 2012 10:11:09 PM
Última modificación	Jun 2, 2012 10:17:03 PM
Modelo de negocio	falso
Identificador CU	CU 24

### Relaciones

Include	
Desde	Crear planificación
Visibilidad	Sin especificar
Estereotipos	Include
Autor	Samuel Sandru
Fecha creación	Jun 2, 2012 10:11:10 PM
Última modificación	Jun 2, 2012 10:17:03 PM

### Flujo de eventos

1.Sistema: El caso de uso es iniciado por  Crear planificación
2.Sistema: Verifica la corrección de los datos
3. <b>if</b> (datos válidos)
3.1.Sistema: Finalizar caso de uso, valor afirmativo
<b>end if</b>

### Extension

3.a.
1. <b>if</b> !(datos válidos)
1.1.Sistema: Informar al usuario sobre los datos incorrectos



end if

### ***Detalles***


<b>Nombre</b>	<b>Valor</b>
Precondición	Que existan datos pendientes para la validación
Postcondición	Que el sistema haya decidido si los datos son correctos o no. En caso negativo, indicar cuáles son incorrectos
Autor	Samuel Sandru



## Guardar trayecto

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Esta funcionalidad nos permite guardar un trayecto para su posterior uso por parte de otros usuarios.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 6:09:11 PM
Última modificación	Mar 7, 2012 7:25:14 PM
Dificultad	Alta
Modelo de negocio	falso
Identificador CU	CU 21


### Relaciones

Extend	
Desde	 Valorar trayecto
Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Sonny
Fecha creación	Mar 7, 2012 6:09:24 PM
Última modificación	Mar 7, 2012 7:15:14 PM

### Flujo de eventos

1. Usuario: Inicia el caso de uso
2. Sistema: Muestra el trayecto actual junto con los datos inherentes: distancia y mapa de altitud
3. Sistema: Guarda el trayecto junto a los datos

### Extension


3.a. Usuario:  Valorar trayecto
1. <a href="#">jump to 3. Sistema: Guarda el c...</a>




## Ver trayectos

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Nos permite ver todos los trayectos correspondientes al área en el que nos encontramos. También nos permite ver los detalles de cada trayecto.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 1:14:03 PM
Última modificación	Mar 7, 2012 7:30:09 PM
Dificultad	Alta
Modelo de negocio	falso
Identificador CU	CU 10

### Relaciones

Extend	
Desde	 Ver detalles trayecto
Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 3:03:38 PM
Última modificación	Mar 7, 2012 6:55:13 PM

### Flujo de eventos

1. Usuario: Inicia el caso de uso
2. Sistema: Obtiene los trayectos relacionados con el contexto
3. Sistema: Muestra los trayectos en el mapa actual
4. Usuario:  Ver detalles trayecto

### Detalles

Nombre	Valor
Precondición	Existe conectividad de datos con la red móvil, de tal forma que los trayectos son obtenibles. El servicio GPS se encuentra encendido.
Autor	Samuel Sandru



## Cambiar idioma

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Esta funcionalidad nos permite realizar el cambio en la interfaz de usuario en función del idioma seleccionado
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 1:44:03 PM
Última modificación	Mar 7, 2012 7:30:09 PM
Dificultad	Media
Modelo de negocio	falso
Identificador CU	CU 18

### *Flujo de eventos*

1. Usuario: Inicia el caso de uso
2. Sistema: Muestra la lista de posibles idiomas
3. Usuario: Selecciona la opción deseada
4. Sistema: Efectúa el cambio al idioma seleccionado

### **Extension**

3.a. Usuario: La opción deseada no existe
1. Sistema: Finaliza el caso de uso



## Cambiar sistema medición

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Esta funcionalidad permite seleccionar entre el sistema métrico y el imperial
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 1:44:34 PM
Última modificación	Mar 7, 2012 7:30:09 PM
Dificultad	Media
Modelo de negocio	falso
Identificador CU	CU 19

### *Flujo de eventos*

1. Usuario: Inicia el caso de uso
2. Sistema: Muestra las dos opciones existentes
3. Usuario: Selecciona el sistema preferido
4. Sistema: Efectúa el cambio en el sistema de medición



## Conmutar control de ritmo

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Esta funcionalidad nos permite activar o desactivar la opción de seguridad que nos ofrece advertencias sobre el ritmo actual en relación con el recomendado
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 1:18:28 PM
Última modificación	Mar 7, 2012 7:30:09 PM
Dificultad	Media
Modelo de negocio	falso
Identificador CU	CU 16

### Flujo de eventos

1. Usuario: Inicia el caso de uso
2. <b>if</b> !ControlRitmoActivo
2.1. Sistema: Activar control de ritmo
3. <b>else if</b> ControlRitmoActivo
3.1. Sistema: Desactivar control de ritmo
<b>end if</b>
4. Sistema: Notificar al usuario sobre el cambio

### Detalles

Nombre	Valor
Precondición	La sesión está en curso para poder utilizar esta función
Autor	Samuel Sandru



## Ver trayecto actual

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Esta funcionalidad realiza una traza del trayecto ya realizado en la sesión actual.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 1:17:41 PM
Última modificación	Mar 7, 2012 7:30:09 PM
Dificultad	Alta
Modelo de negocio	falso
Identificador CU	CU 14

### *Flujo de eventos*

1. Usuario: Inicia el caso de uso
2. Sistema: Recoge los datos referentes que determinan el trayecto
3. Sistema: Muestra al usuario la traza del trayecto actual

### *Detalles*

Nombre	Valor
Precondición	La sesión está en desarrollo para poder realizar esta consulta
Autor	Samuel Sandru



## Comenzar sesión

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Esta funcionalidad debe coincidir con el inicio de la actividad física, es el momento a partir del cual se comenzará la monitorización de los parámetros y del trayecto.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 1:16:05 PM
Última modificación	Mar 7, 2012 7:30:09 PM
Dificultad	Alta
Modelo de negocio	falso
Identificador CU	CU 13

### *Flujo de eventos*

1. Usuario: Inicia el caso de uso
2. Sistema: Inicia la funcionalidad de monitorización de parámetros de sesión
3. Sistema: Inicia la funcionalidad de monitorización de trayecto
4. Usuario: Recibe los parámetros de sesión y trayecto a medida que se avanza en la sesión

### *Detalles*

Nombre	Valor
Precondición	Existe conectividad de datos con la red móvil. El servicio GPS está activado.
Autor	Samuel Sandru





## Ver parámetros de sesión

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Esta funcionalidad le muestra al usuario los parámetros de la sesión en curso.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 1:17:54 PM
Última modificación	Mar 7, 2012 7:30:09 PM
Dificultad	Alta
Modelo de negocio	falso
Identificador CU	CU 15

### *Flujo de eventos*

1. Usuario: Inicia el caso de uso
2. Sistema: Muestra los datos referentes a los parámetros de sesión

### *Detalles*


Nombre	Valor
Precondición	La sesión está en curso para poder mostrar la información deseada
Autor	Samuel Sandru



## Seguir planificación

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Esta funcionalidad nos permite el seguimiento de la planificación creada, transformando un usuario independiente en uno seguidor.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:52:23 PM
Última modificación	Mar 7, 2012 7:02:24 PM
Dificultad	Baja
Modelo de negocio	falso
Identificador CU	CU 4

### Relaciones

Extend	
Desde	 Ver planificación
Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 1:05:38 PM
Última modificación	Jun 5, 2012 5:23:29 PM

### Flujo de eventos


1. Usuario: Inicia el caso de uso
2. Sistema: Muestra al usuario la opción de seguir la planificación
3. Usuario: Elige la opción correspondiente al seguimiento de la planificación
4. Sistema: Modifica el modelo del usuario
5. Sistema: Termina el caso de uso



## Abandonar planificación

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Esta funcionalidad nos permite dejar de seguir una planificación, transformando un usuario seguidor en uno independiente.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:52:58 PM
Última modificación	Mar 7, 2012 7:15:14 PM
Dificultad	Baja
Modelo de negocio	falso
Identificador CU	CU 5

### Relaciones

Extend	
Desde	 Ver planificación
Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 7:06:36 PM
Última modificación	Mar 7, 2012 7:25:14 PM

### Flujo de eventos

1. Usuario: Inicia el caso de uso
2. Sistema: Muestra al usuario la opción de abandonar la planificación
3. Usuario: Elige la opción correspondiente a la desactivación del seguimiento
4. Sistema: Modifica el modelo del usuario
5. Sistema: Termina el caso de uso



## Ver sesión actual

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Esta funcionalidad nos muestra la fecha de la sesión actual y los distintos parámetros de sesión como: distancia y ritmo recomendado
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:53:24 PM
Última modificación	Mar 7, 2012 7:25:14 PM
Dificultad	Media
Modelo de negocio	falso
Identificador CU	CU 6

### Flujo de eventos

1. Usuario: Inicia el caso de uso
2. Sistema: Accede a la planificación
3. <b>if</b> día de entreno
3.1. Sistema: Muestra los datos referentes a la opción actual
<b>end if</b>

### Extension

3.a.
1. <b>if</b> ! día de entreno
1.1. Sistema: Muestra advertencia
<b>end if</b>



## Ver sesión anterior

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Nos permite ver la distancia recorrida y el tiempo transcurrido en la sesión anterior.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:53:33 PM
Última modificación	Mar 7, 2012 7:25:14 PM
Dificultad	Media
Modelo de negocio	falso
Identificador CU	CU 7

### *Flujo de eventos*

1. Usuario: Inicia el caso de uso
2. Sistema: Recuperar los parámetros de la sesión anterior
3. Sistema: Mostrar los parámetros de la sesión anterior y finalizar caso de uso

### *Detalles*

Nombre	Valor
Precondición	Se nos muestra la sesión actual relativa
Autor	Samuel Sandru



## Ver sesión siguiente

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Nos permite ver la distancia a recorrer el ritmo deseado y la fecha de la sesión siguiente
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:53:54 PM
Última modificación	Mar 7, 2012 7:25:14 PM
Dificultad	Media
Modelo de negocio	falso
Identificador CU	CU 8

### *Flujo de eventos*

1. Usuario: Iniciar el caso de uso
2. Sistema: Recuperar los datos referentes a la sesión siguiente
3. Sistema: Mostrar los datos al usuario y finalizar caso de uso

### *Detalles*

Nombre	Valor
Precondición	Se nos muestra la sesión actual relativa
Autor	Samuel Sandru




## Ver trayectos recomendados

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Permite obtener los trayectos recomendados relacionados con la sesión actual.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 1:14:30 PM
Última modificación	Mar 7, 2012 7:30:09 PM
Dificultad	Alta
Modelo de negocio	falso
Identificador CU	CU 11

### Puntos de extensión

Ver detalles trayecto	
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 3:19:34 PM
Última modificación	Mar 7, 2012 6:55:13 PM

### Relaciones

Extend	
Desde	 Ver detalles trayecto
Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 3:19:34 PM
Última modificación	Mar 7, 2012 6:55:13 PM

### Flujo de eventos

1. Usuario: Inicia el caso de uso
2. Sistema: Obtiene los parámetros recomendados de la sesión actual
3. <b>if</b> (existe sesión)
3.1. Sistema: Obtener trayectos recomendados para los parámetros encontrados

3.2.Sistema: Mostrar los trayectos
------------------------------------

3.3.Usuario:  Ver detalles trayecto
--

end if
--------

### Extension

3.a. !(existe sesión)
-----------------------

1.Sistema: Informar al usuario de la inexistencia de la sesión
--





## Modificar planificación

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Permite la modificación de los parámetros para la generación de una nueva planificación.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 12:51:47 PM
Última modificación	Mar 7, 2012 7:05:13 PM
Dificultad	Alta
Modelo de negocio	falso
Identificador CU	CU 2

### Relaciones

Extend	
Desde	Ver planificación
Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 7:04:08 PM
Última modificación	Mar 7, 2012 7:25:14 PM

### Flujo de eventos

1. Usuario: Inicia el caso de uso
2. Sistema: Muestra los parámetros anteriores correspondientes a la planificación
3. Usuario: Modifica los parámetros según sus preferencias
4. <b>if</b> (model element not found)
4.1. Sistema: Crea planificación
4.2. Sistema:  Ver planificación
<b>end if</b>

## Extensión

4.a.
1. <b>while</b> !( (model element not found))
1.1.Sistema: Informar al usuario
1.2.Usuario: Modificar datos
<b>end while</b>
2. <b>jump to</b> <a href="#">4.1. Sistema: Crea planificación</a>

## Detalles


Nombre	Valor
Precondición	El usuario no está conforme con la planificación creada
Postcondición	Se ha creado una nueva planificación y se le ha mostrado al usuario
Autor	Samuel Sandru




## Ver detalles del trayecto

Nombre	Valor
Estereotipos	Caso de uso
Documentación	En el caso de querer obtener más datos sobre un trayecto como comentarios o distancia, esta funcionalidad responde en tal situación
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 1:15:25 PM
Última modificación	Mar 7, 2012 7:02:24 PM
Dificultad	Media
Modelo de negocio	falso
Identificador CU	CU 12

### Relaciones

Extend	
Desde	 Ver trayectos
Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 3:03:38 PM
Última modificación	Mar 7, 2012 6:55:13 PM

Extend	
Desde	 Ver trayectos recomendados
Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 3:19:34 PM
Última modificación	Mar 7, 2012 6:55:13 PM

### Flujo de eventos

1. Usuario: Inicia el caso de uso
-----------------------------------

2.Sistema: Utilizando el identificador del trayecto actual, busca la información relacionada
--

3.Sistema: Muestra la información encontrada
--

### ***Detalles***


<b>Nombre</b>	<b>Valor</b>
Precondición	Existe más de un trayecto en el contexto geográfico actual
Autor	Samuel Sandru




## Valorar trayecto

Nombre	Valor
Estereotipos	Caso de uso
Documentación	Realiza algún comentario sobre el trayecto acabado.
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 6:09:24 PM
Última modificación	Mar 7, 2012 6:15:13 PM
Dificultad	Baja
Modelo de negocio	falso
Identificador CU	CU 22

### Relaciones

Extend	
Desde	 Guardar trayecto
Visibilidad	Sin especificar
Estereotipos	Extend
Autor	Samuel Sandru
Fecha creación	Mar 7, 2012 6:09:24 PM
Última modificación	Mar 7, 2012 7:15:14 PM

### Flujo de eventos

1.Sistema:  Guardar trayecto Inicia el caso de uso
2.Usuario: Provee el comentario sobre el trayecto
3.Sistema: Asigna al trayecto el comentario actual y finaliza el caso de uso

### Detalles

Nombre	Valor
Postcondición	El comentario del usuario tiene una longitud > 0
Autor	Samuel Sandru