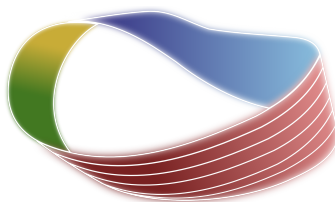




Universidad de Valladolid

E.U. DE INFORMÁTICA (SEGOVIA)

Ingeniería Técnica en Informática de Gestión



**Gesport:
Gestión de Entrenamiento Deportivo.**

Alumno: Diego Cebrián Martínez

Tutor: Juan Antonio Velasco Mate

*Dedicado Pedro, mi entrenador,
que con su trabajo hace llegar a la
excelencia deportiva desde la sombra*

Índice de contenido

1. Estructura de la memoria.....	7
2. Memoria del desarrollo.....	11
2.1 Breve introducción al origen del proyecto.....	13
2.1.1 Idea inicial.....	13
2.1.2 Estructura deportiva de los clubes.....	13
2.1.3 Otras funcionalidades adicionales.....	14
2.1.4 Las relaciones entre los usuarios potenciales.....	14
2.1.5 Software Libre.....	14
2.2 Consideraciones acerca del proceso de análisis.....	16
2.2.1 Ámbito de la aplicación.....	16
2.2.2 Teoría del entrenamiento.....	16
2.2.3 Objetivos del proyecto.....	17
2.2.4 El lenguaje de modelado unificado (UML).....	18
2.2.5 Metodología de análisis.....	19
2.2.6 Herramientas informáticas para el análisis.....	20
2.2.7 Consideraciones acerca del interfaz de usuario.....	20
2.3 Planificación y metodología de control de tiempos.....	22
2.3.1 Desarrollo temporal.....	22
2.3.2 Ciclo de vida.....	22
2.3.3 Metodología.....	23
2.3.3.1 Metodología SCRUM.....	23
2.3.3.2 Control de versiones.....	23
2.3.4 Herramientas informáticas de planificación y seguimiento.....	24
2.3.5 Presupuesto y financiación.....	25
2.4 Consideraciones acerca del proceso de diseño.....	27
2.4.1 Elección de la plataforma.....	27
2.4.1.1 Soluciones multiplataforma.....	27
2.4.1.1.1 Lenguajes mono-plataforma con interfaz de usuario común.....	27
2.4.1.1.2 Interfaz web a través de navegador.....	28
2.4.1.1.3 Tecnologías multiplataforma.....	29
2.4.1.2 La gestión de los datos.....	30
2.4.1.2.1 Gestores de Bases de Datos Orientados a Objetos.....	30
2.4.1.2.2 Gestores de Bases de Datos Relacionales.....	30
2.4.1.2.3 Gestores de Bases de Datos Relacional-Objeto.....	31
2.4.1.2.4 Sistemas intermediarios entre Base de Datos Relacional o Relacional Objeto y Programación Orientada a Objetos.....	31
2.4.1.3 El servidor.....	32
2.4.1.3.1 Windows.....	33
2.4.1.3.2 *NIX.....	33
2.4.1.5 La plataforma elegida.....	34
2.4.1.6 Herramientas informáticas para el diseño y la programación.....	35
2.4.2 El paradigma Modelo Vista Controlador.....	36
2.4.3 Diseño de la base de datos.....	39
2.5 Futuro de Gesport.....	40
2.6 Agradecimientos.....	40

3. Documentación técnica.....	43
3.1 Análisis del sistema.....	44
3.1.1 Glosario de términos deportivos.....	44
3.1.2 Análisis de Requisitos	45
3.1.2.1 Requisitos Funcionales.....	45
3.1.2.1.1 Gestión de entrenamientos.....	45
3.1.2.2 Requisitos no funcionales.....	47
3.1.3 Identificación de las clases de análisis.....	48
3.1.3.1 Clases candidatas.....	48
3.1.3.2 Clases del dominio del problema.....	49
3.1.4 Esquema básico de relaciones entre clases de análisis.....	49
3.1.5 Casos de uso.....	50
3.1.5.1 Actores.....	50
3.1.5.2 Descripción de los casos de uso.....	51
3.1.5.3 Diagrama de casos de uso.....	53
3.1.5.4 Escenarios concretos.....	54
3.1.5.4.1 Tipos de unidades.....	54
3.1.5.4.2 Unidades.....	56
3.1.5.4.3 Factores de conversión.....	58
3.1.5.4.4 Usuarios.....	60
3.1.5.4.5 Temporadas.....	62
3.1.5.4.6 Macrociclos	64
3.1.5.4.7 Mesociclos.....	66
3.1.5.4.8 Microciclos.....	68
3.1.5.4.9 Sesiones.....	70
3.1.5.4.10 Deportistas.....	72
3.1.5.4.11 Feedback.....	73
3.1.5.4.12 Entrada y salida.....	74
3.2 Diseño de la solución final.....	77
3.2.1 Diseño de la infraestructura (diagramas de despliegue).....	77
3.2.2 Diseño de las clases de objetos.....	78
3.2.2.1 Diccionario de clases de diseño.....	78
3.2.2.2 Diagramas de clases.....	90
3.2.3 Diseño del interfaz gráfico de usuario.....	92
3.2.4 Diseño de las pruebas.....	93
3.3 Implementación.....	94
4. Manuales de usuario y administrador.....	97
4.1 Manual de instalación y administración del administrador.....	99
4.1.1 Instalación del sistema operativo de servidor y los componentes necesarios para el funcionamiento de Gesport.....	99
4.1.1.1 Instalación del sistema operativo.....	99
4.1.1.2 Instalación de componentes.....	103
4.1.1.3 Configuración de Apache.....	104
4.1.2 Instalación de Gesport.....	104
4.1.2.1 Instalación de ficheros de Gesport y puesta en marcha de web.....	104
4.1.2.2 Configuración de la base de datos y creación de datos base.....	105
4.1.2.3 Configuración de DNS y acceso a la página.....	107
4.1.2.4 Personalizar Gesport.....	107
4.2 Manual de usuario.....	109

4.2.1 Acceso a la aplicación.....	109
4.2.1.1 Problemas con el acceso a la aplicación.....	110
4.2.1.1.1 Errores en la introducción de datos.....	110
4.2.1.1.2 Olvido de contraseña de acceso.....	110
4.2.2 Solicitud de acceso al programa.....	111
4.2.2.1 Problemas con la solicitud de acceso.....	112
4.2.3 Operaciones del usuario administrador.....	113
4.2.3.1 Operaciones con usuarios.....	113
4.2.3.1.1 Crear un usuario nuevo.....	113
4.2.3.1.2 Edición de usuarios registrados.....	115
4.2.3.1.3 Borrado de usuarios.....	115
4.2.3.2 Operaciones con tipos de unidades de medida.....	116
4.2.3.2.1 Crear un tipo de unidad nuevo.....	116
4.2.3.2.2 Edición de tipos de unidades.....	116
4.2.3.2.3 Borrado de tipos de unidades.....	117
4.2.3.3 Operaciones con unidades de medida.....	118
4.2.3.3.1 Crear una unidad de medida nueva.....	118
4.2.3.3.2 Edición de unidades de medida.....	118
4.2.3.3.3 Borrado de unidades de medida.....	119
4.2.3.4 Operaciones con factores de conversión entre unidades.....	120
4.2.3.4.1 Crear un factor de conversión nuevo.....	120
4.2.3.4.2 Edición de factores de conversión.....	120
4.2.3.4.3 Borrado de factores de conversión.....	121
4.2.4 Operaciones del usuario entrenador.....	122
4.2.4.1 Operaciones con deportistas.....	122
4.2.4.1.1 Crear deportista.....	122
4.2.4.1.2 Editar deportista.....	123
4.2.4.2 Operaciones de planificación y consulta de entrenamientos.....	124
4.2.4.2.1 Creación de periodos de entrenamiento.....	125
4.2.4.2.2 Edición de periodos de entrenamiento.....	126
4.2.4.2.3 Creación de sesiones.....	127
4.2.4.2.4 Visualización de resultados.....	128
4.2.5 Operaciones del usuario deportista.....	129
4.2.5.1 Consultar su entrenamiento diario	129
4.2.5.2 Reportar el resultado del entrenamiento.....	129
4.2.6 Operaciones comunes a todos los usuarios.....	131
4.2.6.1 Visualización y edición del perfil.....	131
4.2.6.2 Reporte de errores y sugerencias.....	132
5. Índices de tablas e ilustraciones del proyecto.....	135
5.1 Índice de Ilustraciones.....	137
5.2 Índice de tablas.....	140
6. Bibliografía.....	145
6.1 Sobre la teoría del entrenamiento deportivo.....	147
6.2 Sobre el análisis y diseño de software.....	147
6.3 Sobre el paradigma Modelo Vista Controlador.....	147
6.3 Sobre el lenguaje de programación PHP y el framework Symfony.....	147
6.4 Sobre la persistencia de los datos en el software.....	147
6.5 Sobre el diseño de interfaces gráficas web.....	147

6.6 Sobre los sistemas operativos y su manejo como administrador.....	148
7. Estructura de ficheros del CD.....	151
7.1 Código fuente de la aplicación.....	154
7.2 Documentación consultada para la realización del programa.....	154
7.3 Programas usados para la realización del programa.....	154
7.4 Logotipos del programa en formato vectorial y mapa de bits.....	155
7.5 Manual de usuario y de administrador.....	155
7.6 Manual del API de Gesport.....	155
7.7 Licencias de programas.....	155
7.8 Diapositivas de la presentación.....	155
7.9 Documentación del III Concurso Universitario de Software Libre.....	156

1. Estructura de la memoria

Descripción de la manera de estructurar la documentación

Gesport - Gestión de Entrenamiento Deportivo

La documentación del proyecto se ha estructurado en 7 partes diferentes para facilitar su comprensión.

La primera es esta descripción de la estructura.

La segunda corresponde con la memoria que describe el proceso de desarrollo del proyecto, el porqué de las decisiones, aspectos organizativos y otros documentos que han sido importantes en el proceso pero que no corresponden a los otros apartados.

La tercera parte es la documentación técnica. Esta documentación es toda la que nos permitiría continuar, modificar o entender la lógica del programa. Comprende los diagramas utilizados en las diferentes fases del desarrollo, así como la documentación del código resultante.

La cuarta parte son los manuales. Hay dos manuales, el de los usuarios y el del instalador. El administrador es un usuario normal con ciertos permisos más.

La quinta parte son los apéndices de tablas e ilustraciones de toda la memoria.

La sexta parte es la bibliografía que se ha utilizado en la documentación del proyecto. Esta bibliografía está ordenada por temas.

La séptima parte es una descripción de la estructura de ficheros del CD-ROM en el que se incluyen otros contenidos, como los binarios, los manuales de usuario y administrador de manera externa, documentos y manuales consultados o los ficheros de los diagramas. Además se ha descrito donde está cada conjunto de ficheros.

2. Memoria del desarrollo

Descripción de las motivaciones, pasos y procesos seguidos en el desarrollo

2.1 Breve introducción al origen del proyecto

2.1.1 Idea inicial

La idea inicial de este proyecto nace de la necesidad detectada por el alumno en su entrenamiento diario del deporte del atletismo. Esta necesidad viene de la carencia de herramientas de comunicación y gestión inteligente de la información recabada por el entrenador y ofrecida por el deportista. Además existen casos en los que entrenadores de atletas de gran nivel entrenan a distancia a sus deportistas, comunicándose mediante el teléfono y el correo electrónico, y teniendo que realizar ambos trabajos duplicados para poder llegar a un análisis correcto de los resultados.

Este programa debería tener la capacidad de controlar y gestionar todos los entrenamientos de los deportistas usando para ello la metodología de cada entrenador. También debería servir de herramienta de comunicación entre los entrenadores y los sujetos entrenados, de manera que un reporte de un entrenamiento de un deportista pueda ser consultado en cualquier momento por el entrenador sin necesidad de acudir al correo electrónico, teléfono u otro medio de comunicación, para luego introducirlo en un software de análisis de datos. Además debería proporcionar métodos para poder incluir módulos que amplíen las funciones del programa de una forma relativamente fácil. Las necesidades iniciales se atribuyen a un club de atletismo con varios entrenadores y datos que no se comparten entre ellos, pero este programa debería ser perfectamente válido para la gestión de cualquier club deportivo, ya sea este de fútbol o de tenis de mesa, haciendo el uso correcto de las extensiones.

2.1.2 Estructura deportiva de los clubes

En este programa nos vamos a ocupar meramente de la gestión deportiva y de los flujos de trabajo e información que relacionan a los sujetos entrenados con sus entrenadores. Si bien se hace esta pequeña introducción a la estructura interna de los clubes teniendo en cuenta la posibilidad de extensión del programa, así como la más que probable aplicación en ese ámbito en los principios del desarrollo. La relación entre entrenador y entrenado es una de las bases del organigrama deportivo y se podría construir una aplicación más completa partiendo de esta base.

Normalmente en un club existen diferentes personas que además desempeñan varios papeles dentro del mismo. Normalmente un entrenador suele ser también deportista en otra categoría o equipo, y a veces coincide que el mismo entrenador se entrena a si mismo.

Además de los diferentes entrenadores siempre hay alguien con la responsabilidad de tomar las decisiones, normalmente llamado director técnico. Esta persona tiene acceso a ciertos datos de todos los usuarios, sin entrar en los entrenamientos personalizados.

Muchas de las figuras que trabajan en club deportivo tienen acceso a datos confidenciales de otras. Los entrenadores a datos médicos de los deportistas, médicos o fisioterapeutas pueden tener informes que guardar. El programa debe cuidar también de conservar esta confidencialidad, e impedir que puedan producirse filtraciones.

Se estudiará la normativa española de obligado cumplimiento en protección de datos, así como los diferentes medios digitales para salvaguardar dicha ley en el funcionamiento del software desarrollado.

2.1.3 Otras funcionalidades adicionales

Aparte de la gestión deportiva los clubes realizan más actividades, pues la logística de los desplazamientos a competiciones, la organización de las mismas, la gestión de los espacios e instalaciones deportivas son también campos importantes a tener en cuenta.

Además la gestión de un club deportivo se puede extrapolar a otras organizaciones que tienen que ver con el deporte sin ser necesariamente clubes. Estos otros posibles destinatarios objetivos de la aplicación son los gimnasios, empresas organizadoras de eventos deportivos, ayuntamientos, empresas de gestión de recintos deportivos. Al tener la idea de ser un programa completamente modular, cualquier otra entidad que pueda necesitar de un software especializado podría beneficiarse de las características.

2.1.4 Las relaciones entre los usuarios potenciales

Para este proyecto se ha pensado en la relación entre el entrenador y el deportista a título individual, en el método de entrenamiento planificado y en la posibilidad de dar soporte a varios entrenadores. Los trabajos datos e informes de los entrenadores son secretos y solo el entrenador y el atleta interesado podrán verlos. De esta manera en un mismo club los diferentes entrenadores de distintos equipos podrían tener una plataforma de trabajo común.

Cuando un deportista en un deporte individual o colectivo termina el entrenamiento lo normal es que regrese a su casa y ya allí, tranquilamente, haga el análisis de su trabajo. Además el entrenador en el campo de trabajo suele tener los medios ni las capacidades para retener toda la información que le ofrecen sus pupilos. Esta condición hace que la aplicación deba poder ser ejecutada desde muchos lugares diferentes, como las diferentes casas de cada deportista, a pesar de tener los datos centralizados en un solo lugar.

No conocemos los sistemas operativos utilizados por los diferentes posibles clientes en sus instalaciones. Es por eso que el programa deberá funcionar, al menos en cliente, en casi cualquier sistema operativo, ya sea de ordenador o de instrumentos portátiles, como PDAs o móviles, así que deberá ser un sistema multiplataforma, ya sea a través de web con navegador, o utilizando tecnologías multiplataforma propiamente dichas, como entornos de desarrollo Qt, programación en .NET (con soporte para plataformas no Windows con Mono), o la opción más conocida del lenguaje Java, que funciona en casi todo tipo de dispositivos, desde ordenadores con diferentes sistemas operativos, Linux, Solaris, Windows, MacOS y dispositivos móviles.

2.1.5 Software Libre

Desde el principio de la realización del proyecto no se han tenido en cuenta las herramientas que se van a utilizar, si bien si que se ha decidido que la licencia tiene que ser alguna de las consideradas Software Libre. La razones de esta decisión son las siguientes:

- En la página 4 del documento de la normativa de proyectos de fin de carrera se especifica que la licencia debe ser tipo GNU o similar.
- Al ser un proyecto educativo desde su concepción, aunque pueda tener otros intereses en su forma final, tiene que poder servir para otros alumnos a aprender tanto de sus posibles virtudes como defectos.
- El autor de este proyecto está comprometido con el uso del software en libertad, tanto de uso, como de modificación, y conocimiento del mismo, y el tutor está de acuerdo en que se realice la distribución libre de este software.

Gesport - Gestión de Entrenamiento Deportivo

- El proyecto se presenta en el III Concurso Nacional de Software Libre, en la categoría de sistemas, pues no hay una categoría para el software de gestión.

2.2 Consideraciones acerca del proceso de análisis

2.2.1 *Ámbito de la aplicación*

En nuestra sociedad española los valores han cambiado mucho en los últimos 30 años. Hoy en día el deporte es visto como algo accesible a todos los públicos. La proliferación de instalaciones deportivas en todas las ciudades y en los más pequeños municipios indican que la práctica del deporte ha pasado a formar parte del común de la sociedad. Debido a esto los gestores de estas instalaciones han tenido que buscar los recursos adecuados para gestionar el buen uso y la organización de las mismas.

Asociado a este gran crecimiento de la práctica del deporte también se ha generado la necesidad de agruparse, además de en el deporte de equipo donde esto es obvio, para obtener ventajas o compartir costes en la instalaciones. Así se han generado federaciones, clubes y asociaciones relacionadas con las prácticas deportivas.

También ha crecido en un gran porcentaje el tiempo dedicado al deporte en los medios de comunicación, de manera que la gente se siente más cercana al deporte. Muchas personas toman como pasatiempo la práctica de diferentes modalidades deportivas esporádicas debido a esta influencia de los medios. En torno a esta nueva necesidad han surgido empresas que se dedican a gestionar actividades deportivas, organizarlas, o que han creado instalaciones vacacionales basadas en el ejercicio físico.

Las empresas, como organizaciones dedicadas a hacer dinero a cambio de servicios, han realizado diferentes sistemas para el control de muchos de estos aspectos, centrándose sobre todo en la gestión de instalaciones, generación de carnés de socios y almacenamiento de unos pocos datos como control de accesos y uso de máquinas. Estos usos de las aplicaciones solo tienen una vertiente económica y no entran a valorar más allá de si las personas tienen derecho a usar la máquina una vez más o si han venido 30 veces al mes a la instalación. Ignoran por completo otros aspectos como los que se refieren a la gestión de los entrenamientos o las comunicaciones entre el club y sus deportistas.

Gesport intentará cubrir ese hueco dejado en los ERP, enterprise resource planning o sistemas de planificación de recursos empresariales, o crear un nuevo concepto de programa: sistema de planificación de recursos deportivos, siempre partiendo de la base de la gestión de entrenamientos y teniendo esta planificación como prioridad y eje de toda la estructura de club.

2.2.2 *Teoría del entrenamiento*

A continuación se van a describir brevemente unos conceptos básicos de teoría del entrenamiento para que el resto del proyecto se comprenda con mayor facilidad.

El entrenamiento es un proceso por el cual se mejoran las capacidades del organismo para alcanzar determinados objetivos mediante unos trabajos realizados.

Solo con esta definición ya tenemos casi todo lo que vamos a estudiar:

- Proceso: se va a realizar a lo largo del tiempo. Una de las cosas más importantes en el entrenamiento es la correcta distribución del tiempo.
- Mejora: el organismo entrenado evoluciona, cambia a mejor.
- Capacidades: en la definición no se habla de capacidades físicas, sino solo de

Gesport - Gestión de Entrenamiento Deportivo

capacidades, se puede entrenar la mente, y el cuerpo.

- Objetivos: no hay entrenamiento sin un objetivo a cumplir.
- Trabajos: para alcanzar esta mejora hay que realizar esfuerzos en la dirección correcta.

A partir de esta definición y con estos elementos cada entrenador trabaja diferentes cosas con sus deportistas en aras de alcanzar esos objetivos propuestos. Para llegar a "adivinar" que tiene que hacer cada sujeto entrenado en cada momento utiliza unas estructuras temporales en las que se trabajan objetivos menores. Estas estructuras son las siguientes:

- Temporada: Suele coincidir con un año lectivo y acabar uno días después de alcanzar o no el objetivo marcado.
- Macro ciclo: Es una subdivisión que se produce en la temporada para realizar trabajos de diferentes tipos. En atletismo se dividen las temporadas en pista cubierta o cross y pista, en fútbol en pretemporada y temporada, y en cada deporte en los tiempos apropiados, según el entrenador, para que la consecución de los objetivos sea más asequible.
- Mesociclo: Divide cada uno de los mesociclos en diferentes fases en las que se evoluciona hacia el objetivo menor marcado para el macro ciclo que las contiene.
- Micro ciclo: Suele considerarse un periodo corto de tiempo donde se introducen las sesiones, de manera que permite un control adecuado.

Así pues, independientemente de la especialidad que se quiera trabajar, el planteamiento de la estructura organizativa temporal es la misma en todos los deportes.

Cuando un entrenador planifica sabe que es lo que quiere conseguir (objetivos), y el como se mide en unidades de volumen e intensidad. El volumen es la cantidad de trabajo realizado (toneladas levantadas, kilómetros corridos...) y la intensidad es el porcentaje del máximo esfuerzo posible con que se realiza. Para la medida de la intensidad se trabaja con intervalos, y se define la cantidad del volumen que se ha trabajado en ese intervalo de intensidad.

La unidad básica del entrenamiento es la sesión. Suele asociarse al trabajo de un día, pero puede haber varias sesiones el mismo día. Cada sesión se compone de varios elementos (calentamiento, estiramientos, trabajos concretos, etc.) que el entrenador describe al deportista para que los ponga en práctica.

Más allá de esta pequeña introducción son los entrenadores los que haciendo uso de sus conocimientos pueden sacar más provecho a la aplicación, definiendo sus intervalos, periodos, ejercicios y sesiones.

2.2.3 Objetivos del proyecto

El objetivo de este proyecto es crear un aplicación modular y extensible que posibilite la gestión de los aspectos deportivos y que permita añadir posteriormente los logísticos, económicos, sociales, legales y organizativos de cualquier organización, ya sea empresa, asociación o club, que tenga que ver con el deporte.

Otro objetivo del proyecto es utilizar correctamente todas las herramientas de análisis, diseño, planificación, desarrollo y pruebas que sean necesarias para culminar con éxito con el menor esfuerzo posible (o lo que es lo mismo avanzar más en el desarrollo en el menor tiempo posible)

2.2.4 El lenguaje de modelado unificado (UML)

El lenguaje de modelado unificado es un lenguaje (gráfico o descriptivo) que se usa principalmente para visualizar, especificar, construir y documentar componentes de software y sistemas de software. El UML no es de uso exclusivo del software ya que con el mismo lenguaje se pueden documentar gráficamente todo tipo de sistemas, desde empresariales hasta de cualquier ingeniería. El UML permite el modelado del sistema completo como modelo en diferentes diagramas, que pueden ser de diferentes tipos.

El lenguaje de modelado unificado no es un método, tan solo es un lenguaje que permite expresar con los conceptos adecuados el resultado de un análisis llevado a cabo con una metodología concreta.

El UML no interviene solo en la fase de diseño de software, a la hora de establecer la clases, objetos, relaciones y contenidos, sino que a través de los diferentes tipos de diagramas también se usa en el análisis, con los diagramas de casos de uso, y en la implementación, si el software usado es el apropiado permite la generación de las clases para facilitar la tarea del programador. Aunque este último caso no sea propiamente UML hoy en día casi todas las herramientas de ingeniería de software propietarias o libres proveen esta característica por defecto.

Los tipos de diagramas que componen el modelo software se dividen en diagramas de estructura, de comportamiento, y de interacción.

Los diagramas de estructura son los que describen los diferentes elementos que hay en el modelo, que relaciones tienen entre ellos, y sus características. Para el desarrollo de Gesport se van a usar los siguientes diagramas de estructura:

- Diagramas de clases, que consisten en la representación de las diferentes clases de los objetos del dominio, sus características (métodos y atributos), y sus relaciones con las otras clases.
- Diagrama de componentes, que va a describir los diferentes elementos reales que van a estar en el sistema así como la interfaz para trabajar con ellos.
- Diagrama de despliegue, que representa los objetos físicos del programa y que realizan cada uno de los componentes que están en ellos.

Los diagramas de comportamiento son los que representan que es lo que sucede. Esto conlleva flujos de información, y relaciones entre los elementos que activan actividades de los elementos destinatarios. En este proyecto se han usado:

- Diagrama de casos de uso: donde se describen a los diferentes actores y casos de uso, funcionalidades del sistema que pueden formar parte de un proceso más largo, que entran en contacto con el sistema modelado y que mensajes se transmiten entre ellos. En alguna bibliografía se desaconseja su utilización para el modelado de objetos, pues distancia el modelo de la vida real del futuro software.
- Diagrama de actividades: que representa en el tiempo, marcando un principio y un final, un proceso completo del dominio del problema.

Los diagramas de interacción son una extensión de los de comportamiento para realzar el flujo de control y de datos entre los diferentes elementos. El que más interés tendría en este proyecto es el diagrama de secuencia, pero es una extensión u otra manera de ver el diagrama de actividades (donde cada actividad se corresponde a un estado, y lo que aporta el de interacción son los mensajes que hay entre los diferentes objetos que cambian de estado), por lo que no se va a desarrollar. En los diagramas de actividades que se van a desarrollar se hará de manera que, sin paso de mensajes al estilo de los más avanzados de interacción, se vean actor y elemento.

Con un buen modelado en todos los aspectos tendremos una gran base para construir un programa sólido y coherente tanto con las necesidades de los usuarios, como con una estructura interna fácil de mantener y ampliar.

2.2.5 Metodología de análisis

Para la realización del análisis de las posibles necesidades organizativas, técnicas y económicas se han realizado una serie de entrevistas con Pedro Lorenzo. Actualmente Pedro es el Director Técnico del Club Atlético Joaquín Blume - Caja-Segovia. Es Licenciado en Ciencias de la Actividad Física y del Deporte por la Universidad Europea de Madrid con la especialización en Atletismo y Golf, Entrenador Nacional de Atletismo por la Real Federación Española de Atletismo y monitor de diferentes disciplinas de ocio y deportivas como Natación, Tiempo Libre, Piragüismo, Método Pilates y Polideportivo. Además ha realizado cursos de especialización en Dietética y Nutrición, Gestión de Instalaciones Deportivas, Esquí, Fútbol, Fútbol Sala, Aquatic Fitness y Fisiología del Ejercicio y ha publicado artículos en diferentes revistas especializadas. Por todo ello, y por su dilatada experiencia laboral en diferentes clubes deportivos como el Real Club de Golf Puerta de Hierro, el Club Atlético Joaquín Blume o el Velox Atletismo creo que es una persona apropiada y que conoce suficientemente los entresijos de este dominio que vamos a abarcar en el proyecto.

Las entrevistas se realizaron en tres fases diferentes.

En la primera se definieron las diferentes necesidades del personal técnico y deportistas a nivel deportivo. En esta fase se definieron los diferentes documentos que intervienen en la planificación deportiva, sus fases y periodos, los datos necesarios en cada periodización y la información que el sujeto debe transmitir al entrenador en forma de realimentación para que este pueda adaptar los futuros entrenamientos. También se estableció el glosario de términos deportivos para aclarar las definiciones que se iban a tratar durante el desarrollo de la aplicación y se tradujo al inglés para hacer el código fuente de la aplicación accesible y extensible a cualquier programador del mundo.

Tras esta entrevista se seguirá una metodología de análisis orientada a objetos. La primera parte consistirá en una especificación de requisitos, donde se van a recoger todos los requisitos funcionales y no funcionales que va tener la aplicación. Estos requisitos se reflejarán en diagramas de casos de uso. De esta manera al finalizar el proyecto podremos comprobar que todas las necesidades marcadas en un principio se están cumpliendo.

Una vez terminada esta fase se comenzó a hacer el análisis mediante diagramas de casos de uso, y, en colaboración con el "cliente" (Pedro Lorenzo), se encontraron los errores de concepto mediante la revisión de la redacción de los requisitos realizada por el analista, y la modificación de los casos de uso que no estén realizados correctamente.

Sobre estos requisitos se estudiarán las clases y objetos que van a intervenir en ella. A este proceso se le llama identificación de las clases del dominio. Para este proceso se utilizará la técnica de la identificación de sustantivos. Consiste en extraer los nombres y grupos nominales de la descripción del problema y las entrevistas con los usuarios, expertos en el dominio y clientes.

Sobre estos nombres obtenidos se realizará una criba donde se eliminarán los que estén duplicados en significado o que representen lo mismo, los que no intervengan realmente como elementos del problema a resolver, los elementos propios del sistema software que vaya a tratar el problema, pantalla, teclado, etcétera, y los que indiquen frecuencia temporal. Además de estos que se eliminan otros tendrán que transformarse pues pueden ser atributos de una clase, en vez de una clase propiamente dicha, representar una acción, que en el sistema se convertirán en métodos, o un estado, que será un valor de un atributo.

2.2.6 Herramientas informáticas para el análisis

Como hemos comentado anteriormente vamos a realizar un análisis exhaustivo de la aplicación que vamos a desarrollar. Se han definido la metodología y el lenguaje que se van a usar en este proceso. Para ayudarnos a tener controlados todos los parámetros que vana pareciendo vamos usar una herramienta Case que nos permita trabajar con los diagramas descritos en el UML.

Las herramientas posibles eran:

- Umbrello: Software libre para el trabajo con UML para el entorno gráfico KDE que utiliza la librerías Qt y que funciona, en teoría, en cualquier sistema operativo.
- Rational Software Modeler de IBM: Software para el trabajo con UML y gestión de proyectos. Funciona en Linux y Windows. Este software tiene un coste de licencia de 1200€ al año.
- ArgoUML de Tigris: Software libre realizado en Java para el trabajo con UML. Funciona en todas las plataformas donde haya una versión de Java disponible. Tiene multitud de extensiones. Se puede integrar en Eclipse IDE. El punto débil es la cantidad de diagramas que soporta y que está basado en UML 1.4.
- UML para Netbeans, de SUN Microsystems: Software libre integrado en el entorno de desarrollo Netbeans. Soporta multitud de diagramas, está basado en Java y se ejecuta en cualquier ordenador con Java. No tiene soporte de importación ni exportación de ficheros XMI, que permiten el intercambio entre diferentes herramientas Case de UML.
- Dia: Editor de diagramas generalista de Gnome, que funciona en otras plataformas gracias a las librerías GTK portadas. No soporta XMI.

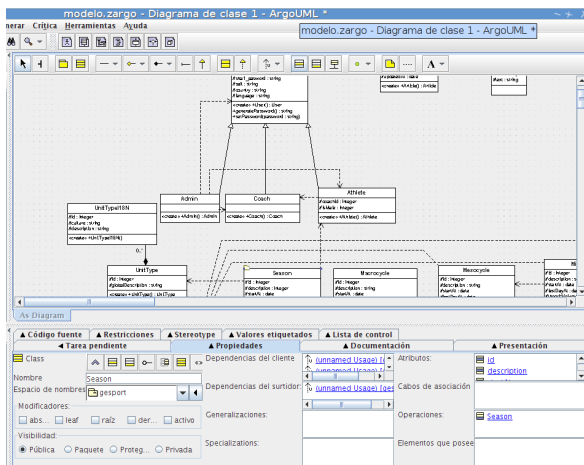


Ilustración 1: Detalle del funcionamiento de ArgoUML

En un principio se eligió Umbrello por ser software libre, soportar XMI y basarse en Qt, ya que el ordenador donde se ha desarrollado el proyecto funciona con KDE sobre Linux y de esta manera, por compartición de librerías, se podía ahorrar memoria. El trabajo con este programa resulta muy intuitivo, pero pero la inestabilidad que hacía que se quedara colgado, y un par de pérdidas de información forzaron el cambio.

Tras este primer intento se probó ArgoUML. El manejo era diferente, con un solo clic para todas las acciones, un sistema de tabulaciones y botón derecho diferentes a los que otros programas nos tienen acostumbrados. Cuando aprendí a manejar la barra inferior de propiedades

el trabajo fue más rápido aún que con Umbrello, avanzando además en la exportación de tablas y diagramas con las extensiones. Facilitó mucho el trabajo.

2.2.7 Consideraciones acerca del interfaz de usuario

Este programa tiene como objetivo el facilitar la gestión de los aspectos deportivos a un club deportivo, entrenador o deportista. Así pues el manejo del mismo tiene que ser lo más sencillo

e intuitivo posible. Para que resulte sencillo se utilizará un sistema de ventanas gráfico como puede ser el de Windows, Qt, Gtk, alguno de los múltiples posibles para Java (AWT, Swing con integración en escritorio) o el interfaz web del navegador. La elección del sistema de ventanas vendrá determinada por sistema base que soportará la aplicación.

Para que sea intuitivo se estudiarán las operaciones de trabajo en el campo de los profesionales de forma que la manera de introducir y obtener los datos sea parecida a la que se realiza en vivo y en directo en la instalación deportiva.

Además se tendrán en cuenta las recomendaciones de usabilidad recogidas en las diferentes organizaciones que tratan sobre el tema, como en las normas de GUI de Gnome Foundation, las recomendaciones para el diseño de aplicaciones Java, o las especificaciones WAI de la W3C de accesibilidad web para personas con discapacidad. El seguimiento de unas recomendaciones u otras dependerá del interfaz finalmente utilizado.

2.3 Planificación y metodología de control de tiempos

2.3.1 Desarrollo temporal

El tiempo estimado de desarrollo de esta aplicación son 9 meses realizada por una sola persona con una dedicación de 7 horas al día.

La primera parte de este tiempo (1 mes) corresponde a la realización de estudios previos de mercado y de funcionamiento de las organizaciones para las que está destinada Gesport. Durante este periodo se han entrevistado a directivos, entrenadores, deportistas y trabajadores de clubes deportivos y otro personal relacionado con el deporte (personal Centro Regional de Medicina Deportiva, técnicos de instituciones como la Diputación de Segovia). Se han estudiado las necesidades anteriormente detalladas.

La segunda fase (3 meses) ha consistido en plasmar todo esa información obtenida en la redacción del análisis, en la creación de los diferentes gráficos y esquemas que provee el Lenguaje de Modelado Unificado para facilitar una cómoda codificación de la aplicación. Se han seguido manteniendo las entrevistas para corregir los posibles errores en los conceptos iniciales. En esta parte se han aportado al proyecto todos los diagramas de Clases y Actividades. Aparte de los gráficos propiamente dichos se han desarrollado las explicaciones de los términos más complejos de temas deportivos y de las relaciones en los diagramas que pudieran no llegarse a entender. Se ha hecho el estudio de las soluciones posibles para los problemas planteados y se han tomado las decisiones relativas a metodología y tecnología de clientes y servidores a usar. En esta etapa también se han diseñado las pruebas a las que se habría de someter cada parte de la aplicación y la aplicación en general.

La tercera fase (4 meses) ha consistido en la implementación y escritura de todos los códigos fuentes necesarios, búsqueda de bibliotecas que ya proveyeran de funciones necesarias (evitar inventar la rueda), diseño de los interfaces gráficos de usuario y pruebas de conexiones entre bases de datos, servidor de aplicaciones e interfaces de usuarios. En esta parte se ha ido avanzando en ciclos por objetivos.

El último mes se ha dedicado a completar la presentación de diapositivas, los manuales de usuario y administrador, terminar detalles y flecos sueltos, y probar más combinaciones posibles que puedan generar error para evitar efectos indeseados en una presentación ante el jurado o un hipotético cliente interesado en adquirir servicios sobre el programa.

Durante todo este periodo de tiempo el alumno ha dedicado una media de 6 horas al día, todos los días de la semana y los fines de semana y festivos 2 o 3 horas.

2.3.2 Ciclo de vida

Se ha utilizado un desarrollo en espiral. Aunque se estudió la posibilidad de realizarlo en cascada por que se tenía conocimiento de todas las áreas del proyecto desde el estado inicial, se ha creído más apropiado hacer un ciclo de la espiral con cada funcionalidad añadida al programa. De esta manera tendremos versiones funcionales de partes del programa en cada ciclo que podrán ser utilizadas por los miembros del club mientras se sigue el desarrollo de otras áreas.

El orden a seguir en esta metodología ha sido el siguiente:

- Generación de estructura de datos maestros: unidades y factores de conversión.
- Sistema de acceso de usuarios.

- Construcción de las estructuras de que engloban a las sesiones (macro, meso y microciclos).
- Sistema gestor de sesiones de entrenamiento para entrenador.

2.3.3 Metodología

2.3.3.1 Metodología SCRUM

Durante el desarrollo se intentará utilizar una técnica parecida a SCRUM.

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutara durante un proyecto.

Los roles están divididos en "cerdo" y "gallina". Los "cerdo" son aquellos que están comprometidos con el proyecto, en este grupo entra el cliente, el equipo de desarrollo y una figura que hace de coordinador. Los "gallina" son los que no forman parte del proceso de producción del software pero pueden tener algo que decir. Estos son los usuarios finales o los clientes y proveedores.

En un periodo de tiempo de 15 a 30 días llamado Sprint el equipo crea un incremento de software *potencialmente entregable*. Al principio de cada Sprint hay una reunión entre todos para definir los requerimientos del siguiente Sprint. Cada día hay una reunión donde se informan entre los miembros del equipo de los logros y dificultades del día, y del plan para el día siguiente.

En la metodología de Gesport no se va a hacer con las reuniones como marca el estándar, pues tan solo hay un desarrollador, pero si en las descripciones diarias de los planes para el día siguiente y logros y dificultades de hoy.

El ciclo de vida en espiral es aplicable perfectamente a este método de trabajo, haciendo coincidir cada ciclo de la espiral con un Sprint.

2.3.3.2 Control de versiones.

Para gestionar el control de las versiones se va a usar el software Subversion, de manera que cada día o cada cambio importante que se haga supondrá un commit o envío al sistema de control de versiones. Cada uno de los envíos a este sistema creará una revisión nueva del estado del proyecto en ese momento.

Los documentos como la presente memoria, los diagramas realizados con la herramienta Case para el análisis y el diseño del software también estarán entre los ficheros controlados con Subversion, de manera que el sistema de control de versiones será compartido por los documentos, el código fuente y el resto de elementos.

Todos los envíos de cambios se realizarán en la rama principal de desarrollo "Trunk", quedando unas ramas secundarias "Branches" y "Tags" reservadas para otras tareas.

Además de las versiones diarias, o periódicas en cada finalización de Sprint o vuelta de espiral se publicará una versión funcional en los términos decididos al comienzo de la evolución. Estas versiones tendrán el formato 0.x, siendo x un entero incremental. No se llegará hasta la versión 1.0, considerada estable y completa, hasta que no se asegure el cumplimiento de todas las funciones establecidas. En estas funciones se incluyen las contenidas en las posibles mejoras del software, ya que en las reuniones previas con los técnicos deportivos se trataron como requerimiento. El índice principal de la versión se llamará *n* y cambiará con los cambios

estructurales grandes, como cambios en el lenguaje de programación o reescritura de gran parte del código. Nunca cambiará como consecuencia de un añadido pequeño de funcionalidades.

Estas versiones *n.x* serán tratadas en el servidor de control de versiones de manera especial, siendo separadas de la rama principal de desarrollo. Se crearán bajo la rama secundaria "Tags", y se introducirán en una carpeta con el nombre de la versión "*vn.x*".

En el caso de hubiera que modificar una versión *n.x* por cualquier motivo (actualización de seguridad, fallo o error de programación subsanado, etc.) se procederá a su denominación *n.x.y*, siendo *y* incremental desde 1. Se procederá a su publicación en "Tags", no como sustitución de la primera, sino en una nueva carpeta "*vn.x.y*"

2.3.4 Herramientas informáticas de planificación y seguimiento

Para llevar a cabo esa metodología se va a contar con las siguientes herramientas:

- El uso de una forja de software (en este caso forja.rediris.es) para tener control de las tareas pendientes, sugerencias de los "gallina", etc....,

- El servidor del control de versiones no ha definido la metodología de versiones de trabajo y lo autores del proyecto hemos definido la metodología de liberación de versiones estables. El servidor de Subversion es público y está alojado en la forja de rediris al igual que las otras herramientas de control. Para consultar la evolución de las diferentes versiones puede hacerlo en https://forja.rediris.es/scm/?group_id=431.

- Se ha usado un sistema de seguimiento y control de horas (eGroupWare, en este caso) con el correo integrado para poder hacer un análisis correcto del tiempo utilizado en cada cosa. Los diferentes aspectos de la forja eran monitorizados, de manera que cualquier correo que llegase a un foro, a una lista de distribución o a una notificación de errores llegaba a este sistema de control de manera automática. En este programa se asignaba el correo de la notificación como una tarea pendiente de resolver. Del proyecto de fin de carrera.

Aplicación	Título	Recursos	Fecha de Inicio Planificado	Fecha de finalización Planificado	Tiempos Planificado	Presupuesto Planificado	Estado	unknown type %L Modificado por	Acciones
	P-2009-0001: Gesport, software para el deporte	Diego Cebrían [dcebrían]	2009/03/04 2009/03/04	2009/03/19 2009/03/31	7.631 d 25 d	1508.90		2009/03/20 02:38 Diego Cebrían [dcebrían]	
	2009/03/19 15:00: Diagramas de clases y documentación	Diego Cebrían [dcebrían]	2009/03/19 2009/03/19	2009/03/19 2009/03/19	2.483	62.08	✓	2009/03/19 17:29 Diego Cebrían [dcebrían]	
	2009/03/19 12:00: Redacción memoria: diagramas de despliegue	Diego Cebrían [dcebrían]	2009/03/19 2009/03/19	2009/03/19 2009/03/19	3.5	87.50	✓	2009/03/19 15:02 Diego Cebrían [dcebrían]	
	2009/03/18 23:00: P-2009-0001: Proyecto fin de carrera	Diego Cebrían [dcebrían]	2009/03/18 2009/03/18	2009/03/19 2009/03/19	4.5	112.50	✓	2009/03/19 13:32 Diego Cebrían [dcebrían]	
	2009/03/18 20:00: Diseño de aplicación para documentación	Diego Cebrían [dcebrían]	2009/03/18 2009/03/18	2009/03/18 2009/03/18	1.5	37.50	✓	2009/03/18 23:30 Diego Cebrían [dcebrían]	
	2009/03/18 15:00: Diagramas varios y documentación	Diego Cebrían [dcebrían]	2009/03/18 2009/03/18	2009/03/18 2009/03/18	3	75.00	✓	2009/03/18 20:39 Diego Cebrían [dcebrían]	
	2009/03/18 12:00: Diagramas de clases y	Diego Cebrían [dcebrían]	2009/03/18 2009/03/18	2009/03/18 2009/03/18	2.34	56.34	✓	2009/03/18 15:24 Diego Cebrían [dcebrían]	

Ilustración 2: Vista de tareas de proyecto de Egroupware

- Se ha realizado un blog donde se cuentan las diferentes maneras de resolver los problemas que se han planteado durante el desarrollo.

2.3.5 Presupuesto y financiación

Para el desarrollo de esta aplicación se ha estimado un coste por hora de 45€ en los trabajos de ingeniería técnica (análisis y diseño) y de 30€ en los trabajos técnicos (programación y pruebas). Se excluye de este presupuesto la instalación y mantenimiento. Aún siendo el mantenimiento la fase de producción de software más longeva y las puestas en marcha las fases más críticas y propensas a fallos, serían las fuentes de financiación directas para costear el desarrollo del programa y se estudiaría cada cliente en particular. En las diferentes fases de la creación del programa se ha contado con el siguiente presupuesto.

Concepto	Cantidad	Coste detalle	Coste Total
- Horas de análisis:	50h		2.250€
*Entrevistas con cliente:	20h	900€	
*Análisis de los requisitos:	30h	1.350€	
- Horas de diseño del software:	40h	1.800€	1.800€
- Horas de diseño de pruebas:	16h	720€	720€
- Horas de diseño de la interfaz:	32h	1.440€	1.440€
- Horas de programación:	60h	1.800€	1.800€
- Horas de pruebas:	20h	600€	600€
- Hardware necesario para pruebas:			2.050€
* PDA	1 unidad	250€	
* Ordenador personal para el desarrollo de la aplicación + garantía ampliada de 3 años	1 unidad	1.800€	
* Servidor para pruebas ¹		0€	
* Servidor de producción para versiones finales en funcionamiento ²		0€	
- Servidores de control de versiones, copia de seguridad y distribución ³		0€	0€
Total			10.660€

Tabla 1: Presupuesto global de desarrollo

El proyecto, según la normativa de la Universidad de Valladolid, será licenciado como software libre al amparo de la GPL v.3. El tiempo de desarrollo del proyecto se considera una inversión para dar un posterior mantenimiento y servicio personalizado a clubes, gimnasios y entrenadores personales. De esta manera la financiación se obtendrá de la repercusión del coste del mantenimiento e instalaciones del software en los potenciales clientes que puedan hacer uso de el,

- 1 Se utilizó el PC de escritorio con el sistema libre de virtualización Qemu y un Sistema operativo cliente de servidor Ubuntu Server para realizar las pruebas.
- 2 No se ha realizado ninguna instalación de aplicación en entorno real. Tan solo un servidor en pruebas en las aulas de la Escuela de Informática de Segovia.
- 3 Alojado en la forja de software de Rediris. Servidor para desarrolladores de software libre y comunidad de los mismo para el intercambio de ideas y colaboración entre proyectos.

así como de las ventas e instalaciones de hardware en los locales de los mismos. Se prevé una amortización total del coste de la inversión en 1 año y 6 meses, con una cartera inicial de 2 clientes (con precio especial por haberse propuesto para la realización de las pruebas y puesta en marcha) y una cartera final del primer año de 7. Se estudiará, según el entorno económico, si es conveniente afrontarlo en tan solo un ejercicio.

2.4 Consideraciones acerca del proceso de diseño

2.4.1 Elección de la plataforma

2.4.1.1 Soluciones multiplataforma

Uno de los requisitos no funcionales de la solución final es que el programa del cliente funcionase en varias plataformas. El motivo de esto era que en una organización concreta suele usarse la misma plataforma en todos los ordenadores, pero la aplicación debe ser exportable a futuros posibles clientes. Si bien la mayoría de los usuarios del posible mercado en el que nos encontramos usa sistemas operativos Microsoft Windows ®, en el caso concreto planteado del Club de Atletismo Joaquín Blume existen al menos dos puestos en los que el sistema operativo es Linux, MacOS y terminal móvil. Ante este problema se han barajado las siguientes soluciones:

2.4.1.1.1 Lenguajes mono-plataforma con interfaz de usuario común

Consiste en crear la aplicación con un interfaz de usuario basado en librerías libres utilizables en cualquiera de estos sistemas operativos de escritorio. El lenguaje de base es mono-plataforma, pero las librerías que usa son comunes a varios sistemas operativos, de manera que el mismo código fuente con diferente compilador (para cada sistema operativo) darían lugar al mismo programa.

El que sean libres viene dado por tres razones principalmente:

- El coste del uso de librerías propietarias.
- El soporte por una comunidad de usuarios y desarrolladores de estas librerías muy involucrados por su voluntariedad.
- Por último, y no menos importante, por el aporte de componentes al movimiento del software libre que puede generar el proyecto.

Estas posibles librerías de interfaz de usuario son GTK (Gimp Tool Kit) o Qt.

- GTK es una implementación de un sistema de ventanas en las que se basa el entorno de ventanas para sistemas UNIX Gnome. Comenzó su desarrollo por los creadores de Gimp, un software de edición gráfica de mapa de bits para sistemas UNIX. Hoy en día tiene capacidades parecidas a Adobe Photoshop ® gracias a las extensiones programables en diferentes lenguajes, aunque con una velocidad de ejecución más lenta debido a que estas extensiones se hacen con lenguajes interpretados. La base de la programación para GTK es C, aunque tiene extensiones para C++ y últimamente ha salido, a través del proyecto Mono, para C#. Esta última manera de programas se considerará en las plataformas interpretadas, puesto que, al igual que con Java, una misma compilación sirve para los tres sistemas operativos.

- Qt es una implementación de una empresa privada llamada Trolltech. En sus inicios tenía licencia privativa, pero su uso para el desarrollo del entorno de ventanas KDE, también para sistemas UNIX y actualmente funcionando en Windows y en Mac sustituyendo a los sistemas por defecto de estos sistemas operativos, hizo que al final Trolltech liberara el código con un doble licenciamiento: si la aplicación que usa Qt es libre el licenciamiento es libre, si la aplicación tiene una licencia no libre entonces hay que pagar para usar las librerías. El lenguaje que se usa con Qt es C++, que permite orientación a objetos y supone una ventaja en el desarrollo sobre GTK.

Es posible que existan otras alternativas, pero estas son las más conocidas, más usadas y con más soporte de la comunidad. Existe la posibilidad de usar GTK+ que es GTK orientado a objetos. El uso de estas tecnologías tiene ciertas ventajas e inconvenientes.

- Ventajas: Se basan en lenguajes compilados, lo que implica una mayor velocidad de ejecución, tienen un interfaz común en todos los sistemas operativos, lo que simplifica, tanto los manuales de usuario, como el aprendizaje de los usuarios,

- Desventajas: Al ser lenguaje compilado necesita una compilación por cada sistema operativo destino. GTK en MacOS solo funciona con el servidor X de UNIX instalado. Al tener la base de datos en un lugar diferente al programa, que se conectaría remotamente, el trabajo sobre ella será lento y en las transferencias con bloqueo se puede bloquear la aplicación si la conexión falla o es lenta. Las diferentes especificaciones de los lenguajes de programación para los diferentes compiladores, no es lo mismo el C de Visual Studio que el de GCC, harían necesarias modificaciones en la escritura del programa para transformarlo de una plataforma a otra. Si bien estas diferencias serían mínimas, harían necesaria una revisión casi completa del código escrito.

2.4.1.1.2 Interfaz web a través de navegador

Esta solución consiste en introducir el software en un servidor web al que tengan acceso los usuarios. Este servidor web ejecutaría los programas en algún lenguaje y ofrecería las respuestas al que consulta o trabaja. Los lenguajes que se soportarían en web son múltiples, ya sean compilados a través de CGI, PHP, Ruby on Rails, JSP o ASP. La base de datos que almacenaría los datos del programa se ejecutaría en el mismo servidor o en un servidor dedicado en la misma red, de manera que las consultas a la base de datos serían bastante rápidas.

Las opciones que se han contemplado para el desarrollo sobre servidor web son:

- PHP: lenguaje orientado a objetos de tipado débil. Se puede ejecutar sobre servidores Windows, Linux, MacOS, Solaris, y casi todos los posibles. Se necesita un servidor web sobre el que se ejecute el interprete PHP. Este puede ser Apache, Roxen, Microsoft Internet Information Server y muchos otros, entre ellos todos los que soporten CGI, pues se puede configurar como programa que funciona sobre CGI. Es un lenguaje de scripting que ha añadido el soporte de objetos desde la versión 4 y que desde la 5 el soporte es casi total. Con determinados frameworks la potencia de PHP en programación puede asimilarse a Java o .NET, con la ventaja o desventaja del tipado débil. Digo desventaja porque se pueden producir asignaciones en la programación de objetos que no corresponden con los que son, pero ventaja, porque el propio motor de interpretación del lenguaje, al menos en sus últimas versiones, llama a los métodos que transforma un objeto en otro si la operación lo requiere y con total transparencia.

- ASP: Actualmente el desarrollo de ASP se realiza en .NET en su mayoría. Los desarrollos en ASP basados en Visual Basic compilado y puesto en servidor ya no se usan mucho por la mayor facilidad de trabajo en .NET. El servidor debe ser necesariamente Windows salvo que se trabaje con el proyecto Mono, en cuyo caso se perdería la posibilidad de trabajar con Visual Studio, que sería una de las mayores ventajas que tendría, ya que tiene asistente para diseño de formularios web muy avanzado. La ventaja sobre PHP es que el código es compilado en vez de interpretado y durante el la ejecución funciona más deprisa.

- JSP: Se trabaja prácticamente igual que en ASP. El único cambio es que ya no depende tanto del sistema operativo ya que la base del software es Java. El programa se ejecuta en un servidor de aplicaciones web sobre Java. Existen múltiples alternativas en el mercado, tanto libres como propietarias, lo que nos da una capacidad de adaptación mucho mayor que ASP. Existen múltiples posibilidades de diseñar el interfaz como una aplicación de gestión al uso, siempre y cuando nuestro entorno de desarrollo las soporte:

- Struts: Plataforma de desarrollo para Java. Consiste en una serie de componentes preparados para su integración en una página web con un sistema de datos y clases detrás. Existe una implementación de Struts.
- JSF: Java Server Faces es una definición de Java para desarrollar una plataforma de desarrollo. La gran diferencia con Struts consiste en la expansibilidad de sus objetos, que permite crear componentes basados en los genéricos. La otra gran diferencia viene dada porque al ser una definición existen diferentes implementaciones del estándar, con lo cual se puede elegir la que mejor nos convenga o mejores extensiones tiene aplicables para nuestra aplicación. El problema viene por el manejo de estos objetos en el entorno de desarrollo, que no se maneja igual que en un interfaz cliente, donde el nuevo objeto creado puede incrustarse con facilidad en las barras de herramientas para su inclusión en el programa (al menos con Netbeans y Eclipse, que son los dos IDEs que se han planteado en el caso de que se realice con esta tecnología).

Al igual que las tecnologías de lenguajes mono-plataforma tienen sus ventajas e inconvenientes:

- Ventajas: Interfaz para el usuario reconocida de antemano, porque casi todos los usuarios tienen la experiencia de navegar por la web anteriormente. Mayor velocidad de conexión a las bases de datos desde el programa pues se encuentran localizadas en la misma red o servidor (dependiendo del caso del cliente). Experiencia del programador en este ámbito de varios años.

- Desventajas: Necesidad de conexión a la red en todo momento durante el manejo del software.

2.4.1.1.3 Tecnologías multiplataforma

Las tecnologías multiplataforma son sistemas que se componen de un compilador del lenguaje y un interprete para cada plataforma destino. Están realizados de manera que una misma compilación sirva para todos los sistemas operativos y arquitecturas de procesadores, pues los mensajes con el sistema operativo los realiza el interprete. Así si en un programa usamos un la posibilidad de varios hilos de ejecución y el procesador y el sistema operativo lo soportan será la maquina virtual que interpreta el programa compilado la que cree las nuevas instancias de ejecución y no el programa por si mismo. De esta manera y con una fuente de datos común se puede trabajar en diferentes ordenadores, con diferentes sistemas operativos y procesadores con un mismo programa.

Las tecnologías multiplataforma disponibles actualmente en el mercado y que cuentan con una documentación abundante son Java y el proyecto Mono, una especificación libre del estándar ECMA para programación, equivalente a la tecnología .NET.

- Ventajas: Un mismo código fuente sirve para todos los sistemas operativos. Hay Plataformas de desarrollo lo suficientemente avanzadas como para integrar la programación de servidor, con el diseño de interfaces de usuario y ayuda en la búsqueda de funciones de librerías avanzadas. Java permite el almacenamiento de certificados de seguridad en sus almacenes, esto es una ventaja respecto a la tecnología de navegador, donde el almacén es compartido por varios usuarios si no se usa correctamente, y respecto a lenguajes compilados, donde el almacenamiento es externo o hay que implementarlo a mano.

- Inconvenientes: Es ligeramente más lento en ejecución puesto que el programa funciona sobre una máquina virtual, otro programa, si bien con Java este inconveniente se está solucionando desde la apertura de su código al empezar a incluirlo en los núcleos de algunos sistemas operativos. La conexión de la base de datos tiene el mismo problema que los lenguajes mono-plataforma, con

la desventaja añadida de que el fallo puede resultar en el sistema operativo, en el programa, o en la máquina virtual que interpreta el código. El aspecto gráfico es diferente entre los diferentes sistemas operativos si se usa la integración con los mismo, o el aspecto es diferente al del resto del sistema operativo. El primer inconveniente dificulta el proceso de aprendizaje, pues existen sistemas operativos que cambian de sitios los menús y los integran a su manera. MacOS X es el caso más llamativo

2.4.1.2 La gestión de los datos

Esta aplicación va a contener una gran cantidad de datos de diferentes deportistas. El diseñador de la aplicación no concibe el producto sin la potencia de un Sistema Gestor de Bases de Datos (SGBD). Para la elección del sistema gestor de bases de datos se han tenido en cuenta diferentes opciones, desde bases de datos orientadas a objetos hasta las soluciones comerciales más conocidas. A continuación vamos a pasar a describir las diferentes posibilidades:

2.4.1.2.1 Gestores de Bases de Datos Orientados a Objetos

Estos sistemas de bases de datos son una relativa novedad. A pesar de que las metodologías orientados a objetos tienen ya cierta solera en el ámbito de la informática, el almacenamiento de la información como objetos no se ha desarrollado mucho. Hay que tener en cuenta que un objeto no es un conjunto de datos, sino los datos de ese conjunto, las operaciones que se pueden realizar con ellos, los mensajes que puede recibir para devolver información y los mensajes que puede recibir para alterar su estado. Con todos estos detalles se nos hace muy difícil la representación de un objeto en una base de datos.

Aún así el mundo de la informática ha evolucionado lo suficiente como para que exista este tipo de software en el mercado. Así tenemos la casi recién nacida JavaDB, DB4o (DataBase for Objects, del 2004), o con mucha más experiencia en el mercado, pero con carácter comercial Cache. Puesto que es un sistema aún bastante nuevo y, a pesar del estándar declarado OQL (Object Query Language), aún las diferentes opciones son demasiado diferentes como para pensar en la posibilidad de una migración de una a otra y demasiado jóvenes como para pensar que en un futuro van a persistir. Es por esta razón, por algunos motivos tecnológicos (se necesitan ejecutar como una clase dentro del sistema, salvo Cache, con lo que tendríamos que programar un servidor de manera imprescindible, y no permiten conexión remota si no es programada en la aplicación a través de un socket) y algunos otros de carácter teórico (el lenguaje de consulta no está basado en una matemática formal como en los sistemas de bases de datos relacionales y relacional – objeto) que se ha descartado este tipo de base de datos en la realización del proyecto.

2.4.1.2.2 Gestores de Bases de Datos Relacionales

Las bases de datos más comunes en el mercado son las bases de datos relacionales. Son sistemas gestores de bases de datos basados en el modelo entidad-relación. Su principal característica es que los lenguajes de consulta, edición y administración se basan en casi todos los casos en el estándar SQL (Structured Query Language). Esto significa que en cualquier base de datos relacional se puede realizar una consulta estándar de búsqueda de datos, de creación o edición de datos, o de administración de permisos de usuarios de una manera única para todas ellas. Este estándar no se cumple al completo en todas las bases de datos, aunque es cierto que los cambios son debido a las mejoras y nuevas aplicaciones de las sistemas gestores que lo modifican.

En este grupo de bases de datos se encuentran MySQL, SQL Server, SQLite o sybase. Soportan mejor el trabajo en red porque tienen integrado un servidor que atiende a las consultas a

través de un socket abierto en el puerto en el programa cliente (en el caso anterior había que hacer un programa en servidor que era el que recogía las consultas y las pasaba a la base de datos). El problema es que no se pueden representar objetos y las diferentes herencias se tienen que construir en tablas relacionadas con condicionantes de claves foráneas. Esto dificulta mucho el desarrollo del diseño de la base de datos en relación con el diseño orientado a objetos de la funcionalidad de la aplicación.

2.4.1.2.3 Gestores de Bases de Datos Relacional-Objeto

Una solución intermedia a estas dos es aplicar las extensiones al SQL que tienen ciertas bases de datos relacionales (Oracle, PostgreSQL) para trabajar con relaciones entre clases. El SQL de estas bases de datos se sale del estándar (lo que no significa mucho problema, pues entre las que son solo relacionales ya hemos comentado que también existen diferencias) y aporta nuevas palabras clave para trabajar con herencia, creación de nuevas implementaciones de tipos de datos y, aún sin estar relacionado con las tablas donde se almacenan estos objetos, permiten extensiones de funcionalidad con lenguajes de programación integrados en la misma base de datos (Extensiones PL o Procedural Language). MySQL 5 ya contempla alguna de las funciones de las bases de datos Relacional-Objeto.

El licenciamiento de una y otra base de datos es una de las grandes diferencias. Oracle, a pesar de tener una licencia de uso gratuito, no es libre. En la licencia de uso gratuito ofrecen soporte para un solo procesador y funcionalidad limitada. La variedad de pago es una licencia de soporte anual del producto sin derecho a actualizaciones en el caso de no renovación. PostgreSQL por el contrario es libre. Si bien no tiene el respaldo de una comunidad de usuarios como pueda ser el conjunto de programadores web que tiene MySQL, en su desarrollo participan multinacionales como Fujitsu o SUN Microsystems, y almacena los datos del sistema de telefonía IP Skype.

Este último tipo de base de datos sería el más apropiado para esta aplicación, ya que, en sistemas de producción, estimo que es más rentable la estabilidad de un producto estandarizado con posibles extensiones tecnológicas, que la extrema innovación de las bases de datos orientadas a objetos y que el trabajoso proceso de transformar un diseño orientado a objetos en un diagrama entidad-relación. Pero aún queda otra posibilidad que con la base de datos relacional de toda la vida puede ser bastante útil.

2.4.1.2.4 Sistemas intermediarios entre Base de Datos Relacional o Relacional Objeto y Programación Orientada a Objetos

Además de las diferentes posibilidades de sistemas gestores de bases de datos, existen en el mercado diferentes interfaces que hacen posible la persistencia de los objetos de una manera sencilla al programador. Con esto quiero decir que se trabaja con el objeto directamente sin tener que hacer declaraciones SQL para obtenerlo, guardarlo o modificarlo. Existen varios programas que realizan esta función de interfaz. Me he fijado en aquellos que por ser software libre podrían encajar con la licencia del proyecto: Hibernate, de RedHat, e iBatis, de Apache para Java y Propel y Doctrine de Sensiolabs para PHP. Aparte está el sistema de persistencia de objetos de Java.

La ventaja de estos programas consiste en que la gestión de la base de datos no es necesaria desde la lógica del programa. Partimos de los objetos y son los propios objetos los que tienen la persistencia adquirida como propiedad, ya sea porque implementa un interfaz que nos la define, o por que herede de alguna clase que ya la tenga implementada. Con una definición de modelo de datos del objeto (explicar los campos que tiene el objeto en un fichero que entienda el sistema, como un XML) una clase puede generar desde la estructura de la tabla donde se van a almacenar los objetos hasta realizar las típicas operaciones de recuperar, guardar y editar el

contenido del objeto sin que el programador escriba ni una sola sentencia SQL. Así conseguimos un programa totalmente orientado a objetos sin constructores ni métodos extraños con lenguajes externos al de programación (SQL, OQL).

Cada uno de estos sistemas utiliza un método para garantizar una persistencia correcta. Hibernate forma parte del proyecto de JBoss de RedHat. Aparte del sistema de persistencia en si mismo tiene añadidos que permiten la indexación de campos para facilitar la búsqueda en diferentes campos de la tabla. Además, a partir de la definición de las clases de datos, el fichero de configuración de conexión, y la equivalencia de las clases con la base de datos ,te crea en el sistema gestor de base de datos la estructura de las tablas que se van a utilizar

iBatis tiene la ventaja de que es más simple de manejar, pues no tiene tantos añadidos. A pesar de ello, la documentación es mucho mayor, así como la comunidad de usuarios, en Hibernate.

Doctrine sería el equivalente a Hibernate en PHP, pero está en Beta.

Symfony es un entorno de trabajo que abarca varios aspectos. No solo integra la persistencia de la que hablábamos antes mediante Propel, sino que facilita la generación de código y está preparado para la una integración total de páginas con AJAX. Tiene integrado el módulo de internacionalización en el propio framework, lo que representa una gran ventaja respecto a los otros que lo basan en la internacionalización del lenguaje base. En este caso realiza la gestión de la relación entre objeto y dato a través de Propel (otro proyecto de software libre) pero se puede integrar directamente con Doctrine, alcanzando un mejor control de las sentencias. Symfony es un entorno de trabajo que permite abarcar todos los aspectos de una aplicación web con una gran sencillez utilizando el patrón Modelo Vista Controlador.

La decisión de usar un sistema intermediario va a permitir escalabilidad en la aplicación. No va a depender de un sistema gestor de base de datos almacenado en un servidor. Cambiando el fichero de configuración el mismo esquema valdría para una base de datos pequeña almacenada en el propio disco de la aplicación (por ejemplo SQLite) como para una base de datos Oracle distribuida en varios servidores. Esto permite además al desarrollador que, una vez que tiene definidas las clases y los esquemas, se dedique solo a programar en el lenguaje de programación elegido, sin tener que preocuparse de la corrección de las sentencias de consulta a la base de datos. Además este sistema tiene la ventaja de que en búsquedas complejas y muy detalladas (en el caso de que no se use el servidor de indexación de Hibernate, que las realizaría con métodos orientados a objetos) se pueden pasar partes de la sentencia del lenguaje de consulta a la base de datos para concretar los resultados.

2.4.1.3 El servidor

Los datos actualizados que maneja el programa estarán almacenados en el servidor de base de datos. Además del propio servidor de los datos es muy posible existan procesos automáticos de cálculos, o incluso la posibilidad de que se consulte, a través de web ciertos datos públicos contenidos en las bases de datos. Dada esta situación la elección de un servidor apropiado es un punto crítico en el posterior funcionamiento correcto del sistema. En esta elección entran en juego el tipo de procesador, el procesador o procesadores, la memoria, la capacidad de almacenamiento y el sistema operativo.

La necesidad de capacidad de cálculo, almacenamiento en memoria y conexiones múltiples dependerá en todo momento del club deportivo y su magnitud. Puede que en ciertos clubes multidisciplinarios se necesite un sistema de dos servidores, uno de almacenamiento de base datos y otro de procesamiento y ejecución del programa, o una granja de varios servidores con las tareas compartidas, de manera que si falla uno el sistema completo siga funcionando, y se pueda añadir capacidad de proceso y almacenamiento según la necesidad.

Uno de los puntos que puede ser clave en el funcionamiento de la aplicación, sea cual sea la magnitud de la organización, es la posibilidad de la virtualización. La capacidad de ejecutar el servidor de aplicaciones y la base de datos en un mismo equipo, pero con sistemas operativos virtualizados da la posibilidad de cambiar en cualquier momento la estructura del sistema con impacto mínimo en el tiempo de migración. Además la seguridad que representaría el poder volver a levantar cualquiera de los servidores (el de la aplicación o el de los datos) en cualquier ordenador que tengamos a mano con la máquina virtual instalada en caso de errores nos da la confianza de tener un tiempo de respuesta a fallos muy rápido.

El caso que nos ocupa es el de un club pequeño, que puede tener 10 o 20 conexiones simultáneas a lo sumo, unos 150 los socios, deportistas, directivos y trabajadores, y un uso del sistema informático por parte de 2 o 3 de los técnicos. Luego las necesidades son muy pequeñas, pero escalables. También puede ser muy interesante la capacidad de crecimiento que aporta al club la aplicación.

La tarjeta de red no será necesariamente más que la estándar de 100Mb/s pues el cuello de botella de la transferencia de datos se encuentra en la línea de conexión a internet de los usuarios con ADSL en el mejor de los casos. En el caso de que la infraestructura sea virtualizada se tratará de dar un interfaz físico de red a cada máquina virtual.

En la elección del sistema operativo tenemos que tener en cuenta, además de que soporte virtualización, que sea totalmente compatible con la base de datos elegida (Oracle o PostgreSQL) y con el servidor de aplicaciones. Si bien el sistema operativo del servidor de aplicaciones podría ser diferente al de almacenamiento de datos, en futuro la formación del mantenedor de sistemas se podría encarecer, luego la elección sería la misma para ambos sistemas. Aquí tenemos las opciones:

2.4.1.3.1 Windows

Sistema operativo de Microsoft ® que en sus diferentes versiones de servidor ha evolucionado positivamente. Windows Server 2008 representa grandes avances de seguridad para un servidor que por la naturaleza cerrada de su código siempre ha sido objeto de ataques. La experiencia del diseñador con Windows Server (2003 en este caso) es positiva en medios de producción con software de gestión de terceros. Sus mejores características son la posibilidad de tener asistentes de todo tipo para configuración de software y hardware y la facilidad de manejo por su gran parecido al los Windows Home y Pro. Su gran desventaja es el incremento del consumo que representa cada incremento de versión, que hace que de una versión de a otra de Windows el hardware que la soporta quede obsoleto. La gran ventaja vendría si la elección del lenguaje de programación es .NET.

2.4.1.3.2 *NIX

Los sistemas operativos de la familia *NIX (UNIX, Linux, Solaris, *BSD, MacOS) tienen un origen común en los laboratorios de AT&T y se han ramificado dando lugar a versiones tanto propietarias, como libres. Tienen la ventaja de que entre la gran gama de núcleos de sistema operativo deferentes todos tienen las mismas herramientas de aplicación en la superficie, consola bash, servidor de ventanas X11, aplicaciones basadas en GTK y Qt y comandos de administración de sistema parecidos. El diseñador de la aplicación tiene larga experiencia en la administración de sistemas Linux de producción y MacOS de escritorio. La experiencia con estos sistemas operativos es muy buena en el ámbito de la estabilidad. Además, la reciente liberación de Java y las incorporaciones al núcleo del sistema operativo Solaris y Linux hacen que si el lenguaje es finalmente Java, se considere esta opción como la más aceptable. Además PostgreSQL nació para sistemas *NIX y, aunque la última versión ya lo soporta, en Windows está aún lejos de tener la

misma estabilidad.

2.4.1.5 La plataforma elegida

La primera cuestión que se va dilucidar en este punto es el interfaz que van a usar los usuarios para acceder a la aplicación. Hemos puesto en algunos de los requisitos que el usuario podría acceder desde diferentes lugares, sistemas operativos e incluso otros dispositivos, como móviles o PDAs. Para que esto sea posible el entorno de trabajo del usuario final tiene que ser web.

En la elección del lenguaje de programación hemos tenido en cuenta la capacidad de soporte externo, tanto de la comunidad como de una empresa, la facilidad de diseño con IDE o plataforma completa de desarrollo (que abarque desde el diseño de las clases, hasta de la interfaz gráfica), la posibilidad de reutilización del código, y la mayor abstracción y parecido de las capacidades del lenguaje con la metodología orientada a objetos. Descartando los lenguajes mono-plataforma por que no tienen el interfaz web, tan solo nos quedan .NET, Java y PHP (de los que han sido tenidos en cuenta), pero como hemos descartado previamente .NET por ser tecnología de Microsoft y tener un coste total de adquisición de la propiedad muy alto, tan solo nos quedarían Java, y PHP. El servidor Java, una vez cargado, es de una ejecución más rápida, pero en cambio consume muchísimos más recursos de memoria que un servidor Apache con PHP5 corriendo sobre él. Tras una intensa sesión de trabajo con PHP5 sin haber llamado en ningún momento al servidor de aplicaciones Tomcat (servidor de Java) estos son los números:

<i>USER</i>	<i>%CPU</i>	<i>%MEM</i>	<i>VSZ</i>	<i>RSS</i>	<i>TTY</i>	<i>STAT</i>	<i>COMMAND</i>
<i>tomcat55</i>	<i>0.1</i>	<i>6.1</i>	<i>297396</i>	<i>77608</i>	<i>?</i>	<i>S1</i>	<i>/usr/bin/jsvc -user</i>
<i>tomcat55</i>							
<i>www-data</i>	<i>0.0</i>	<i>0.9</i>	<i>37420</i>	<i>11808</i>	<i>?</i>	<i>S</i>	<i>/usr/sbin/apache2 -k</i>
<i>start</i>							

Como puede comprobarse la diferencia de uso de la memoria es muy grande, y más si se tiene en cuenta que Tomcat es un servidor de aplicaciones java bastante ligero. En una comparativa parecida Glassfish llegaba al 10% de la memoria total del sistema sin haber hecho ninguna operación con él.

En cuanto a la escalabilidad del sistema por una posible demanda de muchos usuarios el límite no lo establece el servidor de aplicaciones, ya que ambos son escalables. Se puede hacer funcionar una granja de máquinas funcionando bajo el mismo apache o servidor de aplicaciones java. El mayor problema de escalabilidad puede venir de la base de datos, ya que hay alguna de las bases de datos libres que no son fácilmente escalables salvo en sus versiones de pago (MySQL).

Después de haber considerado la opción de una aplicación web tenemos que pensar en el sistema operativo que nos servirá de base a todas estas opciones lo primero que se descartó fue cualquier plataforma basada en Windows tanto por el alto coste total de la implantación y mantenimiento para el cliente, como por las pocas ventajas económicas que aportaría para hipotética empresa que explotara la aplicación. El coste de la estación de trabajo donde se programaría la aplicación incrementaría su coste en 122,85€ solo por el licenciamiento de Windows Vista Bussines si lo adquirimos al comprar el ordenador nuevo o de 479,01€ si ya tenemos el ordenador con un sistema operativo diferente y queremos instalar el nuevo (como es el caso). Para aprovechar las ventajas del Windows utilizaríamos Visual Studio y programaríamos en .NET con lo que habría que adquirir la licencia del IDE de Microsoft. Esto supondría 914,73€ de licencia. En este precio no se ha incluido ningún antivirus, mantenimiento ni seguro por parte de otras empresas que incrementarían notablemente el precio. El coste del licenciamiento de la estación de trabajo es necesario, pues no se puede programar para un servidor Windows sin un

cliente Windows. El coste del sistema operativo de servidor mínimo son 218,18€. Además de todo ello para hacer funcionar esta serie de características es necesario un hardware muy potente, pues, aún en el caso del sistema operativo de servidor Windows Server carga interfaz gráfica para facilitar la administración del mismo.

A la hora de elegir el servidor se plantea la duda entre los sistemas operativos *NIX Open Solaris o Linux. Mac se ha descartado por la obligatoriedad de hardware de Apple ® para hacerlo funcionar. Tanto Open Solaris como algunas distribuciones de Linux tienen programas de certificación que nos permitirían elegir entre profesionales a la hora de mantener los sistemas en funcionamiento. Los sistemas BSD se encuentran en el camino de lograr una certificación para administradores como producto. A fecha de hoy y a pesar de no dudar de la capacidad de los profesionales dedicados al trabajo con este sistema operativo tenemos que descartarle por la ausencia de una formación específica en el mercado para su administración.

De las distribuciones que tienen certificación se ha elegido finalmente Ubuntu Server 8.04 LTS. Ubuntu es la comunidad de usuarios que más ha crecido en el mundo de Linux en los últimos años tanto a nivel de usuario como de servidor. RedHat y sus derivados (CentOS) también se tuvieron en cuenta, pero los precios de las licencias de RedHat y el retardo en las actualizaciones críticas de seguridad de CentOS hicieron que Ubuntu fuese la elegida. Open Solaris tiene una comunidad de usuarios bastante pequeña, presenta incompatibilidades con las distribuciones de Linux, pues el núcleo del sistema y los aplicativos de control son diferentes y presenta un sistema de gestión de actualizaciones bastante lento y desorganizado en su descarga.

Puesto que ya no nos ata la tecnología propietaria de Windows, podemos elegir libremente y en igualdad de condiciones entre Oracle, PostgreSQL o MySQL. Aunque las características de Oracle y PostgreSQL son muy parecidas, tanto en potencia como en modularidad, capacidad de escalar con clusters o granjas de ordenadores, y lenguajes procedurales, y en ambas mayores que MySQL optamos por MySQL que tiene la propiedad de ser libre, y en entornos web bastante más rápida y con menos consumo de recursos que las otras dos. Además al estar bajo una capa de abstracción, si deseamos hacer la aplicación compatible con las otras dos bases de datos no tenemos más que cambiar los ficheros de configuración a la nueva base de datos elegida y generar los contenidos.

El lenguaje de programación elegido es PHP debido a la larga experiencia con el mismo del programador, así como por el amplio soporte de la comunidad, extensa y actualizada documentación y viveza en actualizaciones de seguridad. Pero sobre todo lo elegimos porque nos va a permitir trabajar rápido los prototipos para seguir el modelo de desarrollo en espiral a través del entorno de trabajo Symfony. Además este entorno de trabajo nos va a gestionar la conexión a la base de datos, generación de muchos formularios, separación de Modelo, Vista y Controlador (MVC). Otra de las ventajas de Symfony es que debido a su sistema de abstracción de datos nos va a ser posible cambiar la base de datos sobre la que se ejecuta la aplicación sin ningún problema más allá que las propia necesidad de integridad referencial para esta aplicación.

2.4.1.6 Herramientas informáticas para el diseño y la programación

Todo el trabajo del diseño de la aplicación se va a realizar con la misma herramienta Case que se ha realizado el análisis. Al utilizar el entorno de trabajo Symfony no va a ser necesaria una generación de código, ya que, a partir de herramientas de configuración, se va a generar gran parte del código por sí misma. Así que todos los diagramas de clases, despliegue y colaboración se van a realizar con ArgoUML.

Para la programación se va a elegir un programa que, aunque tengamos que sacrificar memoria, ayude en la elaboración del código. Entre los diferentes editores de código que prestan ayuda hemos evaluado Kdevelop, Eclipse y Netbeans. Todos ellos son software libre y gratuito, aunque solo Netbeans y Eclipse son realmente multiplataforma. Llegado este punto es importante realizar el trabajo en las mejores condiciones posibles para facilitar la posible colaboración de una comunidad de usuario y programadores ya que forma parte también del espíritu del software libre.

De todos los anteriormente mencionados el que mejor soporte actual tiene para PHP es Eclipse + PHP Developer Tools + Subclipse (para el control de versiones con Subversion) + Web Standard Tools. Esta combinación hace que en un mismo fichero mezcla de PHP con HTML al escribir las primeras palabras de la función o etiqueta que deseamos escribir nos aparezca un listado navegable con las opciones posibles. Con las diferentes vistas permite que en el árbol de fichero nos despliegue el contenido de los mismos, véase, clases, funciones, constantes. Hace lo mismo con las hojas de estilos y los fichero XML, necesarios para la configuración.

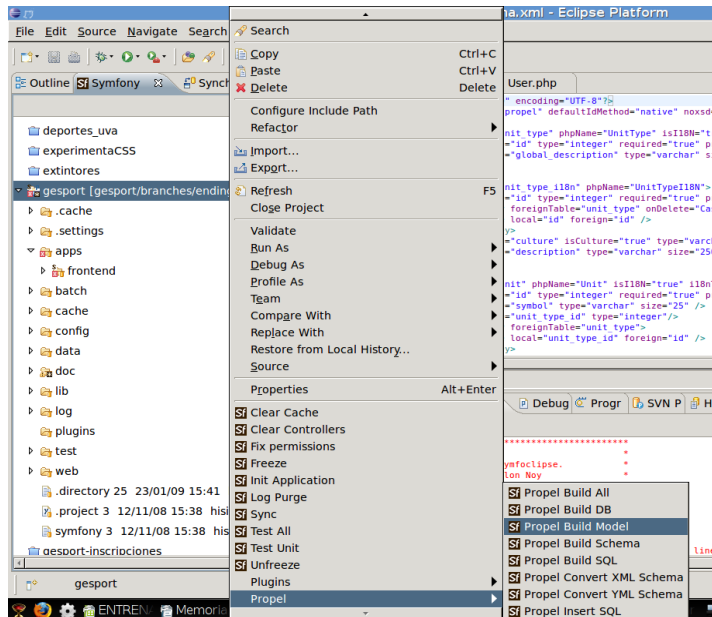


Ilustración 3: Vista de trabajo con Eclipse, coloreado de código y asistente de Symfony

Por si esto fuera poco existen varias extensiones para trabajar con Symfony, facilitando las tareas de las líneas de comando que requiere este framework y proporcionando un editor de YAML, el lenguaje de marcado usado por Symfony internamente. En este caso se ha utilizado Symfoclipse, que fue liberada en su momento como software libre y posteriormente retirada por su autor. Además Eclipse entiende perfectamente los diferentes tipos de documentación introducidos con el estándar PHPDocumentor, así como asignación de tareas en el código.

2.4.2 El paradigma Modelo Vista Controlador

Como hemos visto en el punto 4.1.5, La plataforma elegida, Symfony permite una muy buena implementación del patrón Modelo Vista Controlador. En este patrón se produce una separación lógica de las diferentes tareas que tiene que ejecutar la aplicación. Además esta separación permite utilizar de una manera muy correcta el modelado orientado a objetos.

- El modelo define la representación de la información, sus métodos, la estructura, las relaciones entre ellos. Esta es la parte de orientación a objetos donde se definen las características de los mismos y sus tareas propias.

En Symfony esto se consigue con un fichero llamado schema.xml o schema.yml que detalla el nombre de la clase de objeto, su tabla asociada en la base de datos, las relaciones con

otras tablas u objetos y la herencia en el caso en que se pueda aplicar. La versión de Symfony que usamos utiliza Propel 1.2 para gestionar el modelo y el soporte para herencia, que si bien existe, es limitado y problemático.

Una vez que tenemos definido el fichero `schema.xxx` Symfony automatiza la generación de librerías de control de estos objetos en la base, así como de uno ficheros, que usando la herencia de PHP nos permiten añadir nuestros propios métodos, de manera que en sucesivas revisiones del modelo no tengamos que volver a escribir el código ya escrito. Esto se almacena en la carpeta `/lib/model/`. Los ficheros contenidos allí son las clases que heredan de las contenidas en `/lib/model/om`. Estos últimos son los generados automáticamente por Symfony.

Igualmente que Symfony genera los ficheros PHP de definición de clases también genera el fichero SQL para incluir en la base de datos.

Con el fichero de definición del modelo, los ficheros de definición de clases PHP, y la inclusión de la estructura de datos en la Base de Datos ya tenemos el modelo del paradigma completado.

Cabe destacar que con un mismo modelo de datos podemos construir varias aplicaciones, de manera que podríamos tener en nuestro caso una aplicación web para los usuarios, un API para enganchar con otras aplicaciones, un programa de gestión para el administrador de la aplicación, cada uno con su entorno de trabajo completamente diferenciado.

- Controlador: A la hora de seleccionar, que información mostrar, como usar los métodos de los objetos e interactuar con el modelo, permitir o denegar accesos, y todas las tareas que impliquen interacción entre objetos está el Controlador.

En Symfony es la parte del proyecto que obtiene, transforma, edita, borra o filtra datos. El controlador de un módulo de un aplicación lo define una clase de objeto llamado `actionsNombreModulo` que hereda de `actions`. En este se definen acciones concretas a las que se llama desde los formularios o enlaces. El formato común de llamada es "modulo/acción", pero para evitar que alguien haga llamadas a "usuario/borrar" Symfony, aparte de los ficheros de seguridad, propone una configuración de rutas alternativa. También existen los componentes, que permiten hacer objetos modulares. Estos reciben los datos por paso de parámetros, pues se trata de funciones, en vez de cogerlos del entorno (sesiones) o de la url o los métodos GET o POST del estándar HTTP.

Dentro del controlador intervienen más ficheros que las propias acciones a realizar. La configuración de las rutas, los ficheros de configuración de permisos, los ficheros de validación de formularios, así como la programación de validadores propios, la extensiones de las clases de Symfony que se usan en la construcción de la aplicación o los plugins (paginado de resultados, mejora en conexiones a la base de datos, facilidad en formularios...) son ejemplos de ello.

Una característica propia en Symfony es que el fichero `index.php` es el primer y último fichero que interviene, enviando los datos al corazón de la aplicación, y recibiendo, en última instancia, el resultado que mostrar al usuario. Forma parte del controlador, pues no genera aspecto gráfico, alguno, ni contiene el modelo de datos. Es lo que se llama el controlador frontal y cada aplicación tiene el suyo

- Vista: En el patrón la vista son los ficheros de la aplicación que definen que se muestra y como se muestra la información al usuario.

En Symfony estos forman parte de una aplicación concreta y no del proyecto en general. Hay unos ficheros contenidos en unas carpetas llamadas "templates". Estos ficheros, dependiendo de si se encuentran en la carpeta de aplicación o en un módulo definen la salida de la aplicación completa, del espacio dedicado al módulo en cuestión.

Cada acción tiene su vista asociada en el origen, pero con la configuración o con la programación en el controlador puedes hacer que una determinada acción ejecute la vista de otra. Esto es muy útil en algunos formularios que piden los mismos datos en tiempo de edición y creación.

La vista normalmente será un fichero html, o una parte del mismo, en el que se hacen inclusiones de PHP para conseguir sacar la información de la manera adecuada. Se trata siempre de respetar el html y solo se incluye el php en los casos necesarios, de manera que cualquier diseñador o maquetador de webs pueda variar la plantilla sin tener que saber programar. Un ejemplo sería este:

```
<th><?php echo __('Reported volume'); ?></th>
  <th><?php echo __('Options'); ?></th>
</tr>
</thead>
<tbody>
<?php foreach ($seasons as $season): ?>
<tr>
  <td><?php echo $season->getDescription() ?></td>
  <td><?php echo $season->getStartAt() ?></td>
  <td><?php echo $season->getLastDayAt() ?></td>
  <td><?php echo $season->getTheoricIntensity() ?></td>
```

Como se puede ver en este ejemplo el código de la plantilla es HTML. Cualquier navegador podría leer el fichero aún sin intérprete php, dando una salida por pantalla. Al introducir el intérprete no mostraría los datos en cada salida.

Para comprenderle un poco mejor se muestra un gráfico del funcionamiento concreto de la aplicación real de Symfony a este paradigma, donde se incluye el controlador frontal, que no es otra cosa que el fichero index.php, que se interpone al resto de elementos, ofuscando la ubicación real de los ficheros, y controlando el entorno de ejecución: producción, desarrollo o pruebas.

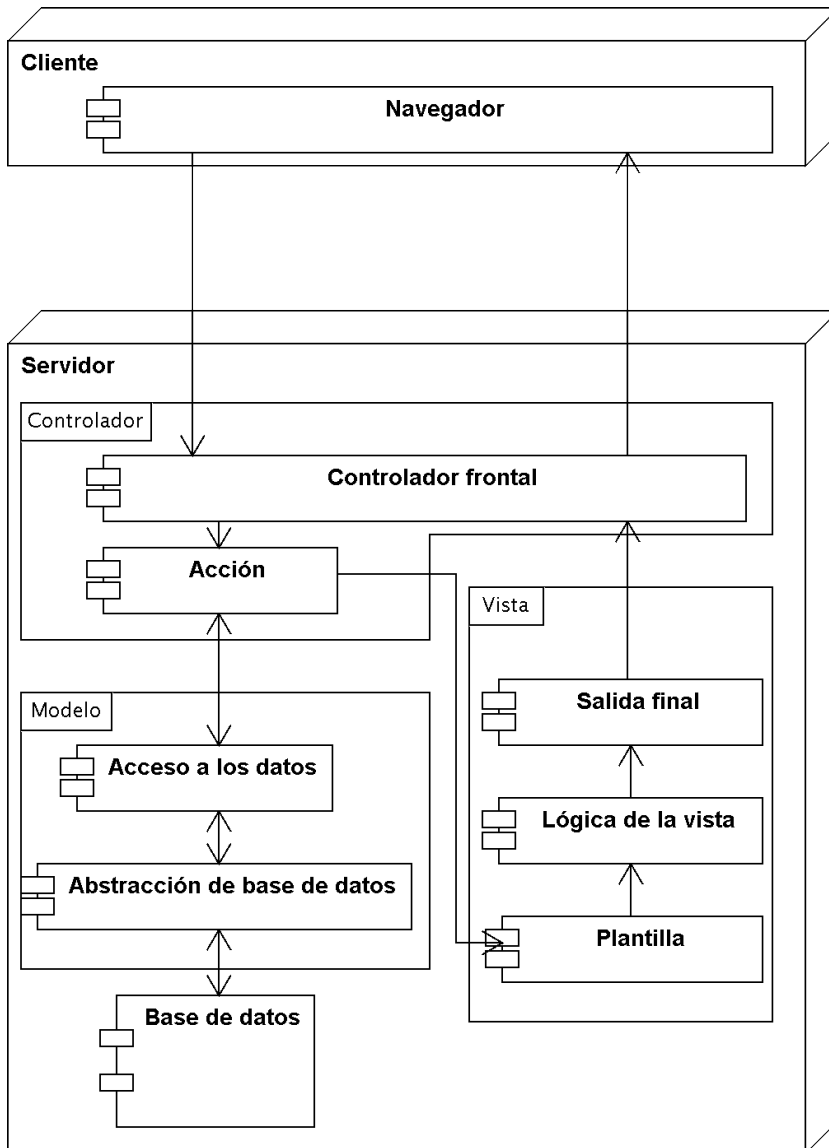


Ilustración 4: Esquema ampliado MVC

2.4.3 Diseño de la base de datos

Al usar Symfony el diseño de la base de datos está íntimamente ligado a los objetos. Se definen los objetos y la base de datos en el mismo momento de la fase de desarrollo. Para ello utilizaremos un fichero de esquema (schema) bien en XML, bien en YML donde definiremos las clases de objetos y las relaciones entre ellas. Genéricamente, cada clase de objeto se corresponde con una tabla de la base de datos. Esta definición se puede cambiar en los ficheros de configuración y utilizar una tabla para definir múltiples objetos o un objeto estar definido en múltiples tablas. En un principio no se hará uso de estas propiedades y, para simplificar, se asignará cada objeto a una

tabla y en aquellos con referencias múltiples simultáneas (relaciones n:n) una tabla intermedia que será representada en el diagrama de clases por una clase funcional.

2.5 Futuro de Gesport

Gesport, tal y como ha sido concebido es tan solo el embrión de un proyecto más ambicioso que abarcará todas las áreas de gestión de los clubes, deportes de equipo, táctica, psicología, técnica,

Esta primera versión completamente funcional solo sirve para llevar una planificación completa de deportistas. Digo solo porque no tiene herramientas auxiliares que aceleren su manejo, interfaces dinámicas con AJAX, generadores automáticos de según que cosas.

Este tipo de elementos podrían encajar bien en una segunda versión de la aplicación.

Además a este futuro hay que añadirle las constantes actualizaciones de posibles problemas de seguridad, migración entre versiones de Symfony para asegurarnos la correcta corrección de fallos, ya no de la aplicación sino del propio framework.

En breve se empezará a utilizar para dar entrenamiento a ciertos atletas del Club de Atletismo Joaquín Blume, de donde se empezarán a recabar nuevas necesidades para que este proyecto sea un proyecto vivo.

2.6 Agradecimientos

Desde espacio quiero agradecer a todos los que me han apoyado, e insistido, en esta consecución de un proyecto iniciado y terminado por mí. Entre estas personas se encuentran toda mi familia, que ya deseaba que acabara el proyecto de fin de carrera, los amigos y la novia.

También quiero agradecer a todos aquellos que de una manera u otra colaboran con el software libre y hacen de esta sociedad algo más humano, donde, a pesar del trabajo que lleva realizar ciertas tareas, estas se comparten sin un ánimo de lucro como objetivo, aunque bien pueden llegar a servir para vivir.

Y, por último y sobre todo, a Pedro Lorenzo, que se ha dedicado a pasar horas junto al autor del proyecto diseñando unos requisitos que esperemos que este programa cumpla.

3. Documentación técnica

Documentación para la comprensión, desarrollo y continuación del proyecto por parte de otros programadores

3.1 Análisis del sistema

3.1.1 Glosario de términos deportivos

Durante el transcurso de las entrevistas surgieron términos técnicos del área del deporte y de la salud que vamos a definir en este punto para que el resto del texto de análisis y posterior diseño y desarrollo se entiendan de la mejor manera posible. También se han traducido estos conceptos al inglés para su posterior inclusión en los diagramas de clases.

Temporada (Season): Periodo determinado de tiempo, generalmente de 9 meses a 1 año, sin necesidad de que sea año natural. En dicho periodo se centra la actividad principal de entrenamientos y competiciones de cada deporte.
Sesión (Session): La sesión es la estructura elemental de todo proceso de entrenamiento. La planificación anual o plurianual de un deportista se basa en la combinación adecuada de las diferentes sesiones.
Microciclo (Microcycle): Es un conjunto de sesiones planificadas en un corto periodo de tiempo.
Mesociclo (Mesocycle): Es un conjunto de Microciclos.
Macro ciclo (Macrocycle): Es un conjunto de Mesociclos. Cada uno de estos ciclos temporales anteriores tiene su razón de ser en el desarrollo de cualidades concretas, en la consecución de objetivos concretos y en las fases competitivas que hay a lo largo de la temporada.
Objetivos (Objective): Son las metas marcadas para conseguir en un determinado tiempo, que puede ser, un microciclo, macrociclo, temporada o periodo plurianual. Pueden ser objetivos principales, secundarios, operativos en función de la meta que busquemos. Siempre tienen que seguir una progresión lógica y asequible.
Carga : La carga constituye la categoría central del entrenamiento, se realiza a través de ejercicios físicos, con una determinada finalidad. El entrenamiento deportivo se basa en la aplicación de cargas de trabajo en las sesiones de entrenamiento y competiciones. Se puede entender como medida fisiológica del organismo, provocada por un esfuerzo físico, con una cierta duración y profundidad (volumen / intensidad).
Periodización: La periodización consiste en distribuir las cargas de forma lógica y progresiva, a lo largo de una temporada, o un periodo determinado de tiempo, en función de los objetivos marcados y propuestos para cada deportista, para conseguir una mayor adaptación y progresión deportiva.
Ejercicio: Movimiento del cuerpo, descrito mediante un texto o dibujo, que, al realizarlo correctamente, consigue unos efectos concretos sobre el sujeto entrenado.
Intensidad (Intensity): Es el porcentaje de trabajo que se aplica durante un esfuerzo* (fza maxima, VO2 max, Velocidad max)
Test: Es una prueba o valoración de cualquiera de las capacidades físicas básicas de un deportista.
Tipos de trabajo (fuerza resistencia, anaeróbico, aeróbico)

Consumo máximo de oxígeno (VO ₂ max): También llamado VO ₂ max., es la cantidad de oxígeno consumida por el cuerpo en condiciones de máximo esfuerzo. Es uno de los mejores (por no decir el mejor) indicadores de la competencia física.
Marca: Resultado obtenido por un deportista en cualquier prueba, test, competición, etcétera.
Espirometría: Es un método de diagnóstico que estudia la función pulmonar en relación a su capacidad funcional y de reserva (cuánto aire se puede introducir en los pulmones y la rapidez con la que se puede expulsar). En la espirometría se evalúa el coeficiente VEF/CV (Volumen espiratorio forzado en el 1º segundo/ capacidad vital).
Test de Esfuerzo: También llamado Prueba de esfuerzo, ECG de ejercicio, etc. Es una herramienta de examen general para evaluar los efectos del ejercicio en el corazón. Durante la prueba se registra la actividad eléctrica del corazón, mientras la persona camina (también se puede hacer con carrera si hablamos de sujetos entrenados) en una cinta sin fin, o pedalea en una bicicleta estática. Se mide la reacción del corazón a una mayor demanda de oxígeno por parte del organismo, provocado por el progresivo incremento de la intensidad del ejercicio.
Monitor de pulso cardíaco o monitor (HRM): 1. Aparato que mide y almacena los datos de la frecuencia cardíaca del deportista que lo porta durante el entrenamiento. En la actualidad aparte de la frecuencia cardíaca media y máxima hay modelos capaces de almacenar el pulso en periodos muy cortos de tiempo, permitiendo la visualización de curvas y gráficas con la evolución del pulso. También existen los que miden la distancia recorrida, la diferencia de altura sobre el nivel del mar e incluso trazan las rutas entrenadas por GPS. 2. Persona encargada de la iniciación deportiva en cualquier categoría. En la iniciación se trata de desarrollar las características físicas básicas sin entrar en la planificación del deportista para la competición. Durante toda esta memoria se tratará de expresar siempre el monitor de visualización de un ordenador como "pantalla" para no dar lugar a equívocos con los anteriormente definidos.

Tabla 2: Glosario de términos deportivos

3.1.2 *Análisis de Requisitos*

3.1.2.1 **Requisitos Funcionales**

En esta sección vamos a describir las necesidades que tiene el cliente. El primer detalle a destacar es que será un software que podrán usar todos los miembros de la organización. Cada uno tendrá acceso a realizar diferentes tareas de cada área según sea su interés real en las mismas. Los diferentes funcionamientos de cada área vienen dados por las entrevistas realizadas al cliente.

3.1.2.1.1 **Gestión de entrenamientos**

El programa debe soportar diferentes tipos de personas encargadas de la gestión deportiva. Para ello tendremos que tener una base, los datos personales. Cada una de estas personas tendrá acceso a ver, añadir, modificar y borrar ciertos datos de las otras personas que intervienen en la gestión. En la gestión de entrenamientos intervienen unos siguientes documentos, que pertenecen a las diferentes personas y que solo ellas podrán consultar. Estos documentos son la esencia del

entrenamiento deportivo.

Documentos necesarios para la correcta gestión de entrenamientos:

1. Documento de datos personales. Donde se definen los datos personales: nombre, apellidos, teléfono, direcciones, fecha de nacimiento, correo electrónico.
2. Documento de datos del deportista. Donde se recoge el deporte que practica, el grupo con el que entrena, el equipo donde milita (para casos en los que haya diferentes equipos dentro de un mismo club: canchas, categorías inferiores, diferentes deportes) y el entrenador.
3. Documento de periodización de la temporada. Donde se describen por microciclos los diferentes aspectos a trabajar durante el mismo, así como el número de kilómetros, kilos o repeticiones trabajados en cada semana. También está reflejado en este documento la progresión que tiene el deportista a la hora de realizar los diferentes ejercicios y, si fuere posible, una utilidad gráfica que muestre la cantidad (volumen) de trabajo por microciclo para ver la curva en un primer vistazo.
4. Documento de programación. Es el que nos describe los ejercicios a realizar en cada día de la semana. A través de este documento se produce la realimentación hacia el entrenador, pues tiene campos abiertos para que el sujeto entrenado describa la realización del ejercicio. En este documento también aparecen los gráficos de las zonas de trabajo físico realizadas: carga (como volumen e intensidad), indicaciones de zonas cardíacas de trabajo y otras estadísticas de utilidad para el profesional del deporte.

En la organización, o club habrá varios entrenadores que entrenarán a cada uno de sus deportistas. Cada atleta tiene personalizados los siguientes documentos, de manera que el entrenador podrá ver cada uno de ellos en relación con el atleta que va a entrenar: 1. Documento de datos personales, 2. Documento de datos del deportista, 3. Documento de periodización, 4. Documento de Programación .

Vamos a proceder a describir las informaciones completas que contienen estos documentos:

3.1.2.1.1 Documento de datos personales:

Al crear un nuevo deportista, socio, o trabajador se generará este documento que contiene todos los datos como nombre, apellidos, teléfono, y el tipo o tipos de papeles que representa en el club (socio, entrenador, deportista, trabajador, director técnico, presidente, secretario, vicepresidente, tesorero, vocal).

Los entrenadores podrán añadir o editar personas de tipo deportista, el director técnico a entrenadores y trabajadores del club, aparte de deportistas, el secretario a los socios y cambiar el estatus de cada uno de ellos según las actas de la junta directiva, el tesorero o los organizadores de pruebas a los proveedores.

3.1.2.1.2 Documento de datos del deportista:

En este documento se describe el grupo al que pertenece, el deportista, su entrenador. El propio deportista podrá cambiar partes de su documento, así como de sus datos personales.

3.1.2.1.3 Documento de periodización:

Consiste en una planificación total de la temporada, luego existe uno diferente por cada deportista y temporada. Contiene:

Gesport - Gestión de Entrenamiento Deportivo

- Nombre de la temporada
- Listado de Macrociclos, Mesociclos y Microciclos.
- Volumen por microciclo.
- Fechas correspondientes a cada microciclo.
- Listado de ejercicios diarios del documento de progresión, con el nivel correspondiente del trabajo en cada microciclo.

El entrenador del atleta es el que edita este documento. No está visible para el atleta normalmente, si bien el entrenador puede hacerlo visible si lo desea.

3.1.2.1.1.4 Documento de programación:

Es el documento donde se tiene el entrenamiento que debe hacer cada día el deportista. Está estructurado en microciclos, y, en cada día que dure el microciclo, se pueden poner hasta 3 sesiones de entrenamiento, que si bien no es lo usual, si que pueden llegar a alcanzarse en deportistas de muy alto nivel.

Cada microciclo tiene:

- La carga de trabajo establecida en la periodización
- La carga efectiva realizada según los datos de realimentación del sujeto entrenado.
- El volumen e intensidad de entrenamiento.
- El ejercicio diario que tiene que realizarse según la periodización.
- Datos que el atleta ofrece como resultados de su entrenamiento para que el técnico pueda comprobar su estado y evolución. Consiste en que el deportista introduzca en una tabla el tiempo dedicado a cada tipo de trabajo de cada sesión.

Según lo indicado anteriormente es el entrenador el que introduce los datos posteriormente a que el programa haya presentado unos ejercicios teóricos según la periodización.

3.1.2.2 Requisitos no funcionales

Para el análisis se han determinado las siguientes condiciones en las que se va a mover la aplicación:

- La aplicación resultante deberá ser accesible a todos los usuarios a través de Internet. Esta disponibilidad podrá ser vía web o por descarga directa de un programa que permita el acceso a los datos pertinentes del servidor central.

- Funcionará desde algún terminal móvil, teléfono o agenda electrónica, para la toma de tiempos, resultados y variación de entrenamientos durante el trabajo de campo para los entrenadores y técnicos. Las funcionalidades de este módulo serán las mínimas posibles para permitir un trabajo con varios deportistas o grupos.

- Tendrá un entorno gráfico de usuario de fácil manejo y preparado para una adaptación a tecnologías de accesibilidad avanzada (personas con problemas de vista, oído o movilidad en sus extremidades).

- Soportará la internacionalización. A la presentación del proyecto funcionará en 3 idiomas, Español, Inglés y Francés, y tendrá la posibilidad de añadir idiomas tan solo con añadir los

ficheros de traducción adecuados. El idioma por defecto de funcionamiento será el Inglés, pero el software detectará del sistema operativo o del navegador el idioma de la persona que está trabajando y en el arranque lo adaptará. Permitirá la elección manual de idioma.

Se trata también de que la gestión de estos recursos sea eficiente y provea las herramientas necesarias de conocimiento de todas las áreas del club a los diferentes usuarios, cada uno en su medida, de manera que tengan un acceso fácil y sencillo a cualquier información de su incumbencia.

Deberá ser de fácil manejo, pues al usuario final no se le presuponen capacidades del manejo de la informática más allá de una alfabetización básica.

Tendrá que ser multiplataforma, pues no se puede asegurar que todos los componentes de una organización tendrán el mismo tipo de ordenador, y menos cuando se piensa en una aplicación para múltiples organizaciones.

Estará obligado por ley a guardar la privacidad de los datos de los usuarios en los niveles que así lo requiera, tendrá que cumplir con la Ley orgánica de Protección de datos y los usuarios podrán borrar o rectificar los datos personales contenidos en los ficheros del programa. Deberá permitir la importación o exportación de firmas digitales. La codificación de datos para permitir la lectura de los mismos por las personas adecuadas será asimétrica.

Deberá presentar la información justa para la toma de decisiones en cada momento. Los datos deberán poder simplificarse en resúmenes, gráficos o esquemas.

3.1.3 Identificación de las clases de análisis

Con los datos que tenemos hasta ahora vamos a identificar las clases del programa. Primero se muestra un listado de candidatas que luego se filtrará. Estas clases que identificamos ahora son las que componen el dominio del problema a resolver, que no tienen porque ser clases efectivas en el diseño y programación de software.

Se corresponden a las clases de objetos que intervienen en todo el ámbito del problema en el mundo real. Así mismo en la fase de diseño se crearán nuevas clases para el intercambio de mensajes, las relaciones, los listados y el trabajo interno de software. Estas últimas clases de objetos se diseñan pensando en la solución software y en esta fase del proyecto estamos analizando el problema.

En este proceso ya se han eliminado todos los conceptos del análisis que se han dejado para sucesivas versiones.

3.1.3.1 Clases candidatas

Documento de datos personales, deportista, nombre, apellidos, entrenador, director técnico, documento de datos del deportista, temporada, documento de periodización, tipo de progresión, macrociclo, periodo, mesociclo, microciclo, sesión, ejercicio diario, documento de progresión, documento de programación, carga, carga de la periodización, carga realizada, horas de entrenamiento, entrada de realimentación, fecha, ejercicio, nivel de intensidad.

3.1.3.2 Clases del dominio del problema

Del listado de clases candidatas hay algunas que por la propia definición del problema no plantean ninguna duda de son clases efectivas en la solución del problema. Las que me han planteado la duda son:

- Nombre, apellidos: en este caso los consideraré atributos de persona, como clase base para todos los tipos de persona que habrá después en el club.

- Deportista, entrenador y el resto de diferentes puestos en la infraestructura del club. Se podría pensar que son objetos de la clase persona con diferentes características. En esta fase de análisis son clases de personas que intervienen en la organización y en la fase de diseño se decidirá la implementación apropiada para ellos estudiando cada caso particular de individualidad en instanciación, o por áreas.

- Ejercicio diario se refiere al ejercicio realizado en una sesión por lo que estaría mejor denominado “ejercicio de la sesión”. Constituiría una clase en la que se podrían incorporar ejercicios predeterminados en el documento de progresión y otros que el entrenador del deportista considere oportunos.

- Fecha. No aparece en los datos personales pero debería aparecer con la fecha de nacimiento. En la fase de análisis es un atributo de la clase persona (fecha de nacimiento) y de la clase prueba o test (fecha de la prueba, competición o test). En la fase de diseño probablemente se convierta en una clase, aunque venga dado por el lenguaje de programación en su especificación, porque necesitará de métodos y condiciones para que la fecha sea válida.

3.1.4 Esquema básico de relaciones entre clases de análisis

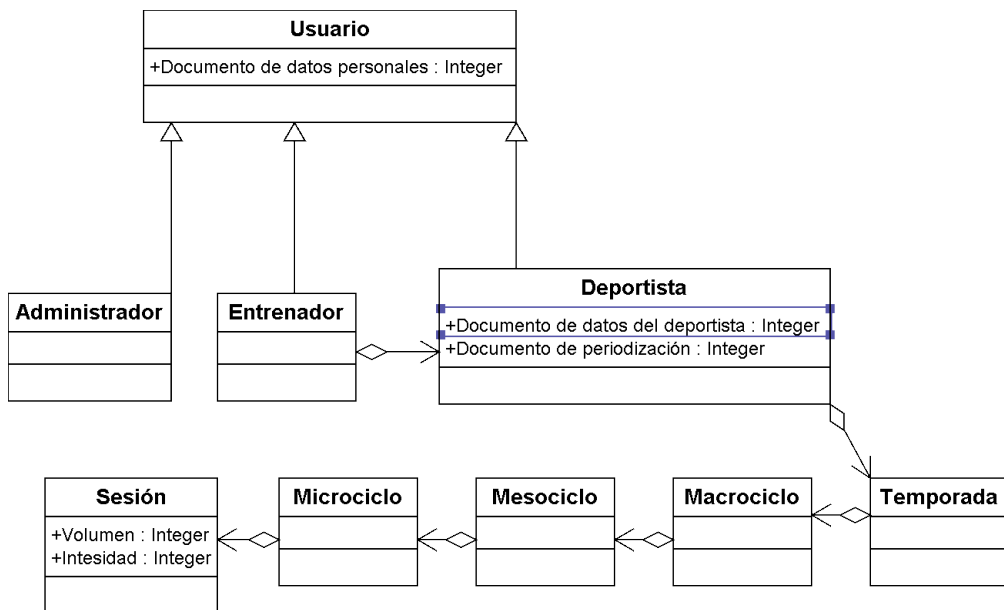


Ilustración 5: Clases obtenidas del análisis

3.1.5 Casos de uso

Con todos los datos recogidos anteriormente se pasan a describir los casos de uso. Se va hacer en el siguiente orden, primero los actores, luego los casos de uso en los que participarán, después se hará el estudio de los escenarios donde pueden crease conflictos.

3.1.5.1 Actores

Nombre de Actor	Administrador
Definición	Es el encargado de la aplicación. No tiene acceso a la totalidad de los datos, solo administra la aplicación y no el sistema completo.
Nombre de actor	Entrenador
Definición	Es el encargado de marcar los planes de entrenamiento a los deportistas. Realiza la planificación y lee los resultados.
Nombre de actor	Deportista
Definición	Sujeto entrenado que pone en práctica los entrenamientos que puede leer y reporta las incidencias y resultados de los mismos.
Nombre de actor	Usuario de internet
Definición	Usuario no registrado en el sistema en el momento de iniciar contacto con el mismo
Nombre de actor	Usuario
Definición	Generalización de usuario del sistema (administrador, entrenador y deportista)

Tabla 3: Actores representados en los casos de uso

3.1.5.2 Descripción de los casos de uso

Nombre del caso de uso 1	Gestión de tipos de unidades
Definición	Actualización, creación, borrado y visualización de los diferentes tipos de unidades de medida que existen como datos básicos para todos los entrenadores y deportistas.
Nombre del caso de uso 2	Gestión de unidades
Definición	Actualización, creación, borrado y visualización de las diferentes unidades de medida que se usarán para comprobar la realización correcta de los entrenamientos.
Nombre del caso de uso 3	Gestión de factores de conversión
Definición	Actualización, creación, borrado y visualización de los diferentes factores de conversión que convierten unas unidades en otras, y no necesariamente del mismo tipo. Esto queda a discreción del administrador pudiendo incluir factores de conversión extraños.
Nombre del caso de uso 4	Gestión de usuarios
Definición	Actualización, creación, borrado de usuarios y visualización de ciertos parámetros de los mismos, dependiendo del actor que lo realice.
Nombre del caso de uso 5	Gestión de temporadas
Definición	Actualización, creación, borrado y visualización de las temporadas a las que tenga acceso según el rol y la relación con los usuarios para los que se planifiquen la temporadas.
Nombre del caso de uso 6	Gestión de macrociclos
Definición	Idéntico a las temporadas.
Nombre del caso de uso 7	Gestión de mesociclos
Definición	Idéntico a los macrociclos y temporadas.
Nombre del caso de uso 8	Gestión de microciclos
Definición	Parecido a los ciclos anteriores y temporadas, pero con funcionalidades diferentes, ya que en última instancia se crean las sesiones de entrenamiento de manera automatizada, una por cada día de duración del microciclo.

Nombre del caso de uso 9	Gestión de sesiones
Definición	Actualización, creación, borrado y visualización de las sesiones del deportista. Dependiendo del rol y de la relación pueden ser diferentes las opciones.
Nombre del caso de uso 10	Gestión de deportistas
Definición	Actualización, creación y visualización de los deportistas.
Nombre del caso de uso 11	Gestión de feedback
Definición	Leer y editar el feedback asociado a cada sesión.
Nombre del caso de uso 12	Entrada y salida al sistema
Definición	Entrar al sistema, salir, adecuar los menús a los permisos adecuados.

Tabla 4: Descripción de los casos de uso

3.1.5.3 Diagrama de casos de uso

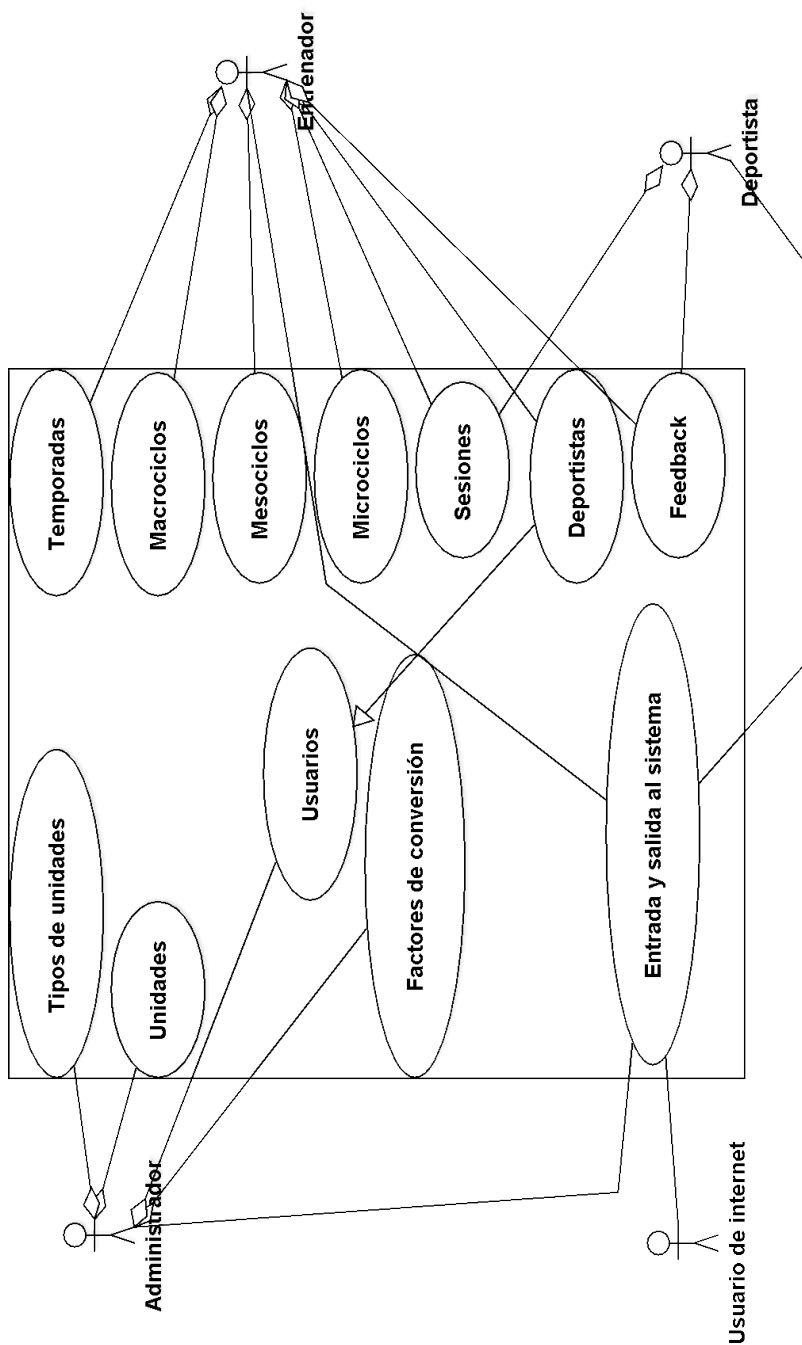


Ilustración 6: Diagrama de casos de uso simplificado

3.1.5.4 Escenarios concretos

Análisis de escenarios concretos de los casos de uso.

3.1.5.4.1 Tipos de unidades

Nombre de Escenario 1.1	Añadir un nuevo tipo de unidad
Precondiciones	
Postcondiciones	1. El tipo de unidad queda guardado
Excepciones	1. No se han rellenado los campos obligatorios.
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Administrador pulsa sobre botón crear tipo de unidad. 2. Administrador introduce los datos correspondientes al tipo de unidad. 3. Administrador pulsa sobre el botón guardar

Tabla 5: Escenario "Añadir nuevo tipo de unidad"

Nombre de Escenario 1.2	Editar un tipo de unidad
Precondiciones	1. El tipo de unidad existe
Postcondiciones	1. El tipo de unidad queda guardado con los nuevos datos
Excepciones	1. No se han rellenado los campos obligatorios
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Administrador selecciona el tipo de unidad que quiere editar. 2. Administrador modifica los campos que cree necesarios. 3. Administrador pulsa sobre el botón guardar

Tabla 6: Escenario "Editar tipo de unidad"

Nombre de Escenario 1.3	Mostrar un tipo de unidad
Precondiciones	1. El tipo de unidad existe
Postcondiciones	
Excepciones	
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Administrador selecciona el tipo de unidad que quiere ver 2. Administrador pulsa sobre botón mostrar

Tabla 7: Escenario "Mostrar tipo de unidad"

Nombre de Escenario 1.4	Borrar un tipo de unidad
Precondiciones	1. El tipo de unidad existe
Postcondiciones	1. El tipo de unidad deja de existir
Excepciones	1. Existen unidades de el tipo de unidad y no se puede borrar.
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Administrador selecciona el tipo de unidad que quiere borrar 2. Administrador pulsa sobre botón borrar 3. Administrador confirma que desea borrar

Tabla 8: Escenario "Borrar tipo de unidad"

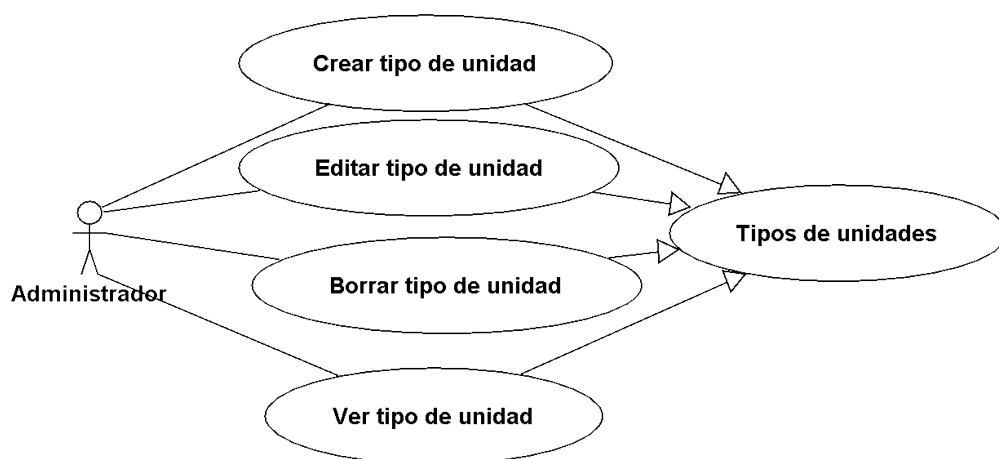


Ilustración 7: Escenarios de tipo de unidad

3.1.5.4.2 Unidades

Nombre de Escenario 2.1	Añadir una nuevo unidad
Precondiciones	
Postcondiciones	1. La unidad queda guardada
Excepciones	1. No se han rellenado los campos obligatorios.
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> Administrador pulsa sobre botón crear unidad. Administrador introduce los datos correspondientes a la unidad. Administrador pulsa sobre el botón guardar

Tabla 9: Escenario "Añadir nueva unidad"

Nombre de Escenario 2.2	Editar una unidad
Precondiciones	1. La unidad existe
Postcondiciones	1. La unidad queda guardada con los nuevos datos
Excepciones	1. No se han rellenado los campos obligatorios
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> Administrador selecciona la unidad que quiere editar. Administrador modifica los campos que cree necesarios. Administrador pulsa sobre el botón guardar

Tabla 10: Escenario "Editar unidad"

Nombre de Escenario 2.3	Mostrar una unidad
Precondiciones	1. La unidad existe
Postcondiciones	
Excepciones	
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> Administrador selecciona el unidad que quiere ver Administrador pulsa sobre botón mostrar

Tabla 11: Escenario "Mostrar unidad"

Nombre de Escenario 2.4	Borrar una unidad
Precondiciones	1. La unidad existe
Postcondiciones	1. La unidad deja de existir
Excepciones	1. La unidad se está usando en la aplicación y no se puede borrar
Iniciado y finalizado por	Administrador
Detalle de operaciones	1. Administrador selecciona la unidad que quiere borrar 2. Administrador pulsa sobre botón borrar 3. Administrador confirma que desea borrar

Tabla 12: Escenario "Borrar unidad"

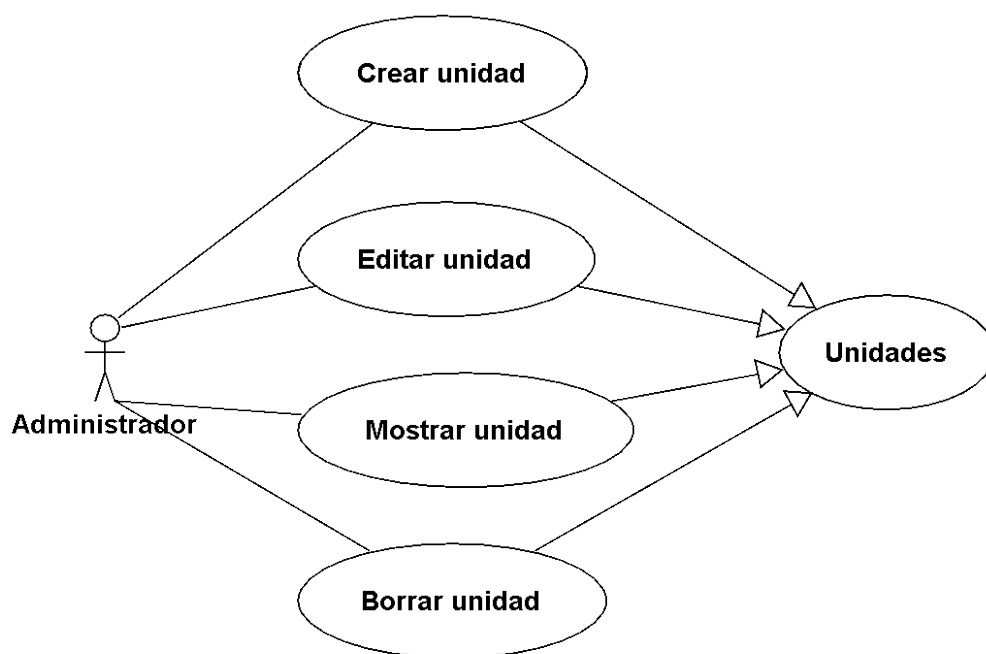


Ilustración 8: Escenarios de unidad

3.1.5.4.3 Factores de conversión

Nombre de Escenario 3.1	Añadir un nuevo factor de conversión
Precondiciones	
Postcondiciones	1. El factor de conversión queda guardado
Excepciones	1. El factor de conversión entre unidades ya existe
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Administrador pulsa sobre botón crear factor de conversión. 2. Administrador introduce los datos correspondientes al factor de conversión. 3. Administrador pulsa sobre el botón guardar

Tabla 13: Escenario "Añadir nuevo factor de conversión"

Nombre de Escenario 3.2	Editar un factor de conversión
Precondiciones	1. El factor de conversión existe
Postcondiciones	1. El factor de conversión queda guardado con los nuevos datos
Excepciones	<ol style="list-style-type: none"> 1. No se han rellenado los campos obligatorios 2. El factor de conversión entre unidades ya existe
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Administrador selecciona el factor de conversión que quiere editar. 2. Administrador modifica los campos que cree necesarios. 3. Administrador pulsa sobre el botón guardar

Tabla 14: Escenario "Editar factor de conversión"

Nombre de Escenario 3.3	Mostrar un factor de conversión
Precondiciones	1. El factor de conversión existe
Postcondiciones	
Excepciones	
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Administrador selecciona el factor de conversión que quiere ver 2. Administrador pulsa sobre botón mostrar

Tabla 15: Escenario "Mostrar factor de conversión"

Nombre de Escenario 3.4	Borrar factor de conversión
Precondiciones	1. El factor de conversión existe
Postcondiciones	1. El factor de conversión deja de existir
Excepciones	
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Administrador selecciona el factor de conversión que quiere borrar 2. Administrador pulsa sobre botón borrar 3. Administrador confirma que desea borrar

Tabla 16: Escenario "Borrar factor de conversión"

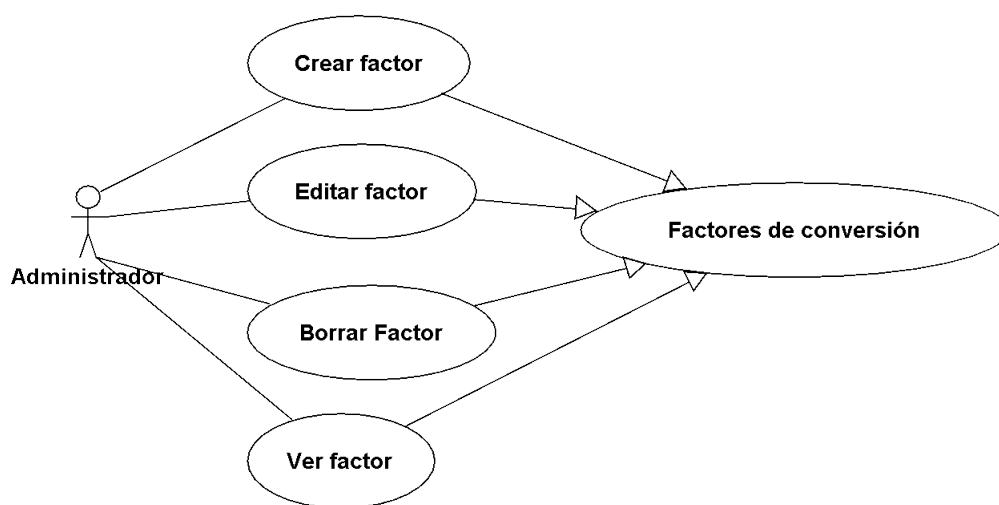


Ilustración 9: Escenarios de factor de conversión

3.1.5.4.4 Usuarios

Nombre de Escenario 4.1	Añadir un nuevo usuario
Precondiciones	
Postcondiciones	2. El usuario queda guardado
Excepciones	1. El correo electrónico es igual al de otro usuario 2. No se han rellenado los campos obligatorios.
Iniciado y finalizado por	Administrador
Detalle de operaciones	1. Administrador pulsa sobre botón crear usuario. 2. Administrador introduce los datos correspondientes al usuario. 3. Administrador pulsa sobre el botón guardar

Tabla 17: Escenario "Añadir nuevo usuario"

Nombre de Escenario 4.2	Editar un usuario
Precondiciones	1. El usuario existe
Postcondiciones	1. El usuario queda guardado con los nuevos datos
Excepciones	1. No se han rellenado los campos obligatorios 2. El usuario no es administrador o no son sus propios datos los que quiere consultar.
Iniciado y finalizado por	Administrador o usuario
Detalle de operaciones	1. Usuario selecciona el usuario que quiere editar. 2. Usuario modifica los campos que cree necesarios. 3. Usuario pulsa sobre el botón guardar

Tabla 18: Escenario "Editar usuario"

Nombre de Escenario 4.3	Mostrar un usuario
Precondiciones	1. El usuario existe
Postcondiciones	
Excepciones	1. El usuario no es administrador o no son sus propios datos los que quiere consultar.
Iniciado y finalizado por	Administrador o usuario
Detalle de operaciones	1. Usuario selecciona el usuario que quiere ver 2. Usuario pulsa sobre botón mostrar

Tabla 19: Escenario "Mostrar usuario"

Nombre de Escenario 4.4	Borrar un usuario
Precondiciones	1. El usuario existe
Postcondiciones	1. El usuario deja de existir
Excepciones	
Iniciado y finalizado por	Administrador
Detalle de operaciones	<ol style="list-style-type: none"> Administrador selecciona el usuario que quiere ver Administrador pulsa sobre botón mostrar

Tabla 20: Escenario "Borrar usuario"

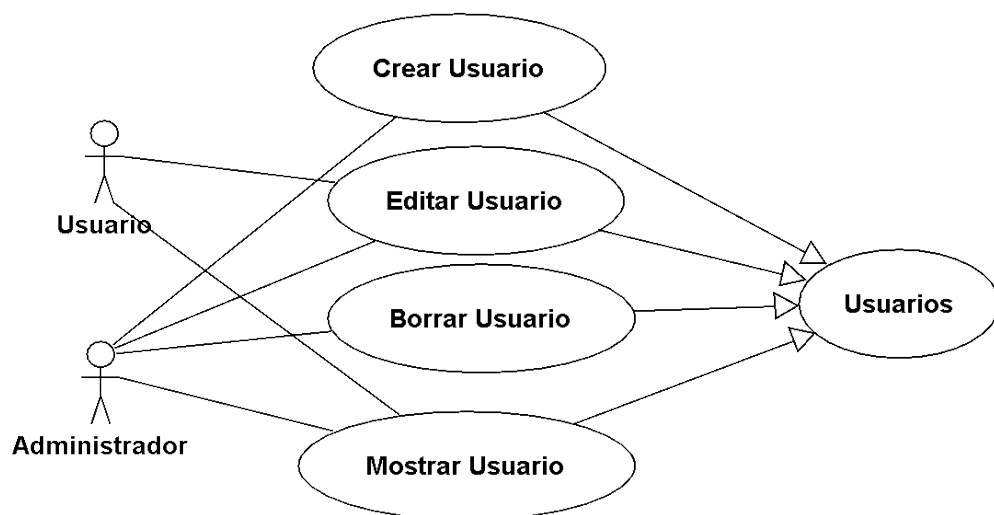


Ilustración 10: Escenarios de Usuario

3.1.5.4.5 Temporadas

Nombre de Escenario 5.1	Añadir una nueva temporada
Precondiciones	1. El deportista al que se le va a asignar existe
Postcondiciones	1. El temporada queda guardada
Excepciones	1. Las fechas se solapan. 2. El nombre de la temporada coincide con otro del mismo deportista. 3. El deportista no pertenece al entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el deportista al que pertenece la temporada. 2. Entrenador introduce los datos correspondientes a la temporada. 3. Entrenador pulsa sobre el botón guardar

Tabla 21: Escenario "Añadir nueva temporada"

Nombre de Escenario 5.2	Editar una temporada
Precondiciones	1. La temporada existe
Postcondiciones	1. La temporada queda guardado.
Excepciones	1. El nombre de la temporada coincide con otro del mismo deportista. 2. Las fechas se solapan con otras temporadas del mismo deportista. 3. Las fechas cambiadas alteran los elementos contenidos en el periodo. 4. El atleta no pertenece al entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona la temporada que quiere editar. 2. Entrenador modifica los campos que cree necesarios. 3. Entrenador pulsa sobre el botón guardar

Tabla 22: Escenario "Editar temporada"

Nombre de Escenario 5.3	Borrar una temporada
Precondiciones	1. La temporada existe
Postcondiciones	1. La temporada deja de existir.
Excepciones	1. Existen elementos contenidos en el periodo, la temporada no se puede borrar. 2. La temporada no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona la temporada que quiere borrar. 2. Entrenador pulsa sobre botón. 3. El sistema pregunta asegurándose de su acción..

Tabla 23: Escenario "Borrar temporada"

Nombre de Escenario 5.4	Mostrar una temporada
Precondiciones	1. La temporada existe
Postcondiciones	
Excepciones	1. La temporada no pertenece a un atleta del entrenador.
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona la temporada que quiere ver 2. Entrenador pulsa sobre botón mostrar

Tabla 24: Escenario "Mostrar temporada"

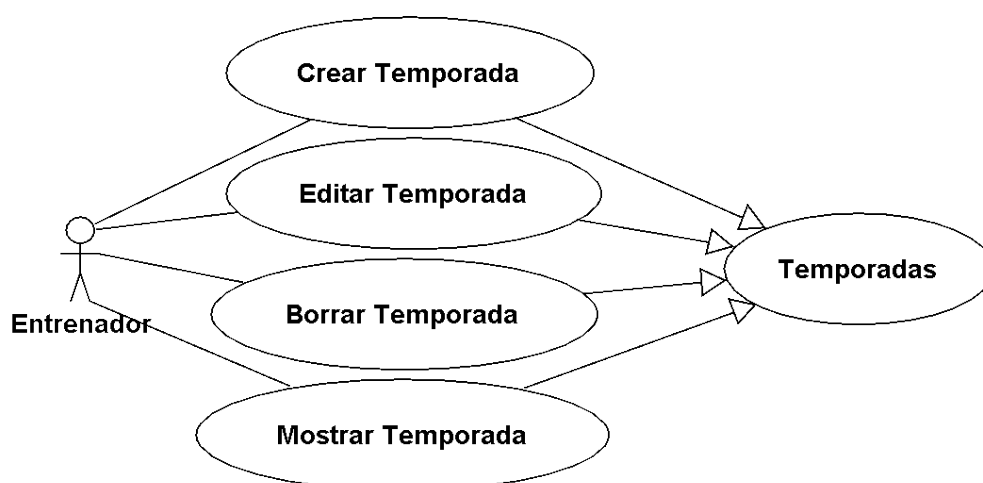


Ilustración 11: Escenarios de temporada

3.1.5.4.6 *Macro Ciclos*

Nombre de Escenario 6.1	Añadir un nuevo macrociclo
Precondiciones	2. La temporada que lo contiene existe
Postcondiciones	1. El macrociclo queda guardado
Excepciones	<ol style="list-style-type: none"> 1. Las fechas se solapan. 2. El nombre del macrociclo coincide con otro de la misma temporada. 3. El macrociclo no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Entrenador selecciona el periodo donde se ubica el macrociclo. 2. Entrenador introduce los datos correspondientes al macrociclo. 3. Entrenador pulsa sobre el botón guardar

Tabla 25: Escenario "Añadir nuevo macrociclo"

Nombre de Escenario 6.2	Editar un macrociclo
Precondiciones	1. El macrociclo existe
Postcondiciones	1. El macrociclo queda guardado.
Excepciones	<ol style="list-style-type: none"> 1. El nombre del macrociclo coincide con otro de la misma temporada. 2. Las fechas se solapan con otros macrociclos de la misma temporada. 3. Las fechas cambiadas alteran los elementos contenidos en el periodo. 4. El macrociclo no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Entrenador selecciona el macrociclo que quiere editar. 2. Entrenador modifica los campos que cree necesarios. 3. Entrenador pulsa sobre el botón guardar

Tabla 26: Escenario "Editar macrociclo"

Nombre de Escenario 6.3	Borrar un macrociclo
Precondiciones	2. El macrociclo existe
Postcondiciones	1. El macrociclo deja de existir.
Excepciones	1. Existen elementos contenidos en el periodo, el macrociclo no se puede borrar. 2. El macrociclo no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el macrociclo que quiere borrar. 2. Entrenador pulsa sobre botón. 3. El sistema pregunta asegurándose de su acción..

Tabla 27: Escenario "Borrar macrociclo"

Nombre de Escenario 6.4	Mostrar un macrociclo
Precondiciones	1. El macrociclo existe
Postcondiciones	
Excepciones	1. El macrociclo no pertenece a un atleta del entrenador.
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el macrociclo que quiere ver 2. Entrenador pulsa sobre botón mostrar

Tabla 28: Escenario "Mostrar macrociclo"

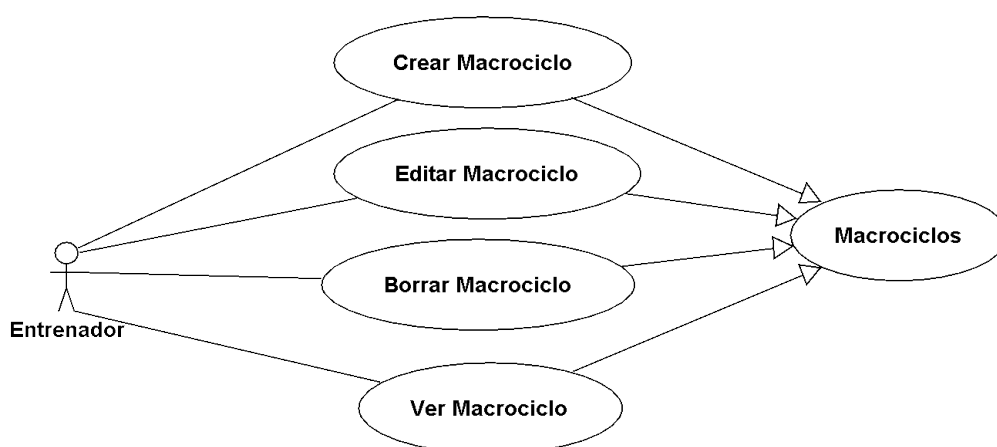


Ilustración 12: Escenarios de Macro ciclo

3.1.5.4.7 Mesociclos

Nombre de Escenario 7.1	Añadir un nuevo mesociclo
Precondiciones	1. El macrociclo que lo contiene existe
Postcondiciones	1. El mesociclo queda guardado
Excepciones	1. Las fechas se solapan. 2. El nombre del mesociclo coincide con otro del mismo macrociclo 3. El mesociclo no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el periodo donde se ubica el mesociclo. 2. Entrenador introduce los datos correspondientes al mesociclo. 3. Entrenador pulsa sobre el botón guardar

Tabla 29: Escenario "Añadir nuevo mesociclo"

Nombre de Escenario 7.2	Editar un mesociclo
Precondiciones	1. El mesociclo existe
Postcondiciones	1. El mesociclo queda guardado.
Excepciones	1. El nombre del mesociclo coincide con otro del mismo macrociclo. 2. Las fechas se solapan con otros mesociclos del mismo macrociclo. 3. Las fechas cambiadas alteran los elementos contenidos en el periodo. 4. El mesociclo no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el mesociclo que quiere editar. 2. Entrenador modifica los campos que cree necesarios. 3. Entrenador pulsa sobre el botón guardar

Tabla 30: Escenario "Editar mesociclo"

Nombre de Escenario 7.3	Borrar un mesociclo
Precondiciones	1. El mesociclo existe
Postcondiciones	1. El mesociclo deja de existir.
Excepciones	1. Existen elementos contenidos en el periodo, el mesociclo no se puede borrar. 2. El mesociclo no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el mesociclo que quiere borrar. 2. Entrenador pulsa sobre botón. 3. El sistema pregunta asegurándose de su acción..

Tabla 31: Escenario "Borrar mesociclo"

Nombre de Escenario 7.4	Mostrar un mesociclo
Precondiciones	1. El mesociclo existe
Postcondiciones	
Excepciones	1. El mesociclo no pertenece a un atleta del entrenador.
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el mesociclo que quiere ver 2. Entrenador pulsa sobre botón mostrar

Tabla 32: Escenario "Mostrar mesociclo"

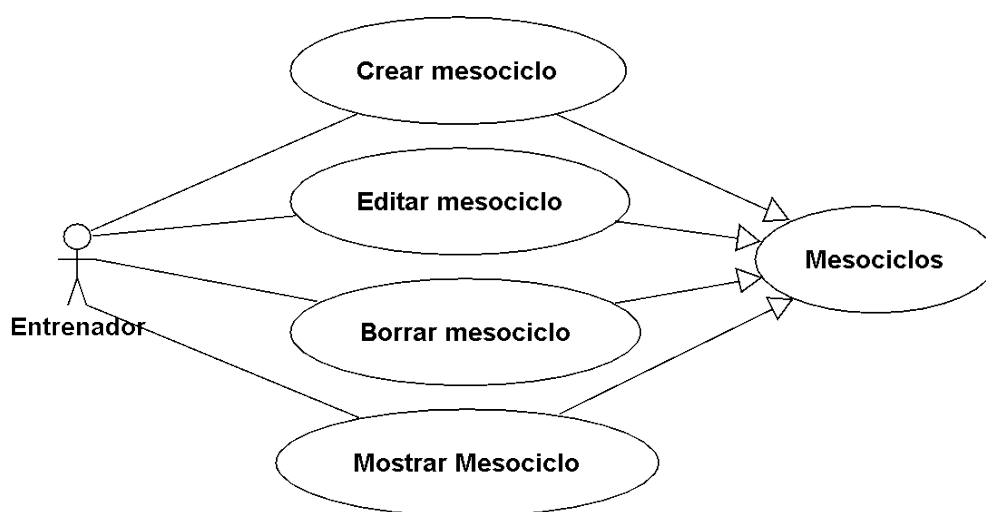


Ilustración 13: Escenarios de mesociclo

3.1.5.4.8 Microciclos

Nombre de Escenario 8.1	Añadir un nuevo microciclo
Precondiciones	2. El mesociclo que lo contiene existe
Postcondiciones	1. El microciclo queda guardado
Excepciones	<ol style="list-style-type: none"> 1. Las fechas se solapan. 2. El nombre del microciclo coincide con otro del mismo mesociclo 3. El microciclo no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Entrenador selecciona el periodo donde se ubica el microciclo. 2. Entrenador introduce los datos correspondientes al microciclo. 3. Entrenador pulsa sobre el botón guardar

Tabla 33: Escenario "Añadir nuevo microciclo"

Nombre de Escenario 8.2	Editar un microciclo
Precondiciones	1. El microciclo existe
Postcondiciones	1. El microciclo queda guardado.
Excepciones	<ol style="list-style-type: none"> 1. El nombre del microciclo coincide con otro del mismo mesociclo. 2. Las fechas se solapan con otros microciclos del mismo mesociclo. 3. Las fechas cambiadas alteran los elementos contenidos en el periodo. 4. El microciclo no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Entrenador selecciona el microciclo que quiere editar. 2. Entrenador modifica los campos que cree necesarios. 3. Entrenador pulsa sobre el botón guardar

Tabla 34: Escenario "Editar microciclo"

Nombre de Escenario 8.3	Borrar un microciclo
Precondiciones	1. El microciclo existe
Postcondiciones	1. El microciclo deja de existir.
Excepciones	1. Existen elementos contenidos en el periodo, el microciclo no se puede borrar. 2. El microciclo no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el microciclo que quiere borrar. 2. Entrenador pulsa sobre botón. 3. El sistema pregunta asegurándose de su acción..

Tabla 35: Escenario "Borrar microciclo"

Nombre de Escenario 8.4	Mostrar un microciclo
Precondiciones	1. El microciclo existe
Postcondiciones	
Excepciones	1. El microciclo no pertenece a un atleta del entrenador.
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el microciclo que quiere ver 2. Entrenador pulsa sobre botón mostrar

Tabla 36: Escenario "Mostrar microciclo"

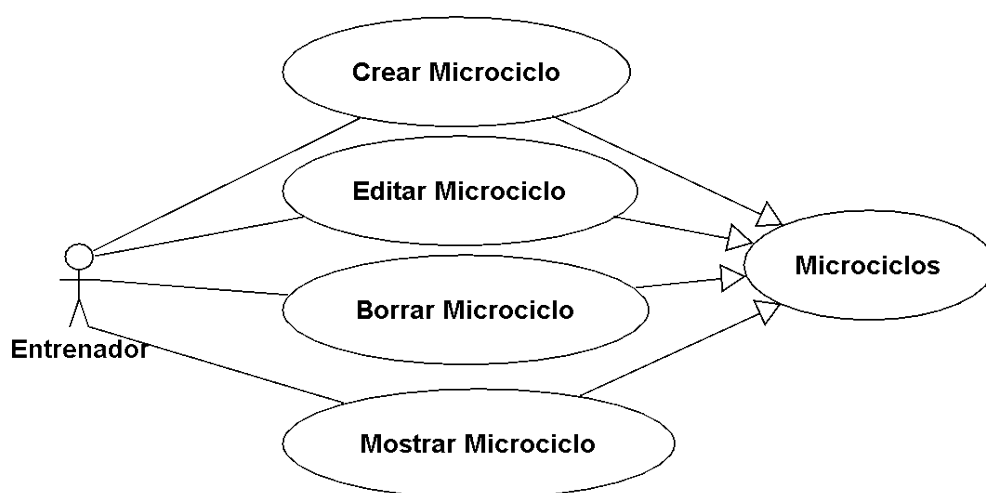


Ilustración 14: Escenarios de microciclo

3.1.5.4.9 Sesiones

Nombre de Escenario 9.1	Añadir una nuevo sesión
Precondiciones	1. El microciclo que la contiene existe
Postcondiciones	1. La sesión queda guardada
Excepciones	1. Las fechas se solapan. 2. El nombre de la sesión coincide con otro del mismo microciclo 3. La sesión no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el periodo donde se ubica la sesión. 2. Entrenador introduce los datos correspondientes al sesión. 3. Entrenador pulsa sobre el botón guardar

Tabla 37: Escenario "Añadir nuevo sesión"

Nombre de Escenario 9.2	Editar una sesión
Precondiciones	1. La sesión existe
Postcondiciones	1. La sesión queda guardada.
Excepciones	1. Las fechas cambiadas alteran las pertenencia al periodo 2. La sesión no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona la sesión que quiere editar. 2. Entrenador modifica los campos que cree necesarios. 3. Entrenador pulsa sobre el botón guardar

Tabla 38: Escenario "Editar sesión"

Nombre de Escenario 9.3	Borrar una sesión
Precondiciones	1. La sesión existe
Postcondiciones	1. La sesión deja de existir.
Excepciones	1. La sesión no pertenece a un atleta del entrenador
Iniciado y finalizado por	Entrenador
Detalle de operaciones	<ol style="list-style-type: none"> 1. Entrenador selecciona la sesión que quiere borrar. 2. Entrenador pulsa sobre botón. 3. El sistema pregunta asegurándose de su acción..

Tabla 39: Escenario "Borrar sesión"

Nombre de Escenario 9.4	Mostrar una sesión
Precondiciones	1. La sesión existe
Postcondiciones	
Excepciones	1. La sesión no pertenece a un atleta del entrenador o al propio atleta
Iniciado y finalizado por	Entrenador o atleta
Detalle de operaciones	<ol style="list-style-type: none"> 1. Usuario selecciona la sesión que quiere ver 2. Usuario pulsa sobre botón mostrar

Tabla 40: Escenario "Mostrar sesión"

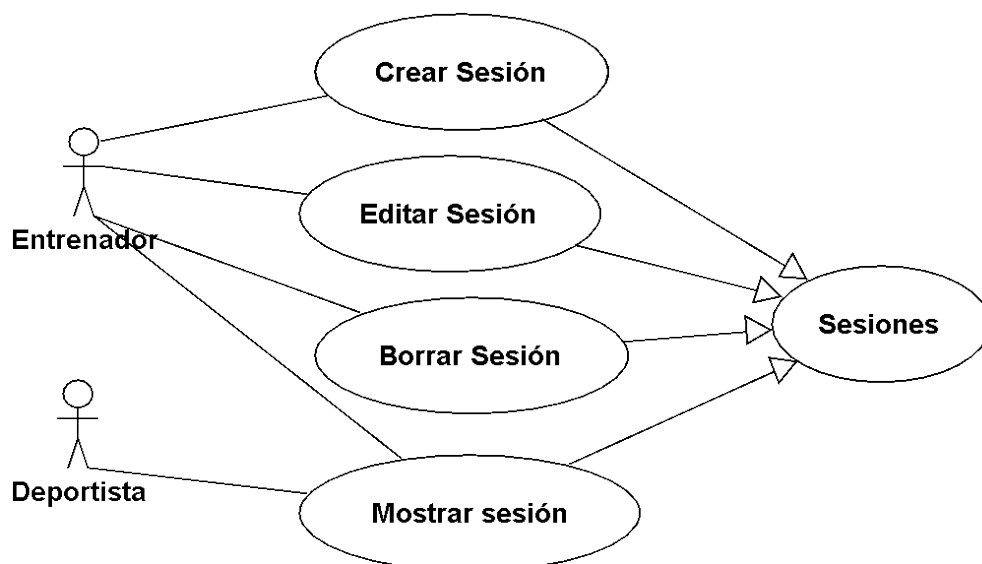


Ilustración 15: Escenarios de sesión

3.1.5.4.10 Deportistas

Nombre de Escenario 10.1	Añadir un nuevo deportista
Precondiciones	
Postcondiciones	3. El deportista queda guardado
Excepciones	1. El correo electrónico es igual al de otro usuario 2. No se han rellenado los campos obligatorios.
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador pulsa sobre botón crear deportista. 2. Entrenador introduce los datos correspondientes al deportista. 3. Entrenador pulsa sobre el botón guardar

Tabla 41: Escenario "Añadir nuevo deportista"

Nombre de Escenario 10.2	Editar un deportista
Precondiciones	1. El deportista existe
Postcondiciones	1. El deportista queda guardado con los nuevos datos
Excepciones	1. No se han rellenado los campos obligatorios 2. El deportista no pertenece al entrenador que lo quiere editar
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el deportista que quiere editar. 2. Entrenador modifica los campos que cree necesarios. 3. Entrenador pulsa sobre el botón guardar

Tabla 42: Escenario "Editar deportista"

Nombre de Escenario 10.3	Mostrar un deportista
Precondiciones	1. El deportista existe
Postcondiciones	
Excepciones	1. El deportista no pertenece al entrenador.
Iniciado y finalizado por	Entrenador
Detalle de operaciones	1. Entrenador selecciona el deportista que quiere ver 2. Entrenador pulsa sobre botón mostrar

Tabla 43: Escenario "Mostrar deportista"

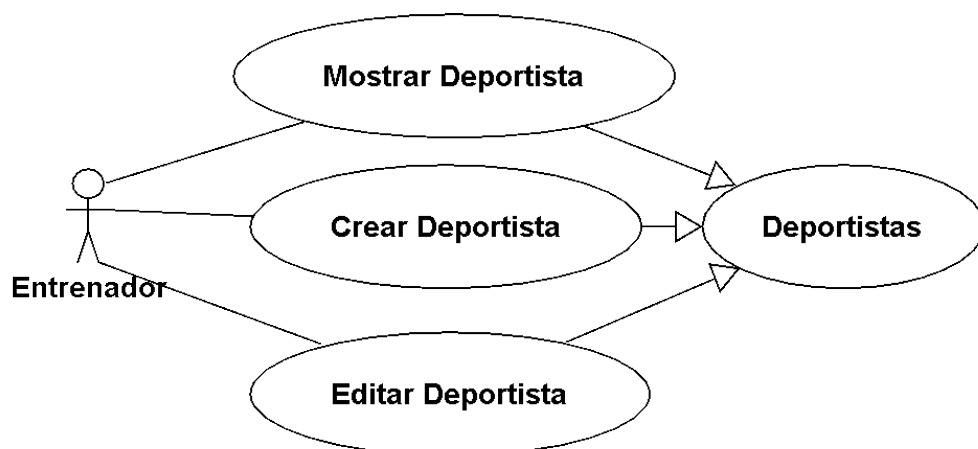


Ilustración 16: Escenarios de deportista

3.1.5.4.11 Feedback

Nombre de Escenario 11.1	Editar un feedback
Precondiciones	1. El feedback o la sesión a la que pertenecen existen
Postcondiciones	1. El feedback queda guardado con los nuevos datos
Excepciones	1. No se han rellenado los campos obligatorios 2. El feedback no pertenece al deportista que lo quiere editar
Iniciado y finalizado por	Deportista
Detalle de operaciones	1. Deportista selecciona el feedback o sesión que quiere editar. 2. Deportista modifica los campos que cree necesarios. 3. Deportista pulsa sobre el botón guardar

Tabla 44: Escenario "Editar feedback"

Nombre de Escenario 11.2	Mostrar un feedback
Precondiciones	1. El feedback existe
Postcondiciones	
Excepciones	1. El feedback no pertenece al deportista o al entrenador del deportista.
Iniciado y finalizado por	Entrenador o deportista
Detalle de operaciones	1. Usuario selecciona el feedback que quiere ver 2. Usuario pulsa sobre botón mostrar

Tabla 45: Escenario "Mostrar feedback"

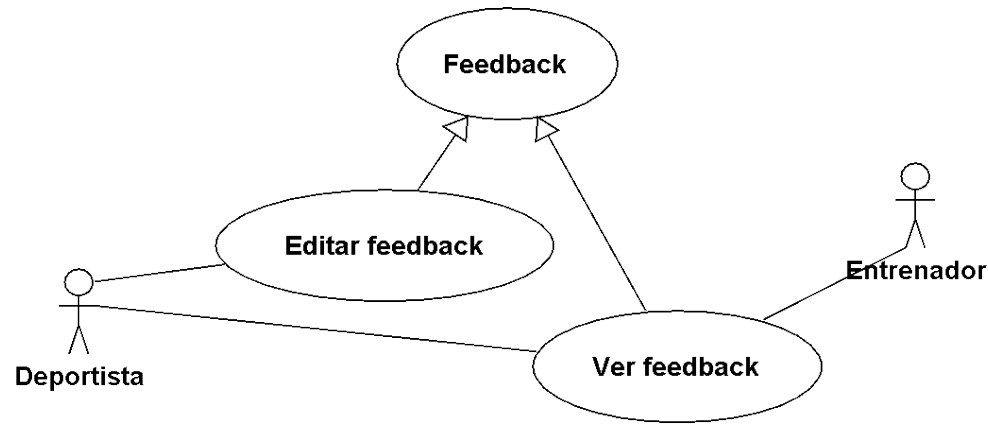


Ilustración 17: Escenarios feedback

3.1.5.4.12 Entrada y salida

Nombre de Escenario 12.1	Entrar al sistema
Precondiciones	1. El usuario que hace la entrada existe
Postcondiciones	1. El usuario puede usar al aplicación de acuerdo a sus credenciales
Excepciones	1. La contraseña está equivocada
Iniciado y finalizado por	Usuario
Detalle de operaciones	1. Usuario pulsa en el botón acceder al sistema 2. Usuario introduce nombre y contraseña 3. Usuario Pulsa botón acceder al sistema

Tabla 46: Escenario "Entrar al sistema"

Nombre de Escenario 12.2	Salir del sistema
Precondiciones	1. El usuario está en el sistema
Postcondiciones	1. El usuario está fuera del sistema, su sesión desconectada y para volver a recabar información de acuerdo a sus permisos tiene que entrar en el sistema
Excepciones	
Iniciado y finalizado por	Usuario
Detalle de operaciones	1. Usuario pulsa en el botón Salir 2. Usuario confirma que quiere salir

Tabla 47: Escenario "Salir del sistema"

Nombre de Escenario 12.3	Solicitud de contraseña
Precondiciones	1. El usuario que hace la solicitud existe
Postcondiciones	1. El usuario recibe una nueva contraseña generada aleatoriamente en su dirección personal de correo electrónico
Excepciones	1. El correo electrónico está equivocado
Iniciado y finalizado por	Usuario
Detalle de operaciones	1. Usuario pulsa en el botón acceder al sistema 2. Usuario pulsa sobre el botón recordar contraseña 3. Usuario introduce correo electrónico 4. Usuario Pulsa botón Recordar

Tabla 48: Escenario "Recordar Contraseña"

Nombre de Escenario 12.4	Solicitar usuario
Precondiciones	1. El usuario no forma parte del sistema
Postcondiciones	1. El administrador recibe un correo para que de de alta al nuevo usuario
Excepciones	1. Dirección de correo errónea 2. Falta algún dato
Iniciado y finalizado por	Usuario de internet
Detalle de operaciones	1. Usuario pulsa en el botón Registrarse 2. Usuario rellena los datos 3. Usuario pulsa sobre el botón Solicitar

Tabla 49: Escenario "Solicitar usuario"

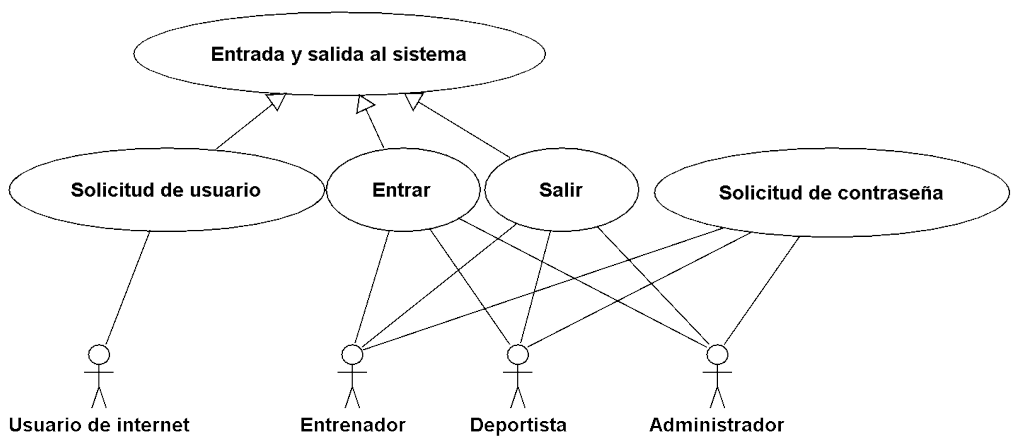


Ilustración 18: Escenarios de entrada y salida del sistema

3.2 Diseño de la solución final

3.2.1 Diseño de la infraestructura (diagramas de despliegue)

A continuación se presentan los diferentes diagramas de la infraestructura sin virtualización (Ilustración 2) y virtualizada (Ilustración 3).

Se ha valorado la inclusión de diferentes sistemas de réplica de la base de datos. En el caso de que alguna de ellas fallara las demás responderían. Aún así se ha desestimado, pues esta opción requiere una configuración, seguimiento y mantenimiento mayores y preferimos delegar la seguridad de los datos en discos con sistemas de ficheros RAID invisibles tanto al SGBD como al sistema operativo. En el caso de un crecimiento exponencial de la cantidad de datos acumulados se optaría por una granja de servidores de gestión de base de datos con un sistema de balanceo de carga.

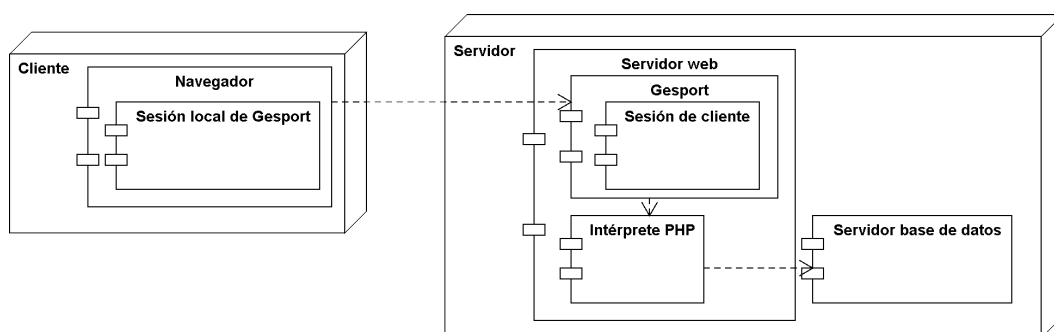


Ilustración 19: Infraestructura de un solo servidor sin virtualización

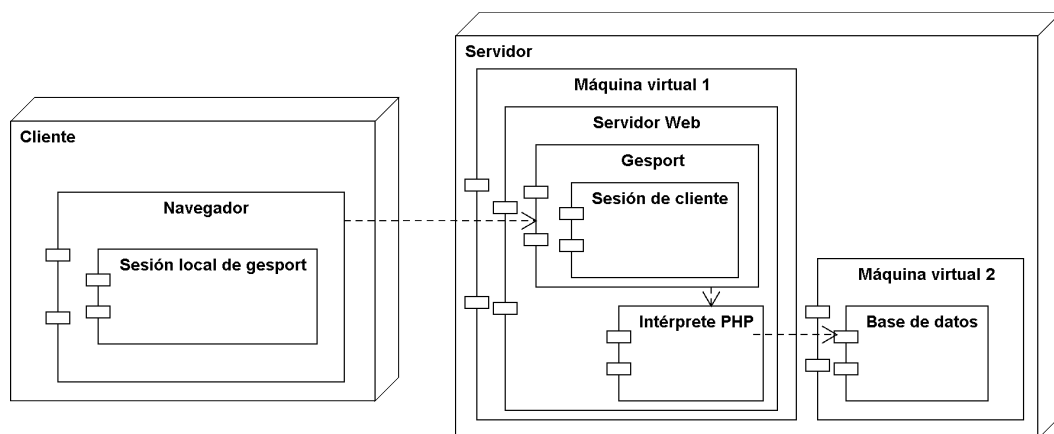


Ilustración 20: Infraestructura virtualizada

A pesar de lo representado en la ilustración 3, que es el ideal de separación de datos, pudiendo en cualquier momento reemplazar una máquina virtual por una real, ganando en escalabilidad y seguridad, para ofrecer un servicio de calidad existen otras opciones que no dan mal resultado y que se basan en un solo servidor.

Estas soluciones se pueden conseguir con un solo servidor Linux con RAID, para duplicidad de datos, con las particiones realizadas de una manera que permitan la extracción de los datos importantes sin afectar al funcionamiento. Esto significaría tener una partición principal del sistema / con las aplicaciones de sistema, una partición de sistema /var, con los datos variables, entre los que incluimos los de la base de datos y una última partición /opt, donde instalaríamos la aplicación.

3.2.2 Diseño de las clases de objetos

Tomando como base las clases de análisis se realiza un proceso de diseño de las aplicación basada en clases de objetos efectivas en el programa. En este diseño ya se tienen en cuenta las propiedades del lenguaje de programación elegido. Además de las clases que intervienen en el dominio del problema habría que definir las clases auxiliares de almacenamiento de datos, de ventanas y paneles del interfaz gráfico de usuarios o de intercambio de mensajes complejos entre clases del dominio. Estas clases vendrán descritas en el diccionario de clases efectivas. En estos diagramas de clases se han omitido los "getters" y "setters", y otros accesos predeterminados a objetos contenidos para centrarnos tan solo en las operaciones que afectan al funcionamiento real de los elementos que componen el sistema.

3.2.2.1 Diccionario de clases de diseño

Clase	Admin		
	Ancestros:	User	
	Atributos:		
	none		
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	Admin	Admin
	Documentación adicional:		
	Al no soportar correctamente la herencia Propel 1.2 se incluirán en la clase ancestral, que tendrá atributos para establecer el tipo.		

Tabla 50: Clase Admin, representa al administrador de la aplicación

Clase	Article		
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	integer	authorId
	protected	date	createdAt
	protected	date	updatedAt
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	Article	Article

Tabla 51: Clase Article, para los artículos que se muestran en la página de inicio

Clase	ArticleI18N		
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	string	culture
	protected	string	title
	protected	string	text
	Métodos:		
	none		

Tabla 52: Clase ArticleI18N, para la internacionalización de los artículos

Clase

Athlete		
Ancestros:	User	
Atributos:		
Visibilidad	Tipo	Nombre
protected	integer	coachId
protected	integer	isMale
Métodos:		
Visibilidad	Resultado	Nombre
public	Athlete	Athlete
Documentación adicional:		
Al no soportar correctamente la herencia Propel 1.2 se incluirán en la clase ancestral, que tendrá atributos para establecer el tipo.		

Tabla 53: Clase Athlete, que representa a los deportistas

Clase	Coach		
	Ancestros: User		
	Atributos:		
	none		
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	Coach	Coach
	Documentación adicional:		
	Al no soportar correctamente la herencia Propel 1.2 se incluirán en la clase ancestral, que tendrá atributos para establecer el tipo.		

Tabla 54: Clase Coach, que representa a los entrenadores

Clase	ConversionFactor		
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	integer	originId
	protected	integer	destinyId
	protected	double	conversionFactor
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	ConversionFactor	ConversionFactor
	Documentación adicional:		
	Factor de conversión entre diferentes unidades de medida, para poder visualizar la información numérica en una escala correcta.		

Tabla 55: Clase ConversionFactor, para la transformación entre unidades

Clase	Macrocycle		
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	string	description
	protected	date	startAt
	protected	date	lastDayAt
	protected	integer	seasonId
	protected	double	theoricVolume
	protected	double	reportedVolume
	protected	integer	volumeUnitId
	protected	double	theoricIntensity1
	protected	double	theoricIntensity2
	protected	double	theoricIntensity3
	protected	double	reportedIntensity1
	protected	double	reportedIntensity2
	protected	double	reportedIntensity3
	protected	integer	intensityUnitId
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	Macrocycle	Macrocycle
	public	double	getIntensity1HighLimit
	public	double	getIntensity2HighLimit
	public	double	getIntensity3HighLimit
	public	double	getIntensity1LowLimit
	public	double	getIntensity2LowLimit
	public	double	getIntensity3LowLimit
	public	Season	getSeason
	Documentación adicional:		
	Representa el macrociclo de la clase de análisis		

Tabla 56: Clase Macrocycle. Representación del ciclo mayor con objetivos

Clase	Mesocycle		
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	string	description
	protected	date	startAt
	protected	date	lastDayAt
	protected		macrocycleId
	protected	double	theoricVolume
	protected	double	reportedVolume
	protected	integer	volumeUnitId
	protected	integer	intensityUnitId
	protected	double	theoricIntensity1
	protected	double	theoricIntensity2
	protected	double	theoricIntensity3
	protected	double	reportedIntensity1
	protected	double	reportedIntensity2
	protected	double	reportedIntensity3
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	Mesocycle	Mesocycle
	public	double	getIntensity1HighLimit
	public	double	getIntensity2HighLimit
	public	double	getIntensity3HighLimit
	public	double	getIntensity1lowLimit
	public	double	getIntensity2lowLimit
	public	double	getIntensity3lowLimit
	Documentación adicional:		
	Representa el mesociclo de la clase de análisis		

Tabla 57: Clase Mesocycle, que determina los periodos para alcanzar los objetivos

Clase	Microcycle		
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	string	description
	protected	date	startAt
	protected	date	lastDayAt
	protected	double	theoricVolume
	protected	double	reportedVolume
	protected	integer	volumeUnitId
	protected	integer	intensityUnitId
	protected	double	theoricIntensity1
	protected	double	theoricIntensity2
	protected	double	theoricIntensity3
	protected	double	reportedIntensity1
	protected	double	reportedIntensity2
	protected	double	reportedIntensity3
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	Microcycle	Microcycle
	public	double	getIntensity1lowLimit
	public	double	getIntensity2lowLimit
	public	double	getIntensity3lowLimit
	public	double	getIntensity1HighLimit
	public	double	getIntensity2HighLimit
	public	double	getIntensity3HighLimit
	public	Season	getSeason
	public	void	saveAndCreate
	Documentación adicional:		
	Representa el microciclo de la clase de análisis. La función saveAndCreate salva el microciclo y añade todas las sesiones que le corresponden por fecha.		

Tabla 58: Clase Microcycle, que describe los trabajos aproximadamente semanales

Clase	Season		
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	integer	description
	protected	date	startAt
	protected	date	lastDayAt
	protected	integer	userId
	protected	integer	volumeUnitTypeId
	protected	integer	intensityUnitTypeId
	protected	double	theoricVolume
	protected	double	reportedVolume
	protected	integer	volumeUnitId
	protected	integer	intensityUnitId
	protected	double	theoricIntensity1
	protected	double	reportedIntensity1
	protected	double	theoricIntensity2
	protected	double	reportedIntensity2
	protected	double	theoricIntensity3
	protected	double	reportedIntensity3
	protected	double	intensity1LowLimit
	protected	double	intensity2LowLimit
	protected	double	intensity3LowLimit
	protected	double	intensity1HighLimit
	protected	double	intensity2HighLimit
	protected	double	intensity3HighLimit
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	Season	Season
	Documentación adicional:		
	Clase que representa la temporada completa, en ella se definen elementos que luego usarán otros periodos, como los limites de las diferentes intensidades.		

Tabla 59: Clase season, que representa la temporada completa de un deportista

Clase	Session		
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	string	description
	protected	string	feedback
	protected	date	date
	protected	integer	microcycleId
	protected	integer	wakeUpHr
	protected	double	theoricVolume
	protected	double	reportedVolume
	protected	integer	volumeUnitId
	protected	integer	intensityUnitId
	protected	double	reportedIntensity1
	protected	double	reportedIntensity2
	protected	double	reportedIntensity3
	protected	double	theoricIntensity1
	protected	double	theoricIntensity2
	protected	double	theoricIntensity3
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	Session	Session
	public	double	getIntensity1lowLimit
	public	double	getIntensity2lowLimit
	public	double	getIntensity3lowLimit
	public	double	getIntensity1HighLimit
	public	double	getIntensity2HighLimit
	public	double	getIntensity3HighLimit
	Documentación adicional:		
	Representa la sesión de la clase de análisis. El feedback es introducido aquí.		

Tabla 60: Clase Session, es el trabajo a realizar en un momento determinado

Clase	Unit		
	Associations:		
	Visibilidad	Tipo	properties
	public	Unit18N	Rolename: (none) Multiplicity: 0..* Navigable: true Ordering: unordered
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	string	symbol
	protected	integer	unitTypedId
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	Unit	Unit
	Documentación adicional:		
	Clase para representar las unidades concretas de medida.		

Tabla 61: Clase Unit, para las unidades de los feedback, volúmenes e intensidad

Clase	Unit18N		
	Associations:		
	Visibilidad	Tipo	properties
	public	Unit	Rolename: (none) Multiplicity: 1 Navigable: true Ordering: unordered
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	string	culture
	protected	string	description
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	void	Unit18N
	Documentación adicional:		
	Clase para la traducción de unidades, tan solo almacena el idioma y la descripción para ese idioma.		

Tabla 62: Clase Unit18N, para las traducciones de las unidades

Clase	UnitType		
	Associations:		
	Visibilidad	Tipo	properties
	public	UnitType18N	Rolename: (none) Multiplicity: 0..* Navigable: true Ordering: unordered
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	string	globalDescription
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	UnitType	UnitType
	Documentación adicional:		
	Unidad para clasificar las unidades según su tipo: longitud, tiempo, volumen, peso, número porcentaje, etc.		

Tabla 63: Clase UnitType, para agrupar los diferentes tipos de unidades

Clase	UnitType18N		
	Associations:		
	Visibilidad	Tipo	properties
	public	UnitType	Rolename: (none) Multiplicity: 1 Navigable: true Ordering: unordered
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	string	culture
	protected	string	description
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	void	UnitType18N
	Documentación adicional:		
	Clase para la internacionalización de los tipo de unidades. (Almacén de nombres según idioma)		

Tabla 64: Clase UnitType18N, para traducir los tipos de unidades

Clase	User		
	Subtypes: Admin , Coach , Athlete		
	Atributos:		
	Visibilidad	Tipo	Nombre
	protected	integer	id
	protected	string	nickname
	protected	string	firstName
	protected	string	lastName
	protected	date	createdAt
	protected	date	bornAt
	protected	string	sha1_password
	protected	string	salt
	protected	string	country
	protected	string	language
	Métodos:		
	Visibilidad	Resultado	Nombre
	public	User	User
	public	string	generatePassword
	public	void	setPassword
		parameters:	__ string password
	Documentación adicional:		
	Usuarios con capacidad de acceso al sistema		

Tabla 65: Clase User, para describir a los usuarios del sistema

3.2.2.2 Diagramas de clases

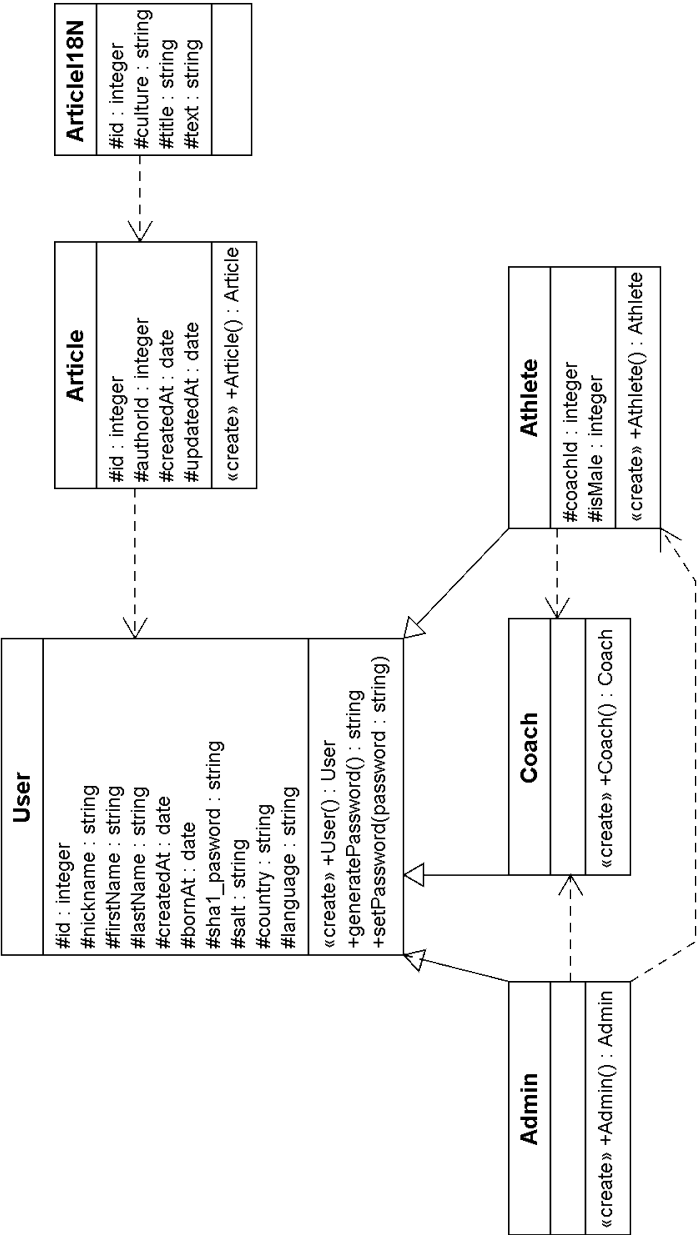


Ilustración 21: Diagrama de clases del ámbito de los usuarios y los artículos de portada

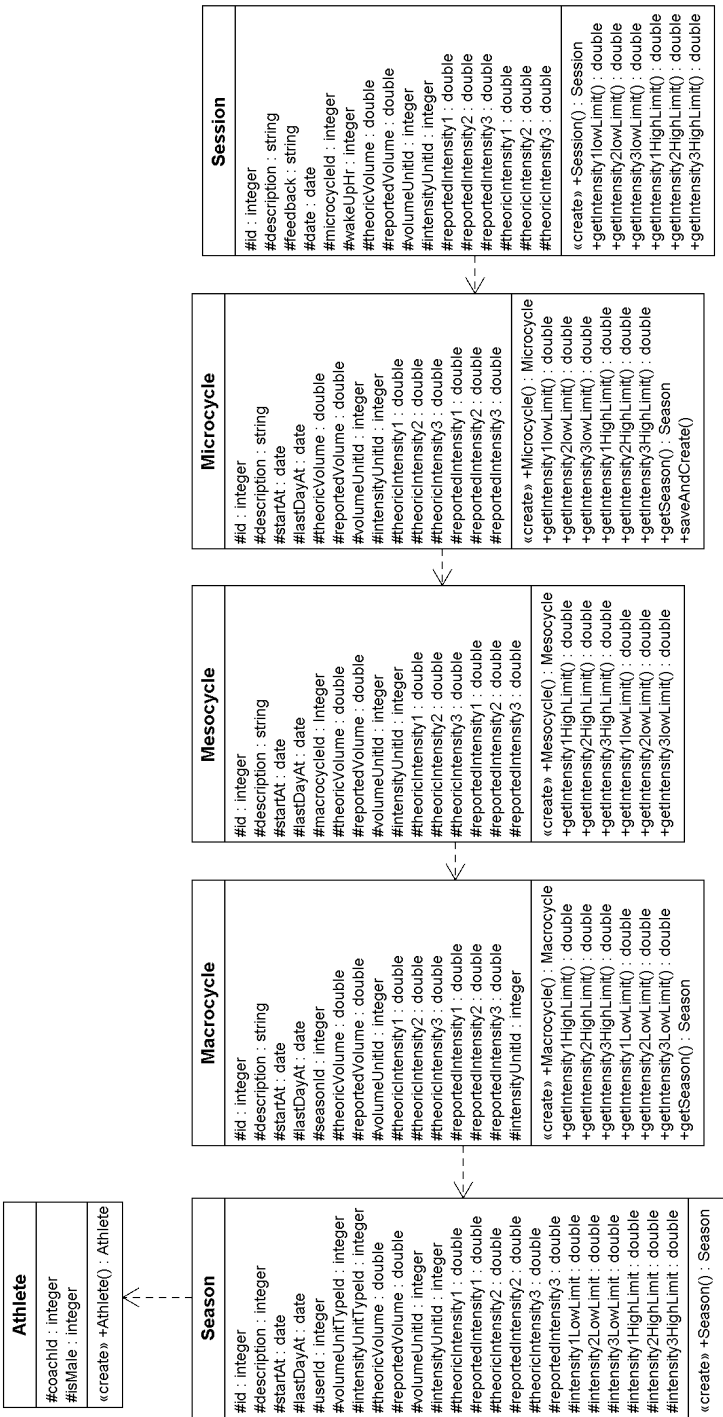


Ilustración 22: Diagrama de clases para los entrenamientos

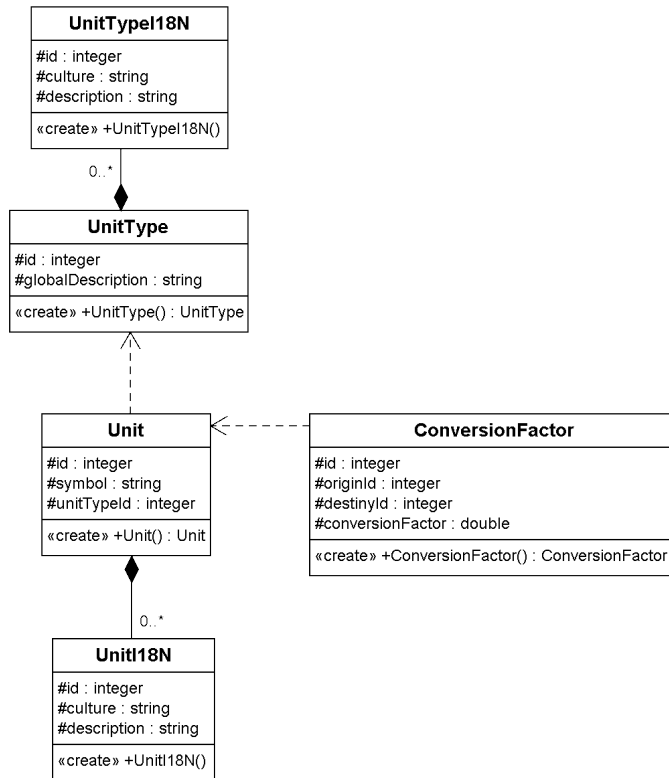


Ilustración 23: Diagrama de clases relacionadas con el sistema de unidades

3.2.3 Diseño del interfaz gráfico de usuario

La aplicación, según las especificaciones anteriores será web. Para que sea muy fácil su manejo se van a seguir los estándares marcados por la W3C en su recomendación XHTML 1.0 (26 Enero 2000) en su versión Transitional, para permitir más libertad a la hora de incluir compatibilidad con navegadores viejos.

El interfaz gráfico de usuario tendrá los siguientes elementos:

- Menú: panel para navegar entre las diferentes funcionalidades de la aplicación
- Logotipo: punto de vuelta a la página principal.
- Barra de estado dinámica: Es una barra donde se podrán poner cosas relacionadas con

la funcionalidad con la que se esté trabajando en el momento. En este caso se va a mostrar la ayuda.

- Pantalla principal de operaciones: donde se vana mostrar resultados de informes, formularios, etc.

Para decidir la posición, color, fondo, tipografía, bordes separación, flujo del texto y otros pormenores de la presentación se van a usar hojas de estilo CSS (Cascade Style Sheet).

La navegación para elegir unas funcionalidades u otras se realizará mediante clic en enlaces y botones de formularios, o mediante navegación por tabuladores. Para rellenar los formularios se podrán usar teclados, físicos o virtuales de dispositivos móviles.

Los espacios del programa original están tratados en porcentaje, de manera que, independientemente del tamaño de la pantalla, se va a ver correctamente situado cada elemento.

3.2.4 Diseño de las pruebas

Se realizarán pruebas funcionales de los diferentes módulos de la aplicación pasando parámetros límites, incluidos en el rango correcto, vacíos y fuera de rango. Estas pruebas se configurarán y automatizarán en la carpeta Gesport/test/functional, donde cada módulo del programa tendrá un fichero con los datos a comprobar.

Las pruebas unitarias se van a omitir por considerar la madurez del proyecto Symfony suficiente y porque en algunas de sus documentaciones se recomienda que las pruebas realizadas sean las funcionales como caso general, y la unitarias solo en casos muy particulares de clases no relacionadas o métodos estáticos.

Además se realizarán las siguientes pruebas al finalizar cada módulo:

- Se realizarán pruebas simultaneas de varias ejecuciones de clases con persistencia para comprobar la integridad de los datos en tiempo de ejecución, bloqueos de acceso y modificación de informaciones, y concurrencia de acceso a datos.
- Se realizarán pruebas del funcionamiento correcto del programa. Para ello se crearán datos de ejemplo desde cero en una base de datos con los datos justos para funcionar sin problemas de integridad referencial. Con esta prueba también se comprobará la seguridad de la instalación, pues el programa en su primera ejecución creará un usuario administrador con una contraseña predeterminada.
- Se realizará otra prueba de concurrencia de acceso a datos con el programa completo para comprobar que el aumento de funcionalidades no afecta al rendimiento de las clases.
- Se realizarán pruebas de funcionamiento del programa con monitores de consumos de procesador, memoria, referencias circulares, consumo de ancho de banda y otras herramientas de optimización de rendimiento de la aplicación. Para ellos se utilizarán las herramientas top, apachetop, y la lectura de los logs de Symfony. Estos datos se analizarán observando que la excepciones se hayan producido en puntos controlados.
- Si el resultado de estas pruebas es correcto se dará el visto bueno del módulo programado.

3.3 Implementación

Para la implementación se ha usado el framework Symfony. Este marco de trabajo genera ciertas clases asociadas a las principales de manera que el diagrama de clases en ejecución quedaría ligeramente diferente. Para usar estas clases en futuros desarrollos se ha creado una documentación adicional, presente en la estructura del CD, en formato HTML y navegable. Consiste en una descripción de todas las clases, ficheros, funciones, acciones, y plantillas que componen la aplicación.

Para más información sobre la documentación del API ver el apéndice de la estructura de ficheros.

A continuación se muestra el diagrama de clases de Usuario con las clases generadas por Symfony, clases auxiliares y métodos que aportan. Se han excluido todos los getters y setters de la clase ancestral BaseUser, que lo único que hacen es devolver y establecer valores en los campos determinados.

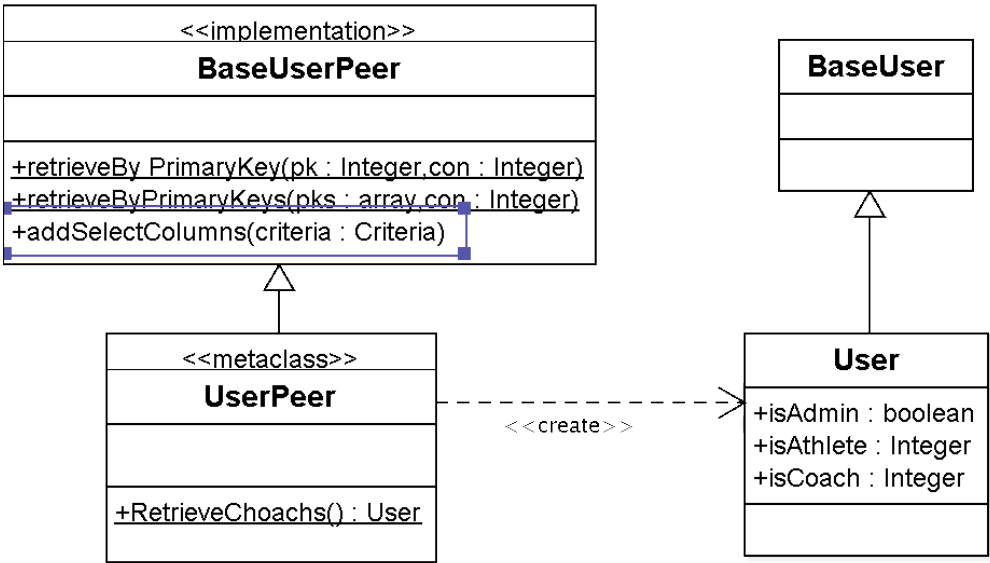


Ilustración 24: Clases adicionales generadas por Symfony

4. Manuales de usuario y administrador

4.1 Manual de instalación y administración del administrador

Advertencia: para el seguimiento de ciertos puntos de este manual es necesario conocer los conceptos básicos de administración de sistemas Linux, operaciones con bases de datos, configuración básica de servidores y conexión remota de ordenadores. Basándonos en esos principios describiremos la instalación completa en un sistema determinado con unos comandos concretos. Este manual describe la instalación de la aplicación de servidor Gesport en una máquina con arquitectura x86 o x64, utilizando la distribución Ubuntu Server LTS 8.04.1 del sistema operativo GNU/Linux, con MySQL en el mismo servidor sin virtualizar, apache 2 como servidor web en la carpeta /opt, recomendada por Debian para software opcional que no provenga de paquetes. Se puede adaptar a otras distribuciones de Linux e incluso a Windows con IIS.

4.1.1 Instalación del sistema operativo de servidor y los componentes necesarios para el funcionamiento de Gesport.

El siguiente manual se refiere a una máquina Intel. Vamos a instalar un sistema operativo Ubuntu Server con todos los programas necesarios para hacer funcionar Gesport: Servidor web apache con módulo de PHP 5 para hacer funcionar Symfony y módulos Include y Rewrite para ocultar al navegador los ficheros donde está el programa, servidor de bases de datos Mysql y por último el programa Gesport.

4.1.1.1 Instalación del sistema operativo

Para instalar Ubuntu Server LTS 8.04.1 descargamos la imagen del CD de instalación de la siguiente dirección <http://ftp.sh.cvut.cz/MIRRORS/ubuntu-releases/hardy/ubuntu-8.04.1-server-i386.iso>. Esta dirección corresponde a un servidor "mirror" (espejo) . En la web de Ubuntu <http://www.ubuntu.com/getubuntu/download> tenemos acceso a diferentes formas de descarga de este mismo sistema operativo.

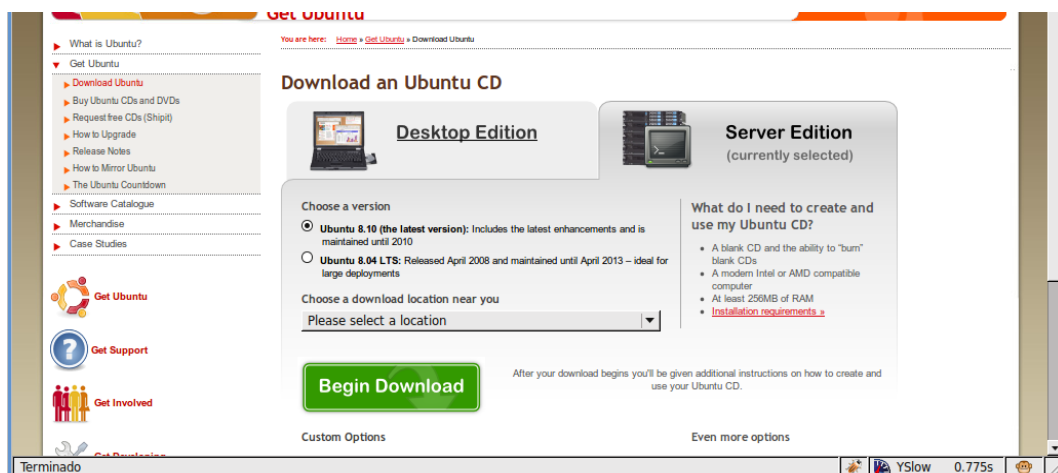


Ilustración 25: Página de descarga de Ubuntu Server

Una vez descargada la imagen la grabamos en un CD con un software de grabación de CDs disponible para nuestro sistema operativo. En el caso de que dispongamos de alguna herramienta de verificación de resumen de la imagen descargada la podemos utilizar para comprobar que el resumen de nuestro CD y el de la web de Ubuntu coinciden antes de comenzar el proceso de grabación. Esto nos asegura que el CD es el mismo que la empresa Canonical ha creado y que no tiene ningún problema de datos perdidos o vulnerabilidades provocadas. Los resúmenes están disponible en <https://help.ubuntu.com/community/UbuntuHashes> en el caso que nos ocupa es 7232c6004ba438890cd09aded162dc8e. Como vemos en la imagen del software de grabación de CDs K3B coincide:

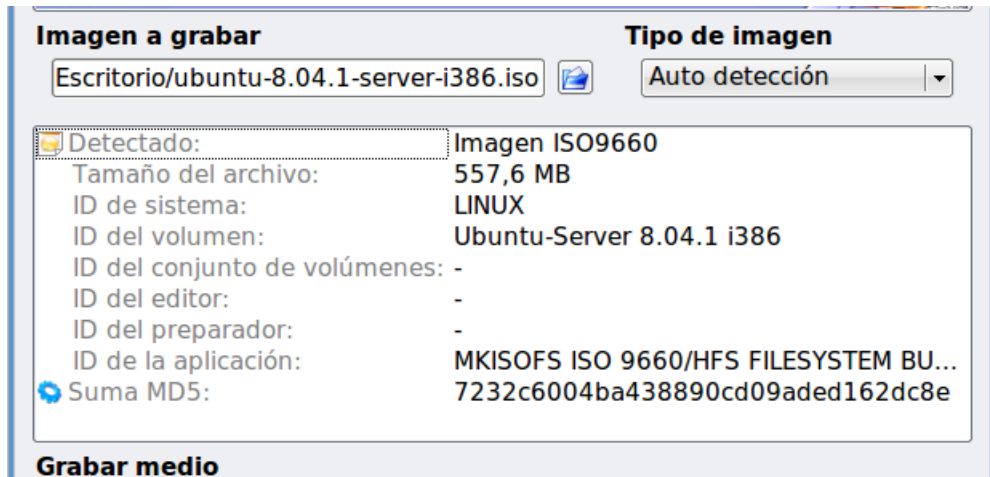


Ilustración 26: Comprobación de resumen de datos de CD de instalación

Cuando se ha grabado y etiquetado, para evitar confusiones, pasamos a instalarlo en la máquina que hará las funciones de servidor. Para ello pulsamos el botón de encendido y, siguiendo las recomendaciones del fabricante de la placa base, accedemos a la configuración de la BIOS, en el caso de que sea un sistema con BIOS, para configurar el orden de arranque y establecer el CD-ROM como dispositivo de primera lectura. Una vez configurado colocamos el CD de instalación de Ubuntu, guardamos y reiniciamos.

En el caso de que este ordenador tenga un sistema EFI, es posible que la configuración para acceder a arrancar desde el CD-ROM se consiga pulsando la tecla C en el momento previo a la carga del sistema operativo con el CD ya dentro.

Sea cual sea el caso una vez arrancado el sistema operativo nos encontramos con unos menús que nos permiten elegir las opciones de la primera ejecución previa a la instalación. Valga como ejemplo el primero, en el que se nos pregunta por el idioma (Ilustración 6). Una vez seleccionado el idioma con el teclado podemos seleccionar la acción a realizar, que en este caso será "Install Ubuntu Server". A partir de aquí comienzan varios procesos como la copia de los ficheros del CD de instalación al disco duro de nuestro servidor o la detección de los dispositivos que tenemos instalados.



Ilustración 27: Menú de selección de idioma en el proceso de instalación de Ubuntu Server

Una vez terminado esto configuramos el nombre de la máquina, "gesport" en este caso, la zona horaria, y como vamos a distribuir el espacio de nuestro disco. Para esta distribución de espacio el instalador nos da varias opciones. El "particionado guiado" o manual. Con el guiado tenemos la opción de usar LVM, que permite una redistribución posterior de nuestro espacio. Si el servidor va a estar dedicado por completo a la aplicación la recomendación es usar "particionado manual". El "particionado guiado" normal que usa todo el disco no tiene en cuenta posibles recuperaciones de sistema con particiones diferentes de datos y ejecutables. El particionado LVM nos obliga a usar un sistema de ficheros determinado, pues al ser su principal característica la movilidad de volúmenes, el sistema de ficheros tiene que tener la característica de poder crecer por delante y por detrás físicamente en el disco duro.

Los requerimientos mínimos para un funcionamiento de Gesport en cuanto a distribución de disco son los siguientes:

- Para los ficheros de arranque, sistema operativo, y ejecutables necesitamos una partición de 1Gb, Primaria, al principio del disco (excepto EFI, que se explicará a continuación), con el punto de montaje "/" y la marca de arranque activada.
- Para los ficheros de datos de las aplicaciones vamos a crear otra partición. Esta partición depende en tamaño de la capacidad de generar datos del club deportivo o empresa para el que la realizamos. El sistema Gesport introduce todas las imágenes y diagramas de ejercicios en la base de datos con la clase Picture, luego el espacio real necesario para el

funcionamiento correcto de la aplicación depende únicamente de este directorio. Esta partición se monta en "/var" y ya es indiferente que sea primaria o lógica. Las tablas de particiones de UNIX soportan hasta cuatro particiones primarias, por lo que no se dan los problemas de Windows con el arranque.

- Para que la instalación de la aplicación sea independiente de la instalación del sistema en "/" se pueden poner particiones diferentes para "/etc" donde se almacenan los ficheros de configuración y "/opt" donde debemos instalar Gesport siguiendo las directrices de Debian y Ubuntu para instalación de software ajeno a la distribución. "/etc" es un directorio donde solo hay ficheros de texto y ocupa bastante poco. En ocasiones las soluciones a problemas de seguridad vienen con pequeñas modificaciones de los ficheros de configuración contenidos en este directorio omitiendo servicios o partes de la configuración vulnerables. Es por esta razón que tendremos nuestro ficheros de configuración de Apache, PHP y MySQL guardados en una copia de seguridad para restaurar mediante comparación de diferencias en los originales de las actualizaciones. En el caso de que el fichero de configuración pueda nutrirse de ficheros externos deberemos configurar los servicios de nuestra aplicación en estos ficheros y no en los principales.
- Para una mejorar sustancial del rendimiento es también recomendable dejar parte del espacio como memoria de paginación en disco. En el caso de Linux hay que destinar una partición de tamaño determinado en tiempo de instalación. Esta partición no tiene punto de montaje y se denomina Swap o área de intercambio. Esta partición debería tener una capacidad entre el tamaño de memoria RAM instalado y el triple de este tamaño.

La marca de arranque activa viene marcada en la primera partición por la letra B (del inglés Boot) detrás de la capacidad de la misma.

Al seleccionar "Finalizar el particionado y escribir los cambios en el disco" comienza el proceso de instalación.

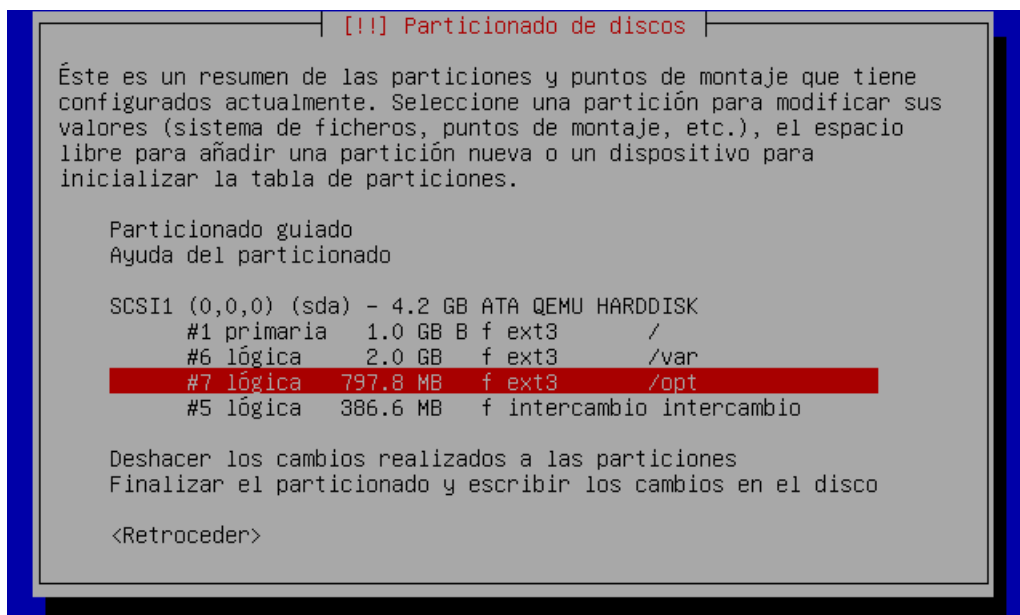


Ilustración 28: Particionado válido para Gesport en una máquina virtual con 4Gb de espacio y 128 Mb de RAM

Tras el proceso de instalación de todos los ficheros de la instalación base comienza el proceso de configuración, donde crearemos un usuario con su contraseña, que en el caso de Ubuntu es también el administrador del sistema. Con el usuario instalado el programa de instalación configura el proxy, si es que usamos uno, y luego nos da a elegir los componentes del sistema que vamos a instalar. Seleccionamos los siguientes:

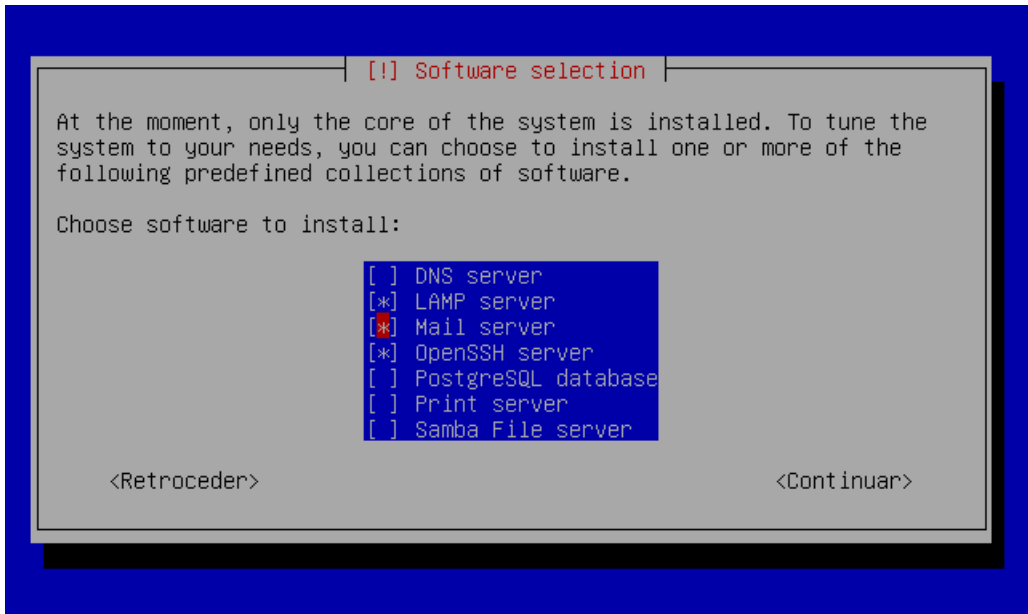


Ilustración 29: Selección de programas a instalar

Los paquetes de software seleccionados son:

- LAMP server: Servidor compuesto por Linux, Apache, Mysql y PHP o Perl.
- Mail server: para notificaciones automáticas del programa.
- OpenSSH server: para poder realizar una administración remota del servidor.

En el proceso de instalación tendremos que poner clave al usuario administrador, a la base de datos MySQL, el nombre de dominio adecuado al servidor, el nombre del usuario administrador y la configuración correcta para conectarse a las diferentes redes del entorno del servidor.

Es posible que, para que el funcionamiento sea totalmente correcto, tengamos que comunicar al administrador de la red donde estamos el nombre de usuario y dirección IP para que configure los DNS con el nombre de dominio adecuado.

4.1.1.2 Instalación de componentes

El comando principal de instalación de componentes en Ubuntu es apt o aptitude. Ambos utilizan las dependencias contenidas en los paquetes .deb, para asegurar el correcto funcionamiento del paquete instalado, y el sistema de repositorios, que permite seleccionar las fuentes de donde obtenemos el software. La diferencia es que aptitude es más completo, pues al instalar una dependencia de algún programa que queremos recordar, al intentar desinstalar, las dependencias que instaló, y las elimina si no son necesarias. Por esta razón utilizaremos aptitude.

Los programas necesarios no instalados se consiguen así:

```
sudo aptitude install php5-cli
```

Aunque en un principio todos los elementos necesarios para el funcionamiento de Gesport podrían parecer ya instalados con esta configuración vamos a utilizar algunas herramientas que no están disponibles por defecto, y que sin ser necesarias podemos usar para facilitar nuestro trabajo:

```
sudo aptitude install phpmyadmin  
sudo aptitude install rsync
```

4.1.1.3 Configuración de Apache

Hay que activar los módulos Include y Rewrite de Apache para un correcto funcionamiento de Symfony y, por lo tanto, de Gesport. Para realizar esta tarea tan solo hay que crear los enlaces simbólicos de la carpeta modules-available a modules-enabled:

```
cd /etc/apache2/modules-enabled  
  
sudo ln -s ../modules-available/include.load include.load  
sudo ln -s ../modules-available/rewrite.load rewrite.load
```

El módulo Include nos permite incluir ficheros de fuera del directorio principal, lo que hará que Gesport pueda incluir los ficheros necesarios para su funcionamiento desde fuera del directorio web. De esta manera cualquier fallo en el servidor que devuelva el código hará inaccesibles los ficheros que se incluyan como referencia en el fichero principal, salvaguardando, al menos las contraseñas de acceso a nuestra base datos guardada en algún fichero PHP.

El módulo Rewrite es el artífice de las direcciones agradables que nos mostrará el navegador en vez de las secuencias con interrogaciones y nombres y valores de datos.

4.1.2 Instalación de Gesport

4.1.2.1 Instalación de ficheros de Gesport y puesta en marcha de web

En la siguiente dirección http://forja.rediris.es/frs/?group_id=431 están los ficheros descargables. Existen dos tipos, "complete" y "for PEAR". La descripción de la instalación se refiere a la instalación "complete" ya que no se recomienda usar el PEAR de Symfony en entornos de producción.

Así pues, accedemos al servidor, ya sea de manera presencial o a través de ssh de manera remota. Descargamos, instalamos y configuramos la última versión de Gesport-complete con los siguientes comandos, como administradores de sistema:

```
#cd /opt  
  
#wget http://forja.rediris.es/frs/download.php/1069/gesport-  
complete-0.0.9.tar.bz2  
  
#tar xvfj gesport-complete-version.tar.bz2
```

Una vez realizado esto tenemos que indicar al servidor apache donde está la carpeta que

tiene que ejecutar y que nombre de dominio vamos a tener. Para ello creamos con nuestro editor favorito un nuevo fichero con un nombre indicativo del sitio.

```
#nano /etc/apache2/sites-available/gesport
```

Lo editamos de la siguiente manera, guardamos y salimos del editor:

```
NameVirtualHost ip_del_servidor:80
```

```
<VirtualHost ip_del_servidor:80>  
  ServerName gesport  
  DocumentRoot "/opt/gesport/web"  
  DirectoryIndex index.php
```

```
  <Directory "/opt/gesport/web">  
    AllowOverride All  
    Allow from All  
  </Directory>  
</VirtualHost>
```

Una vez que tenemos el fichero en la carpeta de sitios disponibles creamos un enlace simbólico hacia este fichero desde la carpeta de sitios activos. El número indica el orden de carga, 000 es para default que carga los servidores básicos, así que por lo menos tendremos que ponerle 001. Si el servidor aloja más aplicaciones podremos elegir el orden de carga de cada una en el inicio de apache. Posteriormente reiniciamos el servidor web:

```
#ln -s /etc/apache2/sites-available/gesport /etc/apache2/sites-enabled/001-gesport
```

```
#/etc/init.d/apache2 restart
```

Para comprobar el funcionamiento correcto de esta parte de la configuración de Gesport tendremos que configurar un ordenador cliente con la dirección IP de la máquina asociada al nombre de servidor y escribir este nombre en el navegador. Como resultado nos debe dar un error, puesto que aún no hay base de datos ni datos dentro.

4.1.2.2 Configuración de la base de datos y creación de datos base

Una vez funcionando el directorio de la aplicación de una manera correcta procederemos a crear un usuario en la base de datos para la ejecución del programa, una base de datos e introduciremos los datos básicos para empezar a funcionar con la aplicación.

Para crear ese usuario podemos hacerlo directamente en la línea de comando o con el programa phpmyadmin si lo instalamos anteriormente. En esta documentación se describen los comandos para bash, ya que el programa lo facilita en sobremanera.

Los siguientes comandos parten del punto anterior y describen lo siguiente:

1. Salir de modo superusuario

```
#exit
```

2. Entrar en la consola de Mysql como superusuario, puesto que vamos a dar permisos:

```
$mysql -u root -p
```

3. Creación del usuario y permisos básicos

```
mysql>CREATE USER 'gesport-app'@ 'localhost' IDENTIFIED BY  
'*****';
```

```
mysql>GRANT USAGE ON * . * TO 'gesport-app'@'localhost' WITH  
MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0  
MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;
```

4. Creación de la base de datos

```
mysql>CREATE DATABASE `gesport-app` ;
```

5. Otorgar permisos al usuario

```
mysql>GRANT ALL PRIVILEGES ON `gesport-app` . * TO 'gesport-  
app'@'localhost' WITH GRANT OPTION ;
```

6. Tras identificarnos de nuevo como superusuario, modificar el ficheros de Gesport que guardan la información de la clase Propel de acceso a la base de datos

```
$sudo su
```

```
password: *****
```

```
#nano /opt/gesport/config/databases.yml
```

En el que tenemos que configurar los parámetros database, username y password respetando los espacios a la izquierda para representar la jerarquía del fichero.

7. Y el fichero que guarda los datos de conexión con la base datos

```
#nano /opt/gesport/config/propel.ini
```

Donde tendremos que configurarlas siguientes lineas

```
propel.database.createUrl = mysql://gesport-app:*****@localhost/
```

```
propel.database.url =  
mysql://gesport:*****@localhost/gesport-app
```

```
propel.output.dir = /opt/gesport
```

8. A partir de aquí solo hay que introducir los datos y limpiar la caché, por si hemos hecho pruebas y hemos metido la pata en la configuración, y tenemos el sistema funcionando

```
#php /opt/gesport/symfony propel-insert-sql
```

```
#php /opt/gesport/batch/load_data.php
```

#php /opt/gesport/symfony clear-cache

Los datos para entrar como administrador son:

Nombre de usuario: admin

Clave: gesport

Se pueden cambiar dentro del programa, pero eso viene reflejado en el manual de usuario.

4.1.2.3 Configuración de DNS y acceso a la página

Con todos los pasos anteriormente realizados tenemos la aplicación funcionando. Ahora quedaría configurar el servidor de nombres de dominio (DNS) para que los el acceso a Gesport suceda de la manera adecuada. Esto significa que al poner el nombre de servidor configurado en el paso de puesta en marcha de la web debería aparecer. Para ello hay varias alternativas:

- Si la ejecución es local (entorno de pruebas) tendremos que dar un alias a nuestra propia dirección modificando el fichero `/etc/hosts`.
- Si va a funcionar en una red local, tendremos que añadir el registro con el nombre asignado en el fichero de configuración de apache, al servidor DNS de la red interna apuntando a la IP del equipo que ejecuta Gesport.
- Si va a funcionar en un servidor abierto a internet tendremos que registrar el dominio correspondiente en un registrador oficial de dominios. Una vez registrado tendremos que hacer que apunte a la/las IPs de nuestro servidor. Dependiendo del tipo de enlace registrado en el registrador de dominios puede ser necesario un sistema DNS interno del servidor para resolver este dominio.

4.1.2.4 Personalizar Gesport

Si con los pasos anteriores ya tenemos funcionando Gesport perfectamente hay que decir que tenemos que configurar algunos otros parámetros para, por ejemplo, que los correos electrónicos lleguen a la dirección de nuestro administrador, en vez de a la por defecto, o para cambiar la barra de título, aspecto de la página, o, en el caso de que estemos actualizando entre versiones, parar la aplicación para evitar daños en los datos.

Los ficheros principales para hacer estas tareas son, desde el directorio donde está instalado Gesport:

- `apps/frontend/config/app.yml`, donde vienen configuradas ciertas variables como los correos electrónicos de los administradores, webmaster y contacto comercial.
- `apps/frontend/config/settings.yml`. Este fichero es el corazón de la aplicación. En el campo `available` de `settings` podemos escribir `off` para dejar el sitio temporalmente apagado y cuando volvamos a estar preparados para ponerlo en funcionamiento ponerlo a `on`.
- `apps/frontend/config/view.yml` es el fichero que describe la salida por pantalla de nuestra aplicación. Podemos ver el título de la ventana de navegador las hojas de estilos que utiliza y las palabras clave y descripción de la cabecera de la página, entre otras.
- `web/css/` es la carpeta donde se sitúan los ficheros `css` que hemos definido en `view`.
- `web/images` es la carpeta donde están las imágenes de Gesport.

4.2 Manual de usuario

4.2.1 Acceso a la aplicación

Para acceder a la aplicación se debe abrir un navegador web. En la barra de direcciones introducir la dirección de la aplicación, en este caso <http://gesport.serincas.com>.

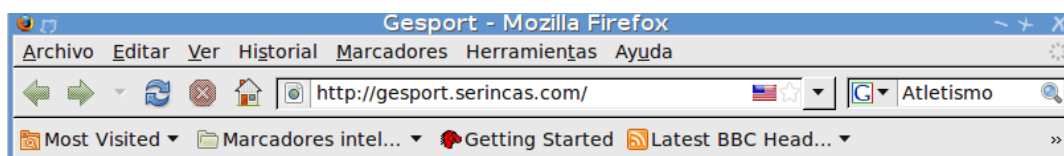


Ilustración 30: Barra de direcciones del navegador

Una vez cargada la página se debe hacer clic sobre el botón acceder al sistema, donde aparecerá un formulario para solicitarle el nombre de usuario y la contraseña.



Ilustración 31: Inicio de Aplicación

Al introducir los datos solicitados correctamente usted quedará listo para operar con el

programa.

Ilustración 32: Pantalla de acceso de usuarios

4.2.1.1 Problemas con el acceso a la aplicación

4.2.1.1.1 Errores en la introducción de datos

En el caso de que falte alguno de los datos de acceso Gesport notificará pertinentemente esta carencia indicando sobre los campos sin rellenar en un color destacado.

En el caso de que el nombre de usuario y contraseña no coincidan con ninguno de la base de datos se notificará también sobre el campo del usuario.

Ilustración 33: Error de acceso

4.2.1.1.2 Olvido de contraseña de acceso

Cuando un usuario olvida su contraseña no puede acceder a consultar y editar sus entrenamientos salvo que realice un proceso de recuperación de la misma. En todas las pantallas de acceso al sistema hay un elemento que dice "No recuerdo mi clave". Al pulsarlo se mostrará un formulario que generará una nueva.

The screenshot shows a web interface with a sidebar menu on the left and a main content area. The sidebar menu includes links for 'Usuario', 'Acceder al sistema', 'Registrarse', 'Acerca de Gesport', 'Acerca de', 'Notificar un error', 'Proponer una sugerencia', 'Concurso Universitario de Software Libre', 'Portada', 'Diego Cebrían', 'Desarrollando Gesport', and 'Serincas'. The main content area is titled 'Usuario' and 'Recibe tu clave por correo electrónico'. It features a label 'Correo electrónico:' followed by a text input field. Below the input field is a button labeled 'Enviar'.

Ilustración 34: Solicitud de regeneración de clave por pérdida

The screenshot shows the same web interface as before, but with an error message. The text input field now contains the email address 'asdfg@1234.es'. Above the input field, a red error message reads: '¡ No hay ningún usuario con este correo, por favor inténtelo de nuevo ¡'. The 'Enviar' button remains visible below the input field.

Ilustración 35: Error en la solicitud de nueva contraseña

The screenshot shows a confirmation page titled 'Confirmación - información de acceso enviada'. The text on the page reads: 'Tu información para entrar se ha enviado a persona@localhost.com. Deberías recibirla en breve, de manera que puedes proceder a formulario de entrada.' The sidebar menu is still visible on the left.

Ilustración 36: Envío satisfactorio de solicitud de nueva contraseña

4.2.2 Solicitud de acceso al programa

Para poder empezar a trabajar con Gesport es necesario que el administrador de de alta al usuario si el usuario es un entrenador y si el usuario es un deportista será su entrenador el propio creador de la cuenta. Para realizar la solicitud existe un enlace que dice "Registrarse". Al hacer clic

sobre él se mostrará un formulario de recogida de datos personales.

Menú

- **Usuario**
 - Acceder al sistema
 - Registrarse
- **Acerca de Gesport**
 - Acerca de
 - Notificar un error
 - Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - Portada
- **Diego Cebrián**
 - Desarrollando Gesport
 - Serincas

Solicitud de una cuenta
Por favor rellene el formulario.

Nombre:

Apellido:

Nombre de usuario deseado:

Correo electrónico:

Fecha de Nacimiento: ...

Usted es (por favor seleccione su futura actividad en Gesport):

Ilustración 37: Solicitud de acceso

Se debe rellenar este formulario y hacer clic en el botón "Enviar Solicitud". Si se han rellenado todos los datos necesarios se terminará la solicitud de ingreso en el programa. El programa envía un correo electrónico al administrador principal, que será el encargado de realizar el alta.

Menú

- **Usuario**
 - Acceder al sistema
 - Registrarse
- **Acerca de Gesport**
 - Acerca de
 - Notificar un error
 - Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - Portada
- **Diego Cebrián**
 - Desarrollando Gesport
 - Serincas

Has solicitado una cuenta
El administrador de la aplicación creará tu cuenta en un breve periodo de tiempo y entonces recibirás un correo electrónico con los datos necesarios para entrar.
El equipo de Gesport

Ilustración 38: Confirmación de envío de correo electrónico

Si el usuario es deportista y quiere usar el programa se le asignará una cuenta de entrenador y deportista de manera que él sea su mismo entrenador.

4.2.2.1 Problemas con la solicitud de acceso

En determinados casos se pueden producir errores con esta solicitud. Estos errores son los siguientes:

- No se han rellenado todos los datos necesario para completar la solicitud.
- El nombre de usuario o correo electrónico ya existen

- La fecha de nacimiento no tiene el formato correcto. Todas las fechas que maneja la aplicación estarán en el formato estándar de su localización según el idioma de manejo.

En estos casos el sistema muestra mensajes de error destacados sobre los campos que fallan o faltan.

Menú

- **Usuario**
 - Acceder al sistema
 - Registrarse
- **Acerca de Gesport**
 - Acerca de
 - Notificar un error
 - Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - Portada
- **Diego Cebrián**
 - Desarrollando Gesport
 - Serincas

Solicitud de una cuenta
Por favor rellene el formulario.

Nombre:

Apellido:

Nombre de usuario deseado:

Correo electrónico: ↓ Es necesario el correo electrónico ↓

Fecha de Nacimiento: ↓ La fecha de nacimiento es necesaria ↓

Usted es (por favor seleccione su futura actividad en Gesport):

Ilustración 39: Errores en solicitud de usuario



Ilustración 40:
Operaciones con
Usuarios

4.2.3 Operaciones del usuario administrador

4.2.3.1 Operaciones con usuarios

Para realizar cualquier operación con usuarios se pulsa sobre el elemento "Usuarios" del menú lateral. Esto nos muestra una lista paginada de los usuarios del sistema.

4.2.3.1.1 Crear un usuario nuevo

Para crear un nuevo usuario se pulsa sobre el botón "Nuevo" del final de la lista. Se rellena el formulario correspondiente, al menos en sus campos obligatorios, marcados especialmente. La marca de la pantalla representada en el manual puede variar según la hoja de estilos utilizada.

4.2.3.1.1.1 Problemas con la creación de usuarios

En el caso de que no se hayan incluido todos los datos necesarios o alguno no tenga el formato correcto el programa volverá al formulario inicial relleno de los datos introducidos (salvo las fechas) y notificando con mensajes los errores correspondientes.

Usuario

Nombre de usuario	Nombre	Apellido	Fecha de creación	Fecha de Nacimiento	Sexo	Es entrenador	Es deportista	Es administrador	Correo electrónico	Entrenador	Idioma	Opciones
admin	Nombre	Apellido	27 de marzo de 2009 01:00:40 CET	20/06/81	Femenino	✓	✓	✓	gesport@localhost.com		es	Mostrar Editar
user1	Persona	Entrenada 1	27 de marzo de 2009 00:47:15 CET	20/06/81	Masculino	✗	✓	✗	persona@localhost.com	Nombre Apellido	es	Mostrar Editar
user2	Sujeto	Entrenado 1	27 de marzo de 2009 00:47:15 CET	20/06/81	Femenino	✗	✓	✗	sujeto@localhost.com	Nombre Apellido	es	Mostrar Editar
user3	Atleta	Fondo 1	27 de marzo de 2009 00:47:15 CET	20/06/81	Femenino	✗	✓	✗	atleta@localhost.com	Nombre Apellido	fr	Mostrar Editar
user4	Nadador	Velocidad 1	27 de marzo de 2009 00:47:15 CET	20/06/81	Masculino	✗	✓	✗	nadador@localhost.com	Nombre Apellido		Mostrar Editar

<< 1 - 2 >>

Nuevo

Ilustración 41: Listado de usuarios

Menú

- Usuario
 - o Salir
 - o Datos de admin
- Administrator
 - o Unidades
 - o Tipos de unidades
 - o Factor de conversión entre unidades
 - o Usuarios
- Entrenador
 - o Deportistas
- Deportista
 - o Ver mi día de entrenamiento
 - o Enviar resultado
- Acerca de Gesport
 - o Acerca de
 - o Notificar un error
 - o Proponer una sugerencia
- Concurso Universitario de Software Libre
 - o Portada
- Diego Cebrián
 - o Desarrollando Gesport
 - o Serincas

Usuario

Nombre de usuario:

Nombre:

Apellido:

Fecha de Nacimiento: ...

Sexo:

Es entrenador: ☐

Es deportista: ☐

Es administrador: ☐

Entrenador:

Correo electrónico:

Nacionalidad:

Idioma:

Ilustración 42: Formulario de creación de usuario

4.2.3.1.2 Edición de usuarios registrados

El usuario se puede modificar pulsando en el elemento "Editar" del listado paginado de usuarios. Gesport mostrará los campos editables de este usuario. Al igual que en el caso de la creación los formatos inválidos de fecha o los campos obligatorios sin rellenar provocarán un error.

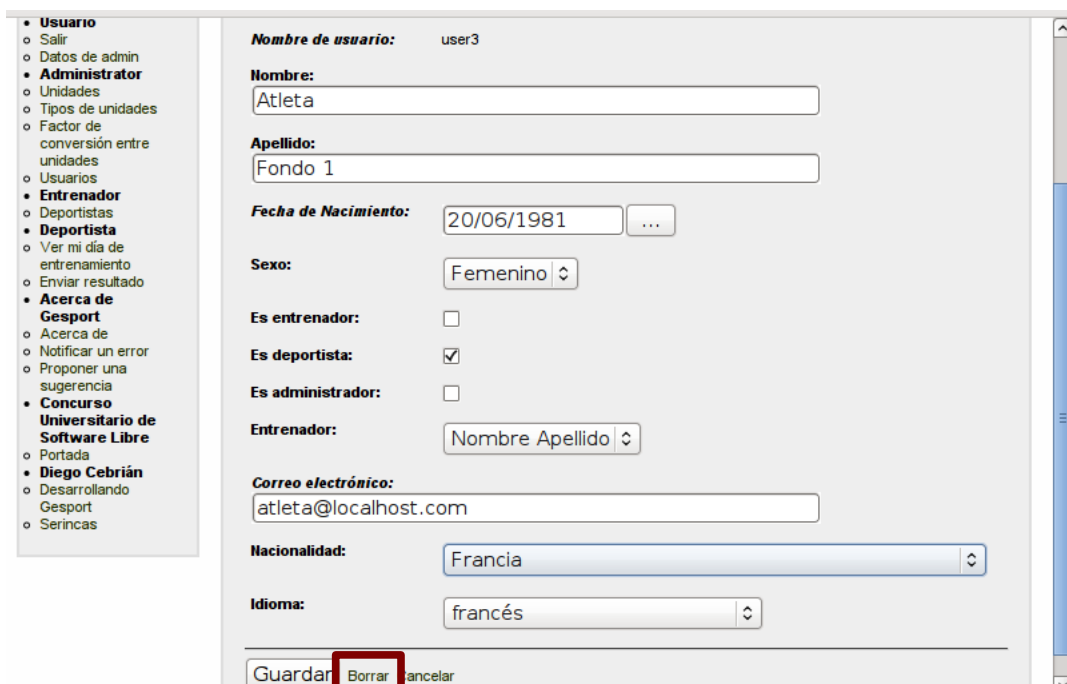


Ilustración 43: Edición de usuarios

4.2.3.1.3 Borrado de usuarios

Para borrar el usuario hay que pulsar sobre el elemento "Editar" de la lista. En el formulario de edición, en la parte inferior hay que pulsar sobre el elemento "Borrar". Tras confirmar el deseo de borrar el usuario y si este no tiene elementos en otras tablas será borrado.



Ilustración 44: Confirmación de Borrado

Menú

- **Usuario**
 - o Salir
 - o Datos de admin
- **Administrador**
 - o Unidades
 - o Tipos de unidades
 - o Factor de conversión entre unidades
 - o Usuarios
- **Entrenador**
 - o Deportistas
- **Deportista**
 - o Ver mi día de entrenamiento
 - o Enviar resultado
- **Acerca de Gesport**
 - o Acerca de
 - o Notificar un error
 - o Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - o Portada
- **Diego Cebrián**
 - o Desarrollando Gesport
 - o Serincas

Usuario

Nombre Apellido no se ha borrado porque aún se está usando en Gesport y no puede ser borrado.

Id:	1
Nombre de usuario:	admin
Nombre:	Nombre
Apellido:	Apellido
Fecha de creación:	27 de marzo de 2009 01:00:40 CET
Fecha de nacimiento:	20/06/81
Sexo:	Femenino
Es entrenador:	✓
Es deportista:	✓
Es administrador:	✓
Entrenador:	

[Editar Lista](#)

Ilustración 45: Error de borrado por usuario con referencias

Menú

- **Usuario**
 - o Salir
 - o Datos de admin
- **Administrador**
 - o **Tipos de unidades**
 - o Factor de conversión entre unidades
 - o Usuarios
- **Entrenador**
 - o Deportistas
- **Deportista**
 - o Ver mi día de entrenamiento
 - o Enviar resultado
- **Acerca de Gesport**
 - o Acerca de
 - o Notificar un error
 - o Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - o Portada
- **Diego Cebrián**
 - o Desarrollando Gesport
 - o Serincas

Ilustración 46:
Menú tipos de unidades

4.2.3.2 Operaciones con tipos de unidades de medida

Para realizar cualquier operación con tipos de unidades se pulsa sobre el elemento "Tipos de unidades" del menú lateral. Esto nos muestra una lista de los tipos de unidades del sistema.

4.2.3.2.1 Crear un tipo de unidad nuevo

Para crear un nuevo tipo de unidad se pulsa sobre el botón "Nuevo" del final de la lista. Se rellena el formulario correspondiente, al menos en sus campos obligatorios.

4.2.3.2.1.1 Problemas con la creación de tipos de unidades

En el caso de que no se hayan incluido todos los datos necesarios o alguno no tenga el formato correcto el programa volverá al formulario inicial rellenando los datos introducidos y notificando con mensajes los errores correspondientes.

4.2.3.2.2 Edición de tipos de unidades

El tipo de unidad se puede modificar pulsando en el elemento "Editar" del listado de tipos de unidades. Gesport mostrará los campos editables de este tipo de unidad. Al igual que en el caso de la creación los formatos inválidos de fecha o los campos obligatorios sin rellenar provocarán un error.

Tipo de Unidad		
Descripción global	Descripción	Opciones
Distance	Distancia	Mostrar Editar
Weight	Peso	Mostrar Editar
Time	Tiempo	Mostrar Editar
counter	Contador	Mostrar Editar
Nuevo		

Ilustración 47: Listado de tipos de unidades

Menú

- **Usuario**
 - o Salir
 - o Datos de admin
- **Administrator**
 - o Unidades
 - o Tipos de unidades
 - o Factor de conversión entre unidades
 - o Usuarios
- **Entrenador**
 - o Deportistas
- **Deportista**
 - o Ver mi día de entrenamiento
 - o Enviar resultado
- **Acerca de Gesport**
 - o Acerca de
 - o Notificar un error
 - o Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - o Portada
- **Diego Cebrián**

Tipo de Unidad

Descripción global:

Descripción:

Ilustración 48: Añadido de un tipo de unidad

Tipo de Unidad

Descripción global:

Descripción:

Ilustración 49: Edición tipo de unidad

4.2.3.2.3 Borrado de tipos de unidades

Para borrar el tipo de unidad hay que pulsar sobre el elemento "Editar" de la lista. En el formulario de edición, en la parte inferior hay que pulsar sobre el elemento "Borrar". Tras confirmar el deseo de borrar el tipo de unidad y si este no se está usando en otro elemento. El mensaje emergente de consulta de borrado será igual que el de "Usuario".

4.2.3.3 Operaciones con unidades de medida

Para realizar cualquier operación con unidades de medida se pulsa sobre el elemento "Unidades de medida" del menú lateral. Esto nos muestra una lista de las unidades de medida del sistema.

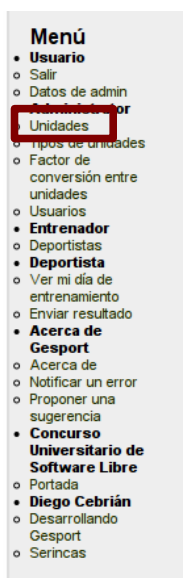


Ilustración 50:
Elemento
Unidades

4.2.3.3.1 Crear una unidad de medida nueva

Para crear una nueva unidad de medida se pulsa sobre el botón "Nuevo" del final de la lista. Se rellena el formulario correspondiente, al menos en sus campos obligatorios.

4.2.3.3.1.1 Problemas con la creación de unidades de medida

En el caso de que no se hayan incluido todos los datos necesarios o alguno no tenga el formato correcto el programa volverá al formulario inicial rellenando los datos introducidos y notificando con mensajes los errores correspondientes.

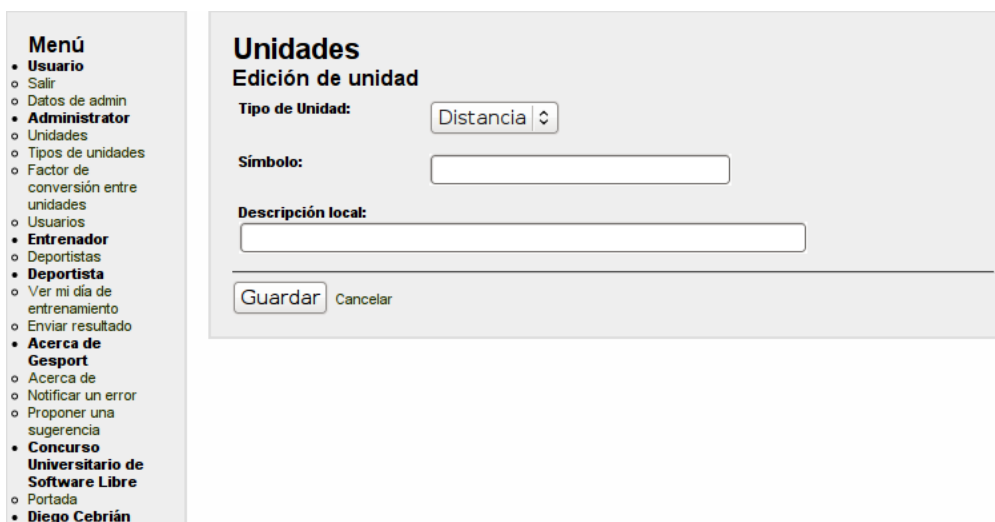
4.2.3.3.2 Edición de unidades de medida

La unidad de medida se puede modificar pulsando en el elemento "Editar" del listado de unidades de medida. Gesport mostrará los campos editables de esta unidad de medida. Al igual que en el caso de la creación los formatos inválidos de fecha o los campos obligatorios sin rellenar provocarán

un error.

Unidades			
Tipo de Unidad	Símbolo	Descripción	Opciones
Distancia	Km	Kilómetros	Mostrar Editar
Distancia	m	Metros	Mostrar Editar
Distancia	Mi	Millas	Mostrar Editar
Peso	Kg	Kilogramos	Mostrar Editar
Peso	Lb	Libras	Mostrar Editar
Peso	Tm	Toneladas	Mostrar Editar
Tiempo	M	Minutos	Mostrar Editar
Tiempo	H	Horas	Mostrar Editar
Tiempo	S	Segundos	Mostrar Editar
Contador	x	Repeticiones	Mostrar Editar
Tiempo	D	Días	Mostrar Editar
Nuevo			

Ilustración 51: Listado de unidades



Menú

- **Usuario**
 - Salir
 - Datos de admin
- **Administrator**
 - Unidades
 - Tipos de unidades
 - Factor de conversión entre unidades
 - Usuarios
- **Entrenador**
 - Deportistas
- **Deportista**
 - Ver mi día de entrenamiento
 - Enviar resultado
- **Acerca de Gesport**
 - Acerca de
 - Notificar un error
 - Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - Portada
- **Diego Cebrián**

Unidades
Edición de unidad

Tipo de Unidad: Distancia

Símbolo:

Descripción local:

Guardar Cancelar

Ilustración 52: Creación de nueva unidad



Menú

- **Usuario**
 - Salir
 - Datos de admin
- **Administrator**
 - Unidades
 - Tipos de unidades
 - Factor de conversión entre unidades
 - Usuarios
- **Entrenador**
 - Deportistas
- **Deportista**
 - Ver mi día de entrenamiento
 - Enviar resultado
- **Acerca de Gesport**
 - Acerca de
 - Notificar un error
 - Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - Portada
- **Diego Cebrián**

Unidades
Edición de unidad

Tipo de Unidad: Peso

Símbolo: Tm

Descripción local: Toneladas

Guardar Borrar Cancelar

Ilustración 53: Edición de unidad

4.2.3.3.3 Borrado de unidades de medida

Para borrar una unidad de medida hay que pulsar sobre el elemento "Editar" de la lista. En el formulario de edición, en la parte inferior hay que pulsar sobre el elemento "Borrar". Tras confirmar el deseo de borrar la unidad de medida y si esta no se está usando en otro elemento. El diálogo de confirmación de borrado será igual que en los casos anteriores.

4.2.3.4 Operaciones con factores de conversión entre unidades

Para realizar cualquier operación con factores de conversión se pulsa sobre el elemento "Factores de conversión" del menú lateral. Esto nos muestra una lista de los factores de conversión del sistema.

4.2.3.4.1 Crear un factor de conversión nuevo

Para crear un nuevo factor de conversión se pulsa sobre el botón "Nuevo" del final de la lista. Se rellena el formulario correspondiente, al menos en sus campos obligatorios.

4.2.3.4.1.1 Problemas con la creación de factores de conversión

En el caso de que no se hayan incluido todos los datos necesarios o alguno no tenga el formato correcto el programa volverá al formulario inicial rellenando los datos introducidos y notificando con mensajes los errores correspondientes.

4.2.3.4.2 Edición de factores de conversión

El factor de conversión se puede modificar pulsando en el elemento "Editar" del listado de factores de conversión. Gesport mostrará los campos editables de este factor de conversión. Al igual que en el caso de la creación los formatos inválidos de fecha o los campos obligatorios sin rellenar provocarán un error.

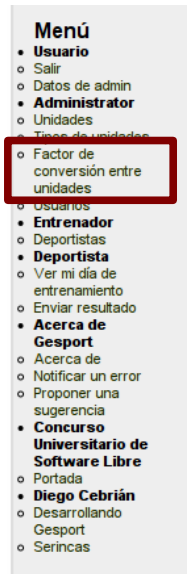


Ilustración 54:
Elemento
Factores de
conversión

Factor de conversión			
Origen	Destino	Factor de conversión	Opciones
Kilómetros	Metros	0.001	Mostrar Editar
Kilómetros	Millas	1.609	Mostrar Editar
Minutos	Horas	0.453592	Mostrar Editar
Minutos	Segundos	1000	Mostrar Editar
Libras	Metros	60	Mostrar Editar
Metros	Millas	60	Mostrar Editar
Horas	Días	24	Mostrar Editar
Origen x Factor de conversión = Destino, Formato: 12341.12345			
Nuevo			

Ilustración 55: Listado de factores de conversión

Menú

- **Usuario**
 - o Salir
 - o Datos de admin
- **Administrator**
 - o Unidades
 - o Tipos de unidades
 - o Factor de conversión entre unidades
 - o Usuarios
- **Entrenador**
 - o Deportistas
- **Deportista**
 - o Ver mi día de entrenamiento
 - o Enviar resultado
- **Acerca de Gesport**
 - o Acerca de
 - o Notificar un error
 - o Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - o Portada
- **Diego Cebrián**

Factor de conversión

Origen':

Destino':

Factor de conversión:

Origen x Factor de conversión = Destino, Formato: 12341.12345

Ilustración 56: Creación de factores de conversión

Menú

- **Usuario**
 - o Salir
 - o Datos de admin
- **Administrator**
 - o Unidades
 - o Tipos de unidades
 - o Factor de conversión entre unidades
 - o Usuarios
- **Entrenador**
 - o Deportistas
- **Deportista**
 - o Ver mi día de entrenamiento
 - o Enviar resultado
- **Acerca de Gesport**
 - o Acerca de
 - o Notificar un error
 - o Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - o Portada
- **Diego Cebrián**

Factor de conversión

Origen':

Destino':

Factor de conversión:

Origen x Factor de conversión = Destino, Formato: 12341.12345

Ilustración 57: Edición de factores de conversión

4.2.3.4.3 Borrado de factores de conversión

Para borrar el factor de conversión hay que pulsar sobre el elemento "Editar" de la lista. En el formulario de edición, en la parte inferior hay que pulsar sobre el elemento "Borrar". Tras confirmar el deseo de borrar el factor de conversión y si este no se está usando en otro elemento.

4.2.4 Operaciones del usuario entrenador

Las operaciones del usuario entrenador van en cascada. Tan solo tiene una opción en el menú principal que es la visualización de deportistas. Al seleccionar esta opción se le presenta una lista con sus deportistas únicamente.

4.2.4.1 Operaciones con deportistas

4.2.4.1.1 Crear deportista

A pesar de no ser administrador el entrenador puede crear usuarios. Todos los usuarios que crea se incluyen directamente en su lista de deportistas. Para crear un deportista nuevo tiene que seleccionar la opción "nuevo" bajo la lista. Se muestra un formulario con los campos obligatorios destacados que se rellena.

Mis deportistas												
Nombre de usuario	Nombre	Apellido	Fecha de nacimiento	Sexo	Es entrenador	Es deportista	Es administrador	Entrenador	Opciones			
user1	Persona	1	Entrenada 1981-06-20 00:00:00	Masculino	✗	✓	✗	Nombre Apellido	Mostrar	Editar	Temporada actual	Temporadas
user2	Sujeto	1	Entrenado 1981-06-20 00:00:00	Femenino	✗	✓	✗	Nombre Apellido	Mostrar	Editar	Temporada actual	Temporadas
user3	Atleta	Fondo 1	1981-06-20 00:00:00	Femenino	✗	✓	✗	Nombre Apellido	Mostrar	Editar	Temporada actual	Temporadas
user4	Nadador	Velocidad 1	1981-06-20 00:00:00	Masculino	✗	✓	✗	Nombre Apellido	Mostrar	Editar	Temporada actual	Temporadas
user5	Johnny	nº5	1986-05-09 00:00:00	Femenino	✗	✓	✗	Nombre Apellido	Mostrar	Editar	Temporada actual	Temporadas
Paginador	Federico	rtaqui	1985-03-05 00:00:00	Masculino	✓	✓	✗	Nombre Apellido	Mostrar	Editar	Temporada actual	Temporadas
nuevo												

Ilustración 58: Listado de deportista de un entrenador del sistema

Menú

- **Usuario**
 - Salir
 - Datos de admin
- **Administrator**
 - Unidades
 - Tipos de unidades
 - Factor de conversión entre unidades
 - Usuarios
- **Entrenador**
 - Deportistas
- **Deportista**
 - Ver mi día de entrenamiento
 - Enviar resultado
- **Acerca de Gesport**
 - Acerca de
 - Notificar un error
 - Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - Portada
- **Diego Cebrián**
 - Desarrollando Gesport
 - Serincas

Usuario

Nombre de usuario:

Nombre:

Apellido:

Fecha de Nacimiento: ...

Sexo:

Entrenador:

Correo electrónico:

Nacionalidad:

Idioma:

Ilustración 59: Formulario de creación de un nuevo deportista

4.2.4.1.2 Editar deportista

Desde el listado de deportistas seleccione la opción "editar". Se muestra un formulario similar al anterior de manera que pueda cambiar los datos de dicho deportista.

Menú

- **Usuario**
 - Salir
 - Datos de admin
- **Administrator**
 - Unidades
 - Tipos de unidades
 - Factor de conversión entre unidades
 - Usuarios
- **Entrenador**
 - Deportistas
- **Deportista**
 - Ver mi día de entrenamiento
 - Enviar resultado
- **Acerca de Gesport**
 - Acerca de
 - Notificar un error
 - Proponer una sugerencia
- **Concurso Universitario de Software Libre**
 - Portada
- **Diego Cebrián**
 - Desarrollando Gesport
 - Serincas

Edición de deportista

Nombre de usuario: user3

Nombre:

Apellido:

Fecha de Nacimiento: ...

Sexo:

Entrenador:

Correo electrónico:

Nacionalidad:

Idioma:

Ilustración 60: Edición de un deportista

4.2.4.2 Operaciones de planificación y consulta de entrenamientos

Al seleccionar la opción mostrar del listado de deportista se muestran los datos del deportista. En ellos se incluyen las temporadas planificadas, además de los datos solicitados en la creación o edición del deportista.

Nadador Velocidad 1

Sexo:Masculino

Correo electrónico:nadador@localhost.com

Fecha de Nacimiento:20/06/81

Temporadas de Nadador Velocidad 1

Descripción	Comienza en	Acaba en	Intensidad teórica	Intensidad reportada	Volumen teórico	Volumen reportado	Opciones
Operaciones							
• Añadir temporada							
• Editar							

Lista de mis deportistas

Ilustración 61: Vista de un deportista sin temporadas planificadas

Persona Entrenada 1

Sexo:Masculino

Correo electrónico:persona@localhost.com

Fecha de Nacimiento:20/06/81

Temporadas de Persona Entrenada 1

Descripción	Comienza en	Acaba en	Intensidad teórica	Intensidad reportada	Volumen teórico	Volumen reportado	Opciones			
2008-2009	1 de septiembre de 2008, lunes	31 de agosto de 2009, lunes	40-70 %	4000 Kilómetros	21 Kilómetros	6220	26	Mostrar	Editar	Añadir macrociclo
			71-85 %	1500 Kilómetros	5 Kilómetros					
			86-100 %	720 Kilómetros	0 Kilómetros					
2009-2010	1 de septiembre de 2009, martes	31 de agosto de 2010, martes	50-75 %	4200 Kilómetros	Kilómetros	6700		Mostrar	Editar	Añadir macrociclo
			76-90 %	1700 Kilómetros	Kilómetros					
			91-100 %	800 Kilómetros	Kilómetros					

Operaciones

- Mostrar temporada actual
- Añadir macrociclo a temporada actual
- Añadir temporada
- Editar

Ilustración 62: Vista de un deportista con una temporada planificada y otra en ejecución

El mostrado y selección de cada uno de los niveles es iterativo hasta llegar a la sesión, donde los formularios y listados cambian. Así se describirán todos los periodos contenidos unos dentro de otros en este punto del manual.

Existen 5 niveles de planificación para cada usuario: Temporada, Macrociclo, Mesociclo,

Microciclo y Sesión. De estos 5 niveles son periodos de tiempo los cuatro primeros y la sesión es la unidad básica de entrenamiento que puede corresponder a un día o incluso haber varias sesiones en un solo día.

4.2.4.2.1 Creación de periodos de entrenamiento

Para añadir un periodo nuevo de entrenamiento se puede realizar de varias maneras diferentes:

- Añadiendo un periodo en el mismo nivel de visualización que estamos. Para ello, bajo e listado de periodos que estamos viendo hay una opción "Añadir periodo".
- Añadiendo un periodo a uno de los periodos contenidos en la vista actual. Para hacer esta operación tendremos que pulsar la opción "Añadir periodo" en la línea de listado de periodos que nos muestra la pantalla.

En ambos casos el resultado es un formulario en el que se nos pregunta por los datos del periodo a añadir. Además este formulario nos presenta un listado con los periodos del mismo nivel para que no se solapen la fechas.

Mesociclo para Persona Entrenada 1 para macrociclo Cross

Mesociclos de Persona Entrenada 1 del macrociclo Cross

Descripción	Comienza en	Acaba en	Intensidad teórica	Intensidad reportada	Volumen teórico	Volumen reportado	Opciones
Preparatorio	1 de septiembre de 2008, lunes	12 de octubre de 2008, domingo	40-70 %	13 Kilómetros		18	Mostrar Editar Añadir microciclo
			71-85 %	5 Kilómetros			
			86-100 %	0 Kilómetros			
Básico	13 de octubre de 2008, lunes	16 de noviembre de 2008, domingo	40-70 %	8 Kilómetros	525	8	Mostrar Editar Añadir microciclo
			71-85 %	0 Kilómetros			
			86-100 %	0 Kilómetros			

Macrociclo: Cross, desde 1 de septiembre de 2008, lunes hasta 31 de marzo de 2009, martes

Descripción:

Comienza en: ...

Acaba en: ...

Intensidad teórica:

40-70 % Kilómetros

71-85 % Kilómetros

86-100 % Kilómetros

Volumen teórico:

Ilustración 63: Formulario de creación de un ciclo de entrenamiento

4.2.4.2.1.1 Problemas en la creación de periodos de entrenamiento

El sistema previene de los solapamientos de fechas, errores y faltas en los datos. Al enviar un formulario con errores de cualquier tipo e programa devuelve el formulario con los datos rellenos (salvo fechas), y con los errores marcados.

CROSS

Mesociclos de Persona Entrenada 1 del macrociclo Cross

Descripción	Comienza en	Acaba en	Intensidad teórica	Intensidad reportada	Volumen teórico	Volumen reportado	Opciones
Preparatorio	1 de septiembre de 2008, lunes	12 de octubre de 2008, domingo	40-70 % 71-85 % 86-100 %	Kilómetros Kilómetros Kilómetros	13 5 0	Kilómetros Kilómetros Kilómetros	18 Mostrar Editar Añadir microciclo
Básico	13 de octubre de 2008, lunes	16 de noviembre de 2008, domingo	40-70 % 71-85 % 86-100 %	300 150 75	8 0 0	Kilómetros Kilómetros Kilómetros	525 8 Mostrar Editar Añadir microciclo

Macrociclo: Cross, desde 1 de septiembre de 2008, lunes hasta 31 de marzo de 2009, martes

Descripción: Precompetitivo

Comienza en: ...

Acaba en: ...

↓ Error de solapamiento: El día que has elegido para que empiece el mesociclo ya pertenece a otro llamado Básico ↓

Intensidad teórica:

40-70 % Kilómetros

71-85 % Kilómetros

86-100 % Kilómetros

Volumen teórico:

Ilustración 64: Error de solapamiento de fechas

4.2.4.2.2 Edición de periodos de entrenamiento

Para editar un periodo de entrenamiento se puede acceder de la misma manera que para la creación, pulsando en la opción editar de debajo del listado para editar el periodo mostrado actualmente, o pulsando la opción editar del listado para editar un periodo subordinado del actual. Los errores y controles producidos son los mismos, solo que además comprueba que los elementos de dicho periodo queden dentro si se modifican las fechas.

Mesociclo Preparatorio de Persona Entrenada 1 para macrociclo Cross

Mesociclos de Persona Entrenada 1 del macrociclo Cross

Descripción	Comienza en	Acaba en	Intensidad teórica	Intensidad reportada	Volumen teórico	Volumen reportado	Opciones
Preparatorio	1 de septiembre de 2008, lunes	12 de octubre de 2008, domingo	40-70 % 71-85 % 86-100 %	Kilómetros Kilómetros Kilómetros	13 5 0	Kilómetros Kilómetros Kilómetros	18 Mostrar Editar Añadir microciclo
Básico	13 de octubre de 2008, lunes	16 de noviembre de 2008, domingo	40-70 % 71-85 % 86-100 %	300 150 75	8 0 0	Kilómetros Kilómetros Kilómetros	525 8 Mostrar Editar Añadir microciclo

Descripción:
Preparatorio

Comienza en: 1/09/2008 ...

Acaba en: 12/10/2008 ...

Intensidad teórica:

40-70 % 250 Kilómetros

71-85 % 150 Kilómetros

86-100 % 50 Kilómetros

Volumen teórico: 450

Guardar **Borrar** Cancelar

Ilustración 65: Edición de ciclos de entrenamiento

4.2.4.2.3 Creación de sesiones

Una vez creado un periodo de tipo macrociclo Gesport genera automáticamente todas las sesiones comprendidas en ese periodo.

Sesiones para Persona Entrenada 1 en el microciclo Preparatorio 1							
Date	Descripción	Intensidad teórica	Intensidad reportada	Volumen teórico	Volumen reportado	Opciones	
1 de septiembre de 2008, lunes		40-70 % Kilómetros 71-85 % Kilómetros 86-100 % Kilómetros	Kilómetros Kilómetros Kilómetros			Mostrar	Editar
2 de septiembre de 2008, martes		40-70 % Kilómetros 71-85 % Kilómetros 86-100 % Kilómetros	Kilómetros Kilómetros Kilómetros			Mostrar	Editar
3 de septiembre de 2008, miércoles		40-70 % Kilómetros 71-85 % Kilómetros 86-100 % Kilómetros	Kilómetros Kilómetros Kilómetros			Mostrar	Editar
4 de septiembre de 2008, jueves	4x500 a 1,35	40-70 % 5 Kilómetros 71-85 % 2 Kilómetros 86-100 % 0 Kilómetros	5 Kilómetros 2 Kilómetros 0 Kilómetros	7	7	Mostrar	Editar
5 de septiembre de 2008, viernes	12341	40-70 % 10 Kilómetros 71-85 % 1 Kilómetros 86-100 % 0 Kilómetros	8 Kilómetros 3 Kilómetros 0 Kilómetros	11	11	Mostrar	Editar
6 de septiembre de 2008, sábado		40-70 % Kilómetros 71-85 % Kilómetros 86-100 % Kilómetros	Kilómetros Kilómetros Kilómetros			Mostrar	Editar
7 de septiembre de 2008, domingo		40-70 % Kilómetros 71-85 % Kilómetros 86-100 % Kilómetros	Kilómetros Kilómetros Kilómetros			Mostrar	Editar
Operaciones <input type="button" value="Añadir sesión"/> <ul style="list-style-type: none"> Mostrar mesociclo actual Editar 							
Lista de mis deportistas							

Ilustración 66: Listado de sesiones de un microciclo

Desde dentro de microciclo se puede pulsar sobre la opción crear sesión para añadir alguna más a las ya existentes, generadas automáticamente.

Los datos solicitados obligatorios se destacan y controlan al igual que en el resto del programa. La opción reportar, marcada por defecto, dice si el usuario tiene que realizar algún reporte del entrenamiento de ese día.

4.2.4.2.4 Visualización de resultados.

Cada nivel de periodización tiene los datos acumulados previstos y reportados por el deportista. En el nivel sesión el entrenador puede ver los comentarios que ha realizado el deportista sobre una sesión dada.

Sesión 5 de septiembre de 2008 para Persona Entrenada 1

Fecha:

5 de septiembre de 2008, viernes

Retroalimentación:

Me he encontrado bien durante los primero 40 minutos, pero al final tenía molestias en el gemelo.

Descripción:

Carrera continua Lenta 50'

Pulso al despertar:

42

Volumen teórico:

11

Volumen reportado:

11

Intensidad teórica:

40-70 % 10 Kilómetros

71-85 % 1 Kilómetros

86-100 % 0 Kilómetros

Intensidad reportada:

8 Kilómetros

3 Kilómetros

0 Kilómetros

Editar

Lista

Ilustración 67: Visualización de un sesión ya reportada por el usuario

4.2.5 Operaciones del usuario deportista

4.2.5.1 Consultar su entrenamiento diario

El deportista una vez identificado tiene en el menú la opción de "Ver mi día de entrenamiento". Al pulsar Gesport le mostrará el entrenamiento de hoy.

4.2.5.2 Reportar el resultado del entrenamiento.

En el menú deportista aparece la opción "Enviar resultado". Al pulsar sobre ella se le muestran los entrenamientos en los que tenga que reportar y no haya reportado el resultado.

Menú

- Usuario
 - o Salir
 - o Datos de admin
- Administrador
 - o Unidades
 - o Tipos de unidades
 - o Factor de conversión entre unidades
 - o Usuarios
- Entrenador
 - o Deportistas
- Deportista
 - o Ver mi día de entrenamiento
 - o Enviar resultado
- Acerca de Gesport
 - o Acerca de
 - o Notificar un error
 - o Proponer una sugerencia
- Concurso Universitario de Software Libre
 - o Portada
- Diego Cebrián
 - o Desarrollando Gesport
 - o Serincas

Sesión 6 de abril de 2009 para Nombre Apellido

Fecha:

6 de abril de 2009, lunes

Descripción:

Retroalimentación:

Pulso al despertar:

Volumen teórico:

Intensidad teórica:

35-70 % Kilómetros

71-85 % Kilómetros

86-100 % Kilómetros

Volumen reportado:

Intensidad reportada:

Kilómetros

Kilómetros

Kilómetros

Guardar

Ilustración 68: Mostrado de entrenamiento y captura de feedback para el entrenador. El interfaz es el mismo para el reporte que para la consulta.

Menú

- Usuario
 - o Salir
 - o Datos de admin
- Administrador
 - o Unidades
 - o Tipos de unidades
 - o Factor de conversión entre unidades
 - o Usuarios
- Entrenador
 - o Deportistas
- Deportista
 - o Ver mi día de entrenamiento
 - o Enviar resultado
- Gesport
 - o Acerca de
 - o Notificar un error
 - o Proponer una sugerencia
- Concurso Universitario de Software Libre
 - o Portada
- Diego Cebrián
 - o Desarrollando Gesport
 - o Serincas

Sesiones para Nombre Apellido

Fecha	Descripción	Intensidad teórica	Intensidad reportada	Volumen teórico	Volumen reportado	Opciones
1 de abril de 2009, miércoles		35-70 % Kilómetros 71-85 % Kilómetros 86-100 % Kilómetros	Kilómetros Kilómetros Kilómetros			Mostrar
2 de abril de 2009, jueves		35-70 % Kilómetros 71-85 % Kilómetros 86-100 % Kilómetros	Kilómetros Kilómetros Kilómetros			Mostrar
3 de abril de 2009, viernes		35-70 % Kilómetros 71-85 % Kilómetros 86-100 % Kilómetros	Kilómetros Kilómetros Kilómetros			Mostrar
4 de abril de 2009, sábado		35-70 % Kilómetros 71-85 % Kilómetros 86-100 % Kilómetros	Kilómetros Kilómetros Kilómetros			Mostrar
5 de abril de 2009, domingo		35-70 % Kilómetros 71-85 % Kilómetros	Kilómetros			

Encontrar:

Anterior

Siguiente

Resaltar todo

Coincidencia

Ilustración 69: Listado de sesiones pendientes por reportar.

4.2.6 Operaciones comunes a todos los usuarios

4.2.6.1 Visualización y edición del perfil

Todos los usuarios tienen un elemento que dice "Datos de Nombre" en el menú usuario. Allí pueden ver sus propios datos y cambiarlos a su gusto (interfaz idiomática, dirección de correo y contraseña).

The screenshot shows a web interface for user management. On the left is a vertical menu titled 'Menú' with the following items: 'Usuario' (highlighted), 'Salir', 'Datos de admin', 'Administrator', 'Unidades', 'Tipos de unidades', 'Factor de conversión entre unidades', 'Usuarios', 'Entrenador', 'Deportistas', 'Deportista', 'Ver mi día de entrenamiento', 'Enviar resultado', 'Acerca de Gesport', 'Acerca de', 'Notificar un error', 'Proponer una sugerencia', 'Concurso Universitario de Software Libre', 'Portada', 'Diego Cebrián', and 'Desarrollando'. The main content area is titled 'Usuario' and displays the following information: 'Nombre de usuario: admin', 'Nombre: Nombre', 'Apellido: Apellido', 'Fecha de Nacimiento: 20/06/81', 'Sexo: Femenino', 'Es entrenador: [check]', 'Es deportista: [check]', 'Es administrador: [check]', 'Entrenador: [check]', 'Correo electrónico: gesport@localhost.com', and 'Idioma y nacionalidad: es'. At the bottom of the main area are two buttons: 'Editar' and 'Cambiar clave', both highlighted with red boxes.

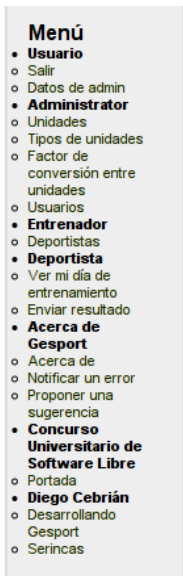
Ilustración 70: Visualización de datos de usuario

Para editar estos datos tienen que pulsar sobre el elemento "Editar" o "Cambiar clave" si es eso lo que desean.

This screenshot shows the password change form. The left menu is identical to the previous one. The main content area is titled 'Usuario' and shows 'Nombre de usuario: admin'. Below this are three input fields labeled 'Antigua clave:', 'Nueva clave:', and 'Repita nueva clave:'. At the bottom of the form are two buttons: 'Guardar' (highlighted with a red box) and 'Cancelar'.

Ilustración 71: Interfaz para el cambio de contraseña

4.2.6.2 Reporte de errores y sugerencias



Gesport está alojado en la Forja de desarrollo de software de RedIris. Si tiene algún problema durante la ejecución del programa, por favor, utilice el elemento "Notificar un error", que le llevará a un foro donde podrá explicar el error que ha encontrado.

Gesport es un programa que ha nacido, pero que sigue vivo, tanto en su ejecución como en su desarrollo. Es por ello que intentamos mejorarlo cada día. Si tiene alguna sugerencia que hacer sobre el programa puede realizarla en elemento "Realizar sugerencia".

Una vez que ha entrado en cualquiera de los foros tiene que pulsar sobre "Comenzar una nueva conversación". Se agradecería mucho que consultase si antes alguien había realizado la misma sugerencia o el mismo reporte de error que va a realizar usted para evitar duplicidades y trabajo innecesario a los desarrolladores de este proyecto.

Ilustración 72:
Sugerencias y
reporte de
errores

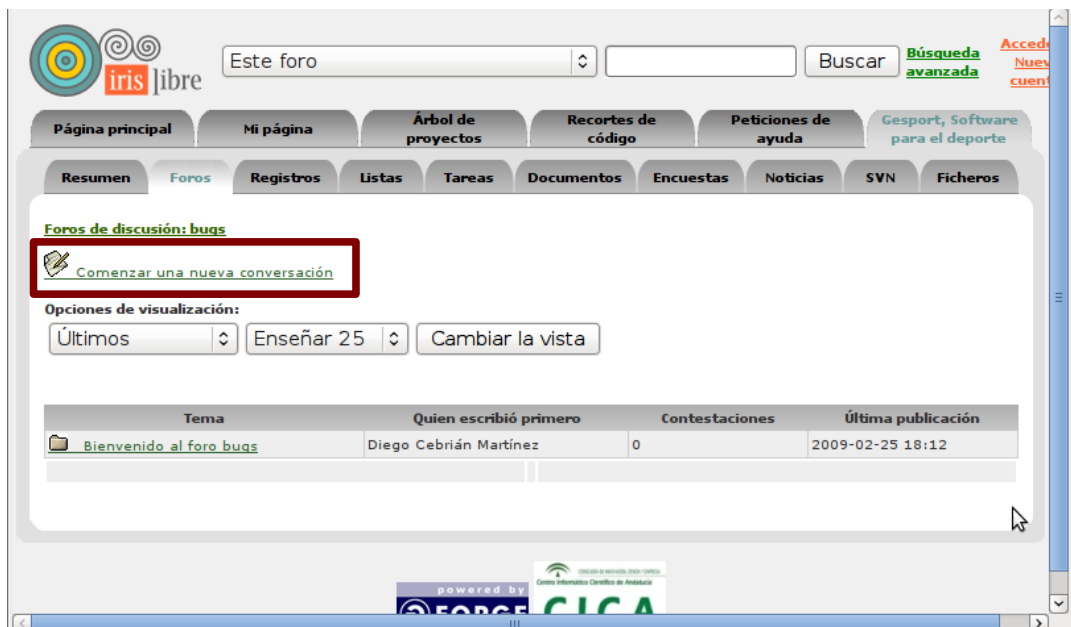


Ilustración 73: Reporte de errores y sugerencias

5. Índices de tablas e ilustraciones del proyecto

Relación de ilustraciones y tablas del proyecto

5.1 Índice de Ilustraciones

Ilustración 1: Detalle del funcionamiento de ArgoUML.....	20
Ilustración 2: Vista de tareas de proyecto de Egroupware.....	24
Ilustración 3: Vista de trabajo con Eclipse, coloreado de código y asistente de Symfony	36
Ilustración 4: Esquema ampliado MVC.....	39
Ilustración 5: Clases obtenidas del análisis.....	49
Ilustración 6: Diagrama de casos de uso simplificado.....	53
Ilustración 7: Escenarios de tipo de unidad.....	55
Ilustración 8: Escenarios de unidad.....	57
Ilustración 9: Escenarios de factor de conversión.....	59
Ilustración 10: Escenarios de Usuario.....	61
Ilustración 11: Escenarios de temporada.....	63
Ilustración 12: Escenarios de Macrociclo.....	65
Ilustración 13: Escenarios de mesociclo.....	67
Ilustración 14: Escenarios de microciclo.....	69
Ilustración 15: Escenarios de sesión.....	71
Ilustración 16: Escenarios de deportista.....	73
Ilustración 17: Escenarios feedback.....	74
Ilustración 18: Escenarios de entrada y salida del sistema.....	76
Ilustración 19: Infraestructura de un solo servidor sin virtualización.....	77
Ilustración 20: Infraestructura virtualizada.....	77
Ilustración 21: Diagrama de clases del ámbito de los usuarios y los artículos de portada.....	90
Ilustración 22: Diagrama de clases para los entrenamientos.....	91
Ilustración 23: Diagrama de clases relacionadas con el sistema de unidades.....	92
Ilustración 24: Clases adicionales generadas por Symfony.....	94
Ilustración 25: Página de descarga de Ubuntu Server.....	99
Ilustración 26: Comprobación de resumen de datos de CD de instalación.....	100
Ilustración 27: Menú de selección de idioma en el proceso de instalación de Ubuntu Server.....	101
Ilustración 28: Particionado válido para Gesport en una máquina virtual con 4Gb de espacio y 128 Mb de RAM.....	102
Ilustración 29: Selección de programas a instalar.....	103
Ilustración 30: Barra de direcciones del navegador.....	109
Ilustración 31: Inicio de Aplicación.....	109

Ilustración 32: Pantalla de acceso de usuarios.....	110
Ilustración 33: Error de acceso.....	110
Ilustración 34: Solicitud de regeneración de clave por pérdida.....	111
Ilustración 35: Error en la solicitud de nueva contraseña.....	111
Ilustración 36: Envío satisfactorio de solicitud de nueva contraseña.....	111
Ilustración 37: Solicitud de acceso.....	112
Ilustración 38: Confirmación de envío de correo electrónico.....	112
Ilustración 39: Errores en solicitud de usuario.....	113
Ilustración 40: Operaciones con Usuarios.....	113
Ilustración 41: Listado de usuarios.....	114
Ilustración 42: Formulario de creación de usuario.....	114
Ilustración 43: Edición de usuarios.....	115
Ilustración 44: Confirmación de Borrado.....	115
Ilustración 45: Error de borrado por usuario con referencias.....	116
Ilustración 46: Menú tipos de unidades.....	116
Ilustración 47: Listado de tipos de unidades.....	117
Ilustración 48: Añadido de un tipo de unidad.....	117
Ilustración 49: Edición tipo de unidad.....	117
Ilustración 50: Elemento Unidades.....	118
Ilustración 51: Listado de unidades.....	118
Ilustración 52: Creación de nueva unidad.....	119
Ilustración 53: Edición de unidad.....	119
Ilustración 54: Elemento Factores de conversión.....	120
Ilustración 55: Listado de factores de conversión.....	120
Ilustración 56: Creación de factores de conversión.....	121
Ilustración 57: Edición de factores de conversión.....	121
Ilustración 58: Listado de deportista de un entrenador del sistema.....	122
Ilustración 59: Formulario de creación de un nuevo deportista.....	123
Ilustración 60: Edición de un deportista.....	123
Ilustración 61: Vista de un deportista sin temporadas planificadas.....	124
Ilustración 62: Vista de un deportista con una temporada planificada y otra en ejecución.....	124
Ilustración 63: Formulario de creación de un ciclo de entrenamiento.....	125
Ilustración 64: Error de solapamiento de fechas.....	126

Gesport - Gestión de Entrenamiento Deportivo

Ilustración 65: Edición de ciclos de entrenamiento.....	127
Ilustración 66: Listado de sesiones de un microciclo.....	128
Ilustración 67: Visualización de un sesión ya reportada por el usuario.....	129
Ilustración 68: Mostrado de entrenamiento y captura de feedback para el entrenador. El interfaz es el mismo para el reporte que para la consulta.....	130
Ilustración 69: Listado de sesiones pendientes por reportar.....	130
Ilustración 70: Visualización de datos de usuario.....	131
Ilustración 71: Interfaz para el cambio de contraseña.....	131
Ilustración 72: Sugerencias y reporte de errores.....	132
Ilustración 73: Reporte de errores y sugerencias.....	132

5.2 Índice de tablas

Tabla 1: Presupuesto global de desarrollo.....	25
Tabla 2: Glosario de términos deportivos.....	45
Tabla 3: Actores representados en los casos de uso.....	50
Tabla 4: Descripción de los casos de uso.....	52
Tabla 5: Escenario "Añadir nuevo tipo de unidad".....	54
Tabla 6: Escenario "Editar tipo de unidad".....	54
Tabla 7: Escenario "Mostrar tipo de unidad".....	54
Tabla 8: Escenario "Borrar tipo de unidad".....	55
Tabla 9: Escenario "Añadir nueva unidad".....	56
Tabla 10: Escenario "Editar unidad".....	56
Tabla 11: Escenario "Mostrar unidad".....	56
Tabla 12: Escenario "Borrar unidad".....	57
Tabla 13: Escenario "Añadir nuevo factor de conversión".....	58
Tabla 14: Escenario "Editar factor de conversión".....	58
Tabla 15: Escenario "Mostrar factor de conversión".....	58
Tabla 16: Escenario "Borrar factor de conversión".....	59
Tabla 17: Escenario "Añadir nuevo usuario".....	60
Tabla 18: Escenario "Editar usuario".....	60
Tabla 19: Escenario "Mostrar usuario".....	60
Tabla 20: Escenario "Borrar usuario".....	61
Tabla 21: Escenario "Añadir nueva temporada".....	62
Tabla 22: Escenario "Editar temporada".....	62
Tabla 23: Escenario "Borrar temporada".....	63
Tabla 24: Escenario "Mostrar temporada".....	63
Tabla 25: Escenario "Añadir nuevo macrociclo".....	64
Tabla 26: Escenario "Editar macrociclo".....	64
Tabla 27: Escenario "Borrar macrociclo".....	65
Tabla 28: Escenario "Mostrar macrociclo".....	65
Tabla 29: Escenario "Añadir nuevo mesociclo".....	66
Tabla 30: Escenario "Editar mesociclo".....	66
Tabla 31: Escenario "Borrar mesociclo".....	67

Gesport - Gestión de Entrenamiento Deportivo

Tabla 32: Escenario "Mostrar mesociclo".....	67
Tabla 33: Escenario "Añadir nuevo microciclo".....	68
Tabla 34: Escenario "Editar microciclo".....	68
Tabla 35: Escenario "Borrar microciclo".....	69
Tabla 36: Escenario "Mostrar microciclo".....	69
Tabla 37: Escenario "Añadir nuevo sesión".....	70
Tabla 38: Escenario "Editar sesión".....	70
Tabla 39: Escenario "Borrar sesión".....	71
Tabla 40: Escenario "Mostrar sesión".....	71
Tabla 41: Escenario "Añadir nuevo deportista".....	72
Tabla 42: Escenario "Editar deportista".....	72
Tabla 43: Escenario "Mostrar deportista".....	72
Tabla 44: Escenario "Editar feedback".....	73
Tabla 45: Escenario "Mostrar feedback".....	74
Tabla 46: Escenario "Entrar al sistema".....	74
Tabla 47: Escenario "Salir del sistema".....	75
Tabla 48: Escenario "Recordar Contraseña".....	75
Tabla 49: Escenario "Solicitar usuario".....	75
Tabla 50: Clase Admin, representa al administrador de la aplicación.....	78
Tabla 51: Clase Article, para los artículos que se muestran en la página de inicio.....	79
Tabla 52: Clase ArticleI18N, para la internacionalización de los artículos.....	79
Tabla 53: Clase Athlete, que representa a los deportistas.....	80
Tabla 54: Clase Coach, que representa a los entrenadores.....	80
Tabla 55: Clase ConversionFactor, para la transformación entre unidades.....	81
Tabla 56: Clase Macrocycle. Representación del ciclo mayor con objetivos.....	82
Tabla 57: Clase Mesocycle, que determina los periodos para alcanzar los objetivos.....	83
Tabla 58: Clase Microcycle, que describe los trabajos aproximadamente semanales.....	84
Tabla 59: Clase season, que representa la temporada completa de un deportista.....	85
Tabla 60: Clase Session, es el trabajo a realizar en un momento determinado.....	86
Tabla 61: Clase Unit, para las unidades de los feedback, volúmenes e intensidad.....	87
Tabla 62: Clase UnitI18N, para las traducciones de las unidades.....	87
Tabla 63: Clase UnitType, para agrupar los diferentes tipos de unidades.....	88
Tabla 64: Clase UnitTypeI18N, para traducir los tipos de unidades.....	88

Tabla 65: Clase User, para describir a los usuarios del sistema.....89

6. Bibliografía

6.1 Sobre la teoría del entrenamiento deportivo

Apuntes del bloque común del curso de monitor polideportivo, Varios autores.

Fundamentos del Atletismo. Por Francisco Gil y Arturo Oliver..

6.2 Sobre el análisis y diseño de software

Requirements Engineering, Ian Sommerville & Pete Sawyer, 1997.

Análisis Orientado a Objetos, Juan Manuel Cueva Lovelle (Universidad de Oviedo)

Diseño Orientado a Objetos, Juan Manuel Cueva Lovelle (Universidad de Oviedo)

ORIENTACIÓN A OBJETOS: CONCEPTOS, TERMINOLOGÍA Y LENGUAJES (PARTE 1)
Miguel Ángel Abián

Orientación a Objetos. Conceptos y terminología (II Parte).Miguel Ángel Abián

6.3 Sobre el paradigma Modelo Vista Controlador

JavaServer Faces Wiki [<http://wiki.java.net/bin/view/Projects/JavaServerFacesSpecFaq>]

Spring reference [<http://static.springframework.org/spring/docs/2.0.x/spring-reference.pdf>]

6.3 Sobre el lenguaje de programación PHP y el framework Symfony

Documentación original de PHP [<http://www.php.net/manual/es/>]

Tutorial de inicio a Symfony Askeet [http://www.symfony-project.org/askeet/1_0/en/]

Symfony 1.0, la guía definitiva, François Zaninotto y Fabien Potencier
[http://www.librosweb.es/symfony_1_0/pdf/]

PHPDocumentor. Varios autores

6.4 Sobre la persistencia de los datos en el software

Propel 1.2 Documentation [<http://propel.phpdb.org/trac/wiki/Users/Documentation/1.2>]

Doctrine ORM for PHP [http://www.doctrine-project.org/documentation/manual/1_0/en/pdf]

Hibernate reference [http://www.hibernate.org/elqNow/elqRedir.htm?ref=http://www.hibernate.org/hib_docs/v3/reference/en-US/pdf/hibernate_reference.pdf]

6.5 Sobre el diseño de interfaces gráficas web

W3c.org XHTML 1.0 Recommendation [<http://www.w3.org/TR/xhtml1/>]

Cascading Style Sheets, level 2 CSS2 Specification [<http://www.w3.org/TR/CSS2/>]

6.6 Sobre los sistemas operativos y su manejo como administrador

Guía de referencia Debian, Osamu Aoki , traducción por Walter O. Echarri.

Ubuntu Server Guide, Canonical [<https://help.ubuntu.com/8.04/serverguide/C/index.html>]

Mysql Reference Manual. MySQL.

7. Estructura de ficheros del CD

Relación de material incluido en el soporte óptico

Se ha evitado el uso de caracteres extraños en todo momento para garantizar la compatibilidad del CD con todos los sistemas operativos y todas las configuraciones locales de caracteres.

La localización de los ficheros de documentación especificados en la normativa está situados en la raíz del soporte óptico. Estos son Titulo.pdf, Memoria.pdf, Solapa.pdf y Resumen.pdf. Además de los ficheros en pdf se adjuntan, bajo licencia Creative Commons de reconocimiento y compartir igual, los ficheros fuentes en Open Document que generaron estos PDFs.

Así queda la siguiente estructura de ficheros:

- /Raíz
 - Software
 - complete
 - Carpetas del programa
 - doc
 - concurso
 - documentacion
 - Entrenamientos
 - PHPDocumentor
 - POO
 - protecciondatos
 - SO
 - symfony
 - UML
 - universidad
 - diseno_grafico
 - licencias
 - programas
 - manuales
 - developers
 - Carpetas del programa
 - doc = que en "complete"
 - Titulo.pdf
 - Titulo.odt
 - Solapa.pdf
 - Solapa.odt

- Memoria.pdf
- Memoria.odt
- Resumen.pdf
- Resumen.odt
- Diapositivas.pdf
- Diapositivas.odp

7.1 Código fuente de la aplicación

Al haberse utilizado un lenguaje interpretado este código fuente coincide con el ejecutable por el mismo interprete. Al haberse utilizado un framework y haber trabajado diariamente en el desarrollo con las librerías enlazadas dinámicamente se van a incluir en la carpeta "Software" las carpetas "complete", que es la versión utilizada para la instalación con la librerías ya incluidas y preparada para funcionar, y la carpeta "developers", que contiene únicamente el código del programa con los enlaces al framework necesarios para funcionar con un enlace simbólico en la configuración del proyecto.

7.2 Documentación consultada para la realización del programa

Symfony provee una estructura de ficheros diseñada para establecer la documentación de manera interna y se ha respetado este aspecto. Toda la documentación del programa estará en la carpeta "doc". Esta carpeta se situará en la raíz de la carpeta que contiene el código fuente.

En la carpeta "doc/documentacion" se han reunido todos los documentos electrónicos consultados durante la realización del programa. Aparecen todos en la bibliografía, pues aunque son fuentes documentales digitales, páginas web de comunidades de usuarios de programadores y artículos escritos sin formato de papel para diferentes sitios de internet, los he considerado fuente fundamental de información a la hora de realizar este proyecto dada la poca cantidad de documentación existente acerca de Symfony fuera del ámbito de la web. De hecho el libro editado por los creadores de Symfony en el que se describen todos los detalles de como trabajar con el está disponible directamente desde su web.

7.3 Programas usados para la realización del programa

Dentro de la misma carpeta "doc" del código fuente se incluyen las herramientas que se han usado para el desarrollo de Gesport. Se sitúan en la subcarpeta "programas". Estos son:

- Eclipse Europa + PDT (PHP Developer Tools) + Symfoclipse
- ArgoUML 0.26
- PHP Documentor.

No se incluyen la distribución Ubuntu ni Symfony. Una al ser demasiado grande para el soporte óptico de CD-ROM y la segunda por formar parte del código fuente en la versión "complete".

7.4 Logotipos del programa en formato vectorial y mapa de bits

Dentro de la carpeta "doc" en "diseno_grafico" está la imagen de concepto del programa, así como los diferentes logotipos que se han usado para representar la imagen de cara a una posible explotación comercial.

7.5 Manual de usuario y de administrador.

Se han realizado impresiones externas de los manuales de la aplicación para su publicación en la web del proyecto. Se incluyen como documentos anejos por el cambio en la paginación y portadas, ya con la imagen corporativa asociada al software. Están situados en la carpeta "doc" del código de proyecto, dentro de la subcarpeta "manuales"

7.6 Manual del API de Gesport.

Para el desarrollo de futuras aplicaciones con base en Gesport se ha documentado el API (Application Program Interface). Este está, al igual que el resto de documentación dentro de la carpeta "doc" en una subcarpeta llamada API. Está organizado por paquetes, entre los que caben destacar lib (el conjunto de librerías básicas del sistema) y frontend (las librerías y clases utilizadas en la aplicación principal).

7.7 Licencias de programas

Gesport, como se ha comunicado anteriormente, se distribuye bajo licencia GNU GPL y su documentación bajo licencia Creative Commons, pero en el desarrollo del proyecto se han usado diferentes programas con sus diferentes licencias:

- Eclipse: Eclipse Public License - v 1.0
- Symfoclipse: Libre uso para proyectos de software libre
- Symfony: MIT
- ArgoUML: BSD
- Linux, PHP Documentor y EGroupWare: GNU GPL
- Apache y Subversion: Apache License 2.0
- PHP: PHP License 3.01

Todas estas licencias se encuentran recogidas en "doc/licencias" dentro del código fuente del proyecto.

7.8 Diapositivas de la presentación

En el mismo directorio raíz se encuentran las diapositivas de la presentación que se realizará ante el jurado. El fichero se llama "Diapositivas.odp". Está en formato Open Document .y se adjunta también una versión en PDF para la correcta visualización si no se dispone del programa para la visualización.

7.9 Documentación del III Concurso Universitario de Software Libre

En la carpeta "doc/concurso" se encuentran todos los documentos presentados para el III Concurso Universitario de Software Libre. Entre ellos se encuentran tanto las bases del concurso, como los datos enviados para la valoración así como un artículo para la revista "Linux Magazine" que será publicado en el número de Mayo o Junio.