

**Ingeniería Técnica de Telecomunicaciones:  
Telemática, Octubre 2009**

Proyecto Fin de Carrera:  
**Gestión de Campeonatos  
de Competición Interna.**



*Universidad Carlos III de Madrid  
Escuela Politécnica Superior*

**AUTORA: Marta Andrés Morena  
TUTOR: Vicente Luque Centeno**



Proyecto Fin de Carrera  
GESTIÓN DE CAMPEONATOS DE COMPETICIÓN  
INTERNA

Autor  
MARTA ANDRÉS MORENA

Tutor  
VICENTE LUQUE CENTENO

La defensa del presente Proyecto Fin de Carrera se realizó el día 14 de 10 de 2009, siendo calificada por el siguiente tribunal:

PRESIDENTE:

SECRETARIO:

VOCAL:

y habiendo obtenido la siguiente calificación:

CALIFICACIÓN:

Leganés, a 14 de Octubre de 2009



## Resumen

El crecimiento de las redes de comunicaciones ha permitido tener acceso a la información y los servicios a un gran número de usuarios. En el ámbito de la oferta y la demanda, las empresas deben ofrecer el máximo número de servicios a través de la red para satisfacer las necesidades de los usuarios.

En este contexto, se presenta el proyecto «**Gestión de Campeonatos de Competición Interna**», que consiste en desarrollar, mediante las redes de telecomunicaciones, un entorno para que todo usuario de la Universidad Carlos III tenga acceso a la información relacionada con las actividades deportivas de competición interna ofertadas por la misma.

La aplicación ofrece, fundamentalmente, soluciones para que el usuario pueda realizar consultas e interactuar vía web con todo lo relacionado con sus actividades deportivas sin que sea necesario que se presente físicamente en la Universidad para realizarlas.

Mediante esta aplicación Web, el único requisito que se necesita para utilizarla es que el usuario disponga de un navegador y conexión a Internet. De esta manera, podrá realizar consultas a cualquier hora y desde cualquier punto.



# Indice General

---

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	3
1.3. Contenido de la Memoria . . . . .	3
<b>2. Tecnologías Usadas</b>	<b>5</b>
2.1. Introducción . . . . .	5
2.2. Tecnologías de Desarrollo WEB . . . . .	5
2.2.1. HTML . . . . .	5
2.2.2. CSS . . . . .	6
2.2.3. Sesiones . . . . .	7
2.2.4. JSP . . . . .	8
2.2.5. Servlets . . . . .	9
2.2.6. JavaScript . . . . .	10
2.2.7. JavaMail . . . . .	10
2.3. Tecnologías sobre la BBDD . . . . .	11
2.3.1. SGBD: SQLServer . . . . .	11
2.3.2. Apache . . . . .	15
2.4. LDAP . . . . .	18
<b>3. Requisitos</b>	<b>23</b>
3.1. Requisitos Iniciales . . . . .	23
3.1.1. <b>R1. Acceso</b> . . . . .	23
3.1.2. <b>R2. Permisos</b> . . . . .	23
3.1.3. <b>R3. Perfil Usuario sin permisos.</b> . . . . .	23
3.1.4. <b>R4. Perfil Delegado.</b> . . . . .	24
3.1.5. <b>R5. Perfil Administrador.</b> . . . . .	24
3.2. Requisitos Finales . . . . .	24
3.2.1. <b>Ampliaciones relacionadas con el Perfil Delegado.</b> . . . .	24

<b>4. Diseño de la Interfaz. Perfiles</b>	<b>27</b>
4.1. Introducción. . . . .	27
4.2. Acceso a la aplicación. . . . .	31
4.3. Perfil Usuario. . . . .	31
4.3.1. <b>Resultados.</b> . . . . .	33
4.3.2. <b>Clasificación.</b> . . . . .	34
4.3.3. <b>Deportividad.</b> . . . . .	37
4.3.4. <b>Goleadores y Anotadores.</b> . . . . .	37
4.3.5. <b>Selección de otro deporte.</b> . . . . .	38
4.4. Perfil Delegado. . . . .	39
4.4.1. <b>Insertar Resultado.</b> . . . . .	40
4.4.2. <b>Horarios Disponibles para Aplazamientos.</b> . . . . .	44
4.4.3. <b>Datos Delegados de tu competición.</b> . . . . .	46
4.4.4. <b>Aplazar Partido.</b> . . . . .	48
4.5. Perfil Administrador. . . . .	51
4.5.1. <b>Confirmar Resultado.</b> . . . . .	52
4.5.2. <b>Datos Delegados de tu competición.</b> . . . . .	54
<b>5. Diseño de Alto Nivel</b>	<b>55</b>
5.1. Selección de tecnologías. . . . .	55
5.1.1. <b>MVC: Modelo Vista Controlador</b> . . . . .	55
5.1.2. <b>Patrón de desarrollo: Servlet y JSP</b> . . . . .	56
5.2. Arquitectura de la aplicación. . . . .	57
5.2.1. <b>Módulo de Almacenamiento.</b> . . . . .	58
5.2.2. <b>Módulo de procesamiento de la información.</b> . . . . .	58
5.2.3. <b>Módulo de visualización.</b> . . . . .	59
<b>6. Diseño de Bajo Nivel</b>	<b>61</b>
6.1. DeporWin: BBDD del Departamento de Actividades Físicas y Culturales. . . . .	61
6.2. Modelo de Datos. . . . .	62
6.3. Objetos utilizados. . . . .	62
6.3.1. <b>Actas.</b> . . . . .	64
6.3.2. <b>Campeonatos.</b> . . . . .	64
6.3.3. <b>Categoría.</b> . . . . .	65
6.3.4. <b>Deporte.</b> . . . . .	66
6.3.5. <b>Equipos.</b> . . . . .	66
6.3.6. <b>Horarios Recursos Semanales.</b> . . . . .	67
6.3.7. <b>Integrantes.</b> . . . . .	68
6.3.8. <b>Inscritos.</b> . . . . .	68
6.3.9. <b>Partidos.</b> . . . . .	69

6.3.10. <b>Personas.</b> . . . . .	70
6.3.11. <b>Recursos.</b> . . . . .	70
6.3.12. <b>Reservas Recursos.</b> . . . . .	71
6.3.13. <b>Sanciones.</b> . . . . .	72
6.3.14. <b>Sexo.</b> . . . . .	72
6.3.15. <b>Temporada.</b> . . . . .	72
6.3.16. <b>Tipos Resultados.</b> . . . . .	73
6.3.17. <b>Tmp Clasificación Liga.</b> . . . . .	73
<b>7. Implementación</b>	<b>75</b>
7.1. Acceso a la aplicación. . . . .	75
7.2. Perfil Usuario. . . . .	76
7.2.1. <b>Resultados y Clasificación.</b> . . . . .	76
7.3. Perfil Delegado. . . . .	76
7.3.1. <b>Insertar Resultados.</b> . . . . .	76
7.3.2. <b>Aplazar Partidos.</b> . . . . .	78
7.4. Problemas y Soluciones del Perfil Administrador. . . . .	78
7.4.1. <b>Confirmar Resultados.</b> . . . . .	79
<b>8. Pruebas</b>	<b>81</b>
<b>9. Planificación y Presupuesto</b>	<b>87</b>
9.1. Planificación . . . . .	87
9.2. Presupuesto . . . . .	87
<b>10. Conclusiones y Trabajos Futuros</b>	<b>91</b>
10.1. Conclusiones . . . . .	91
10.2. Trabajos Futuros . . . . .	95
<b>A. Anexo Base de datos: DeporWin.</b>	<b>97</b>
A.1. Tablas vacías. . . . .	97
A.2. Tablas a modificar y motivos. . . . .	97
<b>B. Entorno de desarrollo utilizado</b>	<b>101</b>
B.1. ECLIPSE . . . . .	101
B.2. TOAD . . . . .	108
<b>C. Manual de instalación del entorno utilizado</b>	<b>115</b>
C.1. CONEXIÓN REALIZADA FUERA DE LA UNIVERSIDAD . . . . .	115
C.1.1. Conexión VPN . . . . .	115
C.2. CONEXIÓN REALIZADA DENTRO DE LA UNIVERSIDAD . . . . .	116
C.2.1. Instalación JAVA . . . . .	116

## INDICE GENERAL

---

C.2.2. Instalación TOMCAT . . . . .	117
C.2.3. Instalación ECLIPSE . . . . .	117
C.2.4. Instalación TOAD . . . . .	118
<b>D. Glosario</b>	<b>119</b>

## Lista de Figuras

---

2.1. Logotipo de Apache Software Foundation . . . . .	15
2.2. Árbol LDAP de la UC3M . . . . .	20
2.3. Tabla comparativa. . . . .	22
4.1. Diagrama UML. Perfil Usuario . . . . .	29
4.2. Diagrama UML. Perfil Delegado . . . . .	30
4.3. Diagrama UML. Perfil Administrador . . . . .	30
4.4. Página de Autenticación . . . . .	31
4.5. Página Principal Usuario . . . . .	32
4.6. Opciones del Usuario. . . . .	33
4.7. Resultados . . . . .	34
4.8. Clasificación . . . . .	36
4.9. Deportividad . . . . .	37
4.10. Goleadores y Anotadores . . . . .	38
4.11. Página Principal Delegado . . . . .	39
4.12. Acciones Delegado . . . . .	40
4.13. Página Principal Inserción . . . . .	41
4.14. Formulario a insertar por el delegado . . . . .	43
4.15. Inserción Errónea . . . . .	43
4.16. Resumen de la inserción . . . . .	44
4.17. Fechas y Horarios. Selección de campus. . . . .	44
4.18. Selección de Pista y Fecha . . . . .	45
4.19. Consulta de la pista . . . . .	46
4.20. Pista Ocupada . . . . .	47
4.21. Datos Delegados Competición . . . . .	47
4.22. Aplazamiento. Selección de grupo . . . . .	49
4.23. Formulario Aplazamientos . . . . .	50
4.24. Email de solicitud del aplazamiento . . . . .	50
4.25. Error al solicitar un aplazamiento. . . . .	51
4.26. Opciones del Administrador . . . . .	52

## LISTA DE FIGURAS

---

4.27. Lista de resultados por confirmar . . . . .	52
4.28. Detalle de la confirmación . . . . .	53
4.29. Resultado Confirmado por parte del administrador . . . . .	54
5.1. Arquitectura de la aplicación. . . . .	57
6.1. Diagrama Relacional . . . . .	63
9.1. Listado de tareas . . . . .	88
9.2. Tareas representadas en barras . . . . .	88
10.1. Simulador virtual. Página de Autenticación. . . . .	93
10.2. Simulador virtual. Página de Resultados. . . . .	94
B.1. Logotipo Eclipse . . . . .	101
B.2. Ruta de creación del proyecto. . . . .	102
B.3. Lenguaje de Programación . . . . .	102
B.4. Creación de Proyecto . . . . .	103
B.5. Configuración Servidor . . . . .	103
B.6. Clases y Paquetes que contiene el proyecto . . . . .	104
B.7. Elementos Java en Eclipse . . . . .	105
B.8. Creación de una clase con eclipse . . . . .	106
B.9. Detección de errores . . . . .	107
B.10. Autocorrección del eclipse . . . . .	107
B.11. Tipos de ejecución . . . . .	108
B.12. Conexión con la BBDD . . . . .	110
B.13. Navegador TOAD . . . . .	110
B.14. scale=0.1 . . . . .	111
B.15. Consulta y Resultados . . . . .	112
C.1. Conexión a la VPN . . . . .	116
C.2. Modificación de las variables de entorno . . . . .	117

## Lista de Tablas

---

4.1. Puntuaciones Deportivas . . . . .	36
8.1. Pruebas Realizadas. . . . .	85
9.1. Presupuesto . . . . .	89

## LISTA DE TABLAS

---

# Introducción

---

Las redes de comunicaciones, y en nuestro caso Internet, no solo permite realizar una labor profesional sino también actividades de ocio desde cualquier punto donde se tenga acceso a la red.

Actualmente, Internet es el medio más rápido, económico y eficaz que facilita una comunicación a tiempo real con el resto del mundo, superando así, al resto de medios de comunicación implantados en nuestra sociedad.

Los medios de comunicación han tenido que adaptarse a las nuevas tecnologías, encontrando en Internet una vía rápida y eficaz para poder subsistir y avanzar a nivel técnico en el siglo XXI.

Desde el año 1998 Internet crece al 100 por cien de su capacidad, y actualmente sigue creciendo al mismo ritmo, teniendo una infraestructura similar a la telefónica.

Este capítulo de introducción consta de tres apartados. En el primero se describe la motivación para el desarrollo del proyecto, a continuación se representan los objetivos iniciales del mismo, y para finalizar se hace un breve resumen de la estructura de la memoria, sus capítulos y contenidos.

## 1.1. Motivación

Este proyecto se desarrolla a partir de la necesidad que le surge al usuario de acceder a un sistema rápido y eficaz de consulta de información sobre las actividades deportivas vía Web. Se facilita, de esta forma, el acceso directo con el departamento encargado de la gestión de las Actividades Deportivas y Culturales

## CAPÍTULO 1. INTRODUCCIÓN

---

(Espacio de Estudiantes), evitando así tener que acceder a la prestación de servicios vía telefónica o teniendo que acudir físicamente a las instalaciones.

Ante el crecimiento de la sociedad de la información, el desarrollo de los medios de comunicación y las ventajas que nos proporcionan los avances tecnológicos, el departamento de actividades deportivas y culturales de la Universidad Carlos III se adapta a las necesidades de sus usuarios para ofrecer una amplia gama de información relacionada con los deportes que se ofertan en la propia universidad.

Actualmente, la página de la Universidad Carlos III de Madrid ofrece una aplicación para la consulta de información de diversas actividades deportivas, pero debido al rápido crecimiento de las tecnologías (como se comentaba en el apartado anterior) ha sido necesario ampliar las funcionalidades de la aplicación para una mayor satisfacción de sus usuarios.

El proyecto que se presenta a continuación, consistirá en mostrar todas los servicios que ya tenía la Universidad en relación con las actividades deportivas, más una ampliación que se realiza para que el usuario pueda interactuar directamente con el administrador, y conseguir una mayor implicación en lo referido a su deporte.

El motivo de no ampliar la herramienta a partir de la existente es porque dicha aplicación fue diseñada por una empresa subcontratada por la Universidad. El hecho de no tener acceso al código ni a la estructura, propicia la creación de una aplicación nueva, en la que sí se han mantenido los mismos registros de las actividades deportivas utilizando únicamente la BBDD ya creada.

Este proyecto se basa en una arquitectura software MVC (Modelo Vista Controlador) que se encarga de separar los datos de una aplicación, la interfaz de usuario y la lógica de control. Este modelo es el más adecuado para proyectos Web que interactúan con los datos de manera dinámica.

Gracias a la actualización de contenidos de la BBDD y al acceso dinámico de los usuarios, siempre obtendrán la información recientemente editada.

Una vez iniciado el curso, y empezado las actividades deportivas, los usuarios pueden beneficiarse de la información totalmente actualizada a cualquier hora y en cualquier punto, asegurando así que el uso de las redes de comunicación beneficia en todo momento al usuario.

## 1.2. Objetivos

Este proyecto se plantea como una forma de integrar en una misma herramienta la funcionalidad de la tecnología Web con el acceso a la BBDD y una presentación de la información de manera adecuada.

A continuación se presentan los objetivos que se marcaron inicialmente para la realización del proyecto:

- El objetivo principal es que el usuario pueda realizar en tiempo real consultas relacionadas con las actividades deportivas vía Web.
- La aplicación debe proporcionar a los usuarios las opciones de consultar las clasificaciones, resultados, deportividad y anotadores/goleadores de todos los equipos, así como la posibilidad de interactuar con el administrador para insertar resultados.
- El siguiente objetivo es crear distintos perfiles con distintas funcionalidades según el tipo de usuario: jugador/delegado del equipo o administrador de las actividades deportivas.
- El objetivo gráfico es mostrar una interfaz de usabilidad sencilla que permita al usuario navegar sin dificultad por ella. Se presupone que el usuario no tiene que tener conocimiento acerca de la tecnología utilizada.
- Ofrecer seguridad para los usuarios mediante validación.
- Finalmente, que la visualización sea correcta en el navegador utilizado y en los distintos dispositivos portátiles. Cumplir unas pautas básicas de accesibilidad.

## 1.3. Contenido de la Memoria

Esta memoria consta de diez capítulos y dos apéndices explicando la funcionalidad, implementación, desarrollo y diseño del Proyecto Fin de Carrera desarrollado.

## CAPÍTULO 1. INTRODUCCIÓN

---

A continuación se pasa a explicar brevemente el contenido de los capítulos:

En el **Capítulo 1** se explica la motivación, objetivos y contenidos de los que consta el proyecto.

El **Capítulo 2** trata sobre el **Estado del Arte** y se encarga de explicar las distintas tecnologías utilizadas a lo largo del desarrollo de la aplicación. Ya sean tecnologías Web, tecnologías de acceso e ínter actuación con la BBDD y tecnologías de seguridad y validación.

El **Capítulo 3** habla sobre los **Requisitos** necesarios para cumplir los objetivos del proyecto.

El **Capítulo 4** desarrolla los **Perfiles** utilizados en la aplicación: Perfil usuario normal (puede ser jugador o no), Perfil Delegado (jugador que es delegado de su equipo) y Perfil Administrador.

En el **Capítulo 5** se habla del **Diseño de Bajo Nivel** donde se detalla la BBDD que se va a utilizar, el modelo de datos de la aplicación y los objetos necesarios para la solución del problema.

El **Capítulo 6** describe el **Diseño de Alto Nivel** donde se justificará el uso de determinadas tecnologías en la aplicación.

El **Capítulo 7** detalla la **Implementación**, explicando el producto desarrollado.

El **Capítulo 8** son las **Pruebas** realizadas en la aplicación, que muestran los resultados del trabajo realizado evaluado como Apto o No Apto.

El **Capítulo 9** calcula **La Planificación y el Presupuesto** de la realización del proyecto, teniendo en cuenta el tiempo dedicado y los recursos empleados.

El **Capítulo 10** trata sobre las **Conclusiones y trabajos futuros**. En él se expresan las conclusiones alcanzadas una vez se ha desarrollado el proyecto y se plantean posibles trabajos futuros como continuación de la línea de este proyecto.

Por último se especifica la **Bibliografía** utilizada y dos apéndices, que describen el entorno de desarrollo y el manual de instalación respectivamente.

# Tecnologías Usadas

---

## 2.1. Introducción

Este capítulo no se llama «Estado del Arte» como tal, ya que a la hora de desarrollar la aplicación se nos dió directamente las tecnologías que debíamos utilizar, sin ser necesario realizar un estudio de las mismas.

Por ese motivo, esta capítulo se llama «Tecnologías Usadas».

A continuación explico cada una de esas tecnologías.

## 2.2. Tecnologías de Desarrollo WEB

### 2.2.1. HTML

HyperText Markup Language, cuyas siglas son HTML, es un lenguaje de marcado utilizado para la construcción de páginas Web.

Un lenguaje de marcado es una forma de codificar el documento que, junto al texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o la presentación del mismo.

El lenguaje de marcado más extendido es HTML, fundamento de World Wide Web [18].

#### Características.

HTML permite:

- Crear lenguajes de codificación descriptivos a través de las etiquetas.

- Definir una estructura de documentos jerárquica, con elementos y componentes interconectados.
- Proporcionar flexibilidad en el conjunto de juegos de etiquetas a utilizar.
- Generar documentos legibles.
- Publicar documentos en línea con encabezamientos, texto, tablas, imágenes...
- Rellenar formularios con información y enviarlos a un servidor que la procesará y realizará cierta tarea con ella.

Como se ha dicho anteriormente, la característica principal de HTML es la utilización de pares de etiquetas de la siguiente forma: <Inicio-Etiqueta></Fin-Etiqueta>. Con el fin de estructurar el texto.

Estas etiquetas indican no sólo la estructura del texto, si no las imágenes, tablas, listas o demás elementos a representar en el navegador.

### **Componentes del lenguaje HTML.**

Un documento HTML 4.0 está formado básicamente por tres partes:

1. Una línea que contiene la declaración de versión de HTML, indicando al navegador como debe interpretar el código de la página.
2. La sección cabecera definida por <head></head>. Suele contener información sobre el documento que no se muestra directamente al usuario.
3. El cuerpo o body delimitado por las etiquetas <body></body>. Define el contenido principal del documento.

### **2.2.2. CSS**

CSS se usa para dar estilo a los documentos HTML y XML (como ya se mencionó los elementos y atributos que especificaban estilo en los documentos desaparecían para XHTML), concretamente para asociar información de estilo a un elemento. Con su utilización se pretende separar el contenido de los documentos de su presentación, permitiendo a los desarrolladores Web elegir el formato de múltiples páginas Web al mismo tiempo.

Para declarar hojas de estilo en un documento se introducen ciertas líneas en la cabecera del documento:

## 2.2. TECNOLOGÍAS DE DESARROLLO WEB

---

```
<head>
<meta http-equiv="Content-Style-Type"
content="text/css" />
</head>
```

Las hojas de estilo pueden ser incluidas dentro del mismo documento:

```
<head>
...
<style type="text/css">
h1 {border-width: 1; border: solid;
text-align: center}
</style>
</head>
```

En un documento externo (a través del elemento link):

```
<head>
...
<link ref="documento.css" rel="stylesheet"
type="text/css" />
</head>
```

O directamente en un elemento (a través del atributo style):

```
<p style="font-size: 12pt; color: blue">
Ejemplo de estilo
</p>
```

CSS funciona mediante reglas que son declaraciones de estilo sobre uno o más elementos. Las hojas de estilo se componen de una o varias de estas reglas aplicadas sobre un documento HTML o XML. Las reglas se forman con el par «propiedad:valor;».

### 2.2.3. Sesiones

En general una Web se compone de una serie de páginas que entre ellas guardan una cierta relación. Como por ejemplo las páginas que requieren de login y password para poder acceder a otras secciones de la aplicación.

Es necesario, por lo tanto, guardar información de usuario para utilizarla en otras partes de la aplicación.

El uso de estas sesiones cobra importancia con la aparición de Servlets y JSP. Mientras que en los objetos «request» y «response» puede ser colocada información para ser enviada entre un JSP y un Servlet, teniendo en cuenta que una vez terminada la solicitud la información se pierde. Con el objeto «session» se mantiene la información hasta que se cierra la sesión.

Las sesiones comienzan cuando el usuario entra en la aplicación y finalizan cuando el usuario se desconecta de la aplicación o sale de la misma.

Durante el tiempo que dura una sesión podemos almacenar en ésta una serie de variables que nos permiten mantener información común entre esas páginas. [3]

### 2.2.4. JSP

JSP (JavaServer Pages) se trata de una tecnología que permite generar contenido dinámico para Web en programación Java generando documentos HTML.

Fue la compañía Sun Microsystems quien generó esta tecnología, actualmente se encuentra en la versión JSP 2.1 aprovechando la especificación de Servlets 2.5.

Con JSP se pueden crear aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP se componen de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. [13]

#### **Funcionamiento**

JSP puede considerarse como una manera alternativa de construir servlets. El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del servlet a generar. Este servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario.

Gracias a la tecnología JSP se pueden separar en niveles las aplicaciones Web, dejando la parte encargada de generar el documento HTML en el archivo JSP.

Otra ventaja es que JSP hereda la portabilidad de Java, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios.

Un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet. La principal diferencia entre los servlets y los JSPt's es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa Java puro que recibe peticiones y genera a partir de ellas una página Web.

A continuación se explica la sintaxis:

- **Variables Implícitas** Variables sin necesidad de ser declaradas.
- **Directivas** Se tratan de etiquetas que configuran como se ejecutará la página JSP.

```
<%@ directiva atributo="valor" %>
```

- **Scriptlets** Permiten declarar variables, funciones y expresiones en Java.

```
<%! int contador = 0; %>
```

```
<% ... código Java ... %>
```

```
<%= contador + 1%>
```

[12]

### 2.2.5. Servlets

Un servlet es un programa que se ejecuta en un servidor y su uso más común es generar páginas Web de forma dinámica a partir de los parámetros de la petición que envíe el navegador Web.

A diferencia de JSP, el servlet es un componente escrito puramente en Java.

#### Funcionamiento

El método obligatorio que debe implementar un servlet es el método

```
service()
```

que procesa una petición devolviendo el resultado al cliente.

Las clases principales que utiliza un servlet son dos:

1. **Clase `HttpServletRequest`:** Representa la comunicación desde el cliente hacia el servidor.
2. **Clase `HttpServletResponse`:** Representa la comunicación desde el servidor hacia el cliente.

[11][10]

### 2.2.6. JavaScript

JavaScript es un lenguaje de programación interpretado (no requiere compilación) utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas Web.

Un ejemplo de la sintaxis:

```
<script type="text/javascript" src="[URL]"></script>
```

Con JavaScript se pueden extender las posibilidades de las páginas Web como, por ejemplo, evitar que se pueda copiar el texto de una página, botones para agregar automáticamente una página a favoritos, crear barras de scroll, abrir popups, cambiar el puntero del mouse, rotar banners, etc.

### 2.2.7. JavaMail

JavaMail es una expansión de Java que facilita el envío y recepción de e-mail desde código java. JavaMail implementa el protocolo SMTP (Simple Mail Transfer Protocol) así como los distintos tipos de conexión con servidores de correo.

Para su utilización es necesario hacer uso del API de JavaMail que proporciona una plataforma y protocolo independiente para la creación de aplicaciones de mensajería y correo. [9]

- SMTP Simple Mail Transfer Protocol (SMTP) es un protocolo Simple de Transferencia de Correo de la capa de aplicación. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre dispositivos. Está definido en el RFC 2821.

### **Funcionamiento**

SMTP se basa en el modelo cliente-servidor, donde un cliente envía un mensaje a uno o varios receptores. La comunicación entre el cliente y el servidor consiste enteramente en líneas de texto compuestas por caracteres ASCII. El tamaño máximo permitido para estas líneas es de 1000 caracteres.

Las respuestas del servidor constan de un código numérico de tres dígitos, seguido de un texto explicativo.

SMTP va por encima del TCP, usando normalmente el puerto 25 en el servidor para establecer la conexión.

### **Protocolo**

las órdenes básicas del SMTP son:

- **HELO:** Para abrir una sesión con el servidor.
- **MAIL FROM:** Para indicar quien envía el mensaje.
- **RCPT TO:** Para indicar el destinatario del mensaje.
- **DATA:** Para indicar el comienzo del mensaje, éste finalizará cuando haya una línea únicamente con un punto.
- **QUIT:** Para cerrar la sesión.

### **Formato del Mensaje**

El mensaje está compuesto por dos partes:

- **Cabecera:** Definen los campos del mensaje, las más típicas son: subject (asunto), from (emisor) y to (receptor).
- **Cuerpo del mensaje:** es el mensaje propiamente dicho.

## **2.3. Tecnologías sobre la BBDD**

### **2.3.1. SGBD: SQLServer**

#### **SGBD**

Los sistemas de gestión de base de datos (SGBD) son un tipo de software dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones

que la utilizan.

### ■ **Objetivos.**

Los objetivos que debe cumplir un SGBD son:

- **Abstracción de la información.** Los SGBD no aportan a los usuarios detalles acerca del almacenamiento físico de los datos. De esta manera, se definen niveles de abstracción.
- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Consistencia.** Un SGBD consistente es aquel en el que se cumplen todas las restricciones especificadas en el esquema de la BBDD.
- **Seguridad.** La información almacenada en una BBDD debe estar bien asegurada únicamente accediendo usuarios con permisos específicos.
- **Manejo de Transacciones.** Las transacciones son programas que se ejecutan como una sola operación, si no se pudiese llevar a cabo esa operación el SGBD debe deshacer la operación y asegurar que la BBDD se quede en el mismo estado en el que se encontraba inicialmente.  
La transacción ha sido realizada completamente cuando se confirma, en caso contrario, no se ha llevado a cabo la operación.
- **Tiempo de Respuesta.** Un SGBD debe asegurar un tiempo de respuesta pequeño en darnos la información solicitada y en almacenar los cambios realizados.

### ■ **Ventajas.**

- Facilidad para operar con grandes volúmenes de datos.
- Simplificación de la programación de equipos de consistencia.
- Organización de los datos con un impacto mínimo en el código del programa.
- Disminuyen los tiempos de desarrollo y aumenta la calidad del sistema desarrollado.
- Proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

### ■ **Inconvenientes.**

- Complejidad.

El software es muy complejo y los usuarios deben tener conocimiento de las funcionalidades del mismo para poder aprovecharlo al máximo.

- Memoria.

La complejidad y la gran cantidad de funciones que tienen hacen que sea un software de gran tamaño, por lo que se requiere de gran cantidad de memoria para poder funcionar correctamente.

- Coste del hardware adicional.

Los requisitos de hardware de un SGBD son relativamente altos, todo lo relacionado con estos equipos supone un gran coste.

[17]

### SQLServer

SQL Server es un Sistema de gestión de Bases de Datos Relacionales (SGBD) basado en el lenguaje Transact-SQL, es capaz de poner a disposición del usuario gran cantidad de datos de manera simultánea.

Microsoft SQL Server es la alternativa de Microsoft al resto de potentes sistemas gestores de bases de datos como Oracle, PostgreSQL, MySQL, etc.

- **Características.**

- *Soporte de Transacciones*

Un SGBD es transaccional si es capaz de mantener la integridad de los datos evitando así que puedan finalizar en un estado intermedio. Cuando por alguna causa el sistema debe cancelar la transacción, se empiezan a deshacer las órdenes ejecutadas hasta dejar la BBDD en su estado inicial, como si la orden de la transacción nunca se hubiese realizado.

- *Escalabilidad*

SQLServer ofrece la propiedad de escalabilidad, que indica la capacidad de manejar el crecimiento continuo de trabajo de manera fluida y estar preparado para cambiar su tamaño sin perder calidad en los servicios ofrecidos.

- *Estabilidad*

SQLServer es una SGBD estable, reaccionando positivamente a los cambios realizados.

- ***Seguridad***

- ***Soporta procedimientos almacenados.***

Un procedimiento almacenado es un programa almacenado físicamente en una BBDD. La ventaja del procedimiento almacenado es que al ser ejecutado, en respuesta a una petición del usuario, es ejecutado directamente en el motor de la BBDD, el cual corre en un servidor separado.

Posee acceso directo a los datos que necesita manipular, y sólo necesita enviar sus resultados de regreso al usuario, deshaciéndose de la sobrecarga resultante de comunicar grandes cantidades de datos salientes y entrantes.

Algunos usos típicos de procedimientos almacenados incluyen la validación de datos integrados a la estructura de base de datos o encapsular un proceso grande y complejo.

Los procedimientos pueden ser ventajosos: Cuando una base de datos es manipulada desde muchos programas externos. Al incluir la lógica de la aplicación en la base de datos utilizando procedimientos almacenados, la necesidad de embeber la misma lógica en todos los programas que acceden a los datos es reducida.

- ***Entorno gráfico.***

Incluye un potente entorno gráfico para la administración que permite el uso de comandos DDL y DML gráficamente.

- ***Cliente/Servidor.***

Permite trabajar en modo cliente-servidor, donde la información de los datos están en el servidor y los clientes sólo acceden a la información.

- ***Administración de información de otros servidores.***

Se permite administrar información de otros servidores, creando así un sistema centralizado, donde puedo consultar información de los servidores configurados para este fin.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos Windows.

[15]



Figura 2.1: Logotipo de Apache Software Foundation

### 2.3.2. Apache

#### ■ Descripción.

El servidor HTTP Apache es un servidor Web HTTP de código abierto para plataformas UNIX, Windows, etc. Implementa el protocolo HTTP/1.1.

El servidor Apache se desarrolla dentro del proyecto HTTP Server de la Apache Software Foundation, 2.1.[4]

Apache presenta, entre otras características bases de datos con autenticación y negociado de contenidos, todas ellas configurables por el usuario. Se criticó la falta de un interfaz gráfica de usuario que ayudase en la configuración del Apache. Sin embargo, se puede decir que Apache es una servidor Web flexible, rápido y eficiente. Continuamente actualizado y adaptado a los nuevos protocolos HTTP/1.1. [5]

#### ■ Uso.

Apache es usado principalmente para enviar páginas web estáticas y dinámicas a través de la red. Muchas aplicaciones web están diseñadas para la utilización de Apache, utilizando características propias del servidor.

Apache es usado para otro tipo de tareas, donde el contenido necesita ser puesto a disposición de una forma segura y confiable. El usuario puede colocar archivos en la raíz de documentos de Apache desde donde pueden ser compartidos.

#### ■ Aceptación en el mercado.

Apache es uno de los servidores que más aceptación tiene en la red, desde 1996 HTTP Apache es el servidor más usado.

Su máxima cuota de mercado la alcanzó en 2005 siendo el servidor empleado en el 70 por ciento de los sitios Web en el mundo.

Actualmente ha sufrido un descenso en su cuota de mercado en los últimos años.

### ■ **Características.**

Entre sus características destacan: [1]

#### - ***Multiplataforma.***

El hecho de que Apache sea multiplataforma significa que funciona en la mayoría de las plataformas actuales, gracias a esta característica se da libertad al programador para que elija la plataforma que más se adapte a sus necesidades.

Por lo tanto, existe una independencia del fabricante del software con el del hardware, lo que hace que el fabricante del hardware esté en constante evolución para ofrecer nuevos productos de calidad a sus usuarios. En caso de disconformidad por parte de los clientes, siempre pueden elegir otra plataforma.

#### - ***Servidor web adaptado al protocolo HTTP/1.1***

El protocolo HTTP consiste en que el cliente establece una conexión, utilizando TCP, con el servidor. Se genera una petición, el servidor responde y se cierra la conexión.

La nueva versión HTTP/1.1 recogida en la RFC 2068 de Enero de 1997 nos explica las principales características que hacen que Apache esté adaptado a este protocolo:

- Conexiones persistentes. Evitando la sobrecarga, ya que no cierra la conexión.
- Varias peticiones simultáneas por parte del cliente, sin esperar a que el servidor conteste de una en una.
- Negociación del contenido que varía según la calidad de la conexión.
- Nuevo método de autenticación, que permite que la contraseña vaya encriptada por la red. [19]

#### - ***Modular.***

Presenta una arquitectura en módulos (explicada posteriormente).

#### - ***Extensible.***

Esta característica nos indica que Apache se puede extender, haciendo un uso mucho mayor del nombrado anteriormente, lo que nos proporciona una solución ideal para páginas de carga media/alta. Apache soporta Dynamic Shared Object (DSO). Gracias a ello se pueden construir módulos que le den nuevas funcionalidades que son cargadas en tiempos de ejecución.

### - *Código abierto.*

El hecho de que Apache sea un software de código abierto tiene la ventaja de que el código fuente esté disponible haciendo posible que usuarios, programadores, y empresas puedan involucrarse en el desarrollo de las aplicaciones y de esta forma el proceso de corrección y detección de errores se lleve a cabo de forma eficiente.

Con el software de código abierto no existe un gasto en licencias si no una inversión en la capacitación del programador.

[16]

### - *Autenticación de diferentes tipos.*

Apache permite la autenticación de usuarios en varias formas. Apache permite el uso de bases de datos DBM para la autenticación de usuarios. De esta forma, se puede restringir el acceso a determinadas páginas de un sitio Web de una forma sencilla y de fácil mantenimiento.

### - *Respuestas personalizadas ante errores del servidor.*

Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.

### - *Creación de contenidos dinámicos.*

Apache permite la creación de sitios web dinámicos mediante:

1. El uso de CGI's.
2. El uso de Server Side Includes (SSI).
3. El uso de lenguajes de Scripting como PHP, javascript, Python.
4. El uso de Java y páginas jsp.

### - *Negociación de contenido*

Apache puede facilitar información en varios formatos para que un determinado cliente pueda interpretarla.

## ■ **Descripción de la Arquitectura en Módulos.**

El servidor Apache es un software estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo (en los ficheros de configuración).

Los módulos de Apache se pueden clasificar en tres categorías:

1. **Módulos Base.** El módulo base es el que contiene las funcionalidades básicas del Apache.

2. **Módulos multiproceso.** Estos módulos se encargan de la unión con los puertos de la máquina, aceptando peticiones, y enviando a los hijos atender a peticiones.
3. **Módulos Adicionales.** Son los módulos que se le añaden al servidor para aportar funcionalidad adicional.

Como explicación más concisa se puede decir que las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Existen varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando le rendimiento y la rapidez del código.

El resto de funcionalidades del servidor se consiguen a través de módulos adicionales que se cargan directamente sin necesidad de volver a instalar el software. [6]

### ■ Direcciones IP y Puertos de Escucha

Cuando Apache se inicia, comienza a esperar peticiones entrantes en determinados puertos y direcciones de la máquina en la que se está ejecutando. Sin embargo, si quiere que Apache escuche solamente en determinados puertos específicos, o solamente en determinadas direcciones, o en una combinación de ambos, debe especificarlo adecuadamente. Esto puede además combinarlo con la posibilidad de usar hosts virtuales, funcionalidad con la que un servidor Apache puede responder a peticiones en diferentes direcciones IP, diferentes nombres de hosts y diferentes puertos. [7]

### ■ Mas información.

Consultar [8].

## 2.4. LDAP

LDAP es el acrónimo de Lightweight Directory Access Protocol (Protocolo Ligero de Acceso a Directorios). Inicialmente LDAP fue pensado como un protocolo para proveer acceso remoto a servicios de directorio, pero actualmente se considera un servicio de directorio.

LDAP es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para realizar búsquedas de información en un entorno de red.

Un directorio es un conjunto de objetos ordenados jerárquicamente. LDAP está formado por un árbol de objetos que contienen a otros objetos, construyendo así una jerarquía, y pudiéndose mover por todo el árbol en la búsqueda de información.

### ■ **Visión general del Protocolo.**

Un usuario inicia una sesión LDAP conectándose a un servidor LDAP, para posteriormente enviarle una serie de peticiones al servidor (puede enviar más de una) y el servidor responde a esas peticiones en el cualquier orden.

Las operaciones más usuales del usuario son:

- Bind: Sirve para autenticarse y especificar una versión del protocolo LDAP.
- Search: Buscar y/o obtener entradas de directorio.
- Add: Añadir una nueva entrada.
- Delete: Borrar una entrada.
- Modify: Modificar una entrada.
- Modify Distinguished Name (DN): Modificar o renombrar una entrada.
- Unbind: Cerrar la conexión

### ■ **Estructura del Directorio.**

El protocolo LDAP accede a directorios que siguen el modelo X.500.

LDAP es un directorio que es un árbol de entradas de directorio, cada entrada está formada por un conjunto de atributos, estos atributos tienen un nombre y uno o más valores.

Cada entrada contiene información de una cierta entidad que se denomina registro. Cada registro está compuesto por lo que se denomina DN (Distinguished Name), que actúa como identificador único del registro, y por uno o más atributos que proporcionan información sobre la entidad que está siendo descrita. Todo esto se puede observar en la figura 2.2

A través de la estructura jerárquica se definen relaciones padre-hijo entre los registros del directorio. El DN de un registro se puede descomponer en dos partes:

1. DN relativo (relative DN o RDN). Consiste en un atributo específico del registro que permite diferenciarlo de otros posibles registros con el mismo registro padre.

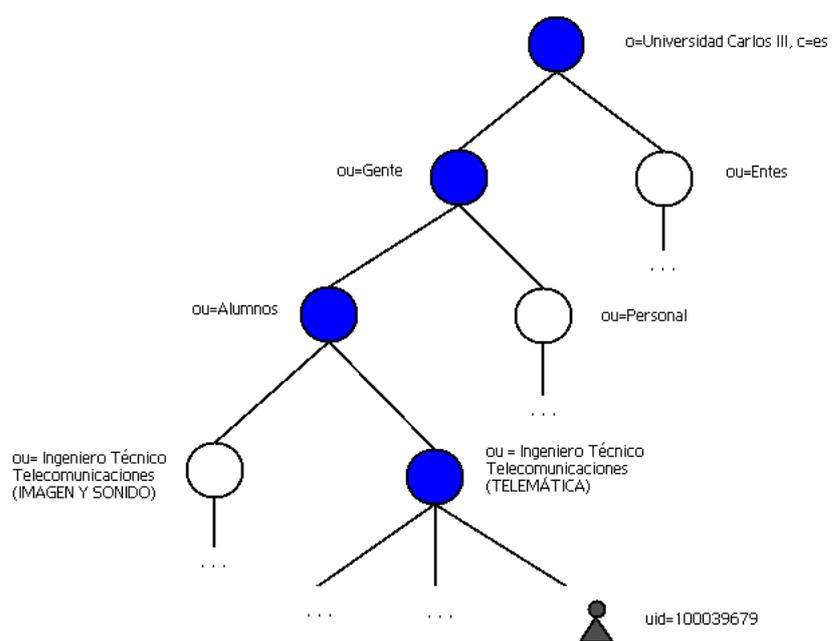


Figura 2.2: Árbol LDAP de la UC3M

2. DN Absoluto. El que representa una parte del directorio LDAP, conociendo el DN Absoluto del padre conocemos la posición del nodo en el árbol.

- **Aplicaciones típicas de LDAP.**

- Almacenar información de una organización. Como ejemplo se puede poner el servidor LDAP de la Universidad que almacena información de sus miembros estructurándola jerárquicamente en personal, alumnos, etc.
- Se puede utilizar el servicio como autoridad central en una red, conteniendo información sobre usuarios y grupos. Un ejemplo de uso es la validación que ofrece la UC3M para entrar a aula global.
- Otro posible uso del servidor LDAP es el almacenamiento de información como si se tratase de una Base de Datos Relacional, la diferencia reside en que LDAP trabaja con estructura jerárquica (relaciones padre-hijo), mientras que una base de datos relacional trabaja con tablas y permite definir relaciones entre ellas (no necesariamente relaciones padre-hijo). El servicio de directorio está optimizado para atender eficientemente consultas de lectura, mientras que una base de datos relacional está preparada para gestionar de forma eficiente no sólo las lecturas, sino también las modificaciones en las tablas.

Teniendo en cuenta lo anterior, el administrador debe estudiar la información a almacenar (si es susceptible de cambios o se considera estática) a la hora de elegir entre un servicio LDAP o una base de datos relacional.

En la figura 2.3 se muestra una tabla comparativa presentando las características de LDAP frente a otras tecnologías.

- **Referencias [14] [2]**

## CAPÍTULO 2. TECNOLOGÍAS USADAS

---

	<b>DNS</b>	<b>NIS</b>	<b>NIS+</b>	<b>FNS</b>	<b>LDAP</b>
NAMESPACE	Hierarchical	Flat	Hierarchical	Hierarchical	Hierarchical
DATA STORAGE	Files/ resource records	2 column maps	Multi columned tables	Maps	Directories [varied] Indexed database
SERVERS	Master/slave	Master /slave	Root master/ non-root master; primary/ secondary; cache/stub	N/A	Master/replica Multi master replica
SECURITY	none	None (root or nothing)	DES Authentication	None (root or nothing)	SSL, varied
TRANSPORT	TCP/IP	RPC	RPC	RPC	TCP/IP
SCALE	Global	LAN	LAN	Global (with DNS)/LAN	Global

Figura 2.3: Tabla comparativa.

# Requisitos

---

En la primera parte de este capítulo se definen detalladamente los requisitos previos necesarios para cumplir las expectativas del proyecto y a continuación los requisitos finales en vista de las ampliaciones de la aplicación.

## 3.1. Requisitos Iniciales

### 3.1.1. R1. Acceso

Para acceder a la aplicación se necesitará un usuario y contraseña de modo que sólo los usuarios autenticados puedan acceder a la misma. Esta validación se realizará a través del Servidor LDAP de la UC3M.

### 3.1.2. R2. Permisos

Existirán tres tipos de usuarios: Usuario normal (sin permisos), Delegado y Administrador.

Cada uno de estos usuarios contará con unos permisos diferentes ya que tienen funcionalidades distintas.

### 3.1.3. R3. Perfil Usuario sin permisos.

Corresponderá con todo usuario de la UC3M que no esté dado de alta en un equipo como delegado. El perfil usuario tendrá únicamente permisos de consulta sobre la BBDD, por lo tanto, la funcionalidad de la aplicación para este perfil corresponde con la consulta de:

- Resultados
- Clasificación
- Goleadores/Anotadores
- Deportividad

### **3.1.4. R4. Perfil Delegado.**

Corresponde con todos aquellos usuarios que estén dados de alta como delegados en algún equipo de la UC3M.

El perfil Delegado dispone de privilegios de modificación sobre la BBDD.

La funcionalidad principal que debe realizar es la de Insertar Resultados de los partidos ya disputados.

### **3.1.5. R5. Perfil Administrador.**

El perfil administrador corresponderá, en este caso, al Departamento de Actividades Deportivas y Culturales de la UC3M (Espacio de Estudiantes).

La funcionalidad principal del administrador es la de Confirmar los resultados insertados previamente por el delegado, así como el correcto funcionamiento de la aplicación.

## **3.2. Requisitos Finales**

A medida que se avanzaba con el proyecto en las distintas sesiones con los tutores, surgieron nuevas necesidades. Después de un estudio de las mismas, se añadieron nuevos requisitos.

Los requisitos finales corresponden con las ampliaciones que se definen a continuación:

### **3.2.1. Ampliaciones relacionadas con el Perfil Delegado.**

Surge la necesidad, desde el punto de vista de los usuarios y del departamento, de realizar funcionalidades relacionadas con el aplazamiento de un partido.

El objetivo fundamental era que los usuarios pudiesen consultar, vía Web, la disponibilidad de las pistas. Para así, poder solicitar el aplazamiento de un partido, ya habiendo consultado el día que propondrían para jugarlo.

Con este fin, se plantea la opción de consulta de horarios y pistas, así como un listado con la información de los delegados de tu competición (para ponerse de acuerdo en el día a solicitar el aplazamiento), y un formulario que deberá cumplimentar el delegado para solicitar ese aplazamiento.

Por lo tanto, los requisitos nuevos se pueden definir como:

- R6. Funcionalidad para consultar las Fechas y Horarios Disponibles para aplazamientos.
- R7. Información de contacto sobre los datos delegados de tu competición.
- R8. Funcionalidad de Aplazar un partido.

## CAPÍTULO 3. REQUISITOS

---

## Diseño de la Interfaz. Perfiles

---

### 4.1. Introducción.

Cuando un usuario nuevo se dispone a utilizar una aplicación por primera vez, tiene ciertas expectativas de como va a funcionar el programa.

Inicialmente se va a realizar un estudio sobre las necesidades que tiene un usuario a la hora de enfrentarse a una nueva aplicación, para posteriormente presentar la aplicación que se ha realizado, intentando cumplir con esas necesidades.

Existen principios relevantes para el diseño e implementación de la Interfaz de Usuario:

- **Anticipación:** La aplicación deberá intentar anticiparse a las necesidades del usuario y no esperar a que el usuario tenga que buscar la información o recopilarla.
- **Autonomía:** La aplicación debe estar a disposición del usuario en cualquier momento temporal y espacial.
- **Sencillez y Claridad:** La aplicación debe ser amigable, sencilla y clara de manejar, de esta manera, el usuario se sentirá cómodo con ella y el aprendizaje sobre la misma se hará con mayor rapidez.
- **Consistencia:** Para lograr una mayor consistencia en la interfaz de usuario y aplicación se va a profundizar en diferentes aspectos ordenados por niveles de mayor a menor:

1. Interpretación del comportamiento del usuario: La Interfaz de Usuario debe entender que quiere hacer en cada momento el usuario.

2. Estructuras visibles: Conjunto de objetos visibles (iconos, botones, ...) que permitan al usuario controlarlos y ahorrar tiempo en ejecutar tareas específicas.
  3. Componente único: Se muestran en un único componente todas las posibles acciones a realizar por el usuario.
  4. Consistencia en el ambiente: La aplicación o interfaz del usuario debe mantenerse en concordancia con el ámbito de trabajo de los usuarios. De esta manera, se parecerá en el manejo a otras aplicaciones que está acostumbrado a utilizar el usuario, y será más fácil entenderla.
  5. Consistencia con la plataforma: La interfaz de usuario debe ser concordante con la plataforma utilizada.
- **Eficiencia de la aplicación:** Se debe tener en cuenta antes la efectividad del usuario que la efectividad de la aplicación. Si el usuario debe esperar la respuesta del sistema por un período de tiempo prolongado, estas pérdidas de tiempo pueden llegar a ser pérdidas económicas. Por lo tanto, las respuestas a las peticiones realizadas por el usuario deben ser contestadas en un período temporal corto para que la aplicación sea eficaz.
  - **Interfaces explorables:** Siempre que sea posible se debe permitir al usuario introducirse en otras opciones de la aplicación o salir de la misma. Es por ello que la interfaz de usuario debe tener un objeto fácil de accionar con el cual poder finalizar la aplicación. Se apoya así al usuario a explorar el terreno sin temores.
  - **Curva de Aprendizaje:** Se pretende que la curva de aprendizaje sea nula para el usuario, lo que significaría que ha alcanzado un dominio total de la aplicación sin esfuerzo.
  - **Estética y Diseño minimalista:** Se debe presuponer que el usuario no puede/quiere tener acceso a unos manuales de uso, la aplicación debe ser sencilla, clara y concisa, con una estética básica y un diseño minimalista que no de problema de comprensión.

En este caso, la aplicación se compone de un sistema interactivo con el usuario. La forma de presentar la información a los usuarios es algo que debe tenerse en cuenta en todo momento ya que determina en gran medida la percepción que el usuario posee de la aplicación. (Como se ha comentado en los pasos previos).

Se ha enfocado desde el punto de vista de un usuario que no conoce la tecnología utilizada y que quiere usar la aplicación con normalidad y sin enfrentarse a ningún problema.

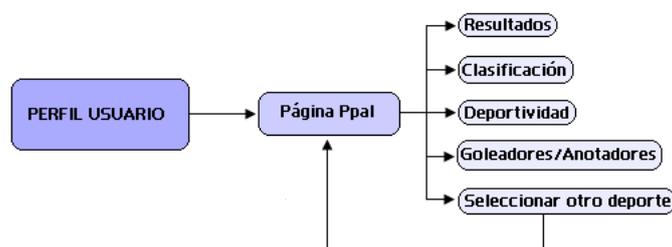


Figura 4.1: Diagrama UML. Perfil Usuario

Se ha creado una interfaz sencilla que ayuda a progresar para conseguir el objetivo final como es la consulta de la información relacionada con las actividades deportivas.

La aplicación desarrollada se encuentra orientada a tres tipos de perfiles:

1. **Perfil Usuario:** Estos usuarios corresponden a todo usuario de la Universidad Carlos III excepto los delegados de los equipos. Entre este grupo de usuarios se encuentran los alumnos inscritos en alguna actividad deportiva, los no inscritos, así como personal docente, personal de departamentos de investigación, etc.
2. **Perfil Delegado:** Estos usuarios corresponden con los delegados de cada equipo dado de alta en las actividades deportivas.
3. **Perfil Administrador:** Corresponde con uno o varios usuarios del Departamento del Espacio de Estudiantes, que se encargarán de gestionar la aplicación y el correcto funcionamiento de la misma.

Por lo tanto, si se hace una clasificación más exhaustiva sobre el tipo de permisos que ofrece la aplicación, van a existir permisos a nivel usuario, permisos a nivel delegado y permisos a nivel administrador. Controlando de esta manera la seguridad de la aplicación frente a las acciones que puede realizar cada uno de ellos.

En las figuras 4.1 4.2 y 4.3 se pueden observar los diagramas UML de cada uno de los perfiles explicados anteriormente, con las distintas funcionalidades de cada uno.

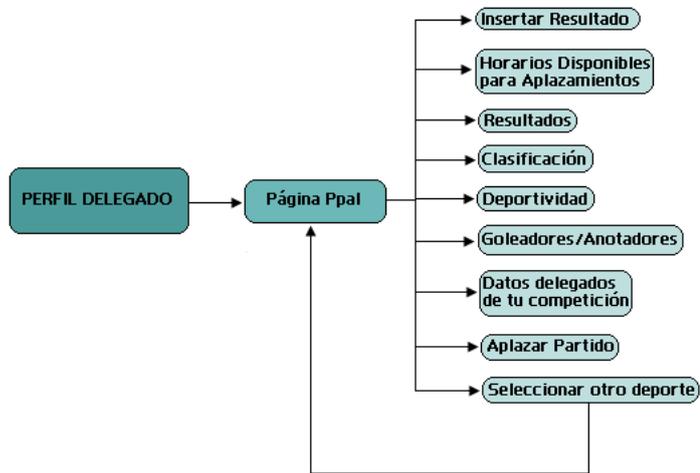


Figura 4.2: Diagrama UML. Perfil Delegado

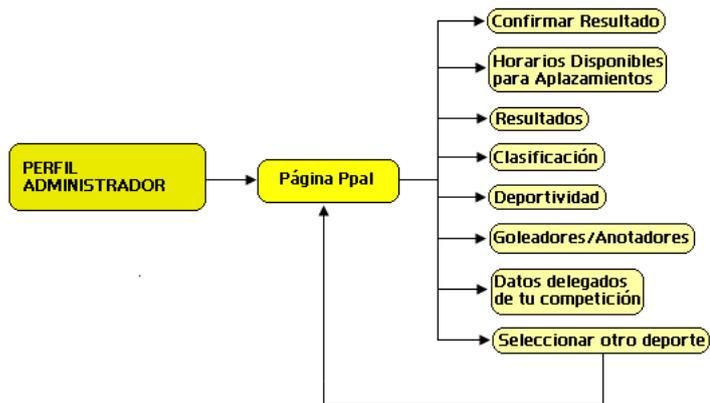


Figura 4.3: Diagrama UML. Perfil Administrador



Figura 4.4: Página de Autenticación

## 4.2. Acceso a la aplicación.

El acceso a la aplicación debe ser restringido, únicamente tendrá acceso el personal de la Universidad Carlos III ya que se utiliza como validación el servidor LDAP de la UC3M.

Dentro de esta primera restricción aparece una segunda restricción relacionada con el tipo de usuario que se esté validando. Como se ha comentado anteriormente, se van a presentar 3 perfiles, y cada uno con permisos diferentes.

Lo primero que se le ofrece al usuario de la aplicación es la página de validación, que podemos observar en la figura 4.4.

Consta de dos campos de texto para introducir el login y la clave respectivamente, con esta información comprobaremos si la validación es o no exitosa (si el usuario pertenece a la Universidad o no). En caso afirmativo, se verificará el tipo de perfil que se encuentra asociado al usuario autenticado.

La siguiente página que se le mostrará al usuario contendrá las opciones que puede realizar en la aplicación, según el tipo de perfil con el que se haya autenticado.

## 4.3. Perfil Usuario.

Una vez que el usuario se haya autenticado y se le asocie con el perfil usuario normal (todo personal de la UC3M que no esté dado de alta como delegado de un equipo), se le ofrece la interfaz de la figura 4.5

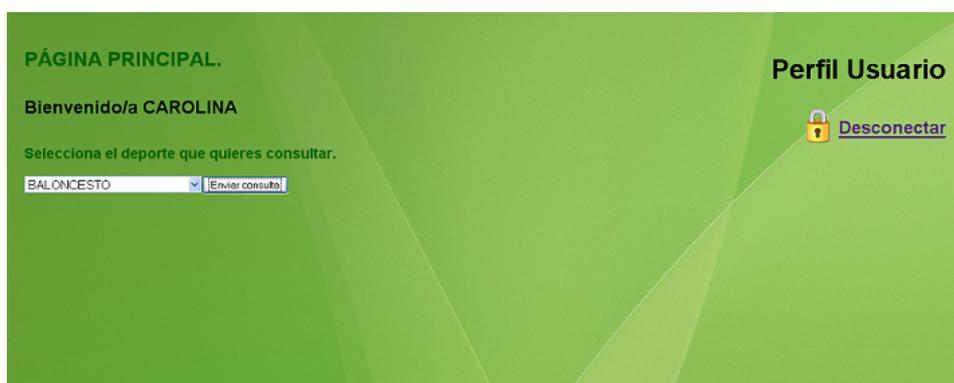


Figura 4.5: Página Principal Usuario

En esta interfaz aparece:

- El título de la misma, para que el usuario sepa en todo momento en que parte de la aplicación se encuentra.
- El perfil con el que se ha conectado.
- La bienvenida, obteniendo el nombre del usuario a través del servidor LDAP de la UC3M, una vez que se ha realizado la validación.
- Un desplegable, con los distintos deportes de LIGA que el usuario puede consultar.
- Desconexión, permite al usuario salir de la aplicación.

Una vez que el usuario haya seleccionado un deporte, se muestran las opciones a realizar sobre el mismo, como se observa en la Figura 4.6.

Las opciones que se ofrecen son las siguientes:

1. Resultados.
2. Clasificación.
3. Deportividad.
4. Goleadores y Anotadores.
5. Seleccionar otro deporte.

Todas estas opciones son únicamente de consulta, ya que el perfil de Usuario no tiene permisos para modificar la información de las actividades deportivas.

A continuación se detalla cada una de estas opciones.



Figura 4.6: Opciones del Usuario.

### 4.3.1. Resultados.

En este apartado se mostrarán los resultados de los partidos disputados de todos los grupos que forman el deporte seleccionado. Una vez seleccionado el deporte, se muestra al usuario no sólo el título de la parte de la aplicación donde se encuentra, si no también el deporte con el que se está tratando.

Se puede observar en esta página, que a la izquierda de la misma, aparece el menú con las opciones que puede realizar el usuario.

Este menú se incluye en toda la aplicación. El objetivo es que el usuario tenga mayor accesibilidad a la funcionalidad que puede realizar. De esta manera, es más intuitivo el manejo de la aplicación dándole mayor facilidad al usuario sobre su uso.

A continuación, se le da al usuario la opción de realizar dos acciones:

1. Seleccionar el grupo que quiere consultar, dentro del deporte en el que se encuentra. En el desplegable creado para este fin aparecen todos los grupos inscritos a ese deporte, de los tres campus que tiene la Universidad Carlos III (Colmenarejo, Getafe y Leganés).
2. Seleccionar la jornada de la que desea consultar el Resultado. De esta forma, se facilita al usuario de la aplicación acceder a una información más concreta.

Una vez realizadas estas acciones, se realiza la consulta a la BBDD y se muestra la información al usuario (Figura 4.7).

Los campos de esta información son:

**RESULTADOS. BALONCESTO**

Bienvenido/a CAROLINA

[Resultados](#)  
[Clasificación](#)  
[Deportividad](#)  
[Goleadores y Anotadores](#)  
[Seleccionar otro deporte](#)

Grupo: BALONCESTO GETAFE 08/09 G1

Jornada: 1 [Enviar consulta](#)

Jornada	Fecha	Hora	Equipo Local	Puntos Local	Equipo Visitante	Puntos Visitante
1	2008-10-28	13:00	L'Squadra Cappezzolo	20	Loitas al Sol	0
1	2008-10-28	14:00	You'll never Drink alone	50	TEAM ROCKET	37
1	2008-10-28	15:00	SOUL TEAM	99	BCB	29
1	2008-10-29	14:00	L.M.M.	22	Pájidos C.B.	19
1	2008-10-29	15:00	Toronto Nábos	21	TIRADORES	12
1	2008-10-31	13:00	CM-RE Fernando de los Rios (fem)	0	FERNANDO DE LOS RIOS A	0

Perfil Usuario [Desconectar](#)

Figura 4.7: Resultados

- Jornada: Campo que especifica la jornada que se está consultando.
- Fecha: Campo que indica la fecha en la que se jugó el partido.
- Hora: Campo que indica la hora en la que se jugó el partido.
- Equipo Local: Nombre del equipo local que jugó el partido, en la fecha y hora señaladas anteriormente.
- Puntos Local: Puntuación del equipo local.
- Equipo Visitante: Nombre del equipo visitante que jugó el partido, en la fecha y hora señaladas anteriormente.
- Puntos Visitante: Puntuación del equipo visitante.

Las puntuaciones vienen dadas por goles (si el deporte seleccionado fue FUTBOL SALA o FUTBOL A7), puntos correspondientes a las canastas encestandas (si el deporte seleccionado fue BALONCESTO) ó sets (si el deporte seleccionado fue TENIS, VOLEIBOL, SQUASH, BADMINTON o PADEL).

### 4.3.2. Clasificación.

En este apartado se mostrará, por jornadas, la clasificación de un grupo concreto dentro del deporte seleccionado por el usuario.

Las acciones que puede realizar el usuario son las mismas que en el apartado anterior. Por lo tanto, una vez seleccionado el grupo y la jornada sobre los cuales se desea realizar la consulta se muestra la información al usuario.

Como podemos ver en la Figura 4.8, los campos que aparecen son:

- Equipo: Nombre del equipo (perteneciente al grupo seleccionado).
- Jornada: Campo que especifica el número de jornada que se está consultando.
- Posición: Campo que especifica la posición en la que está el equipo según la puntuación obtenida en esa jornada.
- Partidos Jugados: Número de partidos que ha jugado ese equipo.
- Partidos Ganados: Número de partidos que ha ganado el equipo.
- Partidos Empatados: Número de partidos que ha empatado ese equipo.
- Partidos Perdidos: Número de partidos que ha perdido ese equipo.
- Tantos a Favor: Campo que especifica los tantos (goles, canastas o sets) marcados por ese equipo.
- Tantos en Contra: Campo que especifica los tantos (goles, canastas o sets) marcados por el resto de equipos al propio equipo.
- Puntos Totales: Puntuación total que se le asigna al equipo según gane, pierda o empate.

Los campos correspondientes a «Partidos Jugados», «Partidos Ganados», «Partidos Empatados», «Partidos Perdidos», «Tantos a favor», «Tantos en contra» y «Puntos Totales» se irán incrementando a medida que aumenten las jornadas, ya que se le suma a los datos de la jornada anterior, los datos de la nueva jornada.

Existe un criterio definido por el departamento del Espacio de Estudiantes de la UC3M para la asignación de los puntos totales, que varía según el deporte y que se encuentra definido en la Tabla 4.1.

En caso de empate se sigue un criterio de desempate. Este criterio consiste en asignarle la primera posición al equipo que haya conseguido más tantos a favor.

En caso de necesitar otro criterio de desempate, contactar con el Departamento de Actividades Deportivas y Culturales.

De la misma manera que en la consulta de los «Resultados», la Clasificación variará según avancen las jornadas.

**CLASIFICACIÓN. BALONCESTO**

Bienvenido/a CAROLINA

[Resultados](#)  
[Clasificación](#)  
[Deportividad](#)  
[Goleadores y Anotadores](#)  
[Seleccionar otro deporte](#)

Grupo: BALONCESTO GETAFE 08/09 G1

Jornada: 1 [Enviar consulta](#)

**Perfil Usuario**  
[Desconectar](#)

Equipo	Jornada	Posicion	Partidos Jugados	Partidos Ganados	Partidos Empatados	Partidos Perdidos	Tantos a Favor	Tantos en Contra	Puntos Totales
SOUL TEAM	1	1	1	1	0	0	99	29	2
You'll never Drink alone	1	2	1	1	0	0	50	37	2
L.M.M.	1	3	1	1	0	0	22	19	2
Toronto Nabos	1	4	1	1	0	0	21	12	2
L Squadra Cappezzolo	1	5	1	1	0	0	20	0	2
TEAM ROCKET	1	6	1	0	0	1	37	50	1
BCB	1	7	1	0	0	1	29	99	1
Pajujos C.B.	1	8	1	0	0	1	19	22	1
TIRADORES	1	9	1	0	0	1	12	21	1
Lolitas al Sol	1	10	1	0	0	1	0	20	1
CM-RE Fernando de los Rios (ferm)	1	11	0	0	0	0	0	0	0
FERNANDO DE LOS RIOS A	1	12	0	0	0	0	0	0	0

Figura 4.8: Clasificación

Deporte	Partido Ganado	Partido Empatado	Partido Perdido
Bádminton	2 puntos	0 puntos	1 punto
Baloncesto	2 puntos	0 puntos	1 punto
Fútbol A7	3 puntos	1 punto	0 puntos
Fútbol Sala	3 puntos	1 punto	0 puntos
Pádel	2 puntos	0 puntos	1 punto
Squash	2 puntos	0 puntos	1 punto
Tenis	2 puntos	0 puntos	1 punto
Voleibol	2 puntos	0 puntos	1 punto

Tabla 4.1: Puntuaciones Deportivas



Figura 4.9: Deportividad

### 4.3.3. Deportividad.

En el apartado de Deportividad se muestra una lista de equipos con una puntuación que representa la deportividad.

El usuario podrá seleccionar un campus y consultar la puntuación del mismo. Como resultado se obtiene un listado con todos los equipos sus puntuaciones.

Esta puntuación se corresponde con la otorgada por el departamento del Espacio de Estudiantes mediante la valoración que el árbitro realiza a cada equipo.

Esa valoración se realiza de la siguiente manera: El árbitro otorga una puntuación del 1 al 10 a cada equipo según la deportividad mostrada en el partido. A esa puntuación se le restarán puntos si el árbitro ha sacado alguna tarjeta amarilla o roja a alguno de los componentes del mismo.

El resultado de esa valoración es la puntuación asignada a cada equipo.

La información se muestra ordenada por puntuación y se puede ver en la figura 4.9.

### 4.3.4. Goleadores y Anotadores.

En esta opción se mostrará, seleccionando un determinado campus, la información correspondiente a todos los equipos, jugadores y puntuación obtenida por los mismos.

**GOLEADORES Y ANOTADORES. BALONCESTO**

Bienvenido/a CAROLINA

Perfil Usuario [Desconectar](#)

Resultados

Clasificación

Deportividad

Goleadores y Anotadores

Seleccionar otro deporte

Campus:

APELLIDOS	NOMBRE	NOMBRE EQUIPO	PUNTOS
GOMEZ POSADA	VICENTE	SOUL TEAM	123
JIMENEZ MARTIN	ALVARO	TEAM ROCKET	117
PERAZA DOMINGUEZ	JUAN MANUEL	SOUL TEAM	95
DE MIGUEL EGIDO	ROBERTO	L' Squadra Cappezzolo	88
INDARTE MARCOS	DANIEL	Toronto Nabos	79
ABEL SESMERO	ALBERTO EUGENIO	TEAM ROCKET	68
OCON RUIZ	MIGUEL	L' Squadra Cappezzolo	67
VIDAL LOPEZ	EDUARDO	You'll never Drink alone	64
SANCHEZ SOCAS	JAVIER	SOUL TEAM	57

Figura 4.10: Goleadores y Anotadores

Esta información aparecerá ordenada por la puntuación obtenida por el jugador. (Figura 4.10).

Los campos de los que se compone esta información son:

- Apellidos: Corresponden con los apellidos del jugador al que se le asigna la puntuación.
- Nombre: Nombre correspondiente al jugador.
- Equipo: Nombre del equipo al que corresponde el jugador.
- Puntuación: Puntos que ha obtenido el jugador en el transcurso de los partidos.

### 4.3.5. Selección de otro deporte.

En este apartado se redirige al usuario a la página principal, con el fin de que pueda seleccionar otro deporte y realizar las mismas acciones, que las explicadas anteriormente.

De esta manera se proporciona al usuario una mayor accesibilidad a todos los deportes y todas las acciones que se pueden realizar en cada uno de ellos.

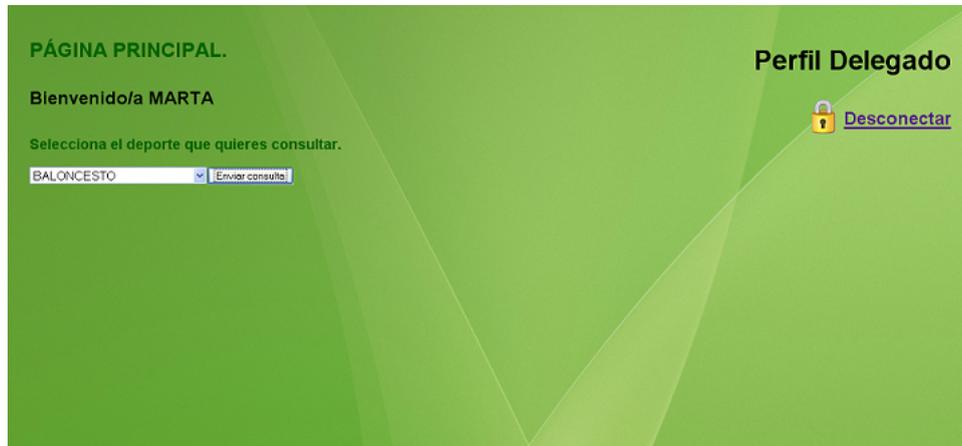


Figura 4.11: Página Principal Delegado

## 4.4. Perfil Delegado.

De la misma manera que en el perfil anterior, una vez que el usuario se autentique, se comprueba con que clase de perfil se está conectando, que en este caso es el «Perfil Delegado».

Un delegado es un miembro del equipo que se va a encargar de representarlo, asumiendo ciertas responsabilidades. Por lo tanto, tendrá más permisos que el resto de integrantes del equipo sobre la aplicación.

Inicialmente, al delegado se le ofrece la interfaz de la página principal (Figura 4.11).

Esta página principal es similar a la del perfil usuario, el único cambio es el perfil con el que se encuentra conectado, que en este caso es el **Perfil Delegado**.

Una vez que el delegado seleccione un deporte concreto, se muestran las opciones que puede realizar sobre ese deporte (figura 4.12).

Las acciones que puede realizar el delegado son las siguientes:

1. Insertar Resultado.
2. Horarios Disponibles para Aplazamientos.
3. Resultados.
4. Clasificación.
5. Deportividad.



Figura 4.12: Acciones Delegado

6. Goleadores/Anotadores.
7. Datos delegados de tu competición.
8. Aplazar Partido.
9. Seleccionar otro deporte.

Como se puede observar, estas opciones no son únicamente de consulta si no también de modificación sobre la BBDD, como se ha mencionado anteriormente, el perfil delegado cuenta con más permisos que el perfil usuario.

Entre las acciones que se observan en el perfil delegado, se puede ver que están incluidas todas las que puede realizar el perfil usuario normal (resultados, clasificación, deportividad, goleadores/anotadores y selección de otro deporte).

A continuación se explicarán las acciones propias del delegado, ya que el resto de acciones funcionan de la misma manera que en el perfil anterior.

### 4.4.1. Insertar Resultado.

La acción de Insertar resultado es exclusiva del delegado del equipo.

La funcionalidad principal es que el delegado pueda, una vez disputado un partido, insertar el resultado del mismo en la aplicación Web. Resultado que, posteriormente, deberá confirmar el administrador, una vez haya recibido el acta del partido por parte del árbitro.



Figura 4.13: Página Principal Inserción

El fin de la creación de esta funcionalidad es que los equipos, representados por el delegado, tengan una mayor participación en todo lo relacionado con su actividad deportiva. Creando, de esta forma, un mayor interés hacia la competición y el Departamento del Espacio de Estudiantes.

Como requisitos indispensables de insertar resultado tenemos:

- Únicamente podrá insertar resultados el delegado del equipo.
- El delegado de un equipo sólo podrá insertar resultados del equipo (o los equipos) en los que sea delegado, nunca del resto de equipos donde no esté dado de alta como delegado.
- Únicamente podrá insertar el resultado del partido en la jornada en la que su equipo jugó como equipo local.

De esta manera, sólo un delegado podrá insertar el resultado de un partido (confiando en que los delegados sean lo suficientemente responsables e inserten el resultado correcto) y al administrador únicamente le llegue la información de un resultado y pueda corroborarlo con el acta en la mano. Confirmando el resultado, si es correcto, o corrigiéndolo si es incorrecto.

Explicados los requisitos que debe cumplir un delegado para insertar un resultado, se mostrará a continuación como realizar la inserción del mismo.

Una vez que el delegado pulse el link «Insertar Resultado» le aparecerá una pantalla que corresponderá a la página principal de insertar resultado (Figura 4.13).

Se puede observar que, a la izquierda, aparece un menú con las acciones que puede realizar el delegado, de la misma manera que se hacía con el usuario.

En el centro de la página se muestra una nota informativa explicándole al delegado que únicamente podrá insertar resultados de la jornada en la que su equipo jugó como equipo local, y a continuación, se muestran dos acciones:

1. Seleccionar el equipo del que se quiere insertar el resultado: Se trata de un desplegable donde aparece una lista con los equipos de los que el usuario es el delegado. Independientemente del deporte seleccionado inicialmente en la aplicación, la acción de insertar resultado permitirá al delegado insertar el resultado de cada uno de sus equipos aunque pertenezcan a deportes diferentes. De esta manera, se pretende aportar comodidad al usuario, ya que podrá insertar los resultados de sus equipos sin la necesidad de tener que cambiar de deporte. Por lo tanto, la opción de insertar resultado es independiente al deporte seleccionado.
2. Seleccionar la jornada en la que se quiere insertar un resultado.
3. Botón Consultar: Ejecuta la búsqueda.

Una vez seleccionado el equipo y la jornada, se buscará el partido concreto que el delegado quiere insertar el resultado. Si el equipo del delegado jugó como equipo local, dejará realizar la inserción, proporcionándole la información del partido, como se observa en la figura 4.14. En caso contrario, no podrá realizar la inserción y le aparecerá un mensaje de error indicándole que la inserción de esa jornada la debe realizar el delegado del equipo contrario, como se muestra en la figura 4.15.

Como se puede observar en la Figura 4.14, se muestra al delegado la información del partido según la jornada seleccionada, entre esta información se encuentra el nombre del equipo local y del equipo visitante, el número de jornada, la fecha y la hora en la que se disputó el partido.

Aparecen dos campos que el delegado debe insertar, la puntuación del equipo local (la de su propio equipo) y la puntuación del equipo visitante. Una vez se hayan rellenado ambos campos, se pulsará el botón guardar.

A continuación se le mostrará al delegado una pantalla con el resumen de la información que se guardará en la BBDD, figura 4.16. Si está de acuerdo, guardará la información esperando la confirmación por parte del administrador. Si no está de acuerdo, cancelará la inserción y se le redigirá a la página principal de insertar resultado.

**INSERTAR RESULTADO** **Perfil Delegado**

Bienvenido/a MARTA  [Desconectar](#)

[Insertar resultado](#)  
[Horarios disponibles para aplazamientos](#)  
[Resultados](#)  
[Clasificación](#)  
[Deportividad](#)  
[Goleadores y Anotadores](#)  
[Datos de los delegados de tu competición](#)  
[Aplazar Partido](#)  
[Seleccionar otro deporte](#)

**Inserta en puntuación los goles marcados, las canastas encestandas o los sets logrados.**

Equipo Local: POLLOS SIN CABEZA  
Puntuación:

Equipo Visitante: DRINK TEAM GETAFE  
Puntuación:

Jornada: 1  
Fecha: 2008-10-29  
Hora: 14:10

Figura 4.14: Formulario a insertar por el delegado

**INSERTAR RESULTADO** **Perfil Delegado**

Bienvenido/a MARTA  [Desconectar](#)

[Insertar resultado](#)  
[Horarios disponibles para aplazamientos](#)  
[Resultados](#)  
[Clasificación](#)  
[Deportividad](#)  
[Goleadores y Anotadores](#)  
[Datos de los delegados de tu competición](#)  
[Aplazar Partido](#)  
[Seleccionar otro deporte](#)

**EN ESTA JORNADA NO PARTICIPAS COMO EQUIPO LOCAL. DEBE INSERTAR EL RESULTADO EL DELEGADO DEL OTRO EQUIPO.**

**Únicamente puedes insertar el resultado de la jornada en la que tu equipo jugó como Equipo Local.**

Selecciona el equipo que quieras insertar resultado:

Jornada:

Figura 4.15: Inserción Errónea



Figura 4.16: Resumen de la inserción



Figura 4.17: Fechas y Horarios. Selección de campus.

### 4.4.2. Horarios Disponibles para Aplazamientos.

Este apartado ha sido creado para que el delegado pueda consultar los horarios disponibles de las pistas, de cualquiera de los tres campus de la UC3M. De esta manera, se facilita al personal del Espacio de Estudiantes, que los usuarios soliciten el aplazamiento de un partido, sabiendo que tienen pistas libres para disputarlo. Se evita que los delegados deban presentarse en las instalaciones para consultar la disponibilidad.

Una vez que el usuario seleccione la acción «Horarios disponibles para aplazamientos», deberá seleccionar el campus que desea consultar (Figura 4.17).

A continuación, debe realizar tres acciones (Figura 4.18):

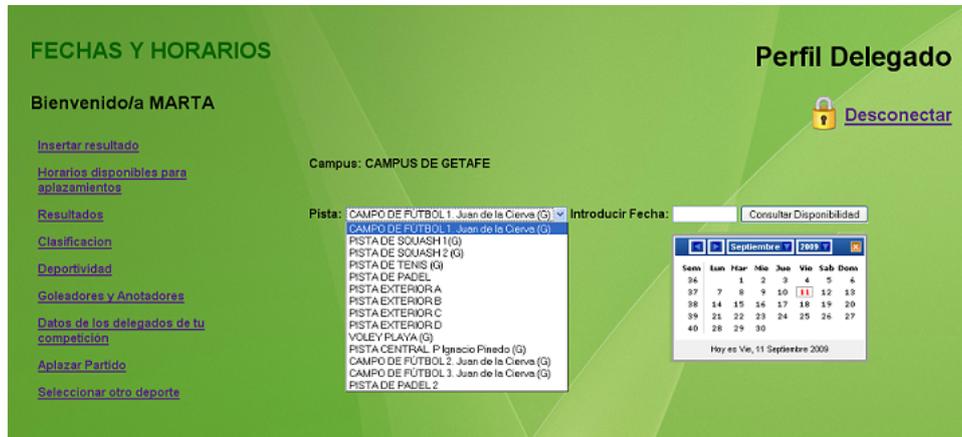


Figura 4.18: Selección de Pista y Fecha

1. Seleccionar la pista: Aparece un desplegable con el conjunto de pistas pertenecientes al campus seleccionado anteriormente.
2. Introducir Fecha: Aparece un mini calendario para que el usuario seleccione el día que quiere consultar.
3. Consultar Disponibilidad: Una vez rellenados los dos campos anteriores, se pulsa el botón de Consultar Disponibilidad para que se realice la consulta con los datos pertinentes.

La información relacionada con la pista aparece en forma de tabla, y contiene información no sólo del día seleccionado si no de toda la semana correspondiente.

El formato que se ha seguido para mostrar la información se puede ver en la figura 4.19.

En la parte superior de la tabla se muestra una cabecera con los días de la semana y sus fechas. Estas fechas corresponden con la semana a la que pertenece el día consultado por el usuario.

En la parte izquierda de la tabla se muestra un intervalo horario, este intervalo corresponde con el horario en el que la pista esta abierta. Cada pista tiene un horario distinto. En el caso de las pistas de futbol, por ejemplo, solo se encuentran abiertas al mediodía. El resto de las pistas suelen estar abiertas todo el día.

Sin embargo, no está siempre la pista disponible para actividades deportivas durante todo el horario de apertura.

Cada día de la semana tiene un horario reservado para uso exclusivo deportivo, es por eso, que en la figura 4.19 se puede ver una zona sombreada en gris indican-



Figura 4.19: Consulta de la pista

do que la pista no se puede reservar (NO RESERVABLE), y una zona sombreada en verde indicando que la pista está libre (en el caso de que esté libre) y se puede reservar (LIBRE.RESERVABLE).

Si la pista estuviese ocupada, figura 4.20, aparecería el intervalo horario en el que está ocupada, en sombreado rojo indicando que está OCUPADA, y por quien (puede describir la competición o poner el nombre de los equipos por los que está ocupada).

#### 4.4.3. Datos Delegados de tu competición.

En este apartado se pueden consultar los datos de los delegados de tu competición.

Esta opción se creó con el fin de que todos los delegados tengan acceso a la información del resto de los delegados de su competición, de esta manera pueden ponerse en contacto para aplazar partidos, elegir nuevas fechas, etc.

La consulta se realiza por campus, y nos muestra la información de todos los delegados del deporte seleccionado. Figura 4.21.

#### 4.4. PERFIL DELEGADO.

**FECHAS Y HORARIOS** **Perfil Delegado**

Bienvenido/a MARTA  [Desconectar](#)

[Insertar resultado](#)  
[Horarios disponibles para aplazamientos](#)  
[Resultados](#)  
[Clasificación](#)  
[Deportividad](#)  
[Goleadores y Anotadores](#)  
[Datos de los delegados de tu competición](#)  
[Aplazar Partido](#)  
[Seleccionar otro deporte](#)

Campus:

Pista:

	Lunes20/04/2009	Martes21/04/2009	Miercoles22/04/2009	Jueves23/04/2009	Viernes24/04/2009
13:00-14:10	Libre Reservable.	Libre Reservable.	Libre Reservable.	Libre Reservable.	Libre Reservable.
14:10-15:20	Ocupada. SEMIFINAL 1: DRINK TEAM Vs	Ocupada. CUARTOS: AULA GLOBAL Vs COGOLLITOS BALOMPIE	Ocupada. CUARTOS: POLLOS SIN CABEZA Vs YAYO VALLECANO	Libre Reservable.	Libre Reservable.
15:20-16:30	Libre Reservable.	Libre Reservable.	Ocupada. CUARTOS: FICOS FC Vs PROYECTO HOMBRE	Libre Reservable.	Libre Reservable.

Figura 4.20: Pista Ocupada

**DATOS DELEGADOS COMPETICION** **Perfil Delegado**

Bienvenido/a MARTA  [Desconectar](#)

[Insertar resultado](#)  
[Horarios disponibles para aplazamientos](#)  
[Resultados](#)  
[Clasificación](#)  
[Deportividad](#)  
[Goleadores y Anotadores](#)  
[Datos de los delegados de tu competición](#)  
[Aplazar Partido](#)  
[Seleccionar otro deporte](#)

Campus:

APELLIDOS	NOMBRE	EQUIPO	EMAIL	TLF.MOVIL
GARCIA DIAZ	JORGE	FICOS F.C.	100083782@ALUMNOS.UC3M.ES	677500989
GONZALEZ SERRANO	ANDRES	ESTRELLA COJA	100073233@alumnos.uc3m.es	678116379
LOPEZ RIVERA	OSCAR	ASTON BIRRAS	100074664@alumnos.uc3m.es	647244343
GAITAN RISCO	DANIEL	TITULCIA F7	100078983@alumnos.uc3m.es	620 868 153
FRALE JORDANA DE POZAS	JACOBO	INTER DE MELON	100072677@alumnos.uc3m.es	687514088
ARNALDOS ORTS	ENRIQUE	FERNANDO DE LOS RIOS A	100079744@alumnos.uc3m.es	636405809
PINEDA	NESTOR JUAN	Netkets Inkjetos A7	100056808@alumnos.uc3m.es	661720443

Figura 4.21: Datos Delegados Competición

Los campos de los que dispone esta opción son:

- Apellidos: Corresponden con los apellidos del delegado.
- Nombre: Corresponde con el nombre del delegado.
- Equipo: Corresponde con el equipo al que pertenece el delegado.
- Email: Corresponde con el correo electrónico del delegado para ponerse en contacto con él.
- Teléfono: Corresponde con el teléfono de contacto.

En el caso de que un delegado quisiera consultar la información de otro deporte donde no esté dado de alta como delegado, se le mostrará una pantalla de error (como en casos anteriores) indicándole que no puede obtener esa información.

Únicamente podrá obtener la información del resto de delegados, del deporte en el que esté dado de alta.

### **4.4.4. Aplazar Partido.**

La sección Aplazar Partido permite al delegado de un equipo solicitar el aplazamiento de un partido.

El aplazamiento se realiza a través de un formulario que el delegado debe rellenar. Este formulario se envía directamente al administrador.

El administrador es el encargado de contestar al aplazamiento, una vez recibido el formulario por parte de los dos delegados de los equipos que quieren aplazar el partido.

Los requisitos para que un delegado pueda solicitar el aplazamiento de un equipo consisten en:

- El Delegado debe solicitar el aplazamiento de un deporte en el que esté dado de alta como delegado.
- El Delegado puede solicitar el aplazamiento con cualquier equipo de cualquier grupo que esté dado de alta en el deporte del equipo. Se realiza de esta manera, con el fin de que este apartado de la aplicación no sea exclusivamente para deportes de liga y se pueda generalizar a la hora de proponer un cambio. Ya sean partidos amistosos, competición interna, liga o copa. Siempre y cuando pertenezcan al mismo deporte.

Para la solicitud del aplazamiento, el delegado debe seguir los pasos indicados en la Figura 4.22.

Posteriormente a la elección del deporte del que se quiere solicitar el aplazamiento, se muestra al delegado una lista con los grupos que forman ese deporte con el fin de que seleccione el grupo contra el que quiere solicitar el aplazamiento.

A continuación debe rellenar el formulario (Figura 4.23), que consta de los siguientes campos:

- Deporte: Seleccionado inicialmente, con este deporte se van a realizar las búsquedas para obtener la información estática del formulario.

The screenshot shows a web interface for 'APLAZAMIENTOS. FUTBOL A7'. The user is logged in as 'MARTA'. The main content area contains the following text: 'Una vez hayáis comprobado la disponibilidad de pistas, rellenad ambos delegados el formulario y esperar la notificación del comite organizador.' Below this is a warning: '¡¡RECORDAD QUE HASTA QUE NO TENGÁIS ESA RESPUESTA EL PARTIDO NO ESTÁ APLAZADO!!'. The form includes a dropdown menu for 'Seleccione grupo (Del equipo contrario)' with the value 'FUTBOL 7 COLMENAREJO 08/09 G1'. There is an 'Aceptar' button. A 'Desconectar' button is visible in the top right corner.

Figura 4.22: Aplazamiento. Selección de grupo

- Grupo: Seleccionado por el usuario, grupo al que pertenece el equipo contra el que quiere solicitar el cambio (puede coincidir con el grupo del equipo del delegado o no).
- Mi Equipo: Nombre del equipo del delegado que solicita el aplazamiento.
- Equipo Contrario con el que se quiere solicitar el cambio: Lista de equipos pertenecientes al grupo seleccionado.
- Fecha Inicial del partido: Campo a rellenar por el delegado indicando la fecha original del partido.
- Hora inicial del partido: Campo a rellenar por el delegado. Indica la hora original del partido.
- Fecha que se propone: Campo a rellenar por el delegado con la fecha que el equipo propone.
- Hora que se propone: Campo a rellenar por el delegado con la hora que se propone para jugar el partido.

Una vez que se ha rellenado y enviado, automáticamente y de manera invisible al usuario ese formulario llega al correo del administrador con el formato de la figura 4.24.

Para el envío del formulario se utiliza el protocolo SMTP de la UC3M.

En el caso que el delegado quiera aplazar un partido de un deporte donde no está dado de alta en ningún equipo como delegado, se mostrará una pantalla de error. Figura 4.25.

**APLAZAMIENTOS. FUTBOL A7** **Perfil Delegado**

Bienvenido/a MARTA  [Desconectar](#)

[Inserir resultado](#)  
[Horarios disponibles para aplazamientos](#)  
[Resultados](#)  
[Clasificación](#)  
[Deportividad](#)  
[Goleadores y Anotadores](#)  
[Datos de los delegados de tu competición](#)  
[Aplazar Partido](#)  
[Seleccionar otro deporte](#)

Una vez hayáis comprobado la disponibilidad de pistas, rellenad ambos delegados el formulario y esperar la notificación del comite organizador.

¡¡RECORDAD QUE HASTA QUE NO TENGÁIS ESA RESPUESTA EL PARTIDO NO ESTÁ APLAZADO!!

Deporte: FUTBOL A7  
Grupo: FUTBOL 7 COLMENAREJO 08/09 G1  
Mi Equipo: POLLOS SIN CABEZA  
Equipo contrario con el que se quiere solicitar el cambio: SERVIA Y TÓ LO NEGRO

Fecha inicial del partido:   
Hora inicial del partido:   
Fecha que se propone:   
Hora que se propone:

Figura 4.23: Formulario Aplazamientos

**Aplazamiento de partido. --FUTBOL 7 COLMENAREJO 08/09 G1--POLLOS SIN CABEZA** [---La Charly---] | X

☆ [100039679@alumnos.uc3m.es](mailto:100039679@alumnos.uc3m.es) [mostrar detalles](#) 11:40 (0 minutos antes) [Responder](#)

\*\*\*\*\* APLAZAMIENTO DE PARTIDO \*\*\*\*\*

DEPORTE: FUTBOL A7  
EQUIPO LOCAL: POLLOS SIN CABEZA  
EQUIPO VISITANTE: SERVIA Y TÓ LO NEGRO  
FECHA PARTIDO: 12/09/2009  
HORA PARTIDO: 12:00  
FECHA PROPUESTA: 12/12/2009  
HORA PROPUESTA: 12:00

[Responder](#) [Reenviar](#)

Figura 4.24: Email de solicitud del aplazamiento



Figura 4.25: Error al solicitar un aplazamiento.

## 4.5. Perfil Administrador.

El administrador se autentica en la aplicación de la misma manera que un usuario o un delegado, pero con login y password especiales.

El administrador debe ser la persona o personas que se encarguen del correcto funcionamiento de la aplicación, así como de la validación de los resultados.

El administrador, por lo tanto, goza de permisos especiales.

Las acciones que puede realizar el administrador son:

1. Confirmar Resultado.
2. Horarios Disponibles para Aplazamientos.
3. Resultados.
4. Clasificación.
5. Deportividad.
6. Goleadores/Anotadores.
7. Datos delegados de tu competición.
8. Seleccionar otro deporte.

Como se puede observar son las mismas que las mostradas en el perfil delegado, con la excepción del «Confirmar Resultado», que es la funcionalidad principal del administrador. Figura 4.26.



Figura 4.26: Opciones del Administrador



Figura 4.27: Lista de resultados por confirmar

### 4.5.1. Confirmar Resultado.

El administrador confirmará los resultados que los delegados hayan insertado, corroborando que sean los correctos.

Las confirmaciones aparecerán en forma de lista (Figura 4.27).

Se puede observar que cada link tiene un asunto concreto indicando la competición y los equipos que participan en el resultado.

El administrador pulsará los distintos links para la confirmación de los resultados. Se le mostrará una pantalla con un resumen de la información que el delegado insertó (figura 4.28), pudiendo modificar los siguientes campos:

**DETALLE** **Perfil Administrador**

Bienvenido/a ADMINISTRADOR  Desconectar

[Confirmar resultado](#)

[Horarios disponibles para aplazamientos](#)

[Resultados](#)

[Clasificación](#)

[Deportividad](#)

[Goleadores y Anotadores](#)

[Datos de los delegados de la competición](#)

[Seleccionar otro deporte](#)

**Información General**

Campeonato/Grupo: FUTBOL 7 GETAFE 08/09 G2

Identificador del Partido: 12522

Fecha Partido: 2008-10-29

Hora Partido: 14:10

Jornada: 1

Nombre Equipo 1: POLLOS SIN CABEZA

Puntuación: 1

Nombre Equipo 2: DRINK TEAM GETAFE

Puntuación: 1

Tipo Resultado: No celebrado

Figura 4.28: Detalle de la confirmación

- Puntuación Equipo Local
- Puntuación Equipo Visitante

Esta modificación la realizará el administrador con el acta en la mano, una vez que la reciba por parte del árbitro. De esta manera se garantiza la veracidad de los datos.

Finalmente, debe concluir indicando el tipo de resultado que se ha producido en el partido (Gana Local, Gana Visitante, Empate...). Una vez confirmado el resultado, el administrador puede realizar tres acciones:

1. Guardar: Se confirma la información guardándola en la BBDD. Se modifican las vistas de Resultados y Clasificación, mostrando la información actualizada.
2. Eliminar: Se elimina esa información de la BBDD.
3. Cancelar: No se modifica la BBDD, porque la información no se confirma. En ese caso, el resultado pendiente de confirmar seguirá apareciendo en la aplicación del administrador.

Una vez que el resultado ha sido confirmado, se le notifica al administrador. Aparecerán, posteriormente, el resto de resultados pendientes de confirmar, o ninguno en caso de que todos estén confirmados. Figura 4.29

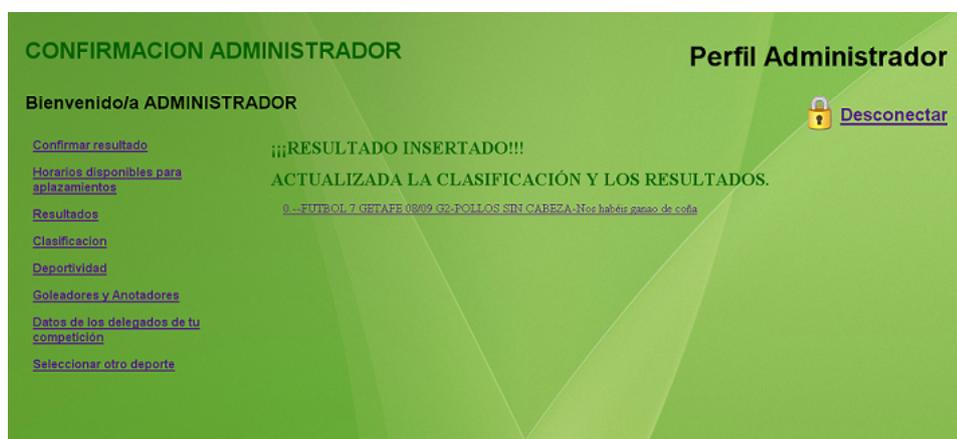


Figura 4.29: Resultado Confirmado por parte del administrador

### 4.5.2. Datos Delegados de tu competición.

En esta sección, el administrador podrá consultar los datos de todos los delegados de cualquier competición.

La diferencia con el perfil delegado radica en que el administrador podrá consultar toda la información de los delegados mientras que el perfil delegado únicamente podrá consultar los datos de los delegados correspondientes a su competición, denegándole el acceso al resto de datos.

## Diseño de Alto Nivel

---

En este capítulo, una vez valoradas algunas de las alternativas existentes en el mercado, se justifican las tecnologías empleadas en el desarrollo de la aplicación. Finalmente, se detalla la arquitectura de la misma.

### 5.1. Selección de tecnologías.

#### 5.1.1. MVC: Modelo Vista Controlador

El modelo vista-controlador se ve frecuentemente en aplicaciones Web ya que separa los datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

En una aplicación Web, la vista es la página HTML, el modelo corresponde con el SGBD y el controlador será el responsable de recibir eventos por parte de los usuarios (servlets).

Uno de los requisitos en la propuesta inicial que se ofrecía del proyecto era usar el MVC, ya que tiene múltiples ventajas en el entorno Web:

- Clara separación entre los componentes del programa: datos, eventos y vistas.
- Permite implementación por separado.
- La conexión entre el modelo y sus vistas es dinámica.
- Sencillez para crear distintas representaciones de los mismos datos.
- Facilidad para la realización de pruebas unitarias o conjuntas.

- Desarrollos más escalables.

### 5.1.2. Patrón de desarrollo: Servlet y JSP

Como patrón de desarrollo se pide, en el informe inicial del proyecto, la utilización de Servlets y JSPs, ya que aportan una serie de ventajas frente a otras tecnologías como CGI, ASP, SSI y PHP.

#### **Ventajas de los Servlets:**

- Favorecen la independencia de la plataforma, ya que están escritos en Java, y por tanto siguen el estándar API.
- Los servlets pueden ser ejecutados en una cantidad enorme de servidores (Apache, Java Web Server, ...).
- Una vez que se tiene un servidor Web, añadir un servlet no supone un coste elevado.
- Son programas rápidos, ya que utilizan hilos (threads) en lugar de procesos.
- Son portables como cualquier otra aplicación de Java e independiente de la plataforma utilizada.
- Aportan seguridad porque son archivos de clases compilados, y no se permite la manipulación, por parte del usuario, de su código fuente.

#### **Ventajas de los JSPs:**

- La parte dinámica está escrita en Java y no en Visual Basic (como ASP), lo que hace que sea más poderosa y fácil de usar.
- Es portable a otros sistemas operativos y servidores Web.
- Gracias a la separación del formato y el contenido se pueden realizar tareas de manera independiente, por un lado, generar la página HTML y por otro insertar el contenido dinámico de la aplicación.
- SSI es una tecnología que incluye piezas definidas externamente dentro de una página Web estática, sin embargo, JSP permite usar servlets en lugar de un programa por separado para generar las partes dinámicas.

## 5.2. ARQUITECTURA DE LA APLICACIÓN.

---

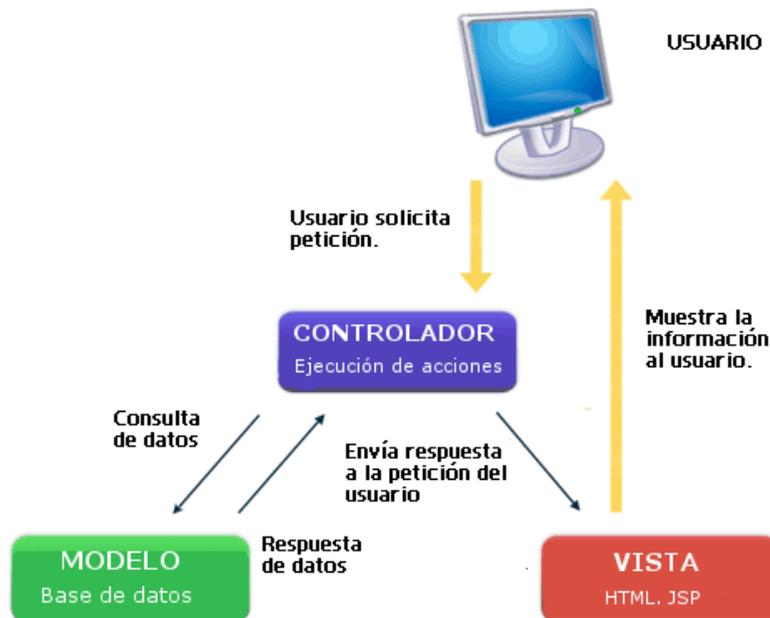


Figura 5.1: Arquitectura de la aplicación.

- La ventaja que presenta un JSP frente a PHP es que el lenguaje de programación utilizado en el primero es Java, el cual es probable que ya se conozca por el usuario (por su extensa API), mientras que en PHP es necesario aprender un lenguaje de programación nuevo.

## 5.2. Arquitectura de la aplicación.

El principal objetivo de este apartado es mostrar cómo se encuentra organizada la aplicación y dar una explicación general sobre las diferentes partes que la componen.

El modelo diseñado para la implementación de esta herramienta viene explicado en el diagrama de la Figura 5.1, que representa de manera global la arquitectura empleada en su desarrollo.

Según el diagrama, se observa como la aplicación se compone de distintos módulos que desarrollarán una función específica dentro de la misma.

La aplicación deberá estar ubicada en un servidor de la UC3M, actualmente se encuentra ubicada en un servidor local creado.

El cliente únicamente realiza peticiones al servidor a través de un navegador Web. El servidor deberá responder a las peticiones realizando consultas al servidor de la BBDD, y obteniendo así los datos solicitados por el cliente. Esta información se volverá a enviar al navegador para mostrársela al usuario.

Cabe destacar tres módulos principales dentro de la aplicación, que corresponden, como hemos dicho anteriormente, con el modelo vista-controlador:

1. Módulo de almacenamiento.
2. Módulo de procesamiento de la información.
3. Módulo de visualización.

### **5.2.1. Módulo de Almacenamiento.**

El módulo de almacenamiento corresponde con el «Modelo» en el diagrama MVC.

Toda la información que maneja la aplicación está almacenada en la BBDD (DeporWin).

La BBDD interactúa con el módulo de procesamiento de la información («Controlador»), que es el encargado de realizar las consultas, búsquedas, inserciones y eliminaciones en la misma.

Se puede ver en la figura 5.1 como la relación existente entre el módulo de almacenamiento y el de procesamiento es bidireccional, ya que por el lado del controlador se realizan las peticiones a la BBDD, y por el lado de la BBDD se responden a dichas peticiones.

### **5.2.2. Módulo de procesamiento de la información.**

Dentro del MVC, este módulo corresponde con el «Controlador».

Este módulo es el más complejo porque tiene que interactuar con un mayor número de tecnologías.

Debe ser capaz de procesar como entrada los datos de la BBDD o la información de los usuarios.

Los elementos de salida habrá que almacenarlos en la base de datos u ofrecerlos directamente al usuario, interactuando así con el modelo de visualización.

## 5.2. ARQUITECTURA DE LA APLICACIÓN.

---

Coopera con el módulo de almacenamiento enviando órdenes de insertar, modificar, consultar o eliminar registros de la base de datos. Y con el módulo de visualización mostrando o recogiendo los datos del usuario.

Es el mediador entre todos los módulos y hace de enlace entre la información presentada a través de la interfaz y la base de datos.

### **5.2.3. Módulo de visualización.**

Este módulo se encarga de presentar la información al usuario y recoger sus acciones en la aplicación. En el MVC se corresponde con la «Vista». La información es presentada mediante páginas HTML, ofreciendo una atractiva y sencilla interfaz para el usuario.

El objetivo es que el cliente no tenga que tener ningún tipo de conocimiento previo sobre el manejo de la interfaz. De esta forma, las transacciones que realiza la aplicación con la base de datos y con el sistema de procesamiento serán ajenas al usuario.



## Diseño de Bajo Nivel

---

En este capítulo se explicará la BBDD a utilizar, se detallará el modelo de datos de la aplicación y los objetos necesarios para la solución del problema.

### 6.1. DeporWin: BBDD del Departamento de Actividades Físicas y Culturales.

DeporWin es una aplicación de gestión de actividades deportivas, ocio y salud. Es, actualmente, la aplicación con la que el departamento de Actividades Deportivas y Culturales (Espacio de Estudiantes) trabaja.

Las características principales son:

- Diseñado como una aplicación Windows. Implica sobre los usuarios una mayor comodidad de uso.
- Escalable: Permite trabajar con diferentes entornos de datos (Microsoft Access, MSDE y SQL Server).
- Doble arquitectura: Consta de un Servidor de Datos y un Servidor de Aplicaciones. Gracias a esta doble arquitectura soporta conexiones locales de los clientes de la red, así como conexiones remotas de los usuarios de la intranet.

Como se comentó en la introducción (*ver Capítulo 1, apartado 1.1.Motivación*), la aplicación Web actualmente publicada trata únicamente sobre competiciones de liga. En ella se pueden realizar un número muy limitado de consultas (Clasificación, Goleadores/Anotadores, Resultados y Deportividad).

Partiendo de esa base se desea realizar una ampliación. Para llevarla a cabo se debería mantener la funcionalidad actual más las ampliaciones pertinentes.

Utilizando el todo momento la BBDD proporcionada por DeporWin, donde actualmente, está almacenada toda la información relacionada con las Actividades Deportivas y Culturales.

En el estudio de estas posibles ampliaciones surge el problema de que la aplicación pertenece a una empresa privada, por lo tanto, no se dispone ni del código fuente ni de la documentación de la BBDD.

La solución propuesta consiste en crear una aplicación completamente nueva, donde se repite la funcionalidad que actualmente está publicada más ampliaciones descritas anteriormente, utilizando únicamente la BBDD proporcionada por DeporWin.

Inicialmente se realizó una conexión con la BBDD, para posteriormente llevar a cabo un estudio analizando la funcionalidad que nos proporcionaba cada una de las tablas.

### **6.2. Modelo de Datos.**

Fruto del estudio de esa BBDD surge el modelo de datos completo a utilizar.

En la Figura 6.1 se puede observar el diagrama relacional utilizado. Se visualizan los campos de cada objeto, sus identificadores y las relaciones existentes entre cada una de ellas.

En este diagrama únicamente aparecen los campos que se han utilizado en la aplicación.

### **6.3. Objetos utilizados.**

Los objetos utilizados en la aplicación, y visualizados en el diagrama relacional de la Figura 6.1, surgen de la necesidad de cumplir los requisitos nombrados anteriormente (*Ver capítulo 3, apartados 3.1 y 3.2*).

A continuación se describen los objetos necesarios en la aplicación, detallando cada uno de los campos y las acciones más importantes que se podrán realizar sobre ellos:

### 6.3. OBJETOS UTILIZADOS.

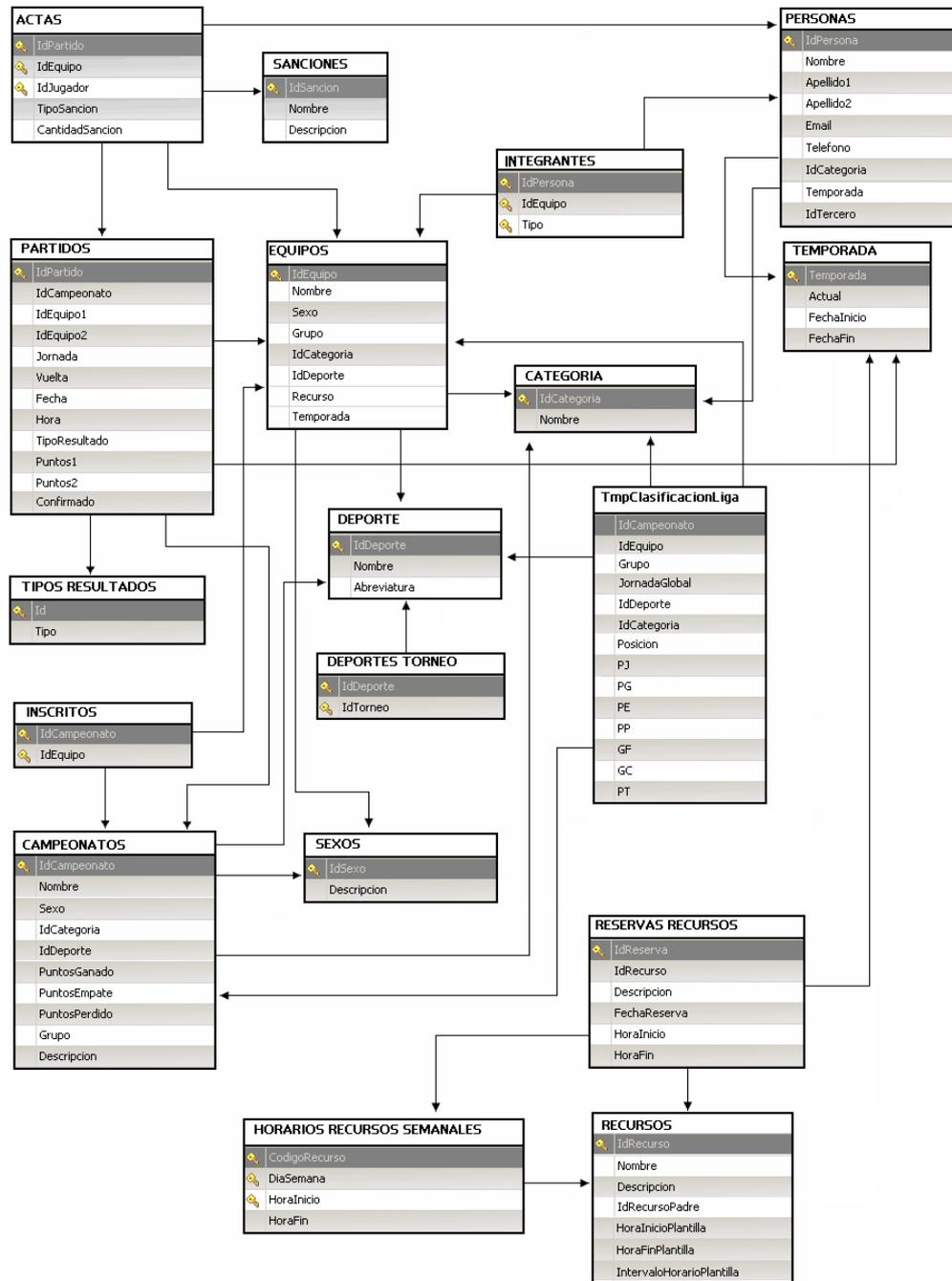


Figura 6.1: Diagrama Relacional

### 6.3.1. Actas.

Un acta es un testimonio escrito de los hechos ocurridos en un partido. El árbitro es el encargado de rellenarla y posteriormente se envía al administrador del departamento para que tenga constancia de lo ocurrido en el partido. En ella se apuntan los tantos y las sanciones (si las hubiese).

#### Campos

- IdPartido: Entero que representa el identificador del partido al que hace referencia el acta.
- IdEquipo: Entero que hace referencia al Identificador del Equipo.
- IdJugador: Entero que hace referencia al Identificador del Jugador al que se le ha puesto la sanción.
- Tipo Sanción: Entero que indica el tipo de sanción impuesta. (Tarjeta Amarilla, Tarjeta Roja...)
- Cantidad Sanción: SmallInt que indica numéricamente la cantidad de sanción impuesta (comprendida entre 0 y 10).

#### Acciones

- Buscar sanción impuesta a un equipo.(Deportividad).
- Consultar los tantos obtenidos por un jugador en los partidos disputados hasta el momento.(Goleadores/Anotadores).

### 6.3.2. Campeonatos.

Un campeonato es una competición deportiva en la que participan un número determinado de equipos, los cuales se enfrentan entre sí para fijar, en función de los resultados obtenidos en cada uno de esos encuentros, una clasificación final.

#### Campos

- IdCampeonato: Entero que hace referencia al identificador del campeonato.
- Nombre: Cadena de texto con el nombre del campeonato.
- Sexo: Entero que especifica de que sexo es el campeonato (femenino o masculino).

- IdCategoria: Entero que describe la categoría a la que pertenece ese campeonato (senior, junior, etc).
- IdDeporte: Entero que relaciona el campeonato con un deporte concreto.
- Puntos Ganado: Entero que representa la puntuación que se le asigna a los equipos de ese campeonato si ganan un partido.
- Puntos Empate: Entero que representa la puntuación que se le asigna a los equipos de ese campeonato si empatan un partido.
- Puntos Perdido: Entero que representa la puntuación que se le asigna a los equipos de ese campeonato si pierden un partido.
- Grupo: Cadena de texto que define los distintos grupos que forman un campeonato.
- Descripción: Cadena de texto que describe el campeonato.

### **Acciones**

- Consultar los campeonatos que forman un deporte, y los grupos que forman un campeonato.
- Consultar los resultados obtenidos por todos los equipos de un campeonato.(Resultados)
- Consultar la puntuación asignada, según el tipo de resultado obtenido por el equipo (si ganan, pierden o empatan), de un determinado campeonato. (En la acción de insertar resultado, se utilizan estos campos para actualizar la clasificación y los resultados).

### **6.3.3. Categoría.**

Dentro de todos los deportes existen diferentes categorías, la más usual es SENIOR por la edad del alumnado.

#### **Campos**

- IdCategoria: Entero que representa el identificador de la categoría.
- Nombre: Cadena de texto que indica el nombre de la categoría.

#### **Acciones**

- Consulta los deportes concretos de una categoría.
- Consulta los equipos correspondientes a una categoría de un determinado campus y temporada.
- Consulta los grupos de un campeonato según la categoría.

### **6.3.4. Deporte.**

La siguiente tabla indica los deportes que se encuentran dados de alta en el departamento de Actividades Deportivas. Entre esos deportes se encuentran los de liga y de copa.

#### **Campos**

- IdDeporte: Entero que representa el identificador del deporte.
- Nombre: Cadena de texto que indica el nombre del deporte.

#### **Acciones**

- Consulta los deportes de liga disputados en la UC3M.
- Consulta los deportes correspondientes a torneos de la UC3M.
- Consulta de la clasificación según el deporte.
- Consulta de los resultados según el deporte.
- Consultar los equipos correspondientes a una temporada según el deporte.
- Consultar la deportividad de un deporte.
- Consultar los goleadores/anotadores de un deporte.

### **6.3.5. Equipos.**

La tabla equipos hace una relación de los equipos que están dados de alta en cualquier competición, campeonato y campus de la UC3M. Pueden existir equipos con el mismo nombre en distintas competiciones y deportes.

#### **Campos**

- IdEquipo: Entero que representa el identificador del equipo.

- Nombre: Cadena de texto del nombre del equipo.
- Sexo: Entero que representa si el equipo es femenino o masculino.
- Grupo: Especifica a que grupo pertenece el equipo.
- IdCategoria: Entero que representa la categoría a la que está inscrita el equipo.
- IdDeporte: Entero que relaciona al equipo con un deporte concreto.
- Recurso: Entero que especifica a que campus pertenece ese equipo.
- Temporada: Cadena de texto que indica a que temporada pertenece el equipo.

#### **Acciones**

- Consultar equipos pertenecientes a un deporte, campeonato, competición.
- Consultar los datos de contacto de un equipo.
- Consultar el equipo al que pertenece un usuario.

#### **6.3.6. Horarios Recursos Semanales.**

Objeto que especifica el horario que una pista está disponible para actividades deportivas, según el día de la semana. Este horario no tiene porque coincidir con el horario de apertura y cierre de la pista, ya que las pistas no se utilizan exclusivamente para fines deportivos.

#### **Campos**

- CódigoRecurso: Entero que corresponde con el identificador del recurso. (Por recurso se refiere a la pista consultada).
- DíaSemana: Entero que representa el día de la semana al que hace referencia.
- HoraInicio: String que define la hora de inicio de la reserva de la pista para actividades deportivas.
- HoraFin: String que define la hora fin de la reserva de la pista para actividades deportivas.

### **Acciones**

- Consultar la hora de inicio y fin de reserva de la pista con fines deportivos.

### **6.3.7. Integrantes.**

Objeto que define a los integrantes de un equipo a partir de los siguientes campos:

#### **Campos**

- IdPersona: Entero que representa a cada una de las personas que forman los distintos equipos. Se trata de un Identificador único pero no corresponde con el login del usuario.
- IdEquipo: Entero que hace referencia al equipo al que pertenece esa persona.
- Tipo: Cadena de texto que especifica el tipo de persona de la que estamos hablando. (Jugador, Delegado, etc.).

#### **Acciones**

- Consulta las personas que pertenecen a un equipo.
- Consulta si la persona es jugador o delegado (Autenticación y perfiles).

### **6.3.8. Inscritos.**

Objeto que relaciona los equipos con un campeonato concreto.

#### **Campos**

- IdCampeonato: Entero que representa el identificador del campeonato al que pertenecen los equipos.
- IdEquipo: Entero que indica el equipo.

#### **Acciones**

- Realización de búsquedas de equipos pertenecientes a un campeonato concreto.

### 6.3.9. Partidos.

Objeto que representa la información referente a un partido.

#### Campos

- IdPartido: Entero que representa el identificador del partido.
- IdCampeonato: Entero que indica a que campeonato corresponde el partido jugado.
- IdEquipo1: Entero del identificador del equipo1 (equipo local).
- IdEquipo2: Entero del identificador del equipo 2 (equipo visitante).
- Jornada: Entero que hace referencia a la jornada del partido.
- Vuelta: Entero que hace referencia a la vuelta del partido.
- Fecha: Cadena de texto indicando la fecha del partido.
- Hora: Cadena de texto indicando la hora del partido.
- Tipo Resultado: Entero explicando que tipo de resultado se obtuvo en el partido (Gana Local, Gana Visitante, etc).
- Puntos1: Entero que representa la puntuación obtenida por el Equipo1.
- Puntos2: Entero que representa la puntuación obtenida por el Equipo2.
- Confirmado: Binario que indica la confirmación del partido por parte del administrador.

Este campo es el único que ha sido añadido en la BBDD para la realización de la aplicación. Era necesario para saber si el partido en cuestión correspondía a un resultado introducido por el delegado, donde el valor de «Confirmado» correspondería a 0, ó por el contrario, el administrador ya lo había confirmado, en cuyo caso su valor sería 1.

#### Acciones

- Consulta, por parte del delegado, del partido que quiere insertar el resultado.
- Consulta y actualización, por parte del administrador, para confirmar el resultado introducido por el delegado.

### **6.3.10. Personas.**

Objeto que define a una persona con los siguientes campos:

#### **Campos**

- IdPersona: Entero que representa el Identificador de una persona.
- Nombre: Cadena de texto con el nombre de la persona inscrita.
- Apellidos: Cadena de texto con los apellidos de la persona inscrita.
- Email: Cadena de texto con el email de contacto de la persona.
- Teléfono: Cadena de texto con el teléfono de contacto.
- IdCategoría: Entero que representa la categoría a la que pertenece.
- Temporada: Cadena de texto que representa en que temporada se ha dado de alta a la persona.
- IdTercero: Cadena de texto que hace referencia al NIA o login de la persona.

#### **Acciones**

- Consultar datos de esa persona. (Datos delegado de tu competición).
- Comprobar a través del login (obtenido de la autenticación) si esa persona es delegado o jugador, con el fin de asignarle un perfil.
- Consultar los equipos en los que está inscrita esa persona.

### **6.3.11. Recursos.**

Objeto que identifica los distintos campus y pistas de la UC3M.

#### **Campos**

- Nombre: Cadena de texto con el nombre de la pista o campus.
- Descripción: Cadena de texto con la descripción de la pista o campus.
- IdRecursoPadre: Entero que identifica a que campus pertenece una pista, en el caso de campus no tiene valor este atributo.

- HoraInicioPlantilla: Cadena de texto que indica la hora de apertura de la pista.
- HoraFinPlantilla: Cadena de texto que indica la hora de cierre de la pista.
- IntervaloHorarioPlantilla: Cadena de texto que indica el intervalo de reserva de la pista.

#### **Acciones**

- Consulta el recurso del que queremos obtener la información sobre una pista. (Fechas disponibles para aplazamientos).
- Consulta los horarios de apertura y cierre de una pista. (Fechas disponibles para aplazamientos).

### **6.3.12. Reservas Recursos.**

Objeto que representa las reservas realizadas en una pista.

#### **Campos**

- IdReserva: Entero que representa el identificador de la reserva.
- IdRecurso: Entero que representa el recurso (pista o campus) al que pertenece la reserva.
- Descripción: Cadena de texto que describe la reserva.
- FechaReserva: Cadena de texto con la fecha de la reserva.
- HoraInicio: Cadena de texto que indica la hora inicial de la reserva.
- HoraFin: Cadena de texto que indica la hora final de la reserva.

#### **Acciones**

- Consulta si una pista está reservada un determinado día en un determinado horario. (Fechas disponibles para aplazamientos).

### **6.3.13. Sanciones.**

Corresponde con la puntuación negativa que le otorga un árbitro a un determinado jugador o equipo. Estas sanciones se ven reflejadas en las actas y en el apartado «Deportividad».

#### **Campos**

- IdSancion: Entero que representa el identificador de la sanción.
- Nombre: Cadena de texto con el nombre de la sanción.
- Descripción: Cadena de texto con la descripción de la sanción (Tarjeta Amarilla, Tarjeta Roja, etc.).

#### **Acciones**

- Calcular la deportividad.

### **6.3.14. Sexo.**

Objeto que indica si se trata con deportes, equipos o competiciones masculinas o femeninas.

#### **Campos**

- IdSexo: Entero que representa el identificador del sexo.
- Descripción: Cadena de texto que describe el sexo referido (Masculino o femenino).

#### **Acciones**

- Consulta de los campeonatos por el sexo, para calcular la deportividad.

### **6.3.15. Temporada.**

Objeto que representa la temporada actual con la que se van a realizar todas las consultas.

#### **Campos**

- Temporada: Cadena de texto que indica las temporadas, (2007/2008, 2008/2009, etc.).
- Actual: Bit que representa si la temporada está o no activa, es decir, únicamente estará activa la temporada actual, en nuestro caso 2008/2009.
- Fecha Inicio: Cadena de texto que indica la fecha de inicio de la temporada.
- Fecha Fin: Cadena de texto que indica la fecha fin de la temporada.

### **Acciones**

- Consulta equipos, campeonatos y categorías de una determinada temporada.
- Consulta partidos, resultados, clasificación, deportividad, goleadores/anotadores, datos de los delegados de la propia competición por temporada.

### **6.3.16. Tipos Resultados.**

Objeto que identifica los tipos de resultados que pueden darse en un partido. (Gana Local, Gana Visitante, Empate, No presentado...).

#### **Campos**

- Id: Entero que representa el identificador numérico del tipo de resultado obtenido.
- Tipo: Cadena de texto que describe al identificador.

#### **Acciones**

- A la hora de confirmar un resultado, por parte del Administrador, le asigna el tipo de resultado obtenido.

### **6.3.17. Tmp Clasificación Liga.**

Objeto que representa todo lo acontecido a una clasificación.

#### **Campos**

- IdCampeonato: Entero que identifica el campeonato dentro de la clasificación.

## CAPÍTULO 6. DISEÑO DE BAJO NIVEL

---

- IdEquipo: Entero que identifica el equipo del que se obtiene la clasificación.
- JornadaGlobal: Entero que representa la jornada de la clasificación.
- IdDeporte: Entero que identifica el deporte dentro de la clasificación.
- IdCategoria: Entero que identifica la categoría de la clasificación.
- Grupo: Cadena de texto que describe a un grupo.
- Posición: Entero que representa la posición de un equipo dentro de la clasificación.
- PJ: Entero que indica los Partidos Jugados por el equipo.
- PG: Entero que indica los Partidos Ganados por el equipo.
- PE: Entero que indica los Partidos Empatados por el equipo.
- PP: Entero que indica los Partidos Perdidos por el equipo.
- GF: Entero que indica los Tantos a Favor obtenidos por el equipo.
- GC: Entero que indica los Tantos en Contra obtenidos por el equipo.
- PT: Entero que indica los Puntos Totales obtenidos por el equipo, estos puntos varían según el deporte.

### **Acciones**

- Consulta la clasificación de un grupo.
- Inserta en la clasificación los nuevos valores según los resultados confirmados por el administrador.

# Implementación

---

En este capítulo se describen los problemas y soluciones encontrados a lo largo del desarrollo de la aplicación.

Se trata de justificar las decisiones tomadas en cuanto a diseño y programación.

## 7.1. Acceso a la aplicación.

Es lógico pensar que el acceso a la aplicación se debe realizar a través del servidor LDAP de la UC3M. Ya que se trata de una herramienta que va a estar a disposición de los usuarios de la universidad.

Los perfiles usuario y delegado se autentican insertando el login y password asignados por la universidad.

Dicho esto, se exponen los problemas encontrados en esta sección.

### ■ Problemas:

- Autenticación del administrador.

El administrador no se corresponde con una única persona, si no con el conjunto de personas que forman el Espacio de Estudiantes.

En la BBDD no existe ningún tipo de información relacionada con el Administrador, es decir, no hay forma de averiguar los usuarios y sus correspondientes contraseñas para llevar a cabo la validación por LDAP.

### ■ Soluciones:

- La solución propuesta e implementada para este problema fue la de asignarles un login y password especiales. De esta manera, todo usuario del Espacio de Estudiantes que quiera acceder a la aplicación con el perfil de Administrador deberá autenticarse con el mismo login y password.

No se realiza validación por LDAP de la UC3M como en los perfiles anteriores.

## 7.2. Perfil Usuario.

### 7.2.1. Resultados y Clasificación.

Como se puede ver en el *Capítulo 4*, apartados: 4.3.1 y 4.3.2, los resultados y clasificación muestran la información correspondiente al grupo y jornada solicitada.

#### ■ Problemas:

- Mostrar la información.

El principal problema que surgió en ambas tablas fue la manera de mostrar la información.

Se pensó que el usuario debería poder hacer una consulta por campos detallados, y no mostrarle toda la información relacionada con su equipo.

#### ■ Soluciones:

- La solución implementada consiste en realizar una consulta por grupo y jornada elegidas por el usuario.

De esta forma, se accede a una información más concreta.

## 7.3. Perfil Delegado.

### 7.3.1. Insertar Resultados.

En la BBDD existe una tabla («Partidos») con la relación de partidos que se van a disputar en una temporada. Cuando el delegado decide insertar el resultado

de un partido, actualiza esa tabla añadiendo las puntuaciones obtenidas en ese partido.

■ **Problemas:**

- Si ambos delegados deciden insertar el resultado.
  - \* Se modificaría la fila correspondiente al partido de la tabla «Partidos» dos veces. Quedando únicamente reflejada la última de las modificaciones.
  - \* Se actualizaría (por parte de un delegado) el partido, y se insertaría un fila con el resultado del otro delegado. El administrador, deberá confirmar una fila y eliminar la otra.
  - \* Si el administrador confirmase ambos resultados por error →Se generaría duplicidad en la BBDD y las tablas resultados y clasificación serían modificadas incorrectamente.
  - \* Enlazar ambos resultados en uno para que el administrador únicamente tenga una opción por confirmar →Problema de relacionar esos resultados en la BBDD.
  - \* No concuerden los resultados introducidos por los delegados →El administrador deberá corregir por duplicado.
- Si un único delegado inserta el resultado.
  - \* ¿Quién de los dos deberá insertarlo?.
- Cómo diferenciar si un partido está pendiente de confirmar o confirmado.

■ **Soluciones:**

- Después de ver los problemas generados si ambos delegados insertan resultados, se llega a la solución de que sólo uno de ellos lo inserte.  
Lo lógico es pensar que deberá insertarlo en delegado del equipo local, ya que será el encargado de hacer cualquier trámite en sus instalaciones. De esta forma, al administrador únicamente le llega la confirmación de un resultado del partido jugado.  
Se evita, de esta forma, que se puedan confirmar partidos duplicados con las consecuencias que provocaría (Duplicidad a la hora de mostrar la información de los Resultados, se modificaría de manera errónea la tabla Clasificación, etc.).

- Para diferenciar si un partido esta confirmado por el administrador o pendiente de confirmar, se inserta un campo en la BBDD en la tabla «Partidos» llamado «Confirmado».

Cuando el delegado inserte un resultado, este campo aparecerá a 0, indicándole al administrador que está pendiente de confirmar. Una vez que el administrador lo haya confirmado aparecerá a 1.

Para más información consultar el *Capítulo 6, apartado 6.3.9.*

### 7.3.2. Aplazar Partidos.

Formulario que el delegado deberá rellenar con el fin de proponer el aplazamiento de un partido.

#### ■ Problemas:

- Enviar formulario de manera transparente al usuario y con autenticación.
- Enviarlo a través del smtp de la uc3m.
- Obtener dirección de correo electrónico del delegado para enviar el formulario.

#### ■ Soluciones:

- La dirección de origen se obtiene de realizar una consulta al servidor LDAP del delegado autenticado y la dirección de destino corresponde con el administrador.
- Después de obtener los datos del smtp de la uc3m, se forma el mensaje el asunto y se envía el correo

## 7.4. Problemas y Soluciones del Perfil Administrador.

El administrador es el encargado de confirmar los resultados introducidos por parte de los delegados de los equipos.

### **7.4.1. Confirmar Resultados.**

■ **Problemas:**

- Modificar correctamente las tablas Resultados y Clasificación una vez confirmado un resultado. El principal problema reside en que los partidos se pueden aplazar. Por lo tanto se pueden jugar partidos sin necesidad de seguir el orden de las jornadas.

A la hora de modificar las tablas Resultados y Clasificación surge un problema ya que la información de ambas se consulta por jornadas.

■ **Soluciones:**

- A la hora de confirmar un resultado, se debe tener en cuenta cual fue el último partido insertado independientemente de la jornada jugada. De esta forma, en la tabla Clasificación se modificarán las puntuaciones siempre respecto al último partido disputado.



## Pruebas

En este capítulo se detalla el Plan de pruebas realizado para comprobar el correcto funcionamiento de la Aplicación desarrollada.

Número	Descripción de la prueba	Apto	No Apto
<b>PRUEBAS GENERALES</b>			
1	Autenticarse	X	
2	Cerrar la sesión	X	
3	Cada usuario se conecta al perfil correspondiente	X	
4	Se trabaja sobre la temporada actual	X	
5	Se obtiene a partir de la validación LDAP el nombre y el correo electrónico del usuario	X	
6	Se obtienen los deportes correspondientes a la liga.	X	
7	Se dá la bienvenida al usuario validado	X	
8	Aparece título de la página indicando la acción seleccionada en cada momento por el usuario.	X	
<b>PERFIL USUARIO</b>			
9	Aparecen todas las opciones que se pueden realizar desde el perfil usuario.	X	
<b>Acciones sobre Resultados</b>			
10	Seleccionar grupos asociados a cada deporte.	X	
11	Elección de Jornada.	X	

CAPÍTULO 8. PRUEBAS

---

12	Mostrar correctamente la información seleccionada y la cabecera de esa información.	X	
13	Información ordenada por fecha y hora.	X	
<b>Acciones sobre Clasificación</b>			
14	Seleccionar grupos asociados a cada deporte	X	
15	Elección de Jornada	X	
16	Mostrar correctamente la información seleccionada y la cabecera de esa información	X	
17	Información ordenada por puntuación total	X	
18	Aplicación del criterio de desempate según los tantos a favor del equipo	X	
<b>Acciones sobre Deportividad</b>			
19	Selección de cualquiera de los tres campus	X	
20	Información ordenada por puntuación total	X	
21	Ejecución correcta de la fórmula aplicada a cada equipo/partido y jugador para obtener la puntuación total sobre deportividad	X	
<b>Acciones sobre Goleadores/Anotadores</b>			
22	Selección de cualquier de los tres campus	X	
23	Información por puntuación total	X	
<b>PERFIL DELEGADO</b>			
24	Aparecen todas las opciones que se pueden realizar desde el perfil delegado.	X	
<b>Acciones sobre Insertar Resultados</b>			
25	Acceso único al equipo del que el usuario es delegado	X	
26	Aparecen los equipos de los que es delegado el usuario	X	
27	En la jornada seleccionada, el equipo del delegado jugó como equipo local.	X	
28	Inserción de todos los datos pedidos	X	
29	Inserción correcta, con el formato solicitado	X	

30	Comprobación de errores. Páginas de error	X	
<b>Acciones sobre Horarios Disponibles para Aplazamientos</b>			
31	Función del desplegable con los tres campus	X	
32	Según el campus seleccionado aparezcan las pistas correspondientes a ese campus	X	
33	Mostrar Calendario	X	
34	Envío de información	X	
35	La respuesta deben ser los horarios y pistas de la semana concreta, correspondiente con la fecha seleccionada	X	
36	Cabecera con el día de la semana y la fecha correspondiente	X	
37	Mostrar intervalo de apertura de la pista	X	
38	Mostrar intervalo de apertura de la pista para actividades deportivas	X	
39	Mostrar intervalo de pista libre (RESERVABLE)	X	
40	Mostrar intervalo de pista ocupada.	X	
41	Mostrar intervalo de pista NO RESERVABLE	X	
42	Mostrar dichos intervalos por tonalidades	X	
43	Que dichas consultas correspondan a las pistas y fechas seleccionadas	X	
<b>Acciones sobre los datos delegados de la competición</b>			
44	Función del desplegable con los tres campus	X	
45	Mostrar información de todos los delegados de un deporte y campus concreto	X	
46	Mostrar página de error en la consulta de la información de delegados que no le corresponde.	X	
<b>Acciones sobre Aplazamientos</b>			
47	Formulario correspondiente a cumplimentar	X	
48	Que se rellenen los campos obligatorios	X	
49	Comprobación de errores de inserción	X	

CAPÍTULO 8. PRUEBAS

50	Comprobación de errores al intentar aplazar un partido cuando no tiene permisos.	X	
<b>Acciones sobre el envío del formulario</b>			
51	Que el asunto del mail se rellene correctamente con los campos introducidos en el formulario	X	
52	Que el texto descriptivo del correo se corresponda con la información introducida en el formulario	X	
53	La dirección origen corresponda con la dirección de correo electrónico del usuario, obtenida de la autenticación	X	
54	La dirección destino corresponde con la del administrador	X	
55	Envío realizado sobre smtp.uc3m.es	X	
56	Se transmite por el puerto correspondiente	X	
57	Comprobación de transmisión correcta o errónea	X	
<b>PERFIL ADMINISTRADOR</b>			
58	Aparecen todas las opciones que se pueden realizar desde el perfil administrador.	X	
<b>Acciones sobre Confirmar Resultados</b>			
59	Se muestre al administrador tantos links como resultados tiene por confirmar	X	
60	Modificación de los campos introducidos por el delegado	X	
61	Confirmar resultado	X	
62	Eliminar resultado	X	
63	Cancelar resultado	X	
64	Modificación de la tabla "Partidos"	X	
65	Modificación de la tabla "Clasificación"	X	
66	Modificación de la tabla "Resultados"	X	
<b>PRUEBAS LIGA Y TORNEOS</b>			
<b>Acciones sobre la Liga</b>			
67	Mostrar deportes de liga	X	
68	Mostrar equipos inscritos en esa liga	X	
69	Consultas y modificaciones de las tablas correspondientes a la liga	X	

---

<b>Acciones sobre la Copa</b>			
70	Mostrar deportes de copa		X
71	Mostrar equipos inscritos en esa copa		X
72	Consultas y modificación de las tablas correspondientes a la copa		X

Tabla 8.1: Pruebas Realizadas.



# Planificación y Presupuesto

---

En este capítulo se detallan las fases que se han llevado a cabo desde el comienzo del desarrollo del Proyecto hasta la finalización del mismo y el coste total de la realización del proyecto.

## 9.1. Planificación

Para planificar la duración del proyecto se ha utilizado el programa Microsoft Project.

Se divide el proyecto en tareas. Para cada tarea, se indica el coste de realizarla medido en días. Cada día se dedica una media de 5 horas.

En la figura 9.1 se muestra un Diagrama de Gantt con las tareas del proyecto, indicando las predecesoras y la duración (fecha inicial y final) de cada una de ellas.

En la figura 9.2 se muestran esas tareas en forma de barras indicando la duración correspondiente.

## 9.2. Presupuesto

Como se puede visualizar en el Diagrama de Gantt de la figura 9.1 se han necesitado 259 días para el desarrollo de la aplicación.

En el cálculo del presupuesto, se tiene en cuenta que cada día se dedica una media de 5 horas y que el coste por hora de un técnico son 20 euros. En la Tabla 9.1 se puede visualizar el coste total del proyecto. Se han agrupada las tareas indicadas en el Diagrama de Gantt en las siguientes fases:

## CAPÍTULO 9. PLANIFICACIÓN Y PRESUPUESTO

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	
Diagrama de Gantt	1	Estudio de las alternativas	20 días	lun 01/09/08	vie 26/09/08	
	2	Estudio de las herramientas a utilizar	20 días	lun 29/09/08	vie 24/10/08	1
	3	Toma de requisitos	15 días	lun 27/10/08	vie 14/11/08	2
	4	Diseño inicial de la herramienta a desarrollar	20 días	lun 17/11/08	vie 12/12/08	3
	5	Documentación de la BBDD	25 días	lun 15/12/08	vie 16/01/09	4
	6	Validación por LDAP	6 días	lun 19/01/09	lun 26/01/09	5
	7	Creación del perfil Usuario con sus opciones	22 días	lun 02/02/09	mar 03/03/09	6
	8	Creación del perfil delegado con sus opciones	28 días	mié 04/03/09	vie 10/04/09	7
	9	Creación del perfil Administrador con sus opciones	14 días	lun 13/04/09	jue 30/04/09	8
	10	Estudio de los requisitos de las ampliaciones	6 días	lun 15/06/09	lun 22/06/09	9
	11	Estudio de la BBDD sobre las ampliaciones	4 días	mar 23/06/09	vie 26/06/09	10
	12	Implementación de las ampliaciones	30 días	lun 29/06/09	vie 07/08/09	11
	13	Pruebas	5 días	lun 10/08/09	vie 14/08/09	12
	14	Estilo de la aplicación	5 días	lun 17/08/09	vie 21/08/09	13
	15	Estudio y documentación sobre LATEX	3 días	lun 24/08/09	mié 26/08/09	14
	16	Documentación	27 días	jue 27/08/09	vie 02/10/09	15

Figura 9.1: Listado de tareas

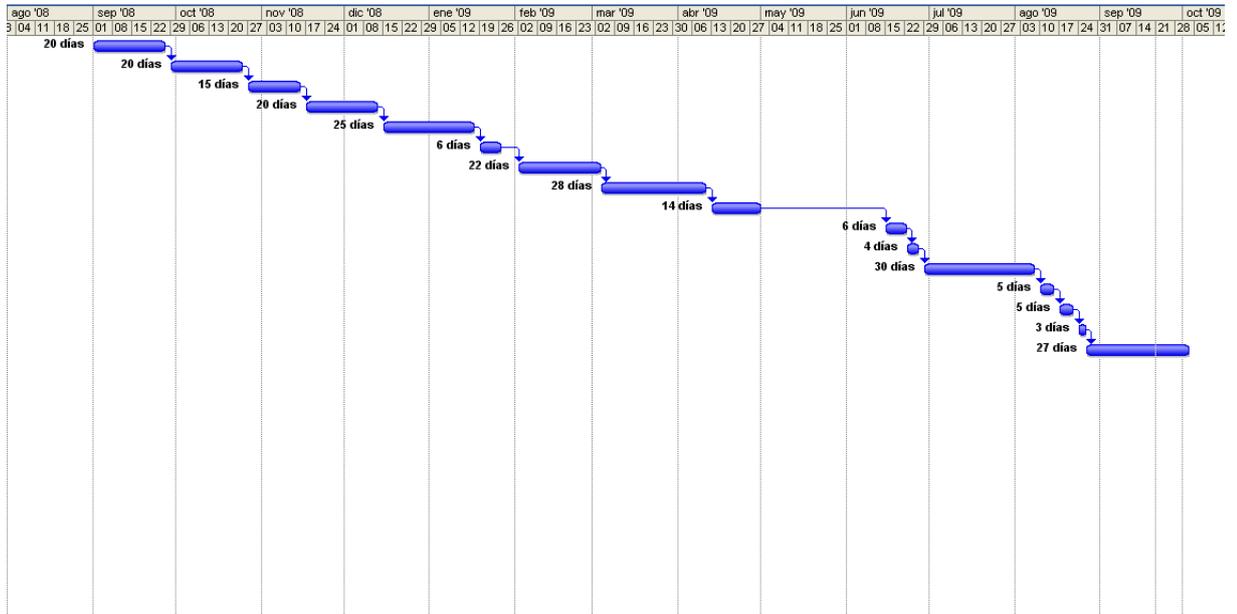


Figura 9.2: Tareas representadas en barras

Descripción de la Fase	Coste en días	Coste en horas	Coste en euros
<b>COSTES INDIVIDUALES</b>			
Estudio	40	200	4000
Análisis	60	300	6000
Implementación	110	550	11000
Pruebas	10	50	1000
Documentación	30	150	3000
<b>COSTES TOTALES</b>			
Coste Desarrollo	250	1250	25000

Tabla 9.1: Presupuesto



## Conclusiones y Trabajos Futuros

---

En este capítulo se realiza un balance del proyecto implementado, analizando los objetivos, las ventajas que ofrece y los puntos débiles que se han encontrado. Aquí también se realiza un estudio sobre algunas líneas de trabajos futuros que se pueden desarrollar siguiendo el hilo de esta aplicación con el fin de hacerlo más completo.

### 10.1. Conclusiones

A continuación se recoge un balance del proyecto implementado, analizando los objetivos logrados y marcados, así como ventajas y debilidades del sistema. En el primer capítulo de la memoria se exponen los objetivos marcados al inicio de este proyecto. Una vez que se ha desarrollado la aplicación se analizan estos objetivos uno a uno con el fin de comprobar si se han cumplido o no.

- *El objetivo principal es que el usuario pueda realizar en tiempo real consultas relacionadas con las actividades deportivas vía Web.*

El objetivo principal de la aplicación ha sido cumplido con éxito, ya que se ha conseguido una herramienta que permite realizar una serie de acciones sobre las actividades deportivas del departamento de «Actividades Culturales y Deportivas» de la UC3M.

- *La aplicación debe proporcionar a los usuarios las opciones de consultar las clasificaciones, resultados, deportividad y anotadores/goleadores de todos los equipos, así como la posibilidad de interactuar con el administrador para insertar resultados.*

## CAPÍTULO 10. CONCLUSIONES Y TRABAJOS FUTUROS

---

Este objetivo también ha sido cumplido, incluyendo una serie de ampliaciones. Ya que, a parte de proporcionar las opciones de consulta nombradas anteriormente, proporciona en el perfil delegado las opciones de insertar resultados y contactar. Y en el perfil administrador la opción de confirmar. (Estas ampliaciones son nombradas en el *Capítulo 3: Requisitos, apartado 3.2* y explicadas en el *Capítulo 4: Diseño de la Interfaz, apartados 4.3, 4.4 y 4.5*).

- *El siguiente objetivo es crear distintos perfiles con distintas funcionalidades según el tipo de usuario: jugador/delegado del equipo o administrador de las actividades deportivas.*

La aplicación desarrollada se encuentra orientada a tres tipos de perfiles. La clasificación entre estos perfiles se realiza basándose en los tipos de permisos de cada usuario. La información que se presenta a los usuarios se encuentra adaptada al tipo de perfil que presenta.

- *El objetivo gráfico es mostrar una interfaz de usabilidad sencilla que permita al usuario navegar sin dificultad por ella. Se presupone que el usuario no tiene que tener conocimiento acerca de la tecnología utilizada.*

Para conseguir este objetivo se presenta al usuario una interfaz simple y trivial que al usarla no dé problemas ni ofrezca grandes retos. Además, debe ser amigable para él y fácil de comprender. Su diseño se ha realizado pensando que los usuarios no van a leer manuales de la herramienta. Por lo tanto, se debía crear una interfaz sencilla, como ha sido el caso.

- *Ofrecer seguridad para los usuario mediante validación.*

La validación se realiza frente al servidor LDAP de la UC3M, esto significa que toda persona externa a la universidad no tendrá acceso a la entrada de la aplicación.

Otra restricción de seguridad es la llevada a cabo para identificar al usuario y asignarle su correspondiente perfil. De esta manera se controla las acciones que puede realizar cada tipo de usuario.

- *Finalmente, que la visualización sea correcta en el navegador utilizado y en los distintos dispositivos portátiles. Cumplir unas pautas básicas de accesibilidad.*

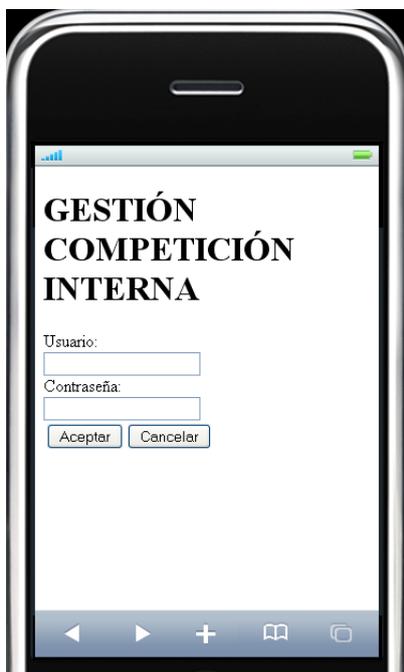


Figura 10.1: Simulador virtual. Página de Autenticación.

Este objetivo se ha llevado a cabo con éxito mediante el validador W3C. La herramienta se visualiza correctamente en distintos navegadores (IE, Opera, Firefox...), después de añadir ciertas reglas para que funcione con cada uno de ellos.

La visualización en dispositivos portátiles se ha generado de manera mucho más sencilla, evitando incluir información no relevante, como iconos, o colores, y mostrando únicamente la información importante para el usuario (Como se puede ver en las figuras 10.1 y 10.2).

Finalmente, se han cumplido unas pautas básicas de accesibilidad llegando a cumplir el nivel AAA con los validadores HERA y TAW.

La aplicación ofrece muchas **ventajas**, a continuación se presentan las más relevantes:

- Una de las ventajas más significativas de la aplicación es la facilidad con la que un usuario puede consultar la información relacionada con las Actividades Deportivas.
- Otra ventaja es la comunicación entre los usuarios con perfil de delegado y el administrador, inexistente actualmente vía Web.



Jornada	Fecha	Hora	Equipo Local	Puntos Local
1	28-10-2008	13:00	L'Squadra Cappezzolo	20
1	28-10-2008	14:00	You'll never Drink alone	50
1	28-10-2008	15:00	SOUL TEAM	99
1	29-10-2008	14:00	L.M.M.	22
1	29-10-2008	15:00	Toronto Nabos	21

Figura 10.2: Simulador virtual. Página de Resultados.

- Se ofrece a los usuarios una mayor participación en todo lo relacionado con su actividad deportiva.
- Otra de las ventajas que ofrece es que al ser una aplicación vía Web sólo se necesita de un navegador y conexión a Internet para poder utilizarla. Los usuarios pueden beneficiarse de ello en cualquier franja horaria y en cualquier parte del mundo.
- El usuario puede fijar sus propios ritmos de aprendizaje, según el tiempo que disponga y los objetivos que se haya fijado.
- Ofrece un sistema conjunto, tanto para usuarios, delegados y administrador, en el que se puede consultar toda la información que se encuentra almacenada correspondiente a cada uno de los deportes.

### **Principales inconvenientes:**

- El administrador deberá seguir el formato que hay actualmente para insertar la nueva información del año académico.
- La Base de Datos no podrá ser modificada sin necesidad de retocar el código.

## 10.2. Trabajos Futuros

Durante el desarrollo del trabajo han ido surgiendo ideas y mejoras que se pueden plantear como posibles trabajos futuros como continuación de la línea de este proyecto. A continuación se identifican algunos de ellos:

**1. Realización de las mismas funcionalidades con las competiciones de copa: Torneos.**

La aplicación presentada en esta memoria es válida, únicamente, para las competiciones de LIGA. Sería interesante que se le diese al usuario la opción de elegir entre competición de liga o copa, y tuviese la misma funcionalidad para ambas.

Esta tarea, actualmente, está implementada en el proyecto. No se ha podido llevar a cabo debido a que la BBDD no está preparada para el tratamiento de torneos.

En un futuro, si se desea implementar esta opción, únicamente deberán insertar los valores siguiendo el mismo formato que en la competición de LIGA. Y en la parte de la visualización, descomentar el código relacionado con los torneos.

**2. Externalizar la aplicación para que pueda ser usada en competiciones interuniversitarias.**

Con esta propuesta y la anterior, estaría cubierto el conjunto de actividades deportivas que ofrece el Espacio de Estudiantes. Ya que no solo existiría la competición interna, si no también la externa.

Las ventajas serían que, aparte de existir una comunicación con los jugadores/delegados pertenecientes a nuestra universidad, aparecería la comunicación con el resto de universidades que participan en competiciones deportivas.

**3. Comunicación mediante chat.**

Se propone la realización de un chat, para que los jugadores y delegados puedan comunicarse a la hora de consultar las pistas disponibles y aplazar un partido. Esta sería una manera rápida y divertida de mantener comunicación relacionada con las Actividades Deportivas.

**4. Ampliar los permisos a los delegados para reservar directamente una pista.**

Actualmente, la aplicación sólo deja consultar la disponibilidad de las pistas. Sería interesante proporcionar más permisos a los delegados para que tuviesen la libertad de consultar una pista libre y reservarla en el momento.

### **5. Adaptación del estilo CSS a los cánones marcados por la universidad.**

En un futuro, el departamento podría adaptar la aplicación a los cánones de estilo marcados por la universidad.

Actualmente, la página Web que presta este servicio no sigue esos cánones. Por lo tanto, es algo opcional por parte del Departamento del Espacio de Estudiantes.

### **6. Recomendaciones de implantación.**

Con el fin de ofrecer una amplia cobertura a todos los usuarios de las actividades deportivas, se debería implantar esta aplicación en un servidor de la UC3M.

Después de la implantación de la aplicación, sería interesante hacer una autoevaluación por parte de los usuarios, para que den su opinión sobre la misma y las cosas que echan en falta. Así como realizar un FAQ (Preguntas Frecuentes).

### **7. Internacionalización.**

La internacionalización es el proceso de diseñar software de manera tal que pueda adaptarse a diferentes idiomas y regiones sin necesidad de cambios de ingeniería ni de código.

Se propone internacionalizar la aplicación para que pueda ser utilizada en diferentes idiomas, sería útil para los alumnos erasmus que estuviesen apuntados en alguna actividad deportiva.

## Anexo Base de datos: DeporWin.

---

### A.1. Tablas vacías.

- Activación campañas.
- Agrupaciones Deportivas.
- Calendario.
- Campaña.
- Clasificación.
- Cuadros.
- Datos Contacto.
- Entidades Deportivas.
- Instalaciones Deportivas.

### A.2. Tablas a modificar y motivos.

- **Campus:** La tabla campus está compuesta de los tres campus de los que consta la UC3M, pero sus identificadores principales no corresponden con las relaciones entre el resto de tablas. La información correcta de campus está en la tabla «Recursos».
- **Clasificación:** La clasificación de un deporte debería aparecer en esta tabla, actualmente está en una tabla llamada «TmpClasifLiga». Es una tabla temporal.

- **Calendario:** En esta tabla podría aparecer el calendario de las competiciones. Actualmente este calendario está en la tabla «Partidos», y a la hora de insertar un partido directamente se actualiza la tabla partidos en vez de insertarse una fila con el resultado, una vez cumplimentado el acta.
- **Cuadros:** En esta tabla podría aparecer el calendario de las competiciones. De la misma manera que en Calendario.
- **Datos Contacto:** Esta tabla se podría rellenar con los datos de cada jugador/usuario, información telefónica, email, etc. Actualmente esta información está ubicada entre las tablas Inscritos y Personas.
- **Deportes:** En esta tabla podría venir definido de alguna forma los deportes correspondientes a liga o a copa.
- **Horarios Campeonatos:** En esta tabla no aparece información relevante aunque sería interesante que apareciese la información relacionada con los horarios que tiene cada campeonato con cada una de las pistas.
- **Horarios RecursosAbsolutos:** Horario que representa el tiempo que está abierta la pista. Sin embargo, este horario no corresponde con el real. El horario de aperturas de las pistas en realidad se encuentra en la tabla «Recursos».
- **Horarios RecursosSemanales:** Horario en que las pistas están abiertas exclusivamente para uso deportivo.
- **Instalaciones Deportivas:** Esta tabla está actualmente vacía, las pistas e instalaciones deportivas deberían estar en esta tabla, actualmente se encuentran en la tabla «Recursos».
- **Participantes:** Tabla obsoleta.
- **Partidos:** En esta tabla está el calendario de cada uno de los partidos, aunque en realidad debería estar en la tabla calendario. En la tabla Partidos únicamente debería aparecer la información cuando el administrador insertase el resultado final del partido una vez teniendo el acta en la mano.
- **Personas:** La tabla personas debería tener relación con la tabla «datos contacto», a parte del «Integrante».
- **Plantilla:** Tabla innecesaria porque no guarda ningún tipo de relación con los horarios relacionados con el resto de tablas.

## A.2. TABLAS A MODIFICAR Y MOTIVOS.

---

- **Plantilla horaria:** Vacía. Debería incluir los horarios absolutos que se encuentran en la tabla «Recursos».
- **Recursos:** Esta tabla debería ser modificada ya que contiene toda la información de todos los campus, todas las pistas, todas las instalaciones. Debería estar en distintas tablas ordenado de manera jerárquica, y no haciendo referencia de unas instalaciones con otras de cada uno de los campus a través de una columna que relaciona el identificador de la pista con un identificador padre, que corresponde con una fila de esta misma tabla.
- **Tablas Temporales:** Correspondientes a todas las que aparecen como tmp, deberían ser tablas de pruebas, pero en realidad se utilizan en la realidad.



## Entorno de desarrollo utilizado

---

En este capítulo se presenta el entorno en el que se ha desarrollado la aplicación, explicando las herramientas utilizadas y sus características más importantes.

Con el objetivo de trabajar simultáneamente con la BBDD SQLServer, el lenguaje de programación Java y un servidor local que nos muestre las páginas Web generadas por la aplicación (el servidor elegido por excelencia para este tipo de aplicaciones es Apache Tomcat) se decidió trabajar con el entorno ECLIPSE, combinando la visualización de la BBDD con el software de TOAD.

A continuación se explican ambos entornos:

### B.1. ECLIPSE

Eclipse es un entorno de desarrollo integrado(IDE) que logra proporcionar un conjunto de herramientas para el programador dentro de un único programa, es decir, las funcionalidades están incluidas, las necesite el usuario o no.

La propia definición que da el proyecto sobre «Eclipse» acerca de su software es: «*Una especie de herramienta universal, un IDE abierto y extensible para todo y nada en particular*». Figura B.1

#### 1. Requisitos



Figura B.1: Logotipo Eclipse

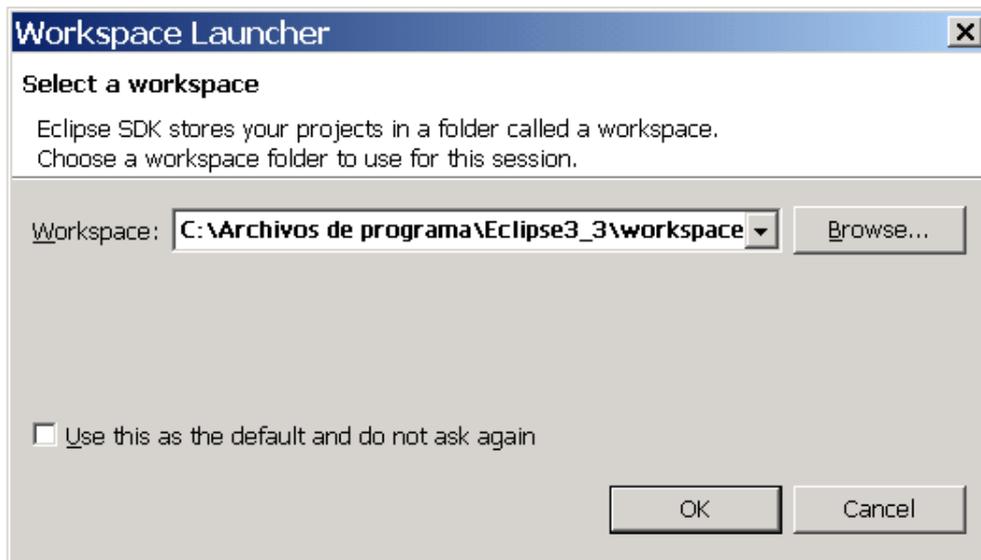


Figura B.2: Ruta de creación del proyecto.

Eclipse únicamente requiere la descarga de un archivo .zip y la ejecución del .exe. A continuación se pasa a configurar el software para nuestra aplicación concreta:

- a) Se elije la ruta donde se guardarán los proyectos que se creen. Figura B.2
- b) Se elije el lenguaje de programación. Figura B.3

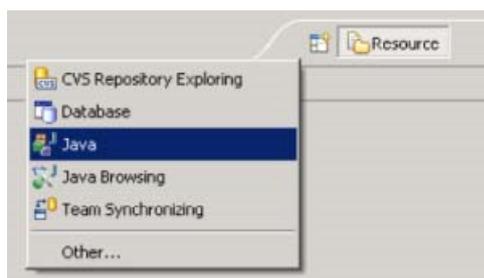


Figura B.3: Lenguaje de Programación

- c) Se crea un proyecto Dynamic Web. Figura B.4
- d) Configuración del servidor. Figura B.5
- e) Creación de clases y paquetes. Figura B.6

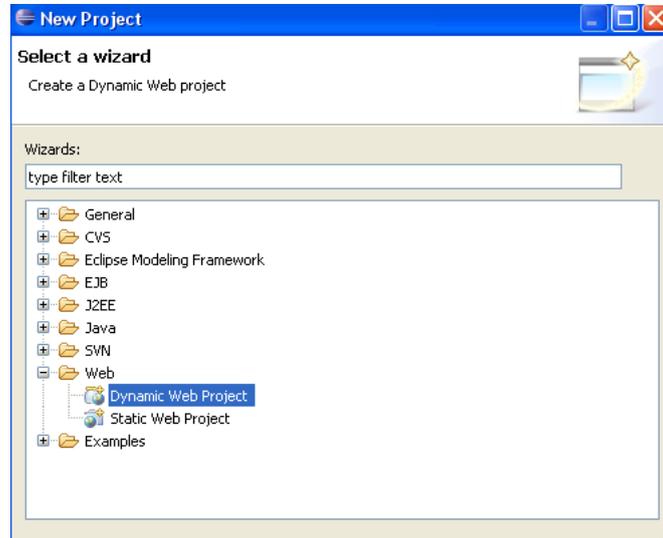


Figura B.4: Creación de Proyecto

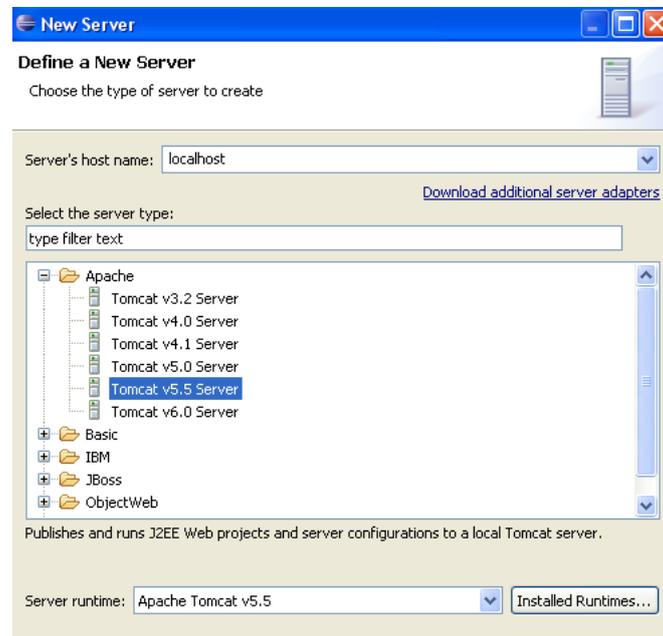


Figura B.5: Configuración Servidor

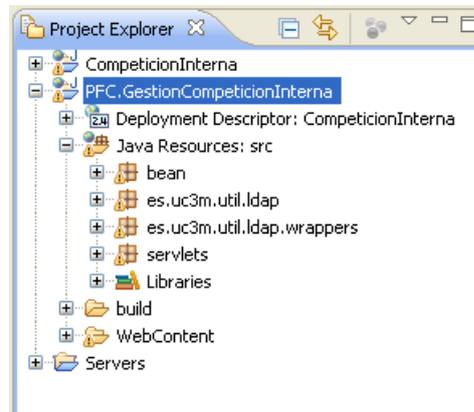


Figura B.6: Clases y Paquetes que contiene el proyecto

## 2. Características

Eclipse ofrece un entorno de desarrollo sencillo de usar, con diversas funcionalidades en un único programa:

- a) Distintos lenguajes de programación.
- b) Editor de texto con resaltado de sintaxis.
- c) Intérprete.
- d) Herramientas de automatización.
- e) Depurador.
- f) Compilación en tiempo real.
- g) Ofrece control de versiones.
- h) Construcción de interfaces gráficas de usuarios.
- i) Procesado de textos en LaTeX.
- j) Aplicaciones en red (Telnet).
- k) Sistemas de gestión de BBDD.

Desde el punto de vista de las aplicaciones clientes, Eclipse provee al programador con frameworks para el desarrollo de aplicaciones gráficas así como aplicaciones Web.

A parte de todo esto, eclipse facilita la instalación de plugins para el desarrollo de editores visuales, editores de diagramas UML, interfaces gráficas (GUI), etc.

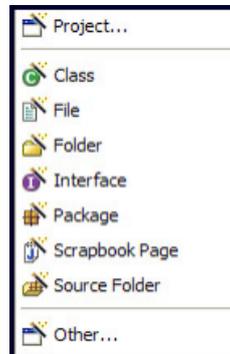


Figura B.7: Elementos Java en Eclipse

EL SDK de Eclipse incluye herramientas de desarrollo Java, ofreciéndose un compilador de Java interno y permitiendo, de esta manera, técnicas de compilación y análisis de código avanzadas (opción debug).

### 3. Funcionamiento

Una vez que la herramienta ha sido instalada, se comienza con la creación del proyecto.

Como se ha comentado anteriormente, un proyecto está formado por paquetes, clases u otro tipo de elementos Java. La creación de estos se puede observar en la figura B.7.

Normalmente, se declaran tantos paquetes como información a almacenar se disponga.

A continuación se crean tantas clases como sean necesarias, como indica la figura B.8

La carpeta fuente especificada debería ser la carpeta definida como «src». Si no se especifica ningún paquete para contener las clases Java, se guardarán dentro de un paquete por defecto. El último campo obligatorio que debería ser rellenado antes de proceder a la creación de la clase Java es el propio nombre de la clase.

Si se desea que la nueva clase contenga un método "main"(es decir, el punto inicial de ejecución del programa), puede añadirse dicho método automáticamente sólo con marcar la casilla con la opción apropiada. También pueden implementarse de esta manera los constructores de la superclase y todos los métodos abstractos heredados.

Una vez que se han creado los paquetes y clases necesarias, el usuario puede comenzar a programar.

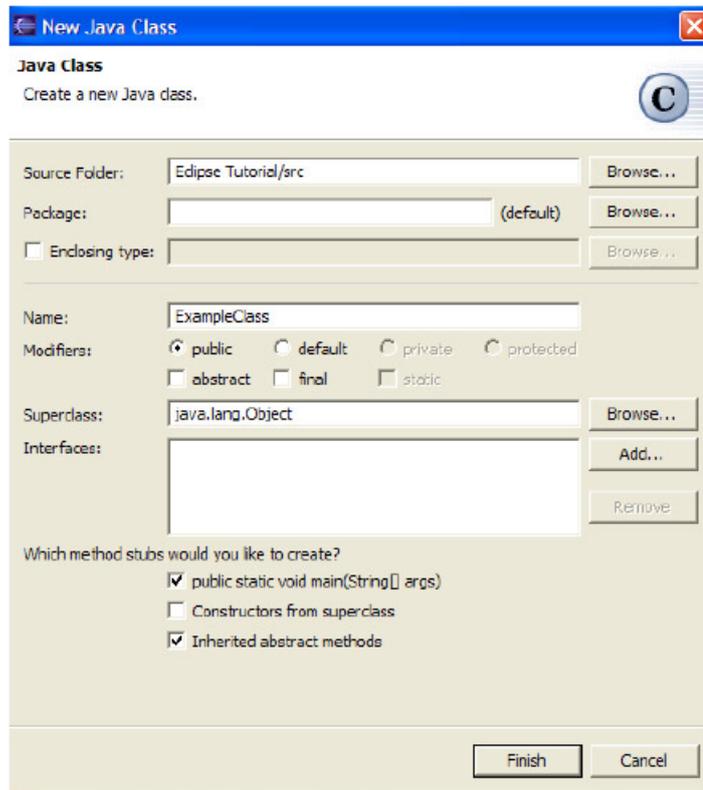


Figura B.8: Creación de una clase con eclipse

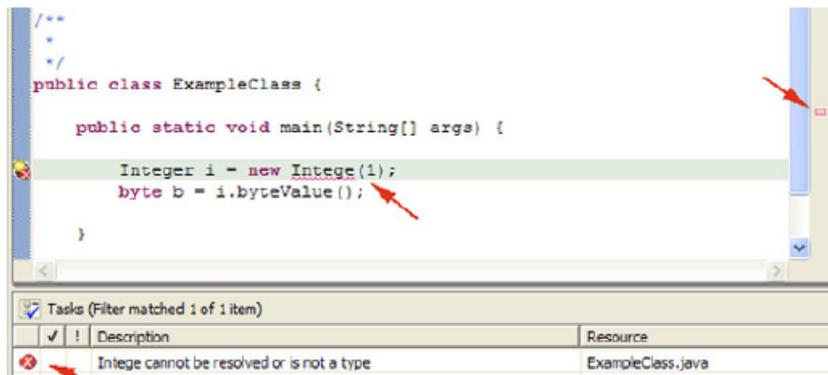


Figura B.9: Detección de errores

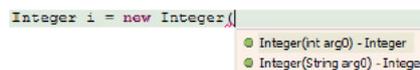


Figura B.10: Autocorrección del eclipse

Las ventajas que ofrece Eclipse a la hora de programar son numerosas:

- Compila y detecta errores a tiempo real. Subraya el fragmento de código erróneo con una línea roja (Figura B.9)
- Autocorregir.
- Autocompletar. Figura B.10
- Genera automáticamente los get y set.
- Consulta la documentación.
- Importa archivos JAR.

Cuando el programa de Java esté completado será hora de ejecutarlo y probarlo.

Para ejecutar un programa dentro de Eclipse hay que seleccionar «Run» del menú principal.

Hay cuatro tipos de configuraciones de ejecución (Figura B.11):

- a) Java Applet (para applets web)
- b) Java Application (para programas normales de Java)
- c) JUnit (casos de prueba)
- d) Run-Time Workbench (otras instancias de Eclipse que permiten probar nuevos módulos de Eclipse)



Figura B.11: Tipos de ejecución

Así pues, para ejecutar un programa de Java normal debería seleccionarse «Java Application» y pulsar el botón «New» para crear una nueva configuración. Dentro de la pestaña «Main» hay que dar nombre a la nueva configuración seleccionar el proyecto que contiene la clase con el método main y seleccionar dicha clase.

Finalmente, hacer clic en el botón «Run» lanzará la ejecución del programa seleccionado.

Si en la ejecución surgiesen errores, Eclipse da la opción de depurar el código (Debug).

Se abre la perspectiva de depuración «Debug», haciendo clic en el margen izquierdo del editor de código aparecerá un menú contextual. Seleccionando «Add/Remove Breakpoint» añadirá o quitará un punto de ruptura, mientras que «Toggle Breakpoint» cambiará el estado de activación del punto de ruptura. Los puntos de ruptura marcan líneas en que la ejecución del programa se detendrá de manera que sea posible comprobar el valor de las variables en ese instante, identificando así posibles errores.

## B.2. TOAD

TOAD es una aplicación de software de desarrollo SQL y administración de bases de datos, considerada útil para los administradores de BBDD ya que tiene una visualización muy sencilla.

Actualmente está disponible para las siguientes BBDD: Oracle Database, Microsoft SQLServer, IBM DB2 y MySQL.

### 1. Requisitos

TOAD es un software fácil de instalar y manejar. El único requisito que se necesita es tener conexión a Internet desde donde se ejecute el programa.

Pasos a seguir para descargar, instalar y configurar el software de TOAD:

a) Descargar el software de la página

[http://www.toadsoft.com/toadsqlserver/toad\\_sqlserver.htm](http://www.toadsoft.com/toadsqlserver/toad_sqlserver.htm)

b) Descomprimir e instalar en el PC.

c) Configuración:

- Server Name →Nombre del servidor al que nos queremos conectar.
- Authentication →Autenticación del servidor.
- Login →Usuario que va a conectarse.
- Password →Contraseña del usuario.
- Database →Por defecto.

Con esto, se quedaría finalizada la configuración del software de TOAD.

## 2. Características

Las principales características de TOAD son:

- Rapidez en la generación de consultas.
- Permite crear y modificar objetos de la BBDD.
- Permite desarrollar y depurar código SQL y PL/SQL.
- Permite la importación/exportación de datos, comparación de esquemas y actualización de estadísticas.
- Ofrece un navegador de esquemas que permite manejar y visualizar todo el diccionario de datos.
- Realizando un click en un objeto se obtiene toda la información detallada del mismo.
- El editor de Toad mejoran la productividad, eliminando errores y reduciendo el tiempo de desarrollo.

## 3. Funcionamiento

Una vez que el programa ha sido instalado, se procede a la conexión con la BBDD.

Como se puede observar en la Figura B.12, el usuario debe seleccionar el nombre del servidor al que se desea conectar e introducir el login y password con el que se va a conectar.

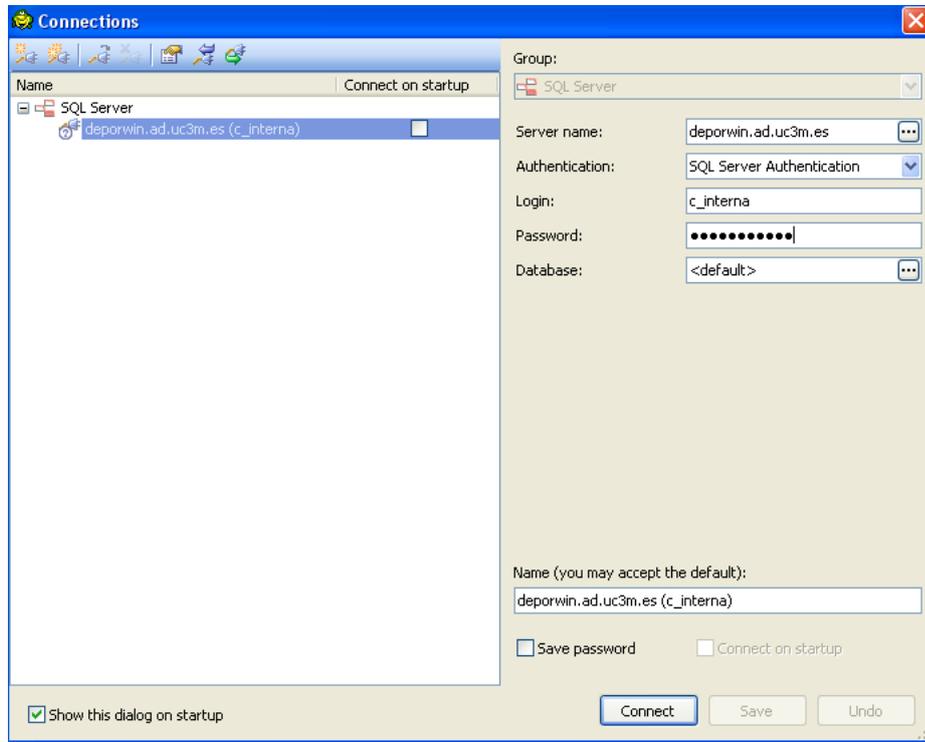


Figura B.12: Conexión con la BBDD

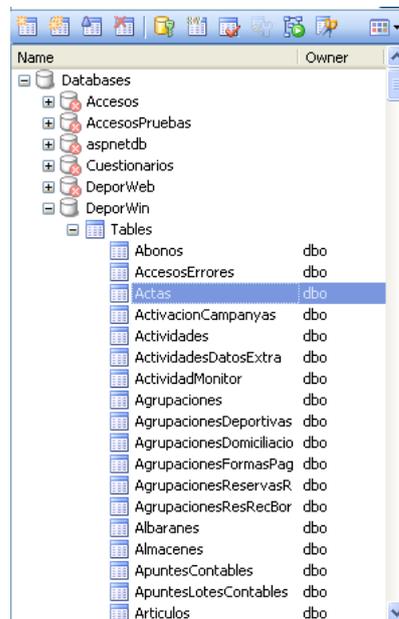


Figura B.13: Navegador TOAD

Toad-For-SQL-Server - S... Flash Player Installation Editor Untitled1 Viewer Table DeporWin.dbo.Actas

Properties Columns **Data** Statistics Indexes Foreign Keys Check Constraints Triggers Uses Used By Permissions

IdPartido	IdEquipo	IdJugador	Dorsal	Puntos	TipoSancion	CantidadSancion
2	1	2057	{null}	1	{null}	{null}
14	1	2057	{null}	40	{null}	{null}
14	1	9036	{null}	20	{null}	{null}
1000	257	23483	{null}	17	{null}	{null}
977	103	20308	{null}	{null}	1	0
972	103	20308	{null}	{null}	1	6
1457	285	22108	{null}	3	{null}	{null}
1457	285	28312	{null}	2	{null}	{null}
1457	336	33039	{null}	1	{null}	{null}
1457	336	36494	{null}	1	{null}	{null}
1116	303	8916	{null}	28	{null}	{null}
1108	301	36931	{null}	4	{null}	{null}
1112	298	15520	{null}	23	{null}	{null}
1774	436	37492	{null}	3	{null}	{null}
1772	433	35021	{null}	3	{null}	{null}
1720	417	34810	{null}	1	{null}	{null}
1221	316	35713	{null}	1	{null}	{null}

Figura B.14: Información detallada de una tabla

## APÉNDICE B. ENTORNO DE DESARROLLO UTILIZADO

---

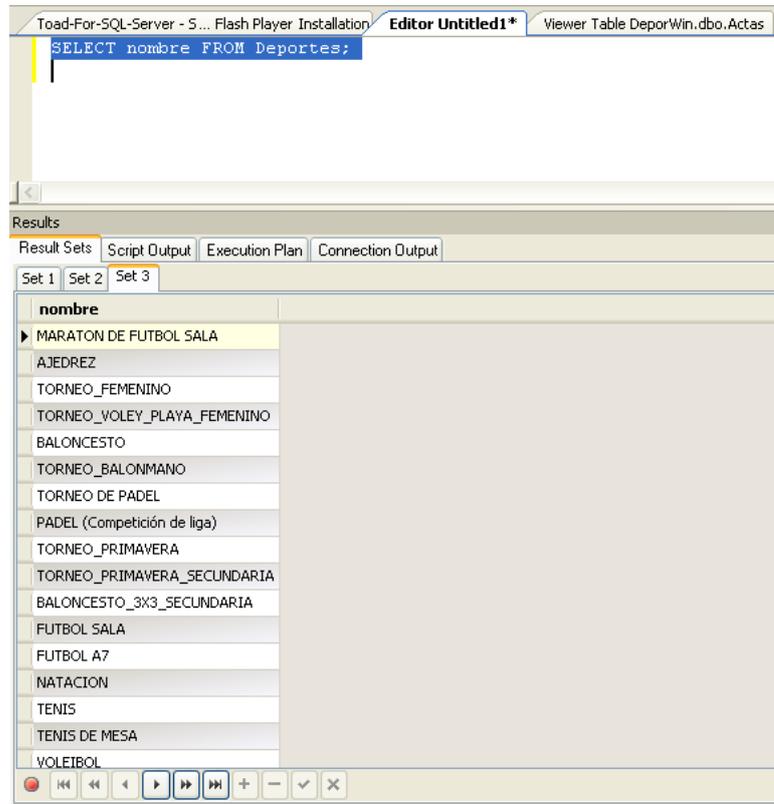


Figura B.15: Consulta y Resultados

Una vez realizada la validación, aparece un navegador (figura B.13) con el fin de que el usuario pueda desplazarse por las tablas y los objetos obteniendo información detallada del mismo (figura B.14).

Para finalizar, se muestra un ejemplo de búsqueda (figura B.15).



# Manual de instalación del entorno utilizado

---

En este capítulo se presenta el manual de instalación del entorno utilizado, con el fin de proporcionar al lector un resumen de como instalar las aplicaciones que han sido necesarias para la realización del proyecto.

## C.1. CONEXIÓN REALIZADA FUERA DE LA UNIVERSIDAD

### C.1.1. Conexión VPN

El objetivo de realizar una conexión VPN se debe a la necesidad de trabajar con una dirección IP de la UC3M para poder realizar la validación LDAP, permitiéndose así la autenticación de los usuarios, y conectarse con el servidor de la BBDD que proporciona el departamento del SIJA, para obtener la información necesaria que se necesita en la aplicación.

Se crea un puente VPN con la UC3M haciendo uso de una guía que proporciona la página de la UC3M, y para que funcione perfectamente se deben añadir ciertas restricciones al antivirus instalado en el PC.

Todo ello viene especificado en la siguiente página: <http://asyc.uc3m.es/index.php?Id=169>

La conexión a la VPN se realiza conectándose al servidor `vpn.uc3m.es` e incluyendo como contraseña el password que el usuario utiliza en el ámbito de la universidad. (Figura C.1).



Figura C.1: Conexión a la VPN

## C.2. CONEXIÓN REALIZADA DENTRO DE LA UNIVERSIDAD

### C.2.1. Instalación JAVA

El lenguaje de programación utilizado en el proyecto es Java, para que su uso sea correcto se deben seguir los siguiente pasos:

1. Descarga software Java. [www.java.com/es/download](http://www.java.com/es/download)
2. Instalación software. La instalación se guarda normalmente en la ruta:

`C:\Archivos de Programa\Java\jre 1.6.0_07`

y en

`C:\j2sdk1.4.2_18`

3. Añadir Variables de Entorno

Se debe añadir la ruta donde se ha instalado el software de Java (las dos rutas anteriormente nombradas) en las variables de entorno del PC. (Figura C.2)

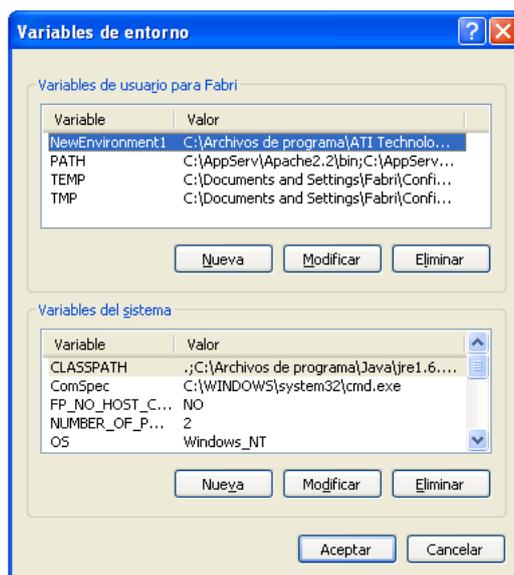


Figura C.2: Modificación de las variables de entorno

### C.2.2. Instalación TOMCAT

Se debe descargar el software de la página: [tomcat.apache.org/download-55.cgi](http://tomcat.apache.org/download-55.cgi)  
Se guardan los ficheros de configuración (normalmente en

C:\Archivos de Programa\Apache Software Foundation\Tomcat 5.5

) y se realizan las configuraciones para que funcione en HTTP/1.1 modificando el puerto en el fichero server.xml

Configuraciones:

```
HTTP/1.1
Connector Port: 8080
User Name: admin
Password: admin
```

### C.2.3. Instalación ECLIPSE

- Inicialmente se debe descargar el software de la página: <http://www.eclipse.org/downloads/>
- Se descomprime en la carpeta donde se quiera tener instalado, normalmente en la ruta:

## APÉNDICE C. MANUAL DE INSTALACIÓN DEL ENTORNO UTILIZADO

---

C:\Archivos de Programa\Eclipse

- Se arranca ejecutando Eclipse.exe, y se pide la ruta por defecto donde queremos que Eclipse vaya guardando los proyectos que vamos creando.

Estos proyectos se guardan en una carpeta llamada workspace dentro de la ruta donde se ha instalado el Eclipse.

- Configuración del Eclipse:

1. Creación de un nuevo proyecto en Java:

```
File/New/Project  
--> Dynamic Web Project
```

2. WebContent: Carpeta donde se guardan los ficheros html y las librerías necesarias para el proyecto.

```
WebContent  
--> demo.html  
--> +META-INF  
--> +WEB-INF
```

3. Configuración del servidor.

- a) Dirigirse a la pestaña de servidores.
- b) Definir un nuevo servidor
- c) Seleccionar Apache Tomcat v5.5 server.
- d) Se seleccionan los proyectos a ejecutar.

### C.2.4. Instalación TOAD

- Inicialmente se debe descargar el software de la siguiente página [www.toadsoft.com/](http://www.toadsoft.com/)

- Configuración de TOAD:

1. Insertar LOGIN

2. Insertar PASSWORD

3. Datos del servidor

- Dirección IP
- Nombre del servidor
- Sistema que gestiona la BBDD →Sql Server.

## Glosario

---

- **ASCII:** American Standard Code for Information Interchange. Código de caracteres basado en el alfabeto latino.
- **ASP:** Active Server Pages. Tecnología de Microsoft para el lado del servidor para páginas web generadas dinámicamente.
- **BBDD:** Base de Datos. Conjunto de datos pertenecientes a un mismo contexto y almacenados para su posterior uso.
- **CERN:** Organización Europea para la Investigación Nuclear.
- **CGI:** Common Gateway Interface. Tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor Web.
- **CSS:** Cascading Style Sheets. lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.
- **DDL:** Data Definition Language. Un lenguaje de definición de datos es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.
- **DML:** Data Manipulation Language. Lenguaje de Manipulación de Datos proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

- **DNS:** Domain Name System. Un servidor DNS permite conectarse con la máquina sin necesidad de usar su dirección IP, basta con ingresar el dominio para que el servidor DNS resuelva y establezca una conexión.
- **DTD:** Document Type Definition. Su función básica es la descripción del formato de datos, para usar un formato común y mantener la consistencia entre todos los documentos que utilicen la misma DTD. De esta forma, dichos documentos, pueden ser validados, conocen la estructura de los elementos y la descripción de los datos que trae consigo cada documento, y pueden además compartir la misma descripción y forma de validación dentro de un grupo de trabajo que usa el mismo tipo de información.
- **GIF:** Formato gráfico utilizado ampliamente en la World Wide Web, tanto para imágenes como para animaciones.
- **HTML:** HyperText Markup Language. Lenguaje de marcado para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores Web.
- **HTTP:** HyperText Transfer Protocol. Protocolo de transferencia de hipertexto usado en cada transacción de la Web.
- **IDE:** Entorno de desarrollo integrado. Programa compuesto por un conjunto de herramientas para un programador.
- **IETF:** Internet Engineering Task Force. Organización internacional abierta de normalización.
- **JPEG:** Joint Photographic Experts Group. JPEG es considerado un formato de archivo. JPEG es el formato de imagen más común utilizado por las cámaras fotográficas digitales y otros dispositivos de captura de imagen.
- **JSP:** JavaServer Pages. Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.
- **LDAP:** Lightweight Directory Access Protocol. Es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

- 
- **MVC:** Modelo Vista Controlador. Patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.
  - **NIS:** Network Information Service. Protocolo de servicios de directorios cliente-servidor desarrollado por Sun Microsystems para el envío de datos de configuración en sistemas distribuidos tales como nombres de usuarios y hosts entre computadoras sobre una red.
  - **PHP:** PHP Hypertext Pre-processor. Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente en la interpretación del lado del servidor.
  - **PNG:** Portable Network Graphics. Formato gráfico basado en un algoritmo de compresión sin pérdida para bitmaps no sujeto a patentes. Este formato fue desarrollado en buena parte para solventar las deficiencias del formato GIF y permite almacenar imágenes con una mayor profundidad de contraste.
  - **SDK:** Software Development Kit. Conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema concreto.
  - **SGBD:** DataBase Management System. Software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.
  - **SMTP:** Simple Mail Transfer Protocol. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (PDA's, teléfonos móviles, etc.).
  - **SQL:** Structured Query Language. Lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.
  - **SSI:** Small-Scale Integration. Hace referencia a los primeros circuitos integrados que se desarrollaron. Cumplían funciones muy básicas, como puertas lógicas y abarcan desde unos pocos transistores hasta una centena de ellos.
  - **TCP/IP:** Familia de protocolos de Internet. Es un conjunto de protocolos de red en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.
  - **UC3M:** Universidad Carlos III.

- **UNIX:** Sistema operativo portable, multitarea y multiusuario.
- **UML:** Unified Modeling Language. Lenguaje de modelado de sistemas de software. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.
- **URL:** Uniform Resource Locator. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.
- **W3C:** World Wide Web Consortium. Consorcio internacional que produce recomendaciones para la World Wide Web.
- **XHTML:** eXtensible Hypertext Markup Language. Lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web.
- **XML:** Extensible Markup Language. Metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

# Bibliografía

---

- [1] Características, May 2005. <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/x31.html>.
- [2] Ldap linux howto, March 2007. <http://tldp.org/HOWTO/LDAP-HOWTO/index.html>.
- [3] Servlets, November 2007. <http://javaweb.osmosislatina.com/curso/servlets.htm>.
- [4] The apache software foundation, August 2009. <http://www.apache.org/>.
- [5] The apache software foundation, September 2009. <http://httpd.apache.org/>.
- [6] Arquitectura del servidor apache, September 2009. <http://www.desarrolloweb.com/articulos/1112.php>.
- [7] Direcciones ip y puertos de escucha, September 2009. <http://httpd.apache.org/docs/2.0/es/bind.html>.
- [8] Documentación del servidor http apache 2.0, September 2009. <http://httpd.apache.org/docs/2.0/es/>.
- [9] Java mail, July 2009. <http://java.sun.com/products/javamail/>.
- [10] Java servlet, September 2009. [http://es.wikipedia.org/wiki/Java\\_Servlet](http://es.wikipedia.org/wiki/Java_Servlet).
- [11] Java servlet technology, September 2009. <http://java.sun.com/products/servlet/>.
- [12] Javaserer pages, September 2009. [http://es.wikipedia.org/wiki/JavaServer\\_Pages](http://es.wikipedia.org/wiki/JavaServer_Pages).
- [13] Javaserer pages technology, September 2009. <http://java.sun.com/products/jsp/>.

## BIBLIOGRAFÍA

---

- [14] Ldap naming service versus other naming services (system administration guide: Naming and directory services (dns, nis, and ldap)), September 2009. <http://docs.sun.com/app/docs/doc/806-4077/6jd6blbej?a=view>.
- [15] Microsoft sql server, September 2009. [http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server).
- [16] ¿por qué código abierto? | abax asesores, September 2009. <http://abaxasesores.com/codigoabierto>.
- [17] Sistema de gestión de base de datos, September 2009. <http://es.wikipedia.org/wiki/SGBD>.
- [18] World wide web consortium, September 2009. <http://www.w3.org/>.
- [19] Alvaro del Castillo San Félix. El servidor de web apache: Introducción práctica. [http://beta.redes-linux.com/manuales/Servidor\\_web/apache.pdf](http://beta.redes-linux.com/manuales/Servidor_web/apache.pdf).