

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS  
INDUSTRIALES Y DE TELECOMUNICACIÓN**

**UNIVERSIDAD DE CANTABRIA**



***Proyecto Fin de Carrera***

**GESTOR WEB DE EVENTOS BASADO EN  
TECNOLOGÍAS DE CÓDIGO ABIERTO  
(WEB MANAGER FOR EVENTS BASED ON  
OPEN-SOURCE TECHNOLOGIES)**

Para acceder al Título de

**INGENIERO TÉCNICO DE TELECOMUNICACIÓN**

**Autor: Rafael Vallejo Alvear**

**Septiembre - 2016**

# INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN

## CALIFICACIÓN DEL PROYECTO FIN DE CARRERA

**Realizado por: Rafael Vallejo Alvear**

**Director del PFC: Jose Luis Bosque Orero**

**Título: “Gestor web de eventos basado en tecnologías de código abierto”**

**Title: “Web Manager for events based on open-source technologies“**

**Presentado a examen el día: 29 de Septiembre de 2016**

para acceder al Título de

## INGENIERO TÉCNICO DE TELECOMUNICACIÓN

### Composición del Tribunal:

Presidente (Apellidos, Nombre): Esteban Stafford Fernández

Secretario (Apellidos, Nombre): José Luis Bosque Orero

Vocal (Apellidos, Nombre): Carmen Martínez Fernández

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del PFC  
(sólo si es distinto del Secretario)

V° B° del Subdirector

Proyecto Fin de Carrera N°  
(a asignar por Secretaría)

## Índice

1 - Introducción	4
1.1 – Motivación	4
1.2 – Planteamiento del problema	4
1.3 – Objetivos	5
1.4 – Estructura del documento	5
2 - Herramientas	7
2.1 – HTML 5	7
2.2 – CSS 3	8
2.3 - PHP 5	8
2.4 – JavaScript	9
2.5 – MySQL	9
3 - Diseño	11
3.1 – Metodología de diseño	11
3.2 – Diseño funcional – Primera versión	12
3.2.1 – Diseño de la base de datos	13
3.2.2 – Diseño del gestor	19
3.3 – Diseño funcional – Segunda versión	23
3.3.1 – Modificaciones en la base de datos	23
3.3.2 – Modificaciones en la estructura de aplicaciones	25
3.4 – Diseño gráfico	25
4 - Implementación	29
4.1 – Implementación: Primera versión	29
4.1.1 – Implementación de la base de datos	29
4.1.2 – Implementación de la estructura	32
4.1.3 – Implementación de la interface gráfica	34
4.1.4 – Pruebas de funcionamiento	35
4.2 – Implementación: Segunda versión	36
4.2.1 – Modelo basado en peticiones AJAX	36
4.2.2 – Módulo de comunicaciones en tiempo real	39
4.2.3 – Préstamo por lector de código de barras	40
4.2.4 – Recarga de los datos en la DB	40
4.2.5 – Pruebas de funcionamiento	41
4.2.6 – Modificaciones a la interface gráfica	41
5 - Casos de uso	43
5.1 – Jornadas Minas Tirith	43
5.2 – Otros casos de uso	46
6 - Conclusiones	47
6.1 – Consideraciones previas	47
6.2 – Posibles mejoras	50
7 - Referencias	54

# 1- Introducción

## 1.1 – Motivación

El presente proyecto nace para dar respuesta a una necesidad: la optimización de los recursos con los que se cuenta para la organización de un evento sociocultural, buscando la reducción de tiempo y esfuerzo en las tareas propias del mismo. Con este objetivo en mente se comenzó a trabajar con la Asociación Juvenil Ciudad Blanca, que cada Diciembre desde diez años viene organizando el Festival de juegos de mesa de Santander ‘Minas Tirith’.

Las jornadas ‘Minas Tirith’ son un evento de carácter gratuito organizado por la citada asociación en colaboración con otras entidades similares. A lo largo de cuatro días de evento ponen a disposición de los asistentes una ludoteca con más de 450 juegos de mesa, en régimen de préstamo, para que puedan probarse dentro del recinto del evento. Además, de manera paralela a la ludoteca, organizan actividades relacionadas con los juegos de mesa y el ocio alternativo. Todo ello se lleva a cabo con un presupuesto mínimo.

El evento mencionado cuenta año a año con una participación creciente por parte de los asistentes, mientras que el grupo de monitores permanece constante. Por ese motivo es imprescindible la búsqueda de soluciones que permitan mejorar su eficiencia, evitando perder así calidad en la atención al público.

Son sus necesidades las que se estudiarán en este capítulo y crearán la base para el desarrollo del proyecto, marcando los objetivos mínimos necesarios para cumplirlas. Por último, el mejor banco de pruebas posible para los resultados del mismo será el propio evento, donde se podrá evaluar el éxito o fracaso del trabajo realizado.

## 1.2 – Planteamiento del problema

La solución elegida debe ser capaz de llevar el control de los participantes en el evento a través de una utilidad de registro, así como de las actividades y del material prestable del mismo. Por otra parte, ha de poder relacionar ambos grupos, creando las conexiones necesarias para llevar el control de los préstamos realizados o el registro de los participantes en las actividades.

Además, es importante que esta solución adoptada ayude a la organización del evento a obtener datos en tiempo real del estado del mismo, con el fin de que puedan ser analizados para tomar las decisiones que garanticen el éxito de la actividad. Además, dada la afluencia habitual de las jornadas y la repetición en alguna de las tareas a realizar, debe servir para agilizar dichas tareas y que el personal del evento,

algo escaso para todo el trabajo que ha de llevar a cabo, pueda gestionar un mayor número de gente.

Finalmente, cabe reseñar que dado el carácter gratuito del evento y que las asociaciones organizadoras no tienen ánimo de lucro, el desarrollo del proyecto debe ser lo más contenido en costes posible, optándose siempre que sea posible por soluciones a coste cero.

Además, la necesidad de involucrar a más de un equipo en las tareas de gestión, por encontrarse los puestos de control en espacios separados, obliga a soluciones capaces de conectarse a una red privada o a internet, siendo esta última opción la preferida por la organización al no contar con un equipo que pueda realizar las tareas de servidor en el espacio del evento.

Dicha conexión a internet se realizará mediante dispositivos móviles en régimen de router wifi, lo que establecerá una limitación en materia de transferencia de datos, que deberá ser lo más contenida posible para garantizar el servicio.

### **1.3 – Objetivos**

El objetivo principal es, en resumen, desarrollar una herramienta online, con capacidad para gestionar y analizar los datos propios del evento, basada en software libre y con bajos requisitos de consumo de datos. Con estas premisas en mente ya se dispone de las bases necesarias para desarrollar el proyecto, que tendrá los siguientes objetivos parciales:

- Análisis y selección de las herramientas de desarrollo.
- Escoger la metodología de diseño que mejor se adapte a las necesidades del proyecto.
- Realizar un diseño de la aplicación, tanto a nivel funcional como de interface con el usuario.
- Implementar el diseño escogido.
- Depurar y comprobar su utilidad en un escenario real.
- Estudiar otros posibles usos de la solución adoptada.

El cumplimiento de los objetivos marcados se irá comprobando a lo largo del desarrollo del documento, prestando siempre atención a que dicho cumplimiento no sobrepase las limitaciones establecidas.

### **1.4 – Estructura del documento**

A lo largo de este texto se explicarán los pasos tomados para el cumplimiento de los objetivos vistos en el capítulo anterior, dividiéndose el mismo en los siguientes apartados:

- **Herramientas:** Análisis y selección de las diferentes herramientas escogidas, justificando su elección frente a otras alternativas.
- **Diseño:** Estudio de las metodologías de diseño disponibles. Recorrido por las diferentes fases del diseño y su desarrollo.
- **Implementación:** Puesta en funcionamiento del diseño realizado. Análisis y solución a los problemas surgidos. Test de funcionamiento.
- **Casos de uso:** Revisión de las diferentes aplicaciones prácticas del desarrollo realizado. Prueba de la herramienta en un entorno real.
- **Conclusiones:** Análisis de los resultados obtenidos y del cumplimiento de los objetivos marcados. Posibilidades de desarrollo posterior.

Estos apartados a su vez se encontrarán subdivididos en diferentes secciones, que facilitarán el recorrido a través del documento. Debido a la cantidad de elementos que compondrán el diseño final y a la imposibilidad de tratar todos ellos en profundidad, este documento se centrará en aquellos más importantes o que por sus características tienen un mayor interés.

## 2- Herramientas

Para el cumplimiento de los objetivos marcados en el capítulo anterior se optará por un diseño compartimentado, que mantenga separadas las capas de estructura, estilo y funcionalidad en diferentes archivos. La explicación a esta decisión podrá verse en el capítulo de diseño, pero es importante mencionarla, ya que para realizar dicha separación es necesario el uso de diferentes tecnologías.

El listado de herramientas necesarias, que se verá en detalle en los próximos apartados, incluye el uso de HTML 5 para la estructura de la aplicación, CSS 3 para dotarla de estilo, PHP 5 y JavaScript con el fin de añadir interactividad y MySQL como gestor de base de datos, así como utilidades complementarias incluidas en ellas.

Cada una de estas tecnologías cumple un objetivo diferente dentro del proyecto, que no puede ser asumido por ninguna de las demás, lo que justifica el uso de todas ellas. Por este motivo, estas son las herramientas que se han seleccionado para llevar a cabo este proyecto.

### 2.1 - HTML 5

“HTML es un lenguaje de descripción de hipertexto compuesto por una serie de comandos, marcas o etiquetas, también denominados ‘Tags’ que permiten definir la estructura lógica de un documento web y establecer los atributos del mismo” [1]

El *Hyper-Text Markup Language* (HTML) nace en 1991 y desde entonces ha crecido hasta convertirse en la base de la programación web tal y como se conoce hoy en día. A lo largo de sus múltiples revisiones ha ido ganando en funcionalidad, adaptándose a los nuevos requerimientos que la revolución tecnológica liderada por el auge de Internet, los dispositivos móviles y el incremento de velocidad de las redes que les dan soporte, han exigido.

Actualmente se encuentra a cargo del W3C, el “*World Wide Web Consortium*”, una organización encargada de definir los estándares de los principales lenguajes web. En este momento, la versión más avanzada de HTML es la 5, publicada en su versión definitiva por el W3C en Octubre de 2014. [2]

Dentro de este proyecto, se utilizará el estándar HTML 5 para realizar la capa de estructura en su totalidad, separando las funciones de estilo de la misma mediante el uso de *Cascade Style Sheet* (CSS 3, que se explica en la siguiente sección), a pesar de que HTML 5 da soporte a las mismas, con el fin de hacer el código modular, entendible y limpio a la par que permitir un mantenimiento más sencillo.

## 2.2 - CSS 3

CSS es, según la W3C, “un mecanismo fácil para añadir estilos (por ejemplo, fuentes, colores, espacios) a los documentos web” [3]. Se trata de un lenguaje descriptivo que otorga atributos a los diferentes elementos descritos en HTML o XHTML. Mediante el uso de pares “etiqueta: valor” define las diferentes características que uno o varios elementos contendrán, permitiendo que el navegador los interprete y dibuje en pantalla de manera adecuada.

Desde su primera especificación completa, surgida a finales del siglo pasado, ha ido ganando uso entre los diseñadores web hasta convertirse en una herramienta indispensable, que simplifica el diseño web y añade un enorme conjunto de funciones plenamente compatibles con otros lenguajes. Su última revisión publicada por parte del W3C es la *CSS Level 3*, que al contrario de las dos especificaciones anteriores, se compone de un conjunto de módulos en constante desarrollo que añaden nuevas características al lenguaje. [4]

En este proyecto se definirá la capa de estilo mediante el uso de esta herramienta, en scripts de código independientes a los usados en las capas de estructura y funcionalidad.

## 2.3 - PHP 5

“PHP es un lenguaje interpretado del lado del servidor que se caracteriza por su potencia, versatilidad, robustez y modularidad. Los programas escritos en PHP son embebidos directamente en el código HTML y ejecutados por el servidor web a través de un intérprete antes de transferir al cliente que lo ha solicitado un resultado en forma HTML puro” [1]

PHP es un lenguaje de programación de código abierto, que al poder ser embebido en el código HTML permite una gran capacidad de interacción entre ambos lenguajes, siendo una de las alternativas más populares a la hora de añadir funcionalidad a una aplicación web.

Dado que la ejecución de los scripts PHP se realiza del lado del servidor otorga dos importantes características. La primera de ellas es la posibilidad de interactuar con las bases de datos, alojadas también en el servidor, opción por la cual PHP sirve a menudo de puente entre dichas bases de datos y la información que se muestra al usuario final.

Por otra parte, la segunda característica es que procesa los datos en la máquina de origen, descargando de trabajo al cliente y enviando únicamente información final ya completamente procesada.

Más adelante se verá que estas dos características son ampliamente utilizadas en el desarrollo de este proyecto, convirtiendo a PHP en una herramienta indispensable para el diseño de la capa de funcionalidad.

## 2.4 – JavaScript

“JavaScript es un lenguaje de programación creado por Netscape con el objetivo de integrarse en HTML y facilitar la creación de páginas interactivas [...]. Es un lenguaje de programación orientado a objetos, diseñado para el desarrollo de aplicaciones cliente-servidor a través de Internet” [5]

JavaScript es un lenguaje interpretado, usado principalmente del lado del cliente. La interpretación del código se realiza por parte del navegador web y forma parte de los estándares definidos por parte del W3C como uno de los elementos básicos de la web bajo el nombre ECMAScript. [6]

Su ejecución en el lado del cliente permite el diseño de aplicaciones dinámicas, con capacidad de responder de manera inmediata a peticiones del usuario. Igualmente, a través del uso del *Document Object Model* o DOM, con el que interactúa especialmente bien y de técnicas como AJAX (*Asynchronous JavaScript And XML*) permite la modificación de elementos de la página web con necesidades mínimas de transferencia de datos, al evitar cargas completas de la página y reducirlo a la edición de bloques individuales, que cuentan con mucho menos código.

De estas es especialmente importante la aportación al proyecto que realiza AJAX, ya que permite establecer un canal de comunicación asíncrono entre el servidor y el cliente, pudiendo realizar peticiones de información puntuales y envíos de pequeños paquetes de datos de forma dinámica. Por otra parte, es una potente herramienta a la hora de realizar validaciones de datos o de añadir interactividad con el usuario.

Es por esto que JavaScript se convierte en un elemento clave del desarrollo de la capa funcional de este proyecto tal y como se verá más adelante.

## 2.5 – MySQL

MySQL es un sistema gestor de bases de datos relacionales de código abierto, que hace uso del *Structured Query Language* (SQL) para la creación y mantenimiento de bases de datos rápidas y flexibles, así como para el acceso, modificación y eliminación de la información que contienen. [7]

“El modelo relacional se basa en el concepto de conjunto. Esquemáticamente, el modelo relacional se puede representar a través de una tabla. [...] Este conjunto tiene atributos (las columnas) y líneas que contienen los valores (los registros). [...] El modelo relacional presenta los datos de forma lógica, es totalmente independiente del

modelo físico. [...] Esta es la mayor ventaja de las bases de datos relacionales, la independencia entre lo lógico y lo físico.” [8]

El lenguaje SQL permite al mismo tiempo realizar la definición lógica de las diferentes tablas relacionales, gestionar el acceso, edición y eliminación de los registros y establecer la seguridad en el acceso a los datos. Dicha versatilidad permite la total administración de la base de datos con un único lenguaje. Por su parte, MySQL cuenta con capacidad de integración en PHP, pudiendo realizarse las llamadas a la base de datos desde scripts programados en este lenguaje.

MySQL es la opción elegida en este proyecto para el almacenamiento de la información, dado que fue concebido para la implementación de bases de datos pequeñas y medianas, como la desarrollada a lo largo de este proyecto. La base de datos es el núcleo y el elemento de más valor de este proyecto. En posteriores capítulos se analizarán la estructura de dicha base de datos y su papel a la hora de cumplir los objetivos marcados.

## 3- Diseño

En el presente capítulo se procederá a analizar la fase de diseño, profundizando en aquellos aspectos más importantes. Tal y como se verá en los próximos apartados, se ha optado por escoger un esquema clásico de diseño, comenzando con la elección de una metodología que se ajuste a las características del proyecto para posteriormente dividir el problema en bloques que permitan abordarlo con mayor facilidad.

### 3.1 – Metodología de diseño

El primer paso del diseño del presente proyecto consiste en escoger la metodología de diseño más adecuada al desarrollo del mismo. Una elección acertada permitirá ahorrar en tiempo y esfuerzo a lo largo del desarrollo, mientras que una elección fallida conllevaría incrementar el coste del proyecto o facilitaría caer en errores de difícil solución una vez el código estuviera implementado. Las opciones disponibles son las siguientes [9]:

- **Diseño en cascada:** En un diseño en cascada, el proyecto se divide en fases claramente diferenciadas que se ejecutan de forma secuencial, admitiendo poca capacidad de feedback. Requieren por lo tanto una fase de estudio muy amplia, en la que queden claramente definidas cada una de esas fases y los objetivos que persiguen, dado que el coste en tiempo y esfuerzo de volver atrás una vez se ha avanzado a una fase posterior son muy grandes.
- **Diseño prototipado:** El diseño prototipado se define por una fase de implementación más dilatada en el tiempo, iterativa, mediante la realización de prototipos no completamente funcionales del proyecto que son sometidos a prueba antes de evaluarse de nuevo y ser mejorados. Esta metodología permite un desarrollo robusto y código de mejor calidad que el que obtendríamos mediante el diseño en cascada, debido a las múltiples fases de prueba a las que es sometido el código. Sin embargo, dado su carácter iterativo y debido a que el código utilizado en esas iteraciones nunca es definitivo es fácil que el diseño se prolongue en el tiempo sin obtener resultados finales.
- **Diseño incremental:** El diseño incremental nace de la unión del diseño en cascada y del prototipado. En esta metodología de diseño, se realizan todas las fases del diseño en cascada en repetidas ocasiones, obteniendo versiones intermedias del producto final que son susceptibles de ser mejoradas y ampliadas en iteraciones posteriores. Esta metodología de diseño busca aunar las principales ventajas de las dos metodologías

anteriores obteniendo versiones funcionales del proyecto, que pueden utilizarse en etapas intermedias del desarrollo y a la vez ser mejoradas con el tiempo mediante el desarrollo de nuevas versiones, que tengan en cuenta la experiencia adquirida en los desarrollos previos. Sin embargo, mediante el uso de esta metodología es sencillo caer en la tentación de posponer los problemas más complicados para versiones posteriores del proyecto.

- **Diseño en espiral:** Al igual que el diseño incremental, el diseño en espiral es una metodología que recoge la mezcla del diseño en cascada y el diseño prototipado. En este caso, el proceso de diseño se asemeja a una espiral, en el que cada 'vuelta' o 'giro' representa un proceso completo del diseño en cascada, que finaliza en una versión funcional del programa. Al comienzo del siguiente giro se evalúan los riesgos y el coste de desarrollar una nueva versión, por lo que al igual que en el diseño en cascada cobra especial relevancia la fase de planificación, siendo por lo tanto menos rápido a la hora de desarrollar que el diseño incremental.

Dado que el proyecto no es de gran envergadura, parece sensato elegir una configuración lo más cercana posible a un diseño en cascada, que permita un desarrollo rápido y controlado de los contenidos y sea capaz de alcanzar rápidamente los objetivos marcados. Sin embargo, al tratarse de un proyecto destinado a un colectivo con pocos recursos, cabe destacar que el coste de retroceder en este tipo de sistema podría desembocar en que el proyecto dejara de ser viable.

Es por eso que la elección de una de las soluciones intermedias parece aún más adecuada. De entre ellas, el diseño en espiral está más enfocado a la evaluación de los riesgos de cada una de las versiones, pero es susceptible de prolongarse en el tiempo más que un diseño de versiones. Además, cualquier versión de código que otorgue una funcionalidad extra es de gran valor para el cliente, lo que finalmente inclina a decantarse por un diseño incremental, en el que se buscará en todo momento realizar el menor número de versiones posibles, acercándolo más al diseño en cascada que al diseño prototipado.

### 3.2 – Diseño funcional – Primera versión

Como se comentó al introducir el capítulo de herramientas, el diseño de la aplicación se realizará de manera compartimentada, de forma que el código quedará dividido de manera efectiva en las siguientes capas:

- **Estructura:** Código HTML que definirá los diferentes bloques que compondrán la página.

- **Estilo:** Archivos CSS que aportarán estilo a cada uno de los bloques definidos por HTML.
- **Funcionalidad:** Código PHP estrechamente ligado a la estructura que otorgará las características de interactividad y procesamiento a los bloques de la página.

El objetivo de esta separación es crear un código más sencillo de entender, y por lo tanto de mantener, pues la separación de las diferentes capas permite ver cada una de ellas con mayor claridad. Esto también implica más facilidad a la hora de editar y mantener la página, ya que no es necesario modificar y adaptar todas las capas para realizar una modificación en una de ellas. Por otra parte, separar las capas permite evitar redundancias, obteniendo un código más ligero y que carga más rápido.

Las ventajas de separar la parte funcional del resto de capas por medio de sentencias 'include' (llamadas a archivos externos con otros fragmentos de código) son descritas por Clint Eccher en su libro 'Diseño Web Profesional', al afirmar que "Los archivos include [...] no solo facilitan que el sitio sea modular, es decir, con diferentes piezas intercambiables, sino, y lo que es más importante, también reducen el tiempo de mantenimiento" [10]

Por otra parte, en el mismo texto se explican las bondades del uso separado de capas de estilo y estructura, señalando que "La página Web resultante es mucho más limpia y fácil de seguir, [...] el tamaño combinado de carga, o peso [...] es mucho menor [...] y la estructura y el estilo se pueden añadir, editar y eliminar con más facilidad". [10]

Finalmente, en 'Diseño de páginas web con XHTML, JavaScript y CSS', Juan Carlos Orós detalla otras virtudes como "disponer de comandos y atributos más potentes y precisos con los que poder maquetar exactamente un documento. [...] Su uso estructurado y razonado permite ahorrar muchas líneas de código". [5]

### 3.2.1 - Diseño de la base de datos

Como se ha comentado a la hora de presentar las herramientas que se utilizarán a lo largo de este proyecto, el elemento de más valor de este desarrollo es su base de datos y por lo tanto se convierte inevitablemente en el eje central del diseño del proyecto.

A la hora de definir la estructura de datos se ha de tener en cuenta que se busca modelar elementos del mundo real, siendo este un buen punto de partida para comenzar a definir las diferentes tablas y relaciones que compondrán la base de datos.

Analizando el tipo de evento en el que se aplicará el software desarrollado en el proyecto es posible identificar tres grandes grupos que permitirán vertebrar la base de datos.

- Las personas que asisten al evento, las cuales se denominarán “asistentes”.
- El conjunto de actividades que conforman el programa del evento, a las que simplemente se llamará “actividades”.
- El material que es prestado a los asistentes formará el tercer eje, que se denominará “ludoteca” al estar compuesto de juegos de mesa.

#### A. *Tabla Asistentes*

Sobre los asistentes se centra uno de los objetivos del proyecto, que es la capacidad de generar estadísticas del evento que ayuden a discernir si dicho evento fue un éxito o no, a la par que ayude a los colectivos organizadores a decidir en qué áreas deben centrarse en ediciones posteriores. Dicha tabla se compondrá de los siguientes campos:

- Nombre y apellidos: Datos necesarios para identificar correctamente al asistente.
- Edad: Necesario para el cálculo de estadísticas que permitan enfocar las actividades y el material ofertado a los asistentes
- Procedencia y encuesta: Permite conocer cuáles han sido los canales de comunicación más efectivos a la hora de informar sobre el evento, a la par que los lugares donde más han funcionado.
- Hora de inscripción: Momento temporal en el que el participante se inscribe, que permite llevar un control de la afluencia y anticipar las cargas de trabajo.

Con estas consideraciones la tabla de asistentes, una vez separados correctamente los campos, queda tal y como se muestra en la Figura 3.1:

<b>Asistentes</b>	
<b>ID_Asistente</b>	<b>INT, Clave principal</b>
Nombre	VARCHAR (30)
Apellidos	VARCHAR (60)
Edad	TINYINT
Codigo_Postal	INT
<b>ID_Encuesta</b>	<b>INT, Clave externa</b>
Fecha	DATETIME

Figura 3.1 – Tabla de asistentes

Como se puede observar, el campo 'encuesta' está referenciado como 'clave externa' y con tipo de datos 'INT' (entero). El motivo es normalizar los datos de los que se dispondrá en el sistema, permitiendo que el tratamiento de dichos datos sea más fluido y reduciendo los errores que puedan cometerse en su inserción. La nueva tabla 'Encuesta' está formada por su correspondiente ID y un campo 'nombre' que define cada uno de los registros.

Por otra parte, con el fin de hacer la base de datos lo más pequeña posible (en términos de espacio de almacenamiento), el campo de edad se define con el tipo de datos 'TINYINT', que según la documentación de MySQL permite valores enteros entre 0 y 255 (*unsigned*), más que suficiente para abarcar el rango de edades lógico de los asistentes. [11]

### B. Tabla Actividades

A continuación se pasará a diseñar la tabla de 'actividades'. Dicha tabla deberá alimentar tanto el programa gestor del evento como su página web, lo cual tendrá que ser tenido en cuenta a la hora de definir sus campos.

- ID y título, que permitirán identificarla tanto a nivel lógico como por el usuario.
- Fecha y hora de comienzo de la actividad, por ser datos importantes que han de mostrarse al público.
- Descripción y duración determinada con el fin de ampliar la información sobre la actividad.
- Número mínimo y máximo de participantes y reservas, por ser datos clave para la organización y permitir una mejor atención al asistente a la hora de inscribirse a la actividad.

La primera versión de la tabla de asistentes quedaría entonces tal y como se muestra en la Figura 3.2:

Actividades	
ID_Actividad	INT, Clave principal
Título	VARCHAR (50)
Día	VARCHAR (10)
Hora	VARCHAR (10)
Duración	INT
Descripción	VARCHAR (2000)
Minpar	TINYINT
Maxpar	TINYINT
Res	TINYINT

Figura 3.2 – Tabla de actividades

En este caso no se definen claves externas, si bien, por la metodología de diseño escogida, no se descarta que se realicen modificaciones futuras.

### C. Tabla Ludoteca

La última tabla principal que se creará es la de 'ludoteca'. Esta tabla contiene igualmente registros para la aplicación gestora y para la web del evento:

- ID y nombre de juego como identificadores del juego.
- Número mínimo y máximo de jugadores, como información útil tanto para los participantes como para los monitores a la hora de recomendar un juego.
- Tiempo de juego y descripción, dado que es interesante mostrar información adicional sobre el juego.
- Código del juego, que se trata de una ID única clave para el gestor, tal y como se verá a continuación.
- Estado y editorial, como información complementaria sobre el juego de valor para la organización. Estos campos serán referenciadas desde tablas secundarias, para normalizar los datos.

Una primera versión de la tabla se muestra en la Figura 3.3:

<b>Ludoteca</b>	
<b>ID_Juego</b>	<b>INT, Clave principal</b>
Nombre	VARCHAR (50)
Autor	VARCHAR (30)
Código	INT
Minjug	TINYINT
Maxjug	TINYINT
Tiempo	INT
Descripción	VARCHAR (2000)
Minage	TINYINT
Estado	INT, Clave externa
Editorial	INT, Clave externa

Figura 3.3 – Tabla de ludoteca

Sobre el código del juego, dado que los juegos presentes en el evento provienen de varias asociaciones e incluso de particulares que los ceden para el mismo, se decide utilizar un sistema que permita identificar al propietario del juego al mismo tiempo que se define inequívocamente el juego.

Es por eso que se establece un código de cinco cifras numéricas, de las cuales las dos primeras identifican al propietario (se evalúa con el cliente que es probable tener más de 10 propietarios diferentes, pero virtualmente imposible superar los 100)

y las tres últimas el número de juego del propietario, pues algunos propietarios cuentan con más de 100 juegos y en el improbable caso de que en algún momento uno o varios superaran el millar, siempre se les podría asignar un segundo código de propietario.

Observando la tabla definida en la Figura 3.3 se puede realizar una primera iteración en su diseño. De cara a normalizar las tablas de una base de datos, una de las condiciones necesarias para que una tabla cumpla la 'Primera Forma Normal' (1FN) es que los datos que contiene cada celda deben ser atómicos, es decir, indivisibles. Esto no se cumple en la tabla de autor, donde un juego puede tener varios autores.

Por este motivo, al margen de las tablas de estado y editorial (ambas con idéntica estructura, contando con una ID y un nombre), se procede a crear una tabla de 'autores', similar a estas dos tablas citadas en cuanto a su estructura. Sin embargo, no se puede crear una relación directa entre juegos y autores, pues un autor puede haber hecho varios juegos y a su vez, como ya hemos comentado, un juego puede tener varios autores.

Esto hace necesaria la creación de una tabla que relacione ambos conceptos, una tabla intermedia con una estructura que permitirá crear relaciones juego-autor, de forma que tanto los juegos como los autores podrán aparecer múltiples veces (ver Figura 3.4):

<b>Juego_Autor</b>	
ID_Juego_Autor	INT, Clave principal
ID_Juego	INT, Clave externa
ID_Autor	INT, Clave externa

Figura 3.4 – Tabla Juego\_Autor

Como se puede ver, el único objetivo de esta tabla es establecer pares Juego-Autor, cumpliendo con los requisitos definidos en la fase de análisis. Dado que la relación entre juegos y autores queda definida ya en esta tabla, esto hace innecesario el campo 'Autor' definido en la tabla de la Figura 3.3, por lo que se procede a su eliminación.

De esta forma la Figura 3.5 presenta el esquema actual de la base de datos:



Figura 3.5 – Base de datos V1

#### D. Otras tablas

Con las tablas establecidas hasta ahora y sus relaciones se puede observar cual es el siguiente paso lógico. Tal y como se ha definido, el esquema consta de tres grandes tablas que forman el núcleo de la base de datos, pero falta relacionarlas entre sí. Los asistentes tienen que poder apuntarse a las actividades y se deben registrar los préstamos de material que se realizan. Esto abre la puerta a dos nuevas tablas.

La primera de ellas es la tabla que relaciona asistentes y actividades, la más sencilla de las dos, dado que únicamente almacena pares de datos ID\_Asierte – ID\_Actividad, con un campo adicional de tipo BIT llamado ‘Borrado’, que se fijará a valor lógico ‘TRUE’ cuando se quiera quitar a alguien de una actividad. En este caso no se opta por borrar directamente el registro, pues el cliente desea guardar el dato para sacar estadísticas de él.

La segunda tabla es algo más interesante. Se elige la denominación ‘préstamos’ para ella, dado que define bien el objetivo de la tabla y en todo momento se busca mantener una nomenclatura clara que permita entender bien la finalidad de cada una de las tablas. Esta vez, además de la relación entre los asistentes y los juegos, ambos por medio de su ID, se añaden los campos de hora de préstamo y hora de devolución. El primero será fijado gracias a la función NOW(), nativa de SQL, mientras que el segundo será establecido originalmente al valor predeterminado ‘00:00:00’ y posteriormente actualizado en el momento de la devolución del material prestado.

De esta forma, además de llevar en todo momento un control del material que se encuentra en préstamo y de los asistentes a los que ha sido prestado, se podrá realizar el cálculo del tiempo que ha estado prestado cada uno de los juegos, siendo este un dato importante a la hora de calcular las estadísticas.

Con estas dos tablas definidas se puede dar por terminada la primera versión funcional de la base de datos, quedando el esquema que se ve en la Figura 3.6:

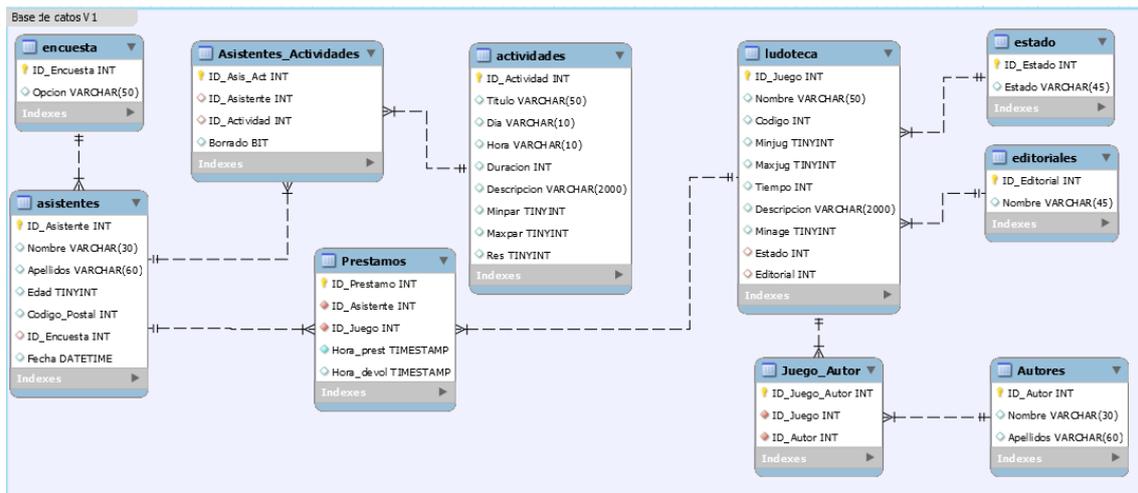


Figura 3.6 – Base de datos V1

Ahora se puede ver cómo se relacionan todas las tablas entre si y se puede pasar a diseñar las dos capas de visualización que usarán estos datos, el gestor y el website, desarrollándose por su interés el primero.

### 3.2.2 – Diseño del gestor

Dentro de las Jornadas ‘Minas Tirith’ existen dos puntos de control, uno denominado “mostrador”, en el que se inscribe a los participantes y se gestionan las actividades y otro llamado “ludoteca”, donde se controla el material y se realizan los préstamos. Es por eso que en el diseño de las aplicaciones del gestor se decide también crear estos dos grupos, de forma que cada uno tenga acceso inmediato a las utilidades que es más probable que necesite.

Finalmente, se añadirá un último grupo de administración, con funciones especiales. Estas son las diferentes utilidades a las que tendrá que tener acceso cada uno de esos grupos:

#### Mostrador:

- Inscripción de participantes
- Inscripción en actividades
- Borrado de inscripción en actividades

#### Ludoteca:

- Nuevo préstamo
- Devolución
- Juegos prestados

#### Administración:

- Creación de actividad

- Edición de actividad
- Inserción de nuevo juego en la ludoteca
- Edición de tablas secundarias

Cada una de estas aplicaciones debe ser lo más sencilla e intuitiva posible, de modo que aquellos monitores que no estén familiarizados con el gestor puedan trabajar con él en un espacio corto de tiempo y a la vez sean ágiles para poder lidiar con grandes volúmenes de gente utilizando pocos recursos.

Es por ello que los diagramas de flujo de las diferentes aplicaciones se estudian para reducir sus elementos de decisión al mínimo posible, acorde al diseño mostrado en la figura 3.7. A continuación se detallarán aquellas que tienen más contenidos.

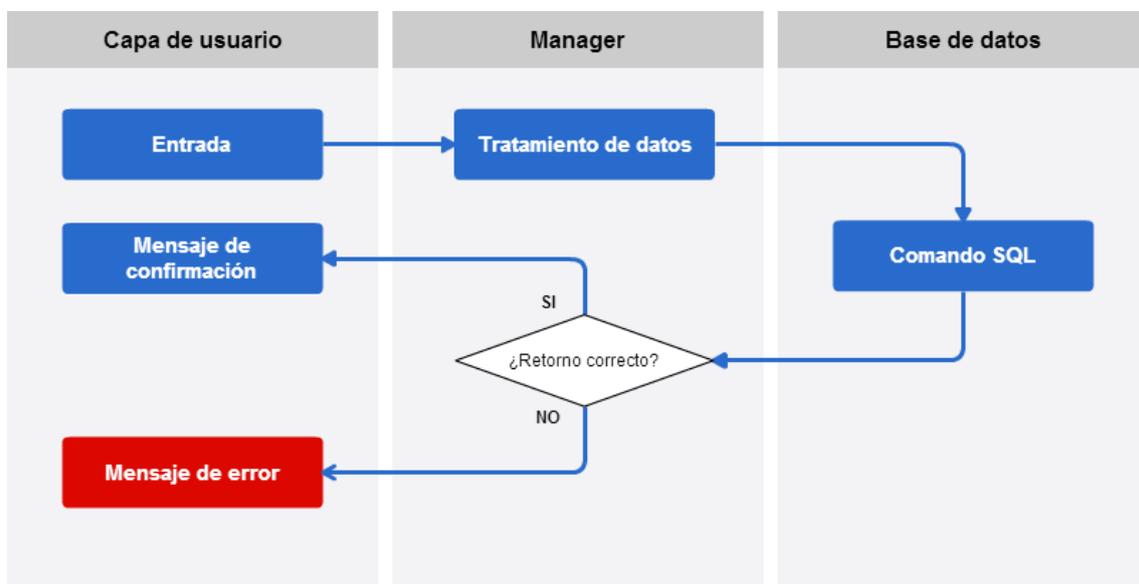


Figura 3.7 – Diagrama de flujo básico de aplicación

Se estudia en profundidad el diseño de una aplicación de cada uno de los bloques, siendo las elegidas la inscripción en una actividad, el nuevo préstamo y la inserción de un nuevo juego en la ludoteca.

### *A. Inscripción a nueva actividad*

En el caso de la inscripción en una actividad, es importante establecer controles para evitar sobrepasar los límites marcados en la actividad. Igualmente, deben mostrarse mensajes adecuados que permitan al monitor dar al participante la mayor información posible.

El primer control que se realizará será seleccionar una actividad válida entre aquellas que ha definido el administrador del sistema. En el código se listarán las actividades que no han comenzado aún y se permitirá la selección de una de ellas, identificada por su ID, siempre que esta contenga plazas libres.

A continuación el monitor deberá introducir una nueva variable, la identificación del usuario que desea apuntarse en la actividad. Dicha identificación será validada a continuación, provocando un mensaje de error si no logra pasar la validación. En caso contrario, se procede a la inscripción del par visitante – actividad en la base de datos.

Por último, se realiza un último tratamiento sobre la información recogida. El programa cuenta la cantidad de usuarios inscritos en la actividad y lo compara con los valores predefinidos en la ficha de la actividad, pudiendo obtener dos posibles casos, que el usuario se inscriba y todavía queden plazas o que sea inscrito como reserva.

En función de la respuesta obtenida, el monitor recibirá un mensaje diferente cada vez, de forma que pueda informar al visitante de la mejor manera posible sobre el estado de su inscripción (Ver figura 3.8).

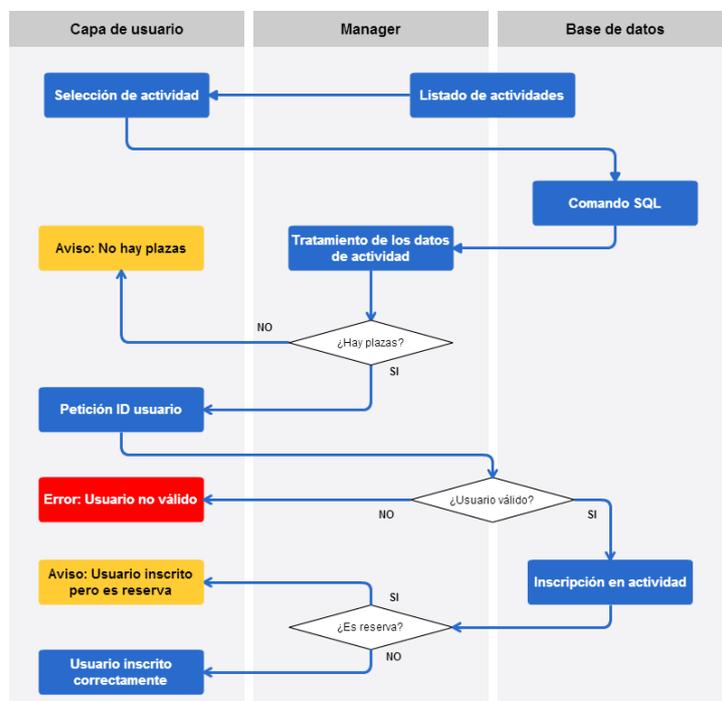


Figura 3.8 – Diagrama de flujo de la inscripción en actividad

### B. Nuevo préstamo

En el caso del nuevo préstamo, todas las validaciones que se realizan en segundo plano, sin intervención del monitor, son aquellas que tienen más importancia. El proceso de préstamos debe ser rápido, requerir muy poca entrada del usuario, ya que se realizan más de 2500 préstamos (y otras tantas devoluciones) a lo largo de 30 horas de evento y muchos de ellos en un tramo muy concentrado de tiempo.

En esta herramienta el monitor deberá introducir dos datos imprescindibles:

- Código del juego a prestar

- Número de usuario al que se realiza el préstamo

Ambas acciones pueden realizarse en una misma llamada, sin necesidad de requerir más trabajo por parte del encargado, momento en el cual empieza el trabajo del programa.

En primer lugar se comprobará si el usuario ya tiene un juego en préstamo, dado que en caso afirmativo debe emitirse un mensaje de error y finalizarse la aplicación. Dicha comprobación se realizará mediante una llamada a la tabla de préstamos en la que se buscarán todos los préstamos asignados a esa ID de usuario que no tengan hora de devolución asignada.

En caso de que la consulta devuelva un valor vacío como respuesta se pasará a la siguiente comprobación, buscando el juego en la tabla de préstamos por si el sistema lo tuviera asignado ya a otro usuario. De ser así, se finalizaría también el proceso, emitiéndose el correspondiente mensaje de fallo. Este punto de control contempla la mejora de la devolución automática del juego y una inscripción en el log indicando la anomalía, de forma que pudiera avisarse al usuario que no realizó la devolución correctamente de la necesidad de llevar a cabo el proceso.

Si pasa ambos puntos de control, el siguiente paso es realizar la inscripción en la tabla de préstamos, anotando la hora de préstamo para poder llevar un control del tiempo de juego. Se enviará igualmente un mensaje afirmativo al monitor y se le ha de devolver a la pantalla de préstamo, con el fin de agilizar el paso del siguiente usuario.

### *C. Inserción de juego*

A la hora de insertar un nuevo juego en la ludoteca, se procederá a la generación automática del código que lo identificará a partir de ese momento. Los datos que requerirá el formulario son aquellos puramente descriptivos, así como la identidad del propietario del juego.

Dicha identidad se facilitará por medio de un desplegable que se alimentará de una nueva tabla creada a tal fin, denominada 'propietarios', que contiene una id, el nombre del propietario y un código único que lo identificará, y que corresponderá con las dos primeras cifras del código del juego.

Con este dato ya es posible montar el código único para la copia del juego. Se establecerá un control para confirmar que dicha cuenta no supera el número máximo de juegos para ese código de propietario (que es de mil) y en caso de tener margen se procederá a concatenar el identificador de propietario con el número de juegos obtenido para obtener el código del nuevo juego. Para terminar se realizará la inserción del nuevo juego en la tabla de ludoteca con los datos proporcionados, finalizando así el proceso.

En el próximo capítulo se podrá estudiar la implementación de esta primera versión de la herramienta. Por otra parte, más adelante en este capítulo se hablará acerca del diseño gráfico de la misma.

### **3.3 – Diseño funcional – Segunda versión**

Tras realizar la implementación de la primera versión, que se ha desgranado en el apartado anterior, se observan una serie de deficiencias, campos de posible mejora y zonas que deben ser optimizadas. Dado que como explicamos en el apartado 3.1 se ha optado por un desarrollo incremental, dichas necesidades justifican la creación de una segunda versión.

A pesar de que los cambios más importantes se llevarán a cabo en la parte de implementación y en la forma de estructurar el código, hay varios cambios a nivel de diseño funcional que merecen ser analizados. Los cambios en la base de datos son los más importantes, por lo que se analizarán primero.

#### **3.3.1 – Modificaciones en la base de datos**

##### ***A. Tabla Ludoteca***

La primera deficiencia que se observa se encuentra en la tabla ‘ludoteca’. Al existir en ocasiones juegos que cuentan con varias copias dentro de la ludoteca, se advierte que esto obliga a introducir datos duplicados, con filas que son idénticas entre si salvo en los campos de estado y código. Además, dado que en esta nueva versión se habilitará la sección de estadísticas, evitar las duplicidades permitirá un mejor cálculo en esta sección, eliminando controles.

La solución encontrada pasa por crear una división en dos tablas. En primer lugar una tabla de “juegos”, que contenga toda la información común propia del juego, que no es susceptible de variar entre copias. Los registros de esta tabla irán diferenciados por su propia ID y se crearán controles para evitar la duplicidad de registros. A esta tabla irán referenciadas las tablas de autores y editoriales, pues a pesar de que un juego puede haber sido editado por más de una editorial, se decide mantener únicamente aquella que comercialice en este momento el material en España o bien en el caso de los juegos importados aquella que lo editó en el país de origen.

La segunda tabla será denominada “copias” y tendrá una referencia a la tabla de juegos. Esta tabla es la que se encontrará vinculada a la de préstamos o tablas secundarias como la de estado y contendrá toda la información que pueda variar entre una copia y otra del mismo juego, como puede ser su código. A pesar de que el código es un valor único que podría ser utilizado como clave primaria de la tabla se decide

crear un registro de IDs similar al usado en el resto de las tablas para mantener la estructura de la base de datos

Relacionada con esta tabla de copias y con la intención de mostrarlo en la parte web de la aplicación como información de interés se crea una relación entre copias e idiomas. Dicha relación se vincula a la tabla de copias dado que la ludoteca cuenta con juegos que tienen diferentes idiomas entre las distintas copias. En este caso se decide utilizar una estructura similar a la usada con la tabla de autores, dado que un juego puede tener sus instrucciones en más de un idioma y un idioma estar presente en más de un juego.

### ***B. Tabla Monitores***

Por otra parte, se decide crear una tabla con datos para los monitores, con el fin de permitir accesos individuales a la herramienta. Con esta medida se pretende blindar el acceso al gestor, estableciendo dos niveles de usuario, uno de administración y otro de uso normal. La tabla de monitores tendrá la siguiente estructura:

<b>Monitores</b>	
<b>ID_Monitor</b>	<b>INT, Clave principal</b>
Nombre	VARCHAR (50)
Apellidos	VARCHAR (80)
User	VARCHAR (30)
Password	VARCHAR (40)
Admin	TINYINT
Presencia	TINYINT

Figura 3.9 – Tabla de monitores

El campo de presencia permite denegar el acceso sin borrar el usuario por parte de la administración. Por otra parte, el campo 'password' se guarda encriptado, de forma que cualquier eventual acceso a los datos no otorga de manera automática las contraseñas de todos los usuarios. Finalmente, el campo 'user' será el encargado de guardar el usuario con el que el monitor realizará el acceso a la herramienta.

Esta nueva tabla permite un nuevo campo en la tabla de préstamos, añadiendo la posibilidad de 'reserva' de material por parte de los monitores para la realización de actividades, una demanda de la organización del evento tras la prueba de la primera versión y que también justifica la creación de esta tabla.

### ***C. Otras tablas***

Para terminar, se procederá a la creación de dos nuevas aplicaciones dentro de la web del evento, un formulario de contacto que permita mandar comentarios a la

organización y una pequeña sección que selecciona un juego cada día y lo recomienda, y que requieren la creación de dos nuevas tablas, una llamada 'feedback' que recoja los comentarios vertidos en la web y otra llamada 'daygame', que guarda los datos del juego mostrado ese día y sirve para determinar si debe saltarse a otro juego o no. Dichas tablas no tienen relación con el resto y salen de la estructura central de la base de datos.

### 3.3.2 – Modificaciones en la estructura de aplicaciones

A la hora de diseñar la estructura de las aplicaciones se realizan pocos cambios, todos ellos obligados por los cambios en las tablas y la forma en la que se implementarán los diseños creados.

Con el fin de reducir la cantidad de información que se envía y se recibe en cada petición (más de 6000 únicamente por el uso normal de las aplicaciones del gestor) se busca implementar los diseños mediante llamadas AJAX, insertando el retorno generado en pequeñas áreas de notificación. Estos cambios se detallarán en el próximo capítulo.

En muchas ocasiones se reiniciarán los campos de datos por medio de JavaScript, evitando la descarga de páginas completas con todo su contenido. Esto permitirá ganar en agilidad a la hora de procesar la información y reducirá la cantidad de datos transmitidos, una necesidad debido a las condiciones de conexión del espacio. En el próximo capítulo se profundizará sobre estos datos.

Con el final de la segunda versión se da por finalizado el proyecto en cuanto a desarrollo se refiere, obteniendo una herramienta plenamente funcional que cumple con los objetivos marcados al inicio de las diferentes etapas que componen el proyecto.

### 3.4 – Diseño del interface gráfico

El trabajo que se ha desarrollado en los apartados anteriores no sirve de nada si no se puede presentar la información de manera clara y accesible, de forma que hasta aquellos usuarios que no cuenten con experiencia de uso en la herramienta puedan desenvolverse con facilidad.

Clint Eccher, en su libro 'Diseño Web Profesional' define las siguientes filosofías de diseño orientadas al apartado gráfico de la página: [10]

- **Filosofía del uso:** "Usabilidad" es un término universal que se puede utilizar para describir cualquier sitio. Representa la facilidad con la que el usuario puede encontrar y procesar información y llevar a cabo ciertas tareas.

La filosofía de la usabilidad entiende este término en su sentido más amplio. Pretende que los sitios sean más fáciles de utilizar por todos los miembros de la comunidad de Internet.

Un método que se utiliza para llevar a cabo este término es despojar a un sitio de todos sus elementos no esenciales, lo que incluye eliminar la mayoría de sus imágenes, si no todas.

- **Filosofía multimedia:** En el otro extremo del espectro del diseño se encuentra la filosofía multimedia. Los sitios multimedia utilizan animación, audio y vídeo para hacerlos más interactivos.
- **Filosofía del ensamblaje:** 'Ensamblaje' puede entenderse como la filosofía de unir imágenes, texto y funcionalidad, así como los formularios, para construir sitios atractivos con imágenes y con un uso más rápido y flexible.

Estos sitios combinan lo mejor de las filosofías de la usabilidad y multimedia, por lo que consiguen diseños profesionales que pueden ver casi todos los usuarios actuales de Internet. El ensamblaje no solo complementa la funcionalidad de un sitio, sino que también permite a los diseñadores utilizar técnicas que la industria del diseño gráfico ha tardado décadas en perfeccionar sin sacrificar rendimiento.

Teniendo en cuenta la descripción que el autor hace de las diferentes filosofías, parece que la elección obvia es la filosofía del ensamblaje. Sin embargo, dadas las características del proyecto donde, como ya se ha comentado, prima la usabilidad y una curva de aprendizaje rápida, se ha optado por adoptar una filosofía de la usabilidad.

En base a eso se opta por establecer un menú lateral, que permite una mayor escalabilidad a la hora de añadir nuevas secciones si futuras iteraciones de diseño lo hicieran necesario, junto con un gran área de información en la que poder mostrar los contenidos de manera clara como se puede ver en la Figura 3.10:

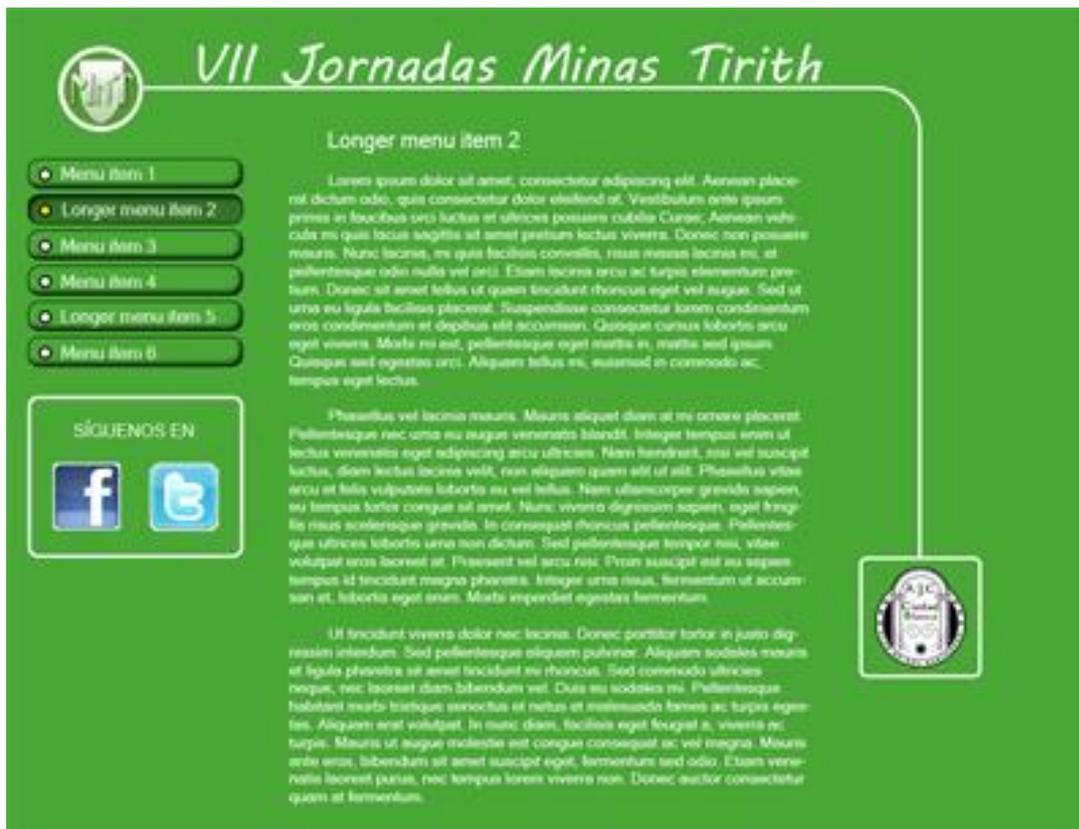


Figura 3.10 – Maqueta de la web según la filosofía de la usabilidad

Sin embargo, si bien esta filosofía se entiende indispensable para la parte privada de la web, aquella que implementará el gestor del evento, se estima que de cara al frontal de la web resulta poco atractiva para los usuarios potenciales de la misma.

Es por eso que para la segunda iteración en la implementación se realiza una mezcla entre ambas filosofías, manteniendo un estilo basado en la usabilidad en la parte privada de la web e implementando un nuevo frontal basado en la filosofía del ensamblaje.

El cambio de imagen es evidente cuando se observa la Figura 3.11, donde puede apreciarse que las imágenes y los bloques de redes sociales ganan peso, al igual que los patrocinadores del evento, en detrimento del texto puro. De esta forma, el usuario que visita la web no se ve abrumado por una gran cantidad de texto, si no que rápidamente puede contar con una serie de imágenes que lo colocan en situación, disponiendo a su vez de opciones para ver la información que antes se mostraba en la página principal.



Figura 3.11 – Maqueta de la web según la filosofía del ensamblaje

Sin embargo, el objetivo del gestor no es resultar atractivo al usuario, al contrario que la web pública, si no resultar práctico para los monitores. Es por eso que un cambio de filosofía de diseño en este apartado sería contraproducente, pues al tener que cargar las imágenes la transferencia de datos sería mayor y podría sobrecargar la red disponible.

## **4- Implementación**

A la hora de realizar la implementación del diseño visto en el anterior capítulo de este proyecto, y una vez elegida una metodología de diseño en espiral, en este capítulo se dividirá la implementación en las dos secciones en las que fue realizada, con las consideraciones previas pertinentes.

En primer lugar es necesario recordar que las premisas básicas del proyecto son la sencillez y la fluidez. Por eso mismo, a la hora de implementar el diseño en su primera etapa se ha optado por realizar una estructura lo más sencilla posible, que permita obtener los primeros resultados rápidamente para que estos sean evaluados y tenidos en cuenta en la segunda iteración del proyecto.

Por otra parte, con el fin de que el código pueda mejorarse posteriormente sin recurrir a un nuevo estudio completo del mismo, este debe estar correctamente documentado en todas sus secciones, con comentarios de texto que ayuden al seguimiento del código y expliquen la funcionalidad de los distintos bloques que lo componen.

Finalmente, la estructura de directorios debe seguir estas mismas premisas, permitiendo separar las diferentes secciones con claridad y generar independencia entre las mismas, de forma que el mantenimiento posterior y los cambios que se realicen en las posteriores iteraciones sean lo más sencillos posible.

Con estos puntos en mente, y una vez finalizada la primera fase de diseño, se procede a realizar la primera iteración del código.

### **4.1 - Implementación: Primera versión**

Al realizar la implementación del diseño obtenido, esta primera versión será más costosa en tiempo y esfuerzo que la segunda, ya que el desarrollo de la misma únicamente contará con la base del diseño realizado, pero no existirá ningún código previo. Por ese motivo este apartado se centrará en la implementación de base, mientras que en secciones posteriores veremos las mejoras sobre el código ya escrito.

#### **4.1.1 - Implementación de la base de datos**

Al igual que hemos hecho durante la fase de diseño, la base de datos centra el primer esfuerzo en la implementación del proyecto, al ser la base sobre la que se construirán el resto de piezas del mismo. Es por eso que la primera labor que se llevará a cabo será la transformación de la tabla relacional diseñada en sentencias SQL de creación de tablas, que generen la estructura de las mismas y sus relaciones, no así sus contenidos por el momento.

## A. Estructura

Para llevar a cabo esta tarea se abordan las tablas una por una, con especial cuidado en los nombres y tipos de datos estudiados durante la fase de diseño. El uso del editor de texto de 'MySQL Workbench' permite ver con claridad las diferentes partes en las que se dividen las sentencias antes de ser ejecutadas en la base de datos, minimizando así el número de errores cometidos en el código. Es común en esta parte del desarrollo encontrar problemas de sintaxis (por ejemplo, el olvido de comas o de paréntesis) o el uso de palabras restringidas dentro del esquema MySQL, que son aquellas que el propio motor de la base de datos restringe por ser necesarias para la realización de sus funciones, como *'date'*, *'time'*, *'check'*... Una lista completa de las palabras reservadas en MySQL puede encontrarse descrita en la web de referencia de MySQL. [12]

Una vez creadas las sentencias MySQL tienen un aspecto como el de la tabla 'ludoteca', que se presenta a continuación:

```
CREATE TABLE ludoteca (  
  ID_Juego INT NOT NULL AUTO_INCREMENT,  
  PRIMARY KEY (ID_Juego),  
  nombre VARCHAR(50) NOT NULL,  
  autor VARCHAR (30),  
  codigo INT NOT NULL,  
  minjug TINYINT,  
  maxjug TINYINT,  
  tiempo INT,  
  descripcion VARCHAR(2000),  
  minage TINYINT,  
  estado INT,  
  editorial INT,  
  FK_estado FOREIGN KEY (estado) REFERENCES estado(ID_estado),  
  FK_editorial FOREIGN KEY (editorial) REFERENCES editoriales(ID_editorial));
```

Nótese la definición de las dos claves externas en los campos 'estado' y 'editorial', que son referenciados a los índices únicos de las tablas correspondientes. De esta forma, el valor de estos campos deberá tomar obligatoriamente uno de los valores del conjunto de claves de esas tablas, fallando las sentencias de inserción de datos en caso contrario. Esto obliga a mantener un orden en la creación de las tablas, comenzando por aquellas que no tienen referencias externas y finalizando por aquellas que si las precisan, una vez las tablas a las que están vinculadas ya han sido creadas. De igual manera se habrá de proceder con la inserción de los datos, para evitar fallos en las sentencias.

Estas sentencias de creación de tabla son lanzadas al servidor a través de llamadas o *'queries'*, bien con el uso de PHP o bien mediante la herramienta gestora de

bases de datos vista anteriormente, MySQL Workbench. Esta última opción fue la escogida para este proyecto, ya que este tipo de herramientas permiten tener una vista actualizada del estado de las tablas, tanto a nivel de estructura como de contenido.

### *B. Carga de datos*

Tras crear el esquema de datos es posible introducir los primeros datos. En esta fase del proyecto se evalúa la posibilidad de comenzar con la inserción de un conjunto de datos de prueba, con el fin de realizar las verificaciones posteriores, o bien los datos definitivos, evitando el desarrollo de un conjunto de datos que sólo se utilizará durante una fase del proyecto.

Una vez estudiadas las implicaciones de cada opción se opta por una solución intermedia, introduciéndose como datos definitivos aquellos que no serán susceptibles de cambio más adelante (estados, editoriales, ludoteca, autores o encuesta) y dejando aquellas tablas más dinámicas (asistentes, actividades, etc...) para el momento en el que se compruebe la funcionalidad de los módulos que deberán insertar, editar y modificar esos datos.

Requiere especial atención la inserción de los datos de la tabla ludoteca, dado que esta es la de mayores dimensiones de la base de datos en esta fase del proyecto. Para realizar la carga de los datos se procede a la creación de una hoja Excel, donde cada columna representa un tipo de dato y cada fila un registro diferente. Con los datos facilitados por la asociación se comienza por realizar la escritura de todos los datos, generando un total de 481 registros.

El siguiente paso consiste en separar aquellos datos que han de ser introducidos en las tablas auxiliares. Para ello, se toman estas columnas, copiándose en una hoja diferente y eliminándose aquellos registros que están duplicados. Una vez hecho esto, es posible guardar dicha hoja en formato .csv, separado con comas, que es fácilmente editable hasta convertirse en sentencias SQL válidas para el gestor de datos. De esta manera se cargan por separado las tablas de autores, estado y editoriales.

Posteriormente, de nuevo en la hoja principal de la tabla de ludoteca, se sustituyen los valores de dichas columnas auxiliares por los correspondientes identificadores que se han asignado a cada una de ellas, de forma que el dato se vincule al registro a través de esta clave única. La sustitución es directa en el caso de los campos 'estado' y 'editorial', no así en el caso de las relaciones entre juego y autor, ya que como se comentó en la fase de diseño, se contempla la posibilidad de que un juego tenga varios autores o un autor varios juegos. Es por ello que en este caso se ha de insertar una tabla intermedia, que relacione ambos datos y que por lo tanto habrá

de generarse una vez cargada la tabla ludoteca, al ser necesarias las ID asignadas a cada juego para este paso.

Por último, se repite el paso realizado con las tablas más pequeñas, generando un .csv que permita separar los campos por comas y ser editado fácilmente para transformarlo en sentencias SQL válidas, que posteriormente son introducidas en la base de datos. En este caso puede darse un pequeño problema con los caracteres especiales, como las comillas simples presentes en algunos nombres o descripciones, que pueden causar conflictos con el lenguaje SQL. Editando estos campos y ‘escapando’ dichos caracteres el problema se soluciona.

De igual manera a como se ha procedido con la tabla ‘ludoteca’ se cargan las demás tablas que van a contener datos definitivos, dejando el uso del resto de tablas para la fase de pruebas.

#### **4.1.2 – Implementación de la estructura**

Una vez se ha terminado con la creación del esquema de la base de datos y la carga de los registros escogidos es momento de comenzar a desarrollar el código que dará vida a la página. Para ello, se separa el trabajo en diferentes módulos, uno por cada aplicación que se va a realizar y otro más para la pantalla principal de administración, con el fin de abordar el problema de manera escalonada.

##### **A. Módulos**

La estructura de los diferentes módulos sigue en todos los casos el esquema estudiado durante la fase de diseño, comenzando por la generación del código HTML básico con el que visualizar la página principal del módulo, pero sin funcionalidad. Posteriormente se aplica el código PHP, dotando de interactividad a la página y finalmente se realizan las conexiones con el resto de páginas del módulo y se aplican los estilos para su correcta visualización.

Los módulos se desarrollan de manera lineal a lo largo del flujo de datos, comenzando por la primera pantalla que visitará el usuario, para luego dar paso a las diferentes casuísticas en el tratamiento de los datos y por último a la presentación del resultado obtenido.

Como módulo interesante a estudiar, por salirse ligeramente del esquema del resto, se encuentra el módulo que dibuja la página principal del gestor. Dicho apartado de la web se encargará también de mostrar el resto de páginas gracias a la gestión que se realiza de la URL de la página, por la cual se envía como parámetro el destino escogido dentro de la web y, en caso de no recibirse nada, se muestra el panel principal. Esta funcionalidad se trata mediante una sentencia ‘switch-case’ que recoge

el atributo de dicho parámetro y lo trata, invocando un script u otro en función el valor obtenido.

La ventaja de obrar de esta manera radica en el ahorro de código que produce, ya que la cabecera, el menú y el marco de la página se genera una sola vez en este módulo, mientras que el resto únicamente llenan de contenido el apartado central de la página. Gracias a esto el mantenimiento y la edición futura de la web serán más sencillos, ya que le otorga independencia a los módulos mientras que los cambios en el script principal se volverán globales.

En relación a los otros módulos, todos siguen el esquema típico de una web por formularios, donde se presentan los diferentes campos que el usuario puede tratar, bien como campos de texto o bien como desplegados de selección, para, una vez rellenos, enviarse a una página de tratamiento que comprobará la coherencia de los datos y los procesará, interactuando con la base de datos a través de llamadas SQL, antes de retornarle el resultado al usuario.

## *B. Seguridad*

En este sentido entran en consideración las primeras medidas de seguridad implementadas en el proyecto, como es la verificación de los campos o el tratamiento de las cadenas de datos para evitar las inyecciones SQL, o dicho de otra manera, el acceso o edición no controlado a los registros de nuestra base de datos, así como cifrados de seguridad para los campos sensibles, como puede ser el campo de contraseña en la gestión de los usuarios, para el que se implementa un cifrado MD5 (nativo del propio código PHP), que se caracteriza por ser un cifrado de una sola dirección.

Este tipo de seguridad es escogida al ser más robusta y complicada de romper por parte de ataques externos. El dato encriptado es el que se guarda en la base de datos, de forma que cualquier acceso no deseado a la misma solo dará como resultado una lista de caracteres indescifrables y no un dato interpretable por el atacante. En este caso, la verificación del dato se realiza aplicando el mismo algoritmo de encriptación a la entrada del usuario y comparando con el registro almacenado. Una coincidencia dará un resultado positivo, mientras que las discordancias serán síntoma claro de un fallo.

Para la elección del algoritmo de encriptación MD5 se ha tenido en cuenta su rapidez de cálculo (clave para obtener una aplicación fluida) y la poca seguridad requerida en esta aplicación, al tener un ámbito de aplicación bastante reducido y poco propenso a ataques. Dicho algoritmo produce resultados de 128 bits, lo que otorga una resistencia a la preimagen de  $2^{128}$  y una resistencia a las colisiones de  $2^{64}$ . [13]. Dichos valores, pese a ser bajos si se comparan con otros algoritmos de encriptación, se entienden suficientes para esta aplicación.

Por otra parte, en lo relativo a la verificación de los datos, cada uno de los scripts que trata la información al recibirse desde el formulario ejecuta unas pautas de comprobación, que incluyen controlar que el tipo de dato es el esperado o que los rangos se encuentran dentro de los valores lógicos para el dato. Una vez verificados, los datos se incluyen en una sentencia SQL que se envía a la base de datos, de la cual se obtiene el resultado para el usuario. Dicho resultado se pinta en la pantalla, junto con un botón para volver de nuevo al inicio de la aplicación.

### 4.1.3 – Implementación de la interface gráfica

A continuación, en el apartado de la implementación gráfica, se hace uso de una maqueta diseñada por medio de software de edición de imágenes, como Photoshop. Dicha maqueta permite tener una visión previa de lo que se desea hacer con la web, pudiendo medirse los diferentes componentes como las áreas destinadas a cada apartado o los menús, simplificando así la edición del código CSS.

Para este apartado la clave se encuentra en separar las diferentes secciones que se pueden ver en la página en ‘bloques contenedores’. Dichos bloques se definen mediante la etiqueta HTML ‘<div>’, que luego se dota de características vinculándola a CSS a través de ‘identificadores’ (id) o clases (class). La diferencia principal entre identificadores y clases es que los primeros deben ser únicos dentro de la página, por lo que no pueden asignarse a más de un bloque. Sin embargo, las etiquetas de clase son útiles cuando se quiere dotar de estilo a varios elementos según un mismo criterio. Un ejemplo de la definición del código podría ser el siguiente:

```
<div id='navegador'>  
  <div class='boton_menu'>Boton 1</div>  
  <div class='boton_menu'>Boton 2</div>  
  <div class='boton_menu'>Boton 3</div>  
  <div class='boton_menu'>Boton 4</div>  
</div>
```

El bloque ‘navegador’, correspondiente a la barra de navegación, es un elemento único que no va a repetirse en la página, por lo que se le puede asignar un identificador. Sin embargo, al haber varios botones de menú, estos adquirirán su estilo por medio de etiquetas de clase.

Una vez se han etiquetado todos los elementos, asignarles estilo es una tarea sencilla con CSS. Simplemente es necesario vincular atributos a las etiquetas, de forma que los elementos las adquirirán. Un ejemplo de aplicación sobre el código visto antes sería el siguiente:

```
#navegador {height: 300px; width: 200px; position: absolute; top: 0px; left: 0px;  
border: solid 1px #000000}
```

```
.boton_menu {position: relative; height: 25px; background-color: #00ff00; color: #ffffff; margin-bottom: 5px; display: block;}
```

De esta forma, el navegador será una caja de 200 píxeles de ancho y 300 de alto, posicionada en la esquina superior izquierda de nuestra pantalla y un borde negro de un píxel de grosor. Por otra parte, cada botón de menú tendrá una altura de 25 píxeles, un fondo de color verde, letra de color blanco, un margen inferior de 5 píxeles y ocupará todo el ancho de la línea. La Figura 4.1 ilustra este ejemplo:

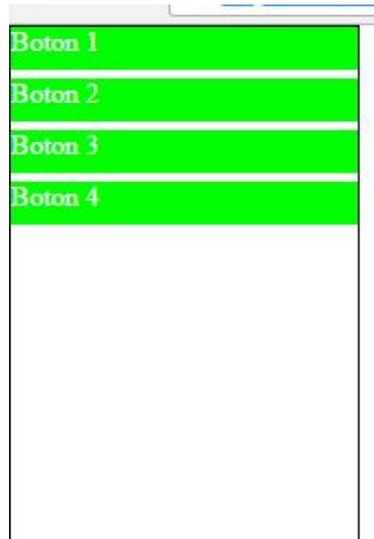


Figura 4.1 – Tratamiento de bloques mediante HTML y CSS

Mediante el uso de la maqueta y Photoshop se toman muestras del tamaño de los elementos, su posición respecto al resto y los colores decididos, completándose la fase de diseño una vez se han creado las diferentes etiquetas y asignado a ellas los atributos correspondientes.

#### 4.1.4 – Pruebas de funcionamiento

Finalmente se realiza la fase de test de los diferentes módulos. Para ello, se prepara una batería de inserciones de usuarios, préstamos y devoluciones, tanto con datos correctos, para comprobar la carga de trabajo en la base de datos y el buen funcionamiento de los módulos como con registros incorrectos entre ellos, para asegurar el buen funcionamiento de las medidas de seguridad.

Para realizar la secuencia se aborda en primer lugar una inserción registro a registro de un total de cuarenta participantes, desde dos equipos y de manera simultánea, registrando los retornos incorrectos para cotejarlos con los esperados y comprobando así también que el sistema soporta la carga de trabajo en paralelo desde diferentes equipos.

En dichas pruebas se encuentra un error en el envío de datos a través del método GET. Dicho método hace uso de la URL para realizar el envío de los diferentes campos del formulario. El bug localizado hace que en ocasiones dicho envío no se haga de manera correcta y el gestor no sea capaz de procesar la información, dejando el registro en blanco.

Para solucionarlo se opta por cambiar al método POST, que envía el dato por otro canal invisible al usuario, encontrándose que de esta manera el dato se recibe y carga correctamente en la base de datos.

Posteriormente, se realizan pruebas de préstamo y devolución a esos participantes, comprobando que los límites implementados funcionan correctamente. Dichos límites incluyen entre otras la imposibilidad de prestar un juego ya en préstamo o el préstamo de dos juegos a un mismo participante. Bajo todas esas casuísticas el programa se comporta correctamente.

Además, se comprueba la inscripción de participantes en las actividades, comprobando que los avisos de retorno se ajustan a los parámetros establecidos de mínimo y máximo número de jugadores, así como la gestión de los reservas. También en esta ocasión el comportamiento es normal y los resultados obtenidos se ajustan a los esperados.

Concluida la fase de test sin encontrar más fallos significativos (más allá de pequeñas correcciones en el código) se da por cerrada la primera iteración del proyecto, siendo esta completamente funcional. Los ejemplos de uso de los módulos más importantes podrán verse en el próximo capítulo, junto con explicaciones detalladas de los mismos.

## **4.2 – Implementación: Segunda versión**

Una vez se ha podido probar sobre el terreno la primera versión del gestor y se han analizado las posibles áreas de mejora se decide en la fase de diseño que, con el fin de mejorar la carga de datos que debe soportar la red, se ha de rediseñar la estructura de la página para adaptarla a peticiones AJAX. Además, se acompañará de un rediseño de la web para hacerla más visual y de la adición de nuevos módulos que incrementen su funcionalidad.

### **4.2.1 – Modelo basado en peticiones AJAX**

De todos los cambios, el de estructura es el más grande por la carga de trabajo que conlleva. Si bien no obliga a reescribir todo el código sí que hará necesarios varios cambios importantes en los diferentes pasos de cada módulo.

## ***A. Implementación***

En primer lugar, en la pantalla principal de cada módulo, se elimina el paso normal de parámetros por medio de formulario. En su lugar, se mantiene la definición de los campos, con la salvedad de que el botón de envío desata una función JavaScript que invoca la llamada AJAX.

Dicha función recoge los parámetros de los campos del formulario accediendo a través de DOM por su ID de referencia y los agrupa en una URL con la que invocará el script PHP que procesará la información. La llamada espera entonces la respuesta del script para vaciar los campos del formulario y escribir en una sección lateral diferente el resultado de la consulta.

El principal ahorro de este método viene al tratar la respuesta del script PHP. Dicha respuesta es un pequeño código HTML que no conlleva la carga total de la página, sino una pequeña ventana de toda la página, compuesta por texto plano. Además, el color de fondo de la ventana y el estilo aplicado ya se encuentra cargado en la clase CSS preexistente, por lo que el tamaño de los datos transmitidos se reduce enormemente. Además, al limpiar ya los elementos del formulario y no cambiar de página previene de recargas adicionales en caso de querer realizar de nuevo la misma función.

## ***B. Comprobación de la mejora obtenida***

Una vez implementados los primeros módulos se utiliza la herramienta para desarrolladores de Google Chrome para medir el volumen de datos de la carga. Dicha herramienta arroja que la carga total de la página conlleva un total de 192KB de datos transferidos (Ver Figura 4.2), que con dos cargas mínimas necesarias en el esquema anterior para realizar cualquier acción supondrían un total de 386KB transferidos por carga.



Figura 4.2 – Carga completa de la página

Sin embargo, al realizar cualquier acción (por ejemplo, al registrar un nuevo usuario, uno de los formularios más grandes de la página) dentro del nuevo esquema, la transferencia de datos se queda muy por debajo de 1KB, siendo en la mayoría de las ocasiones de unos pocos bytes de información, tal y como se muestra en la figura 4.3.

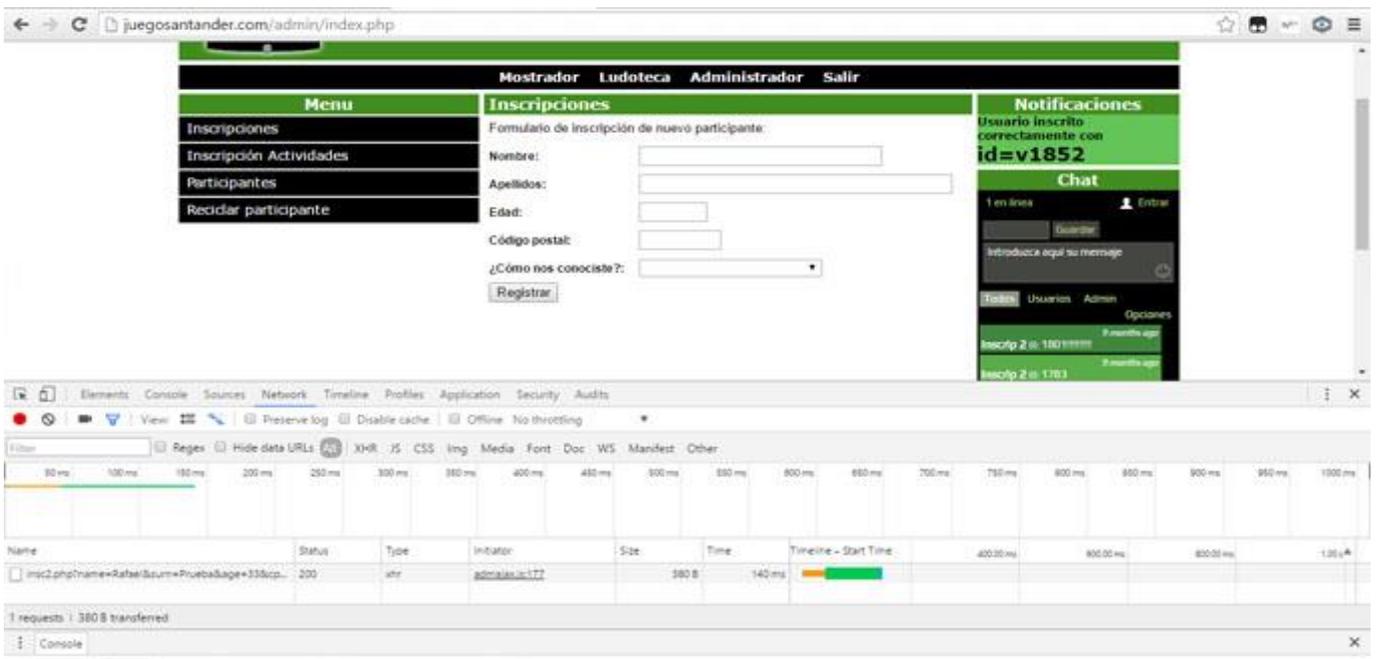


Figura 4.3 – Carga en el nuevo modelo (380B transferidos)

La mejora en este apartado es sustancial si se compara con el volumen de peticiones que van a registrarse a lo largo del evento. Las 1850 inscripciones, 2700 préstamos (y otras tantas devoluciones) o las 250 inscripciones a actividades estimadas generarían un mínimo de 7500 peticiones, o lo que es lo mismo, aproximadamente 2,9GB de datos transferidos a lo largo del evento a través únicamente de redes

móviles. Con el nuevo sistema, dicho volumen de transferencia se reduciría hasta aproximadamente 3MB de información, aproximadamente mil veces menos que en el caso anterior y mucho más manejable con los recursos disponibles.

Dicha mejora se extiende también a la carga de las páginas desde el menú principal, con el fin de evitar la carga repetida de las cabeceras o de las áreas comunes, aumentando así el ahorro obtenido, si bien estas cifras pasan a ser insignificantes al compararlas con las ya vistas. Este cambio conlleva modificaciones en la forma de gestionar el módulo principal de la web, que ya no se registrará por el paso de parámetros desde la URL, al seguir el sistema de llamadas de AJAX.

#### 4.2.2 – Módulo de comunicaciones en tiempo real

Esta reducción permite implementar en la web un chat con el fin de mejorar la comunicación entre los usuarios, para lo que se usará un plugin externo, al ser código ya desarrollado y fácilmente integrable en la herramienta. Tras analizar varios se opta por el ofrecido en <http://yourshoutbox.com/>, al ser fácilmente adaptable al espacio destinado para este elemento, requerir poco esfuerzo de integración y configuración, no consumir apenas recursos y ser gratuita en su versión base.

Para integrarla es necesario realizar una operación de registro en su web, tras la cual se asigna una clave para el uso de la aplicación. Añadiendo la siguiente línea de código en la sección donde se quiera mostrar y configurando las características de la caja contenedora se obtiene el resultado deseado, ilustrado en la Figura 4.4:

```
<script type="text/javascript"
src="//www3.yourshoutbox.com/shoutbox/start.php?key=[clave]"></script>
```



Figura 4.4 – Chat integrado

### 4.2.3 – Préstamo por lector de código de barras

Los cambios en el código abren la posibilidad a realizar una modificación importante en el módulo de préstamos, el cual se adapta al uso con lectores de códigos de barras. Para llevar a cabo esta mejora, además de los cambios en código necesarios, se añaden también códigos de barras al material prestable y a las papeletas de participante.

Este cambio agiliza las operaciones de préstamo y devolución con un cambio mínimo en el código consistente en la codificación de valores para juegos y usuarios separados, de forma que no se crucen los datos por error. Para ello se inserta un valor alfabético al inicio del código de visitante, del que los códigos del material carecen. Dicha codificación se aplica posteriormente en la creación de los códigos de barras que se asignarán tanto a las papeletas de usuarios como al material.

A la hora de implementar esta funcionalidad es posible aprovechar la mayoría del código preexistente, realizando únicamente modificaciones menores. Estas consisten en la eliminación por código del carácter especial que se inserta en las papeletas de los participante, tal y como se ha explicado anteriormente. Además, se adapta el script para que el retorno de carro introducido automáticamente por el lector de códigos no haga saltar de forma indeseada la comprobación de los datos.

### 4.2.4 – Recarga de los datos en la DB

Por otra parte, las modificaciones hechas en la estructura de base de datos llevan a hacer una recarga parcial del contenido de la misma. De todas las tablas a modificar la más importante es nuevamente ludoteca, por el tamaño de la misma. En el momento de la edición esta tabla ha crecido hasta los 533 registros, los cuales descargamos ordenando los campos por el nombre del juego y almacenamos en un fichero, separando los campos por comas. Dicho fichero se cifra como .csv, para poder ser abierto de nuevo por una hoja de datos para su tratamiento.

Se realiza una copia completa de la hoja, separándola de la original y se eliminan los registros con nombre duplicado, de forma que podemos crear la tabla 'juegos' a partir de esta vista. En esta tabla se eliminan los campos propios de las 'copias', como son el código o el estado, y se asignan identificadores únicos a cada registro, que luego irán almacenados como 'id\_juego'.

Por otra parte, de la primera hoja crearemos la tabla 'copias'. En esta se eliminan los registros propios del juego (descripción, número mínimo y máximo de jugadores, duración, nombre...) y se crean dos nuevas columnas, una para el identificador de la copia y otra que relacione el registro de la copia con el de juego. Al haberse mantenido el mismo orden esta asignación es sencilla y no requiere un gran esfuerzo.

Finalmente, procederemos a la carga de los datos de manera similar a como se hizo en la primera versión, almacenando las tablas de nuevo en ficheros .csv separados por comas y realizando las modificaciones necesarias para convertirlos en sentencias SQL válidas que ejecutar en nuestra base de datos.

#### **4.2.5 – Pruebas de funcionamiento**

Los cambios realizados en el diseño de la base de datos obligan a realizar modificaciones en algunos scripts de código, con el fin de adaptar las sentencias de consulta a la estructura de las nuevas tablas. Estos cambios en el código no son importantes, pero obligan a realizar una nueva recarga de los datos en aquellas tablas modificadas y la realización de nuevas secuencias de test para garantizar la coherencia de la información y el buen funcionamiento de los módulos modificados.

Dichas secuencias de test eran obligadas de todos modos, ya que la modificación del esquema de funcionamiento de la web lo requería igualmente, por lo que no supone un esfuerzo añadido y se realizan de manera similar a como se hizo en la primera implementación del código, sin encontrar fallos significativos.

#### **4.2.6 – Modificaciones a la interface gráfica**

En el apartado de diseño gráfico es interesante señalar que se hace un cambio en la parte pública de la web pasando de la filosofía de la usabilidad, orientada a que los datos se muestren de la manera más clara y accesible posible sin importar tanto el estilo a la filosofía del ensamblaje, donde el apartado multimedia y las imágenes cobran más peso.

A la hora de implementar este apartado de nuevo es importante cuidar los tiempos de carga de la web, para que no sean demasiado largos y desanimen al visitante. Esto se consigue mediante dos vías: reduciendo la cantidad de peticiones hechas a la base de datos, de forma que el navegador no tenga que quedarse esperando a que se resuelvan y eligiendo de manera adecuada el formato de las imágenes a mostrar.

Para elegir el formato más adecuado para cada una de las imágenes se hace uso de Photoshop, tomando dichas imágenes de la maqueta realizada y separándolas nuevamente en bloques, de forma que se pueda tratar cada uno de ellos por separado. Esta herramienta cuenta con una opción dirigida especialmente a realizar esta tarea, dentro de la ruta 'Archivo -> Guardar para Web...' se abre una ventana que permite elegir entre diferentes formatos y calidades de imagen, limitando incluso la cantidad de colores disponibles y mostrando en todo momento el peso final de la imagen. Una buena elección para cada imagen puede llevar a conseguir ahorros superiores al 50% en el peso de cada imagen.

Se puede tomar como ejemplo el banner del panel de administración, que al guardarse como .jpg con el 50% de su calidad y aplicando la opción de optimización para web ocupa un total de 13,09KB, mientras que en formato PNG-8 con 24 colores no se aprecia una pérdida de calidad significativa y ocupa únicamente 5,9KB. Este ahorro, llevado a todas las imágenes implica una reducción importante del peso de la web y por lo tanto de su tiempo de carga.

Una vez comprobado que la web se muestra correctamente tanto en su parte frontal como en el gestor y que los módulos que la componen funcionan correctamente se puede dar por finalizada la implementación en esta segunda versión.

## 5- Casos de uso

### 5.1 – Jornadas Minas Tirith

El principal caso de uso y banco de pruebas de la aplicación son las Jornadas ‘Minas Tirith’ de juegos de mesa de Santander, un evento de ocio alternativo que se celebra con carácter anual en la ciudad cántabra, en el Palacio de Exposiciones y Congresos. Sus necesidades son las que se han tomado como ejemplo para el desarrollo de este proyecto y la opinión de su personal la que ha servido para mejorar la herramienta entre iteraciones.

Dicho evento reúne a cerca de 1800 participantes inscritos y pone a disposición de los mismos una ludoteca de más de 500 juegos y más de medio centenar de actividades, realizándose aproximadamente 2700 préstamos de material en sus cuatro días de duración. En los picos de afluencia, cerca de 700 personas se reúnen con más de 170 préstamos por hora realizados y similar número de devoluciones, todo ello gestionado por un grupo de 20 personas.

Es por eso que los procedimientos han de ser fluidos y esta herramienta cobra especial importancia, permitiendo llevar un control de los diferentes procesos y realizarlos con fluidez. En las pruebas realizadas se ha detectado una mejoría palpable en los diferentes departamentos, tanto en la gestión de los usuarios como del material o los préstamos, permitiendo ofrecer mejor información al usuario al encontrarse esta más centralizada. En la Figura 5.1 puede verse a una monitora en el puesto de ludoteca, haciendo uso del gestor durante la prueba.



Figura 5.1 – X ‘Minas Tirith’

En estas circunstancias es posible evaluar el funcionamiento de los diferentes módulos que componen la herramienta, al estar sometidos a condiciones de producción con un uso intensivo y poco margen de error, por ser un servicio que repercute en el visitante del evento y en su impresión del mismo.

En primer lugar analizaremos el puesto situado en la entrada del evento, donde se gestionan los registros de usuarios y las actividades. Dicho lugar no reporta una ganancia inmediata en el procedimiento de inscripción, que sigue llevando aproximadamente el mismo trabajo que con los métodos tradicionales, si en el resto de áreas, gracias a herramientas como el reciclaje de participantes, que permite asignar nuevos identificadores a usuarios que han perdido su acreditación de acceso (algo que ocurre 51 veces a lo largo del evento) como en la gestión de actividades, donde la herramienta les permite llevar un control claro de las plazas disponibles y los horarios de las mismas, con mensajes de avisos que informan del estado de la inscripción.

Además, la siguiente ventaja proviene de la gestión de estadísticas que permite calcular al vuelo datos clave como la edad media de los visitantes, los picos de trabajo en cada una de las áreas (lo que permite repartir mejor el trabajo) y el material más prestado, permitiendo así hacer previsiones para futuras ediciones. Además, la gestión de los datos de la encuesta que se hace a los usuarios permite enfocar mejor la publicidad del evento de cara a futuras ediciones y detectar aquellas acciones que han sido más efectivas.

En el puesto de control de ludoteca la ganancia es incluso mayor. El procedimiento de préstamo o devolución es mucho más rápido, al realizarse de manera asistida por la herramienta, que realiza comprobaciones que de otro modo deberían hacerse manualmente, como la comprobación de que un usuario no toma más de un juego prestado a la vez. Además, el uso de la identificación del material por código de barras agiliza el proceso, que puede realizarse en unos pocos segundos, mejorando la atención en los picos de trabajo y permitiendo gestionar un mayor número de peticiones.

El proceso a través de un gestor web permite colocar también más de un ordenador al mismo tiempo dedicándose a esta tarea, algo necesario en los picos de asistencia y que con carácter previo a la introducción de la herramienta no era posible, al depender de un único equipo que registrara los préstamos. Finalmente, el registro de material prestado permite llevar un mejor control del mismo y realizar un seguimiento de los tiempos de préstamo más habituales, con el fin de pronosticar también el uso del propio material.

Por otra parte se observa que la presencia de una web pública del evento atrae nuevos visitantes y que estos están mejor informados acerca de las actividades que se llevan a cabo en el mismo, buscando directamente aquel material que les resulta de interés o preguntando por las actividades del programa en el momento de la

inscripción. Para la toma de estos datos se tiene en cuenta la impresión del personal del evento con varios años de experiencia.

Finalmente, en el apartado de administración, el uso de las gráficas de estadísticas permite una redacción más rápida y fluida de las memorias de evaluación del evento, al mostrar la información ya procesada y evitando la realización manual de esta labor. La Figura 5.2 es un ejemplo de este trabajo, con la distribución de edades de los participantes de las X 'Minas Tirith'. Por otra parte, las herramientas de gestión de la base de datos, como la introducción de actividades, permite agilizar las labores previas al evento, pudiendo dedicar el tiempo ahorrado a otras tareas.



Figura 5.2 – Gráfica de distribución de edades.

En este punto es importante señalar la importancia de la información en tiempo real para el correcto desarrollo del evento, ya que durante el mismo se superó la previsión de asistentes y por parte de la organización se temió que la cantidad de papeletas de inscripción no fueran suficientes para el desarrollo del evento. El poder consultar la cantidad de usuarios inscritos desde cualquier lugar fue una ayuda inesperada para la gestión del mismo, permitiendo evaluar la magnitud del problema y tomar las medidas adecuadas en función de la progresión observada.

Cabe reseñar que durante el evento no se observaron más que pequeñas deficiencias en el uso de la herramienta, que pudieron solucionarse al vuelo sin interferencia para los usuarios. Dichas deficiencias se debieron a situaciones no contempladas en los test de prueba y se anotan como campos a mejorar en futuras iteraciones de la herramienta.

## 5.2 – Otros casos de uso

Si bien la herramienta parece muy focalizada al primer caso de uso, algo natural al haber sido el detonante de su desarrollo y principal campo de estudio para el mismo, la aplicación desarrollada puede utilizarse para otros casos de uso diferentes al principal:

- **Eventos afines:** El uso de la aplicación no se encuentra restringido al evento anteriormente explicado, pudiendo servir de base a eventos similares de igual manera. La adaptación en este caso sería sencilla, ya que es posible tomar la base de la herramienta, modificándose únicamente los contenidos.
- **Bibliotecas:** El software aquí desarrollado guarda similitudes con el usado en la gestión de bibliotecas, pudiendo ser utilizado de manera satisfactoria para tal fin, al ser una actividad que registra a sus usuarios y produce préstamos de un material inventariado. Los cambios necesarios para adaptar el programa a esta clase de actividad también serían mínimos, al no requerirse en principio funcionalidades adicionales específicas.
- **Videoclubs:** Si bien es un modelo de negocio en declive, de nuevo podría darse como un posible caso de uso de la herramienta desarrollada con modificaciones mínimas. Además podría suponer en este caso una ventaja añadida gracias al ahorro de costes subyacente, por ser una herramienta desarrollada con software libre.
- **Administraciones públicas:** Las administraciones públicas también pueden verse beneficiadas del uso de esta herramienta, principalmente los ayuntamientos, pues existen varios ejemplos de ayuntamientos con servicios de préstamo de material a colectivos registrados, como los de Vitoria, Zaragoza o Santander. En este caso sí que sería necesario realizar pequeñas modificaciones al programa, adaptando el formulario de inscripción al nuevo tipo de usuario, para admitir colectivos.

Como se puede observar, en general el programa podría ser usado por cualquier colectivo o entidad que realizara procesos de préstamo, con la funcionalidad añadida de la gestión de actividades, que puede ser de interés para varios de los grupos anteriormente citados.

## 6- Conclusiones

### 6.1 – Objetivos

Una vez cumplido el ciclo de vida del proyecto y comprobados los resultados obtenidos llega el momento de evaluar el cumplimiento de los objetivos marcados a lo largo del primer capítulo del proyecto, tarea que determinará el éxito o fracaso del mismo.

#### **A. Análisis y selección de las herramientas de desarrollo.**

A lo largo del segundo capítulo hemos estudiado las diferentes herramientas utilizadas para llevar a cabo el desarrollo del proyecto. Todas ellas, imprescindibles para el cumplimiento de los demás objetivos marcados, comparten una característica común, su pertenencia al conjunto de lenguajes *'open source'*, que no requieren el pago de una licencia para su utilización. De esta forma cumplen a la perfección con la política de reducción de costes necesaria para hacer viable la herramienta.

Además, el conjunto de soluciones encontrado permite abordar con éxito otros problemas clave, como son la posibilidad de llevar a cabo tareas de manera simultánea desde varios equipos y una baja carga de transferencia de datos, permitiendo el uso de redes móviles como soporte a los equipos.

Por este motivo se estima que las herramientas seleccionadas cumplen con los requisitos necesarios para dar este objetivo por bueno, siendo las soluciones óptimas para la resolución del problema.

#### **B. Escoger la metodología de diseño que mejor se adapte a las necesidades del proyecto.**

La importancia de escoger la metodología de diseño adecuada puede parecer insignificante en comparación con las otras áreas de trabajo que ha supuesto el proyecto. Sin embargo, una correcta elección puede ahorrar mucho tiempo de desarrollo mientras maximiza el resultado obtenido.

La metodología que se ha elegido ha resultado ser clave para el cumplimiento del resto de objetivos, pues ha permitido obtener una versión funcional de la herramienta desde etapas muy tempranas del desarrollo, que han podido evaluarse y mejorarse a lo largo de la segunda iteración, produciendo un mejor resultado final.

Otras formas de afrontar el problema seguramente no hubieran ofrecido el mismo resultado, o de haberlo hecho no habría sido en el mismo tiempo. Es por eso que la elección realizada se valora como positiva, cumpliendo el objetivo marcado.

### **C. Realizar un diseño de la aplicación, tanto a nivel funcional como de interface con el usuario.**

Afrontar el diseño de una herramienta desde cero es una tarea compleja por la cantidad de factores que se han de tener en cuenta. El número de opciones disponible hace que alcanzar un diseño final adecuado requiera una buena elección del camino a seguir. Superar este objetivo ha sido probablemente el escollo más complejo de todo el proyecto.

El diseño final obtenido ha sido posible gracias a una separación eficaz de las diferentes partes que componían el proyecto, afrontando cada una de ellas por separado, pero sin perder de vista las necesidades de las demás. El hecho de que la base de datos modelase una situación real que podía ser estudiada ha permitido simplificar el problema, al poder disponer de una base sobre la que trabajar.

La solución final conseguida responde a las necesidades planteadas por el desarrollo del evento estudiado, logrando optimizar las tareas que se llevan a cabo y mejorar el control sobre el mismo. Se considera por lo tanto que el diseño realizado es suficiente para dar por cumplido este objetivo, aunque podría mejorarse aún más aplicando las soluciones mostradas más adelante en una hipotética tercera iteración.

### **D. Implementar el diseño escogido.**

La correcta implementación del diseño puede evaluarse fácilmente en función de los siguientes parámetros:

- Simplicidad del código resultante.
- Capacidad de la herramienta de llevar a cabo las tareas demandadas.
- Tiempo necesario para el desarrollo del código.
- Capacidad de admitir mejoras.

El código obtenido ha demostrado ser lo suficientemente simple y robusto como para admitir la corrección de errores en vuelo, sin afectar más que de manera leve al correcto funcionamiento de la herramienta. Además, igual que es sencillo corregir errores, lo es también añadir mejoras al trabajo desarrollado, permitiendo la evolución de la herramienta.

Por otra parte, el gestor ha respondido perfectamente bajo una situación de prueba real en un evento con atención al público, mejorando notablemente la mayor parte de las tareas donde entraba en juego.

Si bien el tiempo de desarrollo no ha sido excesivo, y que gracias a la metodología escogida se ha logrado desarrollar una herramienta funcional en un plazo

razonable de tiempo, la cantidad de scripts resultante ha hecho que en ocasiones la implementación fuera costosa para los resultados obtenidos.

Sin embargo, el hecho de haber producido una utilidad funcional hace que se pueda dar por cumplido el objetivo, con la impresión de que la herramienta todavía tiene recorrido y se puede llegar a una implementación más acertada aplicando alguno de los campos de mejora vistos en la siguiente sección.

#### **E. Depurar y comprobar su utilidad en un escenario real.**

Para garantizar el correcto uso de la herramienta es necesario someterla a procesos de prueba con carácter previo a su lanzamiento final. Estas pruebas deben cubrir las casuísticas esperadas en el entorno real, anticipándose a los problemas que puedan surgir y eliminándolos en etapas del diseño en las que todavía es factible hacerlo.

La aplicación obtenida pasó por una serie de secuencias de test diseñadas para extraer el máximo número de errores posibles y comprobar la funcionalidad de las diferentes áreas que la componen. Si bien estas secuencias de test no han sido infalibles, sí que han permitido extraer los errores más importantes, dejando únicamente aquellos que cuentan con una solución trivial.

Estos errores triviales afloran principalmente en los entornos reales, donde la herramienta es sometida a condiciones de carga de trabajo más elevadas y a diferentes casuísticas no completadas. Si bien durante la fase de prueba en entorno real parte de estas casuísticas se eliminaron, otras siguen permaneciendo como posible elemento de mejora en el futuro.

Dado que la secuencia de test se completó, tanto en su fase de desarrollo como en el entorno real, este punto puede considerarse satisfecho.

#### **F. Estudiar otros posibles usos de la solución adoptada.**

A lo largo del capítulo 5 se han podido ver diferentes opciones fuera de la originalmente pensada que pueden ser objetivo de uso de la herramienta diseñada, bien directamente o bien con pequeños ajustes para adaptarla a las características propias de la aplicación.

Esto permite dar vida al proyecto, ya que al tener más aplicaciones que la que motiva su desarrollo abre las puertas a nuevos clientes y por lo tanto a poder continuar con el desarrollo de las diferentes secciones que lo forman. Especialmente interesante es su aplicación en eventos afines (más de un centenar al año sólo en España) como en bibliotecas (más de 6500 en el territorio nacional), ya que no requiere apenas modificaciones y permite cubrir sus necesidades.

## 6.2 – Posibles mejoras

En este apartado se estudiarán las posibles áreas de mejora dentro del proyecto, desgranándose las razones de cada una de ellas. Al haberse escogido una filosofía de diseño en espiral, dichas mejoras podrían utilizarse en una tercera iteración del proyecto, dotando de especial relevancia a este capítulo.

El motivo por el cual estas mejoras no han sido aplicadas al proyecto es porque se entiende que en las condiciones actuales cumple con los requisitos requeridos, por lo que su aplicación podría llevar a una dilatación temporal del proyecto y por lo tanto a hacerlo inviable en materia de tiempo.

Por otro lado, parte de ellas fueron detectadas en la prueba real realizada a lo largo de las X Jornadas ‘Minas Tirith’, una vez finalizada la segunda iteración del proyecto, por lo que lo lógico es tenerlas en cuenta como mejoras a evaluar en futuras iteraciones.

Todo proyecto web está vivo y la lista de cambios que se pueden realizar es infinita, ya que con el tiempo surgen nuevas necesidades y corrientes de diseño que los justifican. Era por lo tanto necesario establecer un punto en el cual la funcionalidad de la herramienta fuera suficiente como para dar por cerrado el proyecto.

### a. Librería de funciones MySQL

Hoy en día una de las principales preocupaciones a la hora de diseñar aplicaciones web es la privacidad de los datos, evitar que la seguridad de la base de datos no se vea comprometida. En este sentido a menudo surgen nuevas recomendaciones en materia de seguridad por parte de los desarrolladores de los lenguajes de programación.

A partir de la versión 5.5.0 de PHP se declaró obsoleto el uso de las funciones de la librería MySQL, recomendándose el uso de las funciones MySQLi y PDO en su lugar, retirándose de manera definitiva en la versión 7.0.0. [14] Dado que la documentación usada para este proyecto en sus inicios era anterior a esta versión de PHP y la versión de PHP utilizada en la web es la 5.3.10 en el inicio del proyecto se optó por seguir dicha documentación y hacer uso de esta librería.

Un cambio de librería aportaría mayor seguridad a la aplicación, además de protegerla de fallos en caso de subida de versión de PHP a una más reciente. Además, MySQLi permite interfaces orientadas a objetos y múltiples conexiones, permitiendo añadir nuevas funcionalidades a la herramienta, mientras que PDO permite cambios entre diferentes gestores de bases de datos con modificaciones mínimas de código, lo que la convierte en una herramienta muy interesante en caso de pretender migrar la base de datos a otro gestor como ORACLE o POSTGRESQL.

Al no entenderse necesaria la migración de la base de datos, ya que el tamaño reducido y la simplicidad de uso de MySQL hacen que esta sea la opción recomendada, la opción elegida en caso de implementar esta mejora sería MySQLi.

### **b. PHP orientado a objetos**

A partir de la versión 5 de PHP, en la que el código fue reescrito para mejorar el rendimiento del modelo orientado a objetos, el uso de esta funcionalidad en PHP comienza a resultar interesante [15]. Sus principales ventajas son una minimización del código, al permitir agrupar funciones recurrentes dentro de una misma clase, así como una mejor estructura del mismo. Ambas ventajas conllevan un mantenimiento más sencillo del código y mayor facilidad de aplicación de cambios, así como menor probabilidad de errores.

Un cambio de este estilo conllevaría la reescritura de la práctica totalidad del código PHP, pues todo acceso a la base de datos podría hacerse igualmente a través de llamadas a los métodos de los objetos, lo que afectaría a todas las secciones de la web. Sin embargo, las ventajas se consideran lo suficientemente atractivas como para merecer la pena el cambio en futuras iteraciones de la herramienta.

### **c. Revisión de las secciones**

A pesar de la gran optimización llevada a cabo en la segunda iteración de la web, el estudio de su uso abre la puerta a futuras mejoras que conlleven menor número de acciones por parte del usuario para la realización de las diferentes tareas y por lo tanto optimicen aún más sus acciones. El gran número de acciones repetidas que se llevan a cabo hace que pequeñas mejoras en este apartado permitan grandes ganancias en tiempo en el uso de la herramienta, una de las claves del proyecto debido a los requisitos de uso vistos en los capítulos 1 y 5.

Una posible mejora podría encontrarse en la gestión de los préstamos. Agrupar los préstamos y las devoluciones en la misma pantalla, posible dado que son pequeños formularios en ambos casos, permitiría evitar el cambio de pantallas, pues su uso ha demostrado que ambas funciones se alternan constantemente, llevando a numerosos cambios de sección que podrían evitarse.

Igualmente, en este caso, automatizar tareas como la envió de los datos o establecer campos generales, donde sea el propio código el que interprete el valor introducido y lo categorice en consecuencia, permitirían optimizar aún más los préstamos y devoluciones, requiriendo menos trabajo por parte del usuario de la herramienta.

#### **d. Diseño adaptativo**

Aunque en la segunda iteración del proyecto se realizó un profundo cambio en el apartado visual de la herramienta, las corrientes de diseño cambian constantemente y en poco tiempo quedan obsoletas. Hoy en día el uso de los dispositivos móviles para acceder a aplicaciones web es generalizado y un rediseño gráfico, modificando la web para adaptarla a los estándares del diseño adaptativo, permitiría abrir su uso a toda clase de dispositivos.

Dicho cambio sería igualmente costoso en materia de tiempo de trabajo, pues conllevaría la reescritura de toda la estructura de la web y la revisión de su hoja de estilos, así como una gran batería de pruebas para garantizar su funcionamiento en los distintos dispositivos. Sin embargo, esta optimización permitiría adaptar la herramienta a una forma completamente diferente de uso sin perder en ningún momento su funcionalidad original.

En este caso, el uso del framework Bootstrap para HTML, CSS y JavaScript permitiría realizar dicho cambio, ya que define una estructura que adapta automáticamente el tamaño de los contenedores al tamaño de la pantalla, además de ofrecer utilidades para modificar la propia estructura de la página en función del dispositivo que la invoca.

#### **e. Interactividad**

Actualmente en los sitios web se presta mucha atención a la interactividad con el usuario. Otorgar plataformas para que el usuario exprese sus opiniones e interactúe con otros usuarios es un elemento casi obligado en un ecosistema fuertemente influenciado por las redes sociales y la forma en la que estas han cambiado la percepción del usuario sobre la web.

Aunque en la segunda versión de la herramienta se han dado los primeros pasos para integrar estas funcionalidades, con un chat en la parte de administración o módulos de conexión con redes sociales y un formulario de contacto en la parte pública, es aún posible avanzar más en esta materia. Para ello se pueden habilitar plataformas de contacto entre los usuarios y espacios para que valoren su experiencia o los juegos a los que han jugado, avanzando un paso más hacia este modelo y mejorando la experiencia del usuario.

#### **f. Casos de uso no contemplados**

A pesar de haberse estudiado el diagrama de flujo de las aplicaciones para abarcar todos los casos de uso esperables, en ocasiones surgen excepciones al uso normal que no han sido contempladas durante la fase de diseño y producen deficiencias en el uso de la herramienta.

Analizar dichos casos en incluirlos en los procedimientos, bien dentro de los sistemas de control si se trata de fallos o en las opciones de las herramientas si presentan nuevas situaciones que deben ser contempladas es la última opción de mejora presentada en este proyecto.

En este caso la carga de trabajo que conlleva es variable, en función de cuantos de estos casos surjan, pero durante la prueba funcional de la herramienta se ha mostrado que no son excesivos, a pesar de la carga a la que ha sido sometida la aplicación y que su solución conlleva menos trabajo que otras propuestas de mejora presentadas en este apartado, pudiendo incluso combinarse con alguna de ellas.

## 7- Referencias

A continuación se detallan las diferentes referencias utilizadas a lo largo del presente proyecto:

- [1] Ángel Cobo, Patricia Gómez García, Eliana Rocha Blanco, and Daniel Pérez González, *PHP y MySQL : tecnologías para el desarrollo de aplicaciones web*. Madrid: Díaz de Santos, 2005.
- [2] World Wide Web Consortium. (2014, Octubre) Sitio web del W3C. [Online]. <https://www.w3.org/TR/2014/REC-html5-20141028/>
- [3] World Wide Web Consortium. (2016, Septiembre) Sitio web del W3C. [Online]. <https://www.w3.org/Style/CSS/>
- [4] World Wide Web Consortium. (2016, Septiembre) Sitio web del W3C. [Online]. [https://www.w3.org/standards/techs/css#w3c\\_all](https://www.w3.org/standards/techs/css#w3c_all)
- [5] Juan Carlos Oros, *Diseño de páginas web con XHTML, JavaScript y CSS.*: RA-MA S.A. Editorial y Publicaciones, 2008.
- [6] World Wide Web Consortium. (2016) Sitio web del W3C. [Online]. <https://www.w3.org/standards/webdesign/script>
- [7] Oracle Corporation. (2016, Septiembre) Sitio web de MySQL. [Online]. <http://dev.mysql.com/doc/refman/5.5/en/>
- [8] Eric Godoc, *SQL. Los fundamentos del lenguaje.*: ENI, 2014.
- [9] Centers for Medical & Medicaid Services. (2008, Marzo) sitio web de CMS.org. [Online]. <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>
- [10] Clint Eccher, *Diseño Web Profesional.*: Anaya Multimedia, 2011.
- [11] Oracle Corporation. (2016) Sitio web de MySQL. [Online]. <http://dev.mysql.com/doc/refman/5.5/en/integer-types.html>
- [12] Oracle Corporation. (2016) Sitio web de MySQL. [Online]. <http://dev.mysql.com/doc/refman/5.5/en/keywords.html>
- [13] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of applied cryptography.*: CRC Press, 1996.

[14] The PHP Group. (2016) Website de PHP. [Online].  
<http://php.net/manual/es/function.mysql-connect.php>

[15] The PHP Group. (2016) Sitio web de PHP. [Online].  
<http://php.net/manual/es/oop5.intro.php>