

## AGRADECIMIENTOS Y DEDICATORIAS

Me gustaría comenzar este apartado de agradecimientos, dándole las gracias a mi director de proyecto Joan Carles Gil y a mi ponente Xavier Llinas por ofrecerme la posibilidad de hacer este trabajo.

Por otro lado quisiera dar las gracias especialmente a Mónica y Oscar porque sin ellos la presentación de este trabajo hubiera sido completamente imposible.

Por último quisiera agradecer especialmente al departamento de AC de la UPC por ofrecerme facilidades para que acabe este proyecto.

Ahora que acabo este paso en mi camino me gustaría dedicarle este trabajo que simboliza esa culminación a mucha gente que ha contribuido a que esta carrera llegue a su fin. El problema es que no me gustaría parecer *Pedro Almodóvar* en la entrega de los Oscar, así que intentaré ser breve y no dejarme a nadie en esta brevedad.

Primero de todo le dedico este proyecto a mi familia. A mi madre que me ha aguantado todo y más, a mi hermano que aunque sólo me da dolores de cabeza es mi hermano, a mis tíos que han estado y están ahí siempre para todo lo que necesito, a mis primos que más que primos son hermanos, a mi abuela que por fin ve cómo uno de sus nietos acaba una carrera,

Quiero dedicarle este proyecto a mis amigos que en mayor o menor medida siempre han estado ahí cuando los he necesitado y que me han visto disfrutar y sufrir con esta carrera: a mis compañeros inseparables de fatigas que en su día fueron bautizados como “pavosdelculo” (Anthony, Deivid, Férran, Isis, Lora y Róger), a aquellos que no siempre veo tanto como quisiera (Neus, Núria, va por vosotras), a Aitor que ha estado ahí compartiendo noches en vilo y que me mostró que Windows no es la solución, a mis compañeros de Cometa que han aguantado mis cambios de humor constantes, a mis compañeros de teatro que han tenido que aguantar mis faltas por razones de estudio, a Cristina, Eva, Joan, Jordi y compañía que compartieron el inicio de esta aventurilla conmigo y a todos los que no han salido todavía y han estado ahí que son muchos, que digo muchos, muchísimos.

¡Muchas gracias a todos por todo lo que me habéis ayudado a lo largo de este camino!



# 1. CONTENIDOS

---

<b>1. CONTENIDOS.....</b>	<b>7</b>
<b>2. INTRODUCCIÓN.....</b>	<b>9</b>
2.1. Definición .....	9
2.2. Historia.....	9
<b>3. METODOLOGÍA .....</b>	<b>10</b>
3.1. Decisión .....	10
3.2. Ventajas de la metodología escogida.....	14
3.3. Desventajas de la metodología .....	14
<b>4. INFORME DE DEFINICIÓN.....</b>	<b>15</b>
4.1. Oportunidad del proyecto .....	15
4.2. Situación actual.....	15
4.3. Objetivos .....	16
4.4. Alcance .....	17
<b>5. PLANIFICACIÓN .....</b>	<b>19</b>
<b>6. DECISIONES TECNOLÓGICAS .....</b>	<b>23</b>
<b>7. ESTIMACIÓN ECONÓMICA DEL PROYECTO .....</b>	<b>28</b>
7.1. Coste del Hardware .....	28
7.2. Coste software .....	29
7.3. Coste de desarrollo del proyecto .....	30
7.4. Estimación Total .....	32
<b>8. VISIÓN TÉCNICA DEL SISTEMA .....</b>	<b>33</b>
8.1. Usuarios del sistema .....	33
8.2. Arquitectura del sistema .....	34
8.3. Posibles mejoras de la aplicación.....	44
<b>9. CONCLUSIONES.....</b>	<b>45</b>
<b>10. ANEXO A – RESUMEN FUNCIONAL .....</b>	<b>46</b>

10.1.	Introducción .....	46
10.2.	Documentación de referencia.....	47
10.3.	Marco de referencia.....	48
10.4.	Descripción del proceso .....	50
10.5.	Entrada/Salida de información.....	61
<b>11.</b>	<b>ANEXO B – MANUAL DE USUARIO .....</b>	<b>62</b>
11.1.	INTRODUCCIÓN A BIBLIOREF.....	62
11.2.	CONFIGURACIÓN DE LA BASE DE DATOS.....	63
11.3.	ELECCIÓN DEL ODBC .....	68
11.4.	LOGIN DE USUARIO .....	69
11.5.	COMBINACIÓN CON MICROSOFT GROOVE .....	71
11.6.	INTRODUCCIÓN DE INFORMACIÓN .....	80
11.7.	BUSCADOR.....	98
11.8.	IMPORTACIÓN DE INFORMACIÓN.....	100
11.9.	EXPORTACIÓN DE INFORMACIÓN.....	103
11.10.	OTRAS FUNCIONALIDADES .....	105
<b>12.</b>	<b>BIBLIOGRAFÍA .....</b>	<b>108</b>

## **2. INTRODUCCIÓN**

---

### **2.1. Definición**

Una aplicación de gestión bibliográfica, es una herramienta de ayuda a la gestión de la información generada por una entidad.

Con el paso del tiempo la sociedad se ha dado cuenta de la ineficiencia del tratamiento de la información sobre papel, y de las enormes ventajas que ofrecen las soluciones tecnológicas para este campo.

Así, la gestión documental se ha convertido en una necesidad y un problema para cualquier organización. La mayoría de las organizaciones necesitan acceder y consultar de forma frecuente la información archivada. En otros es la importancia de los documentos o el volumen de información lo que estimula a buscar nuevas soluciones innovadoras que ofrezcan ventajas y valor añadido sobre los sistemas tradicionales de archivo y almacenamiento.

Un sistema de gestión documental es un sistema informatizado, un conjunto formado por una o más aplicaciones, utilizado para localizar y almacenar documentos electrónicos y/o imágenes de documentos soportados en papel.

### **2.2. Historia**

Tradicionalmente la gestión bibliográfica o de la información se ha realizado de forma manual. No ha sido hasta hace un par de décadas que con la masificación de las Bases de Datos no han empezado a aparecer diferentes herramientas para la ayuda en estas tareas.

Actualmente existen en el mercado multitud de herramientas para la gestión documental. Por otro lado no existen tantas para el procesamiento bibliográfico de esa información. Y de ahí la necesidad de crear herramientas específicas para esta labor.

### 3. METODOLOGÍA

---

#### 3.1. Decisión

Para el desarrollo de este proyecto se ha utilizado una metodología de “*Extreme Programming*” o “*Programación Extrema*”.

El Programación Extrema consiste en acentuar la simplicidad y la comunicación entre el desarrollador y el usuario final.

De este modo, cuando existen cambios en los requisitos iniciales, hecho que de forma natural aparece en cualquier proyecto del ámbito informático, se intenta adaptar el desarrollo de la manera más eficiente posible y que otorgue mejores resultados.

Las características fundamentales de este método son:

- **Desarrollo iterativo e incremental:** el desarrollo el proyecto va avanzando de forma que la aplicación cada vez tiene más funcionalidades implementadas.
- **Pruebas unitarias continuas:** en lenguajes orientados a objetos, como el utilizado para el desarrollo de este proyecto, para cada una de las clases construidas se realizan varias pruebas. A medida que se avanza en el desarrollo y se van añadiendo funcionalidades a la aplicación se van realizando pruebas con el conjunto de clases especificado y sus interacciones.
- **Programación en parejas:** Aunque no es imprescindible, la programación en parejas permite que exista una discusión del código sobre la marcha. Esto desemboca en una mayor calidad del código y por lo tanto una mayor calidad del Software obtenido. Por tanto en esta metodología se le da mayor importancia a la calidad obtenida que a la productividad inmediata. Obviamente en este proyecto no se ha respetado este punto, aunque como ya se ha mencionado, no es imprescindible para que haya una metodología de Programación Extrema.

- **Frecuente interacción del equipo de programación con el cliente o usuario.** La comunicación entre el equipo de desarrollo y los usuarios finales de la aplicación permite corregir al vuelo ciertos aspectos del diseño de la aplicación, de este modo aumenta el dinamismo del desarrollo y se evitan mayores cambios en el futuro.
- **Corrección de todos los errores antes de añadir nuevas funcionalidades.** Derivada de la particularidad de las pruebas unitarias, en esta metodología se apuesta por un avance de calidad y seguro en el software, por tanto antes de añadir una nueva funcionalidad hay que asegurarse que la anterior versión del código es correcta y funciona sin problemas.
- **Propiedad del código compartida.** Derivada de la programación en parejas o grupos, en esta metodología la propiedad de cada parte del código no es de una única persona. Cada persona del equipo de desarrollo debe saber y poder modificar cualquier parte del código.
- **Refactorización del código,** es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- **Simplicidad en el código:** es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Derivado de la utilización de esta metodología, el ciclo de vida del software que se ha seguido, ha sido un ciclo de vida *Espiral* (Figura 1).

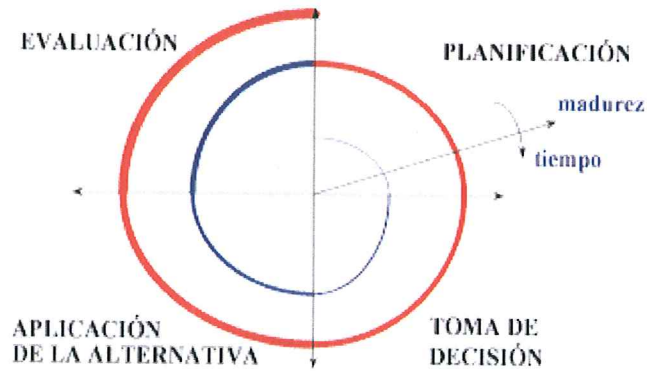


Figura 1

El ciclo de vida *Espiral* se basa en una evolución del ciclo de vida *Prototipado*.

Este ciclo de vida se basa en la creación de prototipos que van evolucionando escalaramente debido a las fuentes de incertidumbre, a diferencia del *Prototipado* que va creando prototipos nuevos cada vez y a cada cambio.

A menudo la fuente de incertidumbre es el propio cliente/usuario, que aunque sepa claramente lo que desea, no es capaz de definirlo en todos sus aspectos sin ver como influye en otros.

Así cada una de las fases del proyecto se va repitiendo escalaramente durante el desarrollo del proyecto. A continuación se define cada una de las clases y una serie de objetivos que la caracterizan:

1. *Fase Definición* (¿Qué hacer?)

- Estudio de **viabilidad**.
- **Conocer los requisitos** que debe satisfacer el sistema (funciones y limitaciones de contexto).
- Asegurar que los **requisitos son alcanzables**.
- Formalizar el **acuerdo** con los usuarios.
- Realizar una **planificación** detallada.



2. *Fase Diseño (¿Cómo hacerlo?)*

- Identificar **soluciones tecnológicas** para cada una de las funciones del sistema.
- Asignar **recursos** materiales para cada una de las funciones.
- Proponer (identificar y seleccionar) **subcontratas**.
- Establecer métodos de **validación** del diseño.
- **Ajustar las especificaciones** del producto.

3. *Fase Construcción (Hacerlo y probarlo)*

- **Generar el producto** o servicio pretendido con el proyecto.
- **Integrar los elementos subcontratados o adquiridos externamente**.
- **Validar** que el producto obtenido satisface los requisitos de diseño previamente definidos y **ajustarlo**.

4. *Fase Mantenimiento y operación (Mantenerlo )*

- **Operación:** asegurar que el uso del proyecto es el pretendido.
- **Mantenimiento** referido al mantenimiento no habitual.

### **3.2. Ventajas de la metodología escogida**

La principal ventaja de la metodología escogida para el desarrollo de esta herramienta es su flexibilidad y dinamismo.

Además de eso, el hecho de que permita avanzar de forma escalár y tener una cota de tiempo acotada, ha facilitado el hecho de tener una herramienta funcionalmente operativa a lo largo del desarrollo que en cualquier momento.

Por otro lado debido a la posibilidad de reunirse con los usuarios finales se creyó conveniente el hecho de tener una alta permeabilidad a cambios que naturalmente aparecen durante el transcurso del proyecto, ya sea por una mala definición pobre inicial por parte de los usuarios como por un desconocimiento inicial del entorno por parte del equipo productivo.

### **3.3. Desventajas de la metodología**

Una de las desventajas que presenta está metodología, paradójicamente se contrapone a una de sus principales ventajas. El exceso de dinamismo y flexibilidad puede desviar el proyecto de sus objetivos iniciales.

Por otra parte, la necesidad de una comunicación constante entre los clientes/usuarios y el equipo de desarrollo no siempre queda cubierta debido a las ocupaciones y agendas por ambas partes.

## **4. INFORME DE DEFINICIÓN**

---

### **4.1. Oportunidad del proyecto**

La oportunidad de este proyecto, nace de la necesidad de los investigadores y docentes del departamento de Organización de Empresas de la Universidad Politécnica de Cataluña, para la gestión de alguna forma ágil de la información que van consiguiendo para su posterior procesado.

Así mismo puede ser extensivo a otros ámbitos fuera de este departamento debido a que el problema es similar para todos los investigadores.

La gestión de la información es una tarea más o menos mecánica y que se lleva proporcionalmente una parte importante del tiempo de investigación. Este tiempo podría ser destinado a la obtención de más información o a un análisis más exhaustivo de la información obtenida y por lo tanto ser más productivo.

Bajo esta premisa, la necesidad pasaba por automatizar de alguna forma la clasificación de la información encontrada y su posterior localización, para de esta forma después reducir el tiempo perdido en la clasificación y poder destinarlo a un aumento de la productividad.

Así pues el proyecto básicamente necesitaba la creación de una herramienta que permitiera a los usuarios de la misma utilizar un repositorio de información donde por diferentes conceptos pudieran clasificar y después localizar la información de la que disponían.

### **4.2. Situación actual**

En el área de estudio de la economía regional, como en la mayoría de áreas de estudio, la información se recoge principalmente de artículos, "*papers*", libros y trabajos de tesis que se encuentran documentados en bases de datos especializadas.

En el caso que nos atañe, la información obtenida suele buscarse principalmente en 2 grandes bases de datos existentes en el mercado y a las que la Universidad Politécnica de Cataluña tiene acceso a través de una suscripción.

A continuación se muestra algún dato sobre las bases de datos más utilizadas:

- Econlit
- ISI Web of Knowledge

El proceso que siguen los investigadores y docentes en esta disciplina suele basarse en:

1. Búsqueda de información en las diferentes fuentes. Ya sea en las bases de datos mencionadas anteriormente, o ya sea en diversos libros o tesis.
2. Recopilación de la información que a priori parece interesante.
3. Procesamiento de esa información manualmente sobre papel.
4. Generación de hipótesis, basándose en la información procesada previamente
5. Análisis de las hipótesis
6. Conclusiones.

En todo este proceso, la información suele estar procesada normalmente en papel, con lo que las búsquedas sobre información ya procesada, suele ser lenta y en ocasiones incluso compleja.

Todo ello deriva en la consecuente pérdida de tiempo y eficiencia. Por tanto el problema existente actualmente es lo que se intenta paliar con este proyecto.

### **4.3. Objetivos**

El objetivo principal de este proyecto es la creación de una herramienta que permita un ligero procesamiento de la información bibliográfica obtenida de búsquedas especializadas, así como la clasificación y la posterior búsqueda de la misma.

Se desea por otra parte una implantar una herramienta que permita la interacción con otras aplicaciones existentes para el estudio de datos bibliográficos como por ejemplo SITKIS.

Por lo tanto la aplicación ha de dar solución a los siguientes puntos:

- Creación de repositorio de información obtenida
- Clasificación de la información a raíz de varios campos, autor, título de la información obtenida, medio que la contiene, fechas....
- Búsqueda de la información a través de los campos que componen las diferentes fichas de información introducidas
- Importación automática de información a través de varias bases de datos especializadas
- Interacción con herramientas existentes en el mercado para el procesamiento bibliográfico como en el caso de SITKIS

#### **4.4. Alcance**

Este punto queda mayormente documentado en el Apéndice A – Resumen funcional de la aplicación.

A continuación se muestra un esquema del alcance del proyecto en base a ese Resumen funcional, aprobado por los responsables del proyecto en el departamento de Organización de Empresas.

El alcance del proyecto se basa principalmente en la posibilidad de:

- **Creación, modificación y borrado de fichas de información bibliográfica**, esto incluye la recopilación de datos relevantes sobre la misma que puedan facilitar el trabajo de investigación posteriormente. Es decir, dentro de cada ficha se ha de poder guardar los autores de la información, el medio que contiene a la información, las palabras clave que tiene asociadas, etc.
- **Búsqueda avanzada de la información**. Hablamos de búsqueda avanzada refiriéndonos a que la aplicación ha de ser capaz de realizar búsquedas acotadas por los diferentes campos que hemos introducido en las fichas de información. De esta forma se pretende agilizar el tiempo de búsquedas de información ya procesada.

- **Importación automática de la información.** Se contempla la posibilidad de importar información automáticamente desde 2 bases de datos especializadas y mencionadas anteriormente, *ISI Web of Knowledge* i *Econlit-Ebsco*.
- **Exportación de la información recogida en la aplicación hacia herramienta de tratamiento bibliográfico SITKIS**

La aplicación por tanto se ha elaborado basándose en estos puntos y en este orden de importancia de los mismos.

## 5. PLANIFICACIÓN

A continuación se muestra el diagrama temporal de planificación inicial del proyecto.

Lógicamente y como cabe esperar, las horas invertidas en cada una de las fases del proyecto no coincide exactamente con las que se han realizado realmente. De todas formas, debido a que se realizó un estudio previo basándose en otros proyectos realizados con anterioridad, la planificación no dista mucho de la realidad.

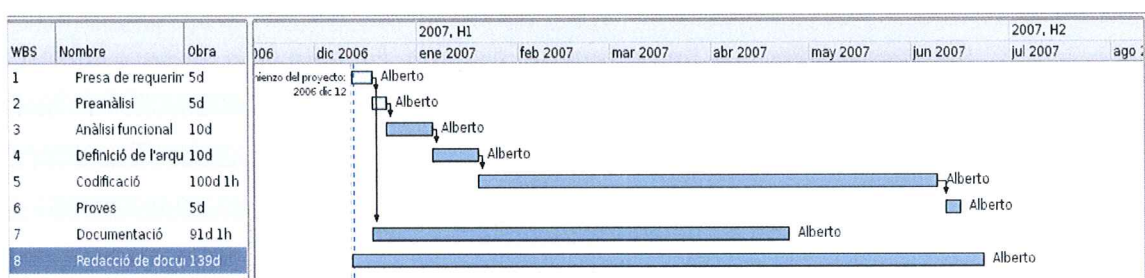


Figura 2

A continuación se muestra una descripción detallada de todas la tareas y subtareas de las que se compone el proyecto:

### 1. Toma de requerimientos.

- 1.1. Esta fase consiste en realizar una serie de reuniones con los clientes/usuarios de la aplicación para detectar las necesidades tecnológicas que tienen. Es parte importante del proyecto ya que sin una buena toma de requisitos se puede llegar un resultado final del proyecto que no es el esperado.
- 1.2. En este proyecto la toma de requisitos se valoró en un esfuerzo inicial de 5 días de trabajo. Se estima una carga de trabajo de unas 2 horas diarias de media para la toma de requerimientos, por lo tanto la carga de trabajo de esta fase se sitúa en torno a las **10 horas** de trabajo.

## **2. Preanálisis.**

2.1. En este punto se hace un análisis previo de los requerimientos y se consensúa con los clientes/usuarios para determinar que la toma de requisitos ha sido positiva.

2.2. Para la elaboración del preanálisis de este proyecto se estimó un esfuerzo inicial de 5 días. La carga de trabajo diario para esta fase del proyecto se situó en torno a las 2 horas diarias, lo que hace un total de **10 horas** de trabajo aproximadamente.

## **3. Análisis funcional.**

3.1. La fase de análisis funcional se caracteriza por una definición minuciosa de las funcionalidades que ha de tener el sistema. En esta fase se contemplan todos los requerimientos especificados en las fases anteriores y una vez contemplados se añaden aquellas funcionalidades que pueden mejorar el sistema y que no se salen del alcance del proyecto.

3.2. Para la elaboración del análisis funcional de la aplicación se estimaron 10 días de trabajo. Durante esta fase se estimó una carga de trabajo diario de unas 5 horas diarias, eso nos sitúa en una carga de **50 horas** de trabajo.

## **4. Diseño de la arquitectura.**

4.1. El diseño de la arquitectura consiste en plasmar y definir las unidades lógicas que intervendrán en la aplicación. Un buen diseño permite una flexibilidad mayor a la hora de hacer cambios, así como una mayor posibilidad de escalabilidad del proyecto.

4.2. En este proyecto se ha intentado hacer un diseño en 3 capas Presentación, Dominio y Consistencia y realizarlo de la forma más modular posible para poder afrontar los cambios que fuesen saliendo durante y después del transcurso del mismo. Para ello se estimó un esfuerzo que englobaba tanto el diseño conceptual de la aplicación como el diseño de la persistencia del mismo. El esfuerzo por tanto se valoró en 10 días. Durante esos días se estimo que se estaría una media de 5 horas diarias para la realización del diseño de la aplicación. Con estos datos obtenemos una carga de **50 horas**



## **5. Codificación**

5.1. Esta fase del proyecto consiste precisamente en codificar lo que previamente se ha diseñado. Es la fase más larga de todo el proyecto, aunque no por ello quiere decir que sea la más importante conceptualmente. Su importancia radica en que si no hay una codificación no hay una aplicación que cubra el proyecto.

5.2. Para la fase de codificación se estimaron aproximadamente unos 100 días de trabajo. Para esta fase se estimó que no se rendiría más de 5 horas diarias. Eso nos sitúa en un esfuerzo para esta fase de unas **500 horas** de trabajo.

## **6. Pruebas.**

6.1. Aunque durante el transcurso de la codificación, y más aún teniendo en cuenta el tipo de metodología que se iba a utilizar en el proyecto, se van realizando pruebas unitarias, siempre es bueno indicar un periodo de pruebas global, en el que se someterá a la aplicación a un seguido de baterías de pruebas para conseguir el mínimo de errores en la misma.

6.2. El esfuerzo durante esta fase del proyecto se ha estimado en unos 5 días. Durante el periodo de pruebas se estimó una carga de trabajo diario de unas 2 horas aproximadamente para someter la aplicación a pruebas. Eso hace que la carga horaria estimada de esta fase se vea en torno a las **10 horas** de trabajo.

## **7. Documentación.**

7.1. Esta fase consiste en la búsqueda de documentación necesaria para la elaboración del proyecto. Es una fase un tanto irregular porque suele tener mayor carga de trabajo al inicio del proyecto que al final. Engloba tanto la documentación necesaria para entender el área de estudio, como la documentación técnica necesaria para la elaboración de la aplicación.

7.2. En proyectos como el que nos atañe, un Proyecto Final de Carrera, suele tener una carga un tanto mayor de lo habitual, debido a la inexperiencia o simplemente debido a que las restricciones de tecnología te obligan a plantearte la solución al problema de una forma de la que no estás habituado. La carga de trabajo para esta fase se valoró en unos 90 días aproximadamente. Teniendo en cuenta la irregularidad de la carga a lo largo

del proyecto se estimó una media de unos 30 minutos diarios de documentación, por lo tanto la carga ascendería aproximadamente a unas **50 horas** de esfuerzo.

#### 8. Redacción de documentación.

8.1. En cualquier proyecto informático, existen una serie de entregables al cliente/usuario, como por ejemplo el manual de usuario o la documentación técnica (sobre todo si se trata de *software libre*). En proyectos como este, además se presenta un componente docente que obliga la presentación de una memoria del proyecto explicando completamente el proyecto y el modo de proceder con el mismo.

8.2. Esta fase al igual que la anterior se trata de una fase irregular, normalmente su carga de trabajo se suele concentrar al final de la misma. De todos modos, haciendo la estimación se asignaron de media 140 días de trabajo. Teniendo en cuenta que de media se había planeado estar 1,5 horas diarias, la carga por horas de trabajo asciende a **100 horas** aproximadamente.

Teniendo en cuenta estos esfuerzos, se valoró el proyecto en un total aproximado de **780 horas** de esfuerzo. Más adelante veremos como este esfuerzo se ve traducido en un coste económico aproximado del proyecto.

## 6. DECISIONES TECNOLÓGICAS

---

En este apartado se pretenden definir cuales han sido las decisiones tecnológicas que se han utilizado, el porqué de las mismas y las posibles alternativas de implementación que se han descartado en el proyecto.

Para la elección tecnológica del proyecto se ha tenido que tener en cuenta el perfil de los usuarios de la aplicación. En este caso se trata de usuarios con un bajo perfil tecnólogo y por lo tanto con pocas posibilidades de un mantenimiento mínimo o de una instalación algo más compleja.

La idea del departamento de Organización de Empresas, por otra parte, era la de tener una aplicación en la que pudieran interactuar varias personas con la aplicación. Inicialmente esto nos hizo pensar que la implantación de una aplicación distribuida sería la mejor opción, pero al estudiar las posibilidades de las que disponía actualmente el departamento esta idea tuvo que ser rechazada.

Finalmente se optó por la creación de una herramienta de implantación local, y se llegó a la conclusión que se utilizaría un software complementario para el intercambio de los ficheros generados así como de la base de datos.

Así pues, se pasa a detallar cada una de las tecnologías utilizadas y sus alternativa:

### 1. Tecnologías utilizadas.

#### 1.1. Entorno

- El entorno de desarrollo y ejecución de la aplicación se denota por un sistema operativo Microsoft Windows XP. Se trata del entorno en el que actualmente se encuentra el departamento y el que se ha utilizado para el desarrollo de la aplicación.

#### 1.2. Lenguaje de programación escogido

- Para el desarrollo de la aplicación, “ **BiblioRef** ”, se optó por un lenguaje de programación **Java**.
- **Java** es un lenguaje de programación orientado a objetos, desarrollado a principios de los años 1990 por *Sun Microsystems*. Actualmente se ha liberalizado completamente bajo licencia GNU General Public License

(GPL) todo el software de Java. Motivo por el cual se creyó una buena opción en la elección de este lenguaje para el desarrollo.

- La elección de un lenguaje bajo licencia GPL nos exime de pagar ningún coste por licencia en el desarrollo de aplicaciones bajo este lenguaje. Eso afectará positivamente en la evaluación económica del proyecto.
- La versión de **Java** utilizada en esta aplicación ha sido la última que existe actualmente en el mercado 1.6.
- Para la ejecución de aplicaciones desarrolladas en este lenguaje, es necesaria la instalación de un software de interpretación conocido como *Máquina Virtual de Java*. Este software se puede adquirir a través de sus desarrolladores *Sun Microsystems*.

### 1.3. Elección de Base de Datos

- Por requerimientos específicos del departamento, la elección de la base de datos se fijó en un Sistema Gestor de Bases de Datos (SGBD) propietario como el de *Microsoft Access*.
- La versión utilizada para la creación de la base de datos de la aplicación ha sido indistintamente *Microsoft Access 2003 / 2007*.
- La elección de este SGBD se basa en el conocimiento previo de los usuarios de la aplicación final.
- Esta elección añadirá un coste adicional de licencias en el estudio económico.

### 1.4. Software destinado a la compartición de ficheros

- Para poder dotar a la aplicación de cierto carácter de aplicación distribuida en la que pudieran trabajar diversas personas, ha sido necesario el estudio de aplicaciones destinadas a la compartición en grupo de ficheros.
- Exactamente lo que se necesitaba y se pretendía de estas aplicaciones era el poder compartir la base de datos y los ficheros asociados a cada ficha de artículo.
- Para esto finalmente, y tomando como premisa que el software ya había sido utilizado anteriormente por los usuarios de la aplicación se optó por una aplicación propietaria, *Microsoft Groove*.

- *Microsoft Groove*, es un programa de colaboración para la ayuda de trabajos en grupo que permite compartir ficheros. Actualmente se encuentra integrado dentro de la suite de ofimática *Microsoft Office*.
- La forma de trabajo con *Groove* es similar a la forma de trabajo utilizado alguna aplicación de versiones de ficheros. Cada uno de los integrantes del proyecto se descarga los ficheros en su máquina y cuando acaban de trabajar suben los cambios al espacio compartido del grupo. Estos cambios son detectados automáticamente por el programa que de manera simultánea avisa al resto de los integrantes del grupo para que se actualicen su versión.

Así pues, la opción escogida para montar la aplicación fue básicamente, montar una aplicación de ámbito local, con la consideración que para poder compartir información entre diferentes usuarios deberían pasarse la base de datos y los ficheros asociados para poder trabajar sobre un único repositorio de información.

Esto planteaba problemas de trabajo concurrente. Esta solución no contempla la posibilidad en que dos usuarios estén trabajando al mismo tiempo sobre un repositorio de datos. Al estar montado en un entorno local, cada usuario realmente está trabajando con una base de datos propia, aunque más adelante la comparta. Eso significa que si dos usuarios hacen cambios al mismo tiempo, la versión de los ficheros que prevalecerá será la del último usuario que suba sus cambios al espacio compartido.

## **2. Tecnologías alternativas.**

### **2.1. Entorno**

- Una de las alternativas se basa en montar la aplicación en un entorno Web. Este entorno permite utilizar la aplicación en cualquier dispositivo que tenga acceso a un navegador de Internet. Además ofrece la posibilidad de trabajar desde cualquier punto sin necesidad de estar en un PC con acceso al espacio compartido como en el caso anterior.

- Para la creación del entorno Web tan sólo es necesario un servidor (entendemos por servidor, aquel hardware capaz de servir peticiones Web y que se encuentra constantemente conectado a la red) que tenga instalado un servidor Web ( software encargado de escuchar las peticiones Web, derivarlas para su tratamiento y devolver la respuesta al usuario que ha hecho la petición)
- Para el montaje de este entorno se estudió la posibilidad de instalar un servidor Web *Apache HTTP Server 2.2* junto con 2 de sus módulos *mod\_rewrite* y *mod\_PHP*
- *Apache HTTP Server 2.2* se trata de un servidor Web libre bajo licencia GPL. Esto implicaría un cierto ahorro sobre la plataforma anterior en la que el entorno pasaba por un sistema operativo propietario.

## 2.2. Lenguaje de programación

- La alternativa al lenguaje de programación de la aplicación pasaba entre otros a la utilización de PHP.
- *PHP Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*). Es un lenguaje interpretado utilizado normalmente para la creación de aplicaciones para servidor o creación de contenido dinámico Web.
- Las ventajas que ofrece PHP entre otras son: que se trata de un lenguaje multiplataforma, con conectividad a la mayoría de gestores de bases de datos, destacando su conectividad a MySQL. PHP tiene una gran comunidad que le respalda y por lo tanto es muy fácil encontrar información y soluciones aplicables. Además ofrece la posibilidad de implementar programación orientada a objetos y se trata de un lenguaje muy ágil que ofrece pocas restricciones.
- *PHP* se trata de un proyecto libre al igual que *Java* y que por tanto no añadiría ningún tipo de restricciones de licencias y no incrementaría el coste del proyecto.

### 2.3. Elección de Base de Datos

- Para la elección de la base de datos se estudiaron 2 posibilidades existentes en el mercado. Ambas soluciones de código libre.
- Las 2 posibilidades pasaban por los gestores de bases de datos MySQL o PostgreSQL.
- Ambas posibilidades eran coherentes, aunque finalmente la opción elegida hubiera sido MySQL debido a su agilidad y poco consumo de recursos del servidor.
- MySQL, tiene una excelente conectividad con PHP y junto con APACHE HTTP Server ofrecen una excelente solución para la creación de aplicaciones distribuidas.

La solución alternativa para el desarrollo y posterior uso de la aplicación, por tanto, pasa por montar un entorno Web, que permitiría el trabajo concurrente entre los usuarios de la aplicación sin necesidad de software intermedio. Además permitiría una mayor movilidad debido a la posibilidad de acceso remoto que esto supondría.

El problema de esta alternativa y de ahí su descarte, es que para esta solución, existe, condición *sine qua non*, la necesidad de un mantenimiento del entorno Web. Posibilidad que por otra parte no tienen actualmente los usuarios de la aplicación.

Como dato optimista, cabe contemplar la posibilidad que la existencia de un entorno Web, también permitiría la migración de la solución actual utilizando un servidor de aplicaciones como *Tomcat* y definiendo una nueva capa de presentación. El resto de la lógica se podría aprovechar completamente de la solución actual desarrollada en lenguaje *Java*.

## 7. ESTIMACIÓN ECONÓMICA DEL PROYECTO

En este apartado se realizará una estimación económica de los costes de desarrollo del proyecto.

Para ello, se valorará por separado, el coste de las licencias de software utilizado, si procede, el coste de amortización del hardware utilizado, y el coste del desarrollo del proyecto.

### 7.1. Coste del Hardware

Pasemos a valorar el apartado de maquinaria utilizada para el desarrollo del proyecto.

Debido a que los requisitos mínimos de la aplicación no son para nada elevados, el desarrollo de la aplicación se ha hecho en un PC de sobremesa de clase media. A continuación se detallan las características del PC en el que se ha desarrollado la aplicación, para más tarde pasar a ver un posible plan de amortización y por lo tanto un coste aproximado del hardware necesario para el desarrollo del proyecto:

Características del Ordenador Personal	Costes
<ul style="list-style-type: none"> <li>• AMD64 Opteron 3.2Ghz</li> <li>• 1024Mb DDR2 RAM</li> <li>• 180Gb 7200rpm SATA HD</li> <li>• nVIDIA® Ge-Force™ 6600 128Mb</li> <li>• Monitor Benq 19"</li> </ul>	<p>507 €</p> <p>aprox.</p>

Realizando una amortización en 4 años:

Valor	Plazo de amortización	% Imputable	Coste Imputable
507€	42 meses	$(6 / 42) * 100 = 14,3\%$	72,4 €
			<b>TOTAL: 72,4 €</b>






## 7.2. Coste software

En este apartado se evaluará cual ha sido el coste económico de la utilización del software para el desarrollo de la aplicación.

Se ha de tener en cuenta que al tratarse de software privativo, el coste de sus licencias debe contemplarse tanto en el desarrollo de la herramienta como en su posterior mantenimiento.

Para la realización de la aplicación, y como ya se ha comentado anteriormente, se han utilizado tecnologías de la familia *Microsoft* que añaden un cierto coste en el proyecto. A continuación se pasa a describir cuales han sido las herramientas utilizadas y su coste asociado:

Herramienta	Coste
	164,62€
	565,00€
	164,62€
<b>TOTAL: 894,24€</b>	

Del mismo modo que sucedía antes el coste total del software se ha de amortizar durante el tiempo de utilización. Haciendo una estimación del tiempo de amortización a 36 meses tenemos el siguiente cálculo:

Valor	Plazo de amortización	% Imputable	Coste Imputable
894,24€	36 meses	$(6 / 36) * 100 = 16,6 \%$	149 €
			<b>TOTAL: 149 €</b>

En esta estimación encontramos sólo el coste asociado al desarrollo del proyecto y no al de su mantenimiento.

### 7.3. Coste de desarrollo del proyecto

Por último y antes de generar la valoración económica definitiva, cabe contemplar el coste asociado al desarrollo del proyecto.

La estimación para este apartado se ha hecho en base a unas tablas de precios/hora distribuidas por tareas y roles que se utilizan actualmente como método presupuestario en empresas del mercado.

Hay que tener en cuenta que aunque el desarrollo de todo el proyecto sea de una única persona, esta toma diferentes roles a lo largo de todo el proyecto y es habitual en el ámbito empresarial así reflejarlo y por lo tanto imputar los costes del proyecto dependiendo del rol desempeñado en cada momento.

Así pues tendremos unas tablas que definirán el porcentaje de horas destinado a cada tarea, dividiendo entre tareas de análisis gestión o pruebas.

Esforç dedicat a: (percentatge %)	
Anàlisis	15,00%
Proves	5,00%
Gestió	10,00%
Marge desviació	10,00%

Taula de Rols i Preu/hora	
Gestor	80,0 €
Analista	60,0 €
Programador	40,0 €

<b>Client :</b>	Dep. Organització d'Empreses de la UPC	<b>Data</b>	06 / 2007
<b>Descripció del projecte :</b>	Eina d'ajuda a l'estudi de bases de dades bibliogràfiques		
<b>Nom del document</b>	Estimació econòmica BiblioRef		

Nom de la tasca (Projecte Fet a mida ó Projectes de Formació)	Est. Optim.	Est. Normal	Est. Pess.	Hores	Anàlisis (hores)	Proves (hores)	Gestió (hores)	Hores	Dies
Adquisició de coneixement	50	50	50	50	no aplica	no aplica	5	55,00	9
Anàlisis funcional	50	50	50	50	no aplica	no aplica	5	55,00	9
Definició arquitectura de l'aplicació	50	50	50	50	no aplica	no aplica	5	55,00	9
				0					
Gestió de Fitxes d'Article	45	50	75	53,33333	8,00	2,67	5,33	69,33	11,56
Gestió de Fitxes de Medi	35	45	60	45,83333	6,88	2,29	4,58	59,58	9,93
Gestió d'Autors	20	20	30	21,66667	3,25	1,08	2,17	28,17	4,69
Gestió de Paraules Clau	20	20	3	17,16667	2,58	0,86	1,72	22,32	3,72
Gestió d'Autors	20	20	30	21,66667	3,25	1,08	2,17	28,17	4,69
Gestió de Categories	20	20	30	21,66667	3,25	1,08	2,17	28,17	4,69
Gestió de Cites	20	20	30	21,66667	3,25	1,08	2,17	28,17	4,69
Gestió de Referencies	30	30	40	31,66667	4,75	1,58	3,17	41,17	6,86
				0					
Procés d'importació	30	35	45	35,83333	5,38	1,79	3,58	46,58	7,76
Procés d'exportació	30	35	45	35,83333	5,38	1,79	3,58	46,58	7,76
				0					
Creació de vistes	70	80	100	81,66667	12,25	4,08	8,17	106,17	17,69
				0					
Creació de documentació (memoria, manual)	100	100	150	108,3333	0,00	0,00	0,00	108,33	18,06
				0					

646 h.	58 h.	19 h.	54 h.	778 h.	130 d.
--------	-------	-------	-------	--------	--------

Preu	25853,33	3492,00	776,00	4304,00	<b>TOTAL</b>	<b>34.425,33 €</b>
------	----------	---------	--------	---------	--------------	--------------------

Así pues tal y como se muestra en la anterior tabla, el coste de desarrollo de todo el proyecto oscila en torno a los **34.425,33 €**

#### **7.4. Estimación Total**

La estimación económica total, será por tanto la suma de los costes calculados con anterioridad.

Así pues, el coste total del proyecto asciende a los **34.646,73 €**.

Este es el coste estimado que supondría el proyecto en el mercado. Ahora bien, al tratarse de un proyecto de carácter docente no existe ningún coste a carga del departamento o la universidad en concepto de nada.