



UNIVERSIDAD DE ZARAGOZA  
CENTRO POLITÉCNICO SUPERIOR



## **Ingeniería Informática**

Proyecto Final de Carrera

---

# **Publindex: Análisis de la calidad de publicaciones de investigación**

---

**Autor:** Sara Subijana Gracia

**Director:** Eduardo Mena Nieto

*Departamento de Informática e Ingeniería de  
Sistemas*

Septiembre de 2010



*Dedicado a mi abuela Elisa,  
gracias por mandarme tus fuerzas allí donde estés.  
Siempre te llevaré conmigo.*



## Agradecimientos

Quiero agradecer todo su esfuerzo y dedicación a mi director, Eduardo, que siempre ha estado disponible para responder a mis dudas y que me ha hecho aprender cómo hacer las cosas mejor. Gracias por ayudarme a que esta memoria no parezca una novela del Señor de los Anillos y por invitarme a tantas consumiciones en tu oficina particular.

Agradecer su ayuda a todo el grupo de investigación Sistemas Informáticos Distribuidos (SID), que hicieron una gran labor de pruebas desde las primeras versiones de la aplicación ayudándome así a mejorarla.

Mil Gracias a ti, Rafa, por darme todo tu apoyo y confianza en los momentos de desesperación y flaqueza. Sin tus ánimos, tu cariño y paciencia no hubiera afrontado las cosas de la misma manera que lo he hecho. Te quiero.

Gracias a mi Comando, es especial a mis chicas, los momentos de desconexión, cervezas y risas con vosotros que me daban fuerzas renovadas para afrontar los días siguientes mucho más relajada. Espero que no nos separemos nunca, por muy lejos que estemos las unas de las otras.

Quiero agradecer a mis chicos Deloitteiros su compañía y amistad durante el periodo que estuve trabajando con ellos. El conocerlos fue una de las mejores experiencias que gané mientras estuvimos en Walqatraz.

Gracias a mi familia, en especial a mis padres y mi abuelo, que siempre han creído en mí y se han interesado por este proyecto, aunque no entendieran qué estaba haciendo. Espero que os sintáis muy orgullosos de vuestra ingeniera.



---

# Memoria

Publindex: Análisis de la calidad de  
publicaciones de investigación

---





# PUBLINDEX: ANÁLISIS DE LA CALIDAD DE PUBLICACIONES DE INVESTIGACIÓN

## **RESUMEN**

El proyecto Publindex comprende el análisis, diseño, gestión e implementación de una aplicación web de consulta cuyo uso está dirigido a todo el personal perteneciente al ámbito de la investigación científica.

La calidad de una conferencia o revista de investigación indica cuál es el nivel de exigencia que se aplica a los artículos enviados para ser publicados en estas revistas y eventos. Cuanto mayor es la calidad de una publicación más referencias se hacen a ella, menor número de artículos publica y más prestigio tiene estos trabajos. Existen índices en la Web que dan valoraciones a los eventos y revistas de divulgación científica siguiendo un criterio propio. Por ejemplo, JCR es una fuente de calidad de revistas que asigna un factor de impacto numérico a cada una; CORE es un índice de conferencias y revistas que asigna valoraciones basadas en letras (A, B, C...)...

La motivación de este proyecto surge como consecuencia de la manera en la que el personal investigador recopilaba la información de la calidad a través de estos índices. Consultar múltiples sitios web resultaba tedioso y a veces requería emplear cierto tiempo. De esta necesidad surge la idea de Publindex, cuya finalidad es la de desarrollar una aplicación, que permita consultar de una manera sencilla, rápida y centralizada la calidad de las publicaciones, almacenando para ello la información de ciertos índices de calidad.

La aplicación fue construida de manera progresiva. En un primer momento, se desarrolló un sistema que permitía consultar el índice JCR. Posteriormente, se incorporó CORE, requiriéndose un nuevo análisis, diseño e implementación debido a que en un primer momento se había desarrollado un sistema muy específico para un índice concreto. Finalmente, se añadió SJR sin suponer problemas, ya que el sistema ya se había adaptado para poder almacenar y manejar la información de varias fuentes de calidad.

La consulta de la calidad puede realizarse a través de tres tipos de analizadores diferenciados y dependientes entre sí que constituyen las tres principales funcionalidades de la aplicación. El primero en desarrollarse fue el analizador de publicaciones, que busca las valoraciones de calidad de una publicación concreta dado su nombre, siglas o ISSN. Posteriormente se desarrolló el analizador de listados de publicaciones a través de un conjunto de referencias bibliográficas y el analizador de la calidad de un autor a través de las citas de su trabajo.

Además de las funcionalidades presentadas, la aplicación destaca por incorporar métodos de consulta online de información y por la usabilidad que hizo que el sistema fuera utilizado desde sus primeras versiones por el personal investigador del grupo de Sistemas Informáticos Distribuidos (SID).



## Tabla de contenido

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>1</b>
1.1	OBJETIVOS .....	1
1.2	TERMINOLOGÍA EMPLEADA.....	2
1.3	HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS.....	2
1.4	RESUMEN.....	3
<b>2</b>	<b>INFORMACIÓN GENERAL DEL PROYECTO .....</b>	<b>5</b>
2.1	METODOLOGÍA.....	5
2.2	FASES DEL PROYECTO .....	6
<b>3</b>	<b>ANÁLISIS .....</b>	<b>7</b>
3.1	ÍNDICES DE CALIDAD.....	7
3.1.1	<i>Journal Citation Reports (JCR)</i> .....	7
3.1.2	<i>Computing Research and Education (CORE)</i> .....	8
3.1.3	<i>Scientific Journal Rankings (SJR)</i> .....	9
3.1.4	<i>Computer Science Conference Ranking (CSCR)</i> .....	9
3.2	DESCRIPCIÓN DEL SISTEMA .....	10
3.2.1	<i>Análisis de calidad de una publicación</i> .....	10
3.2.2	<i>Análisis de un listado de publicaciones</i> .....	11
3.2.3	<i>Análisis de la calidad de un autor</i> .....	11
3.2.4	<i>Mantenimiento de datos</i> .....	11
3.3	TRABAJO PREVIO.....	12
3.4	CAPTURA DE REQUISITOS .....	12
3.4.1	<i>Requisitos funcionales</i> .....	12
3.4.2	<i>Requisitos no funcionales</i> .....	13
3.5	CASOS DE USO.....	13
3.6	MODELO DE DOMINIO.....	15
<b>4</b>	<b>DISEÑO DEL SISTEMA .....</b>	<b>17</b>
4.1	HOMOGENEIZACIÓN DE LA APLICACIÓN .....	17
4.2	ARQUITECTURA DEL SISTEMA .....	17
4.3	DISEÑO DE LA BASE DE DATOS .....	18
<b>5</b>	<b>PROTOTIPO (IMPLEMENTACIÓN) .....</b>	<b>21</b>
5.1	DECISIONES DE IMPLEMENTACIÓN.....	21
5.2	TECNOLOGÍAS UTILIZADAS.....	22

5.2.1	<i>Java</i> .....	22
5.2.2	<i>MySQL</i> .....	22
5.2.3	<i>Struts y Apache Tomcat</i> .....	23
5.2.4	<i>Hibernate</i> .....	23
5.2.5	<i>JSP</i> .....	24
5.2.6	<i>JUnit</i> .....	24
5.2.7	<i>JFlex y CUP</i> .....	24
5.3	EVOLUCIÓN DEL PROTOTIPO .....	25
5.3.1	<i>Evolución de la aplicación</i> .....	25
5.3.2	<i>Evolución de la base de datos</i> .....	27
5.3.3	<i>Diseño final</i> .....	28
5.3.4	<i>Evaluación de la implementación del extractor de publicaciones a través de citas bibliográficas</i> .....	30
5.4	PRUEBAS REALIZADAS .....	32
<b>6</b>	<b>CONCLUSIONES</b> .....	<b>33</b>
6.1	CRONOGRAMA .....	33
6.2	SEGUIMIENTO DEL PROYECTO .....	35
6.3	PROBLEMAS ENCONTRADOS .....	36
6.4	POSIBLES AMPLIACIONES .....	36
6.5	OPINIÓN PERSONAL .....	37
<b>7</b>	<b>BIBLIOGRAFÍA</b> .....	<b>39</b>
7.1	REFERENCIAS WEB .....	39
7.2	REFERENCIAS BIBLIOGRÁFICAS .....	39
	<b>ANEXOS</b> .....	<b>41</b>
	<b>ANEXO A: CAPTURA DE REQUISITOS</b> .....	<b>43</b>
	<b>ANEXO B: CASO DE ESTUDIO</b> .....	<b>59</b>
	<b>ANEXO C: IMPLEMENTACIÓN</b> .....	<b>115</b>
	<b>ANEXO D: FASE DE PRUEBAS</b> .....	<b>155</b>
	<b>ANEXO E: MANUAL DE USUARIO</b> .....	<b>189</b>

# 1 Introducción

---

Actualmente, existen en la Web diversos índices que asignan una cierta valoración a las publicaciones de investigación, siguiendo distintos criterios. Sin embargo, ya que todos estos rankings no se encuentran centralizados, la tarea de recopilar los índices de calidad de una publicación es una tarea que requiere de un cierto tiempo, además de necesitar que el usuario conozca los distintos métodos de valoración de cada uno de dichos índices. Cuando se trata de analizar la calidad de una lista de publicaciones, esta tarea se vuelve muy costosa.

El propósito de este proyecto consiste en la realización de una aplicación que permita analizar la calidad de revistas y conferencias de investigación a través de valoraciones ofrecidas por estos índices de calidad. Para ello, la aplicación reunirá la información de aquellos índices que se consideren más fiables y actuales.

A lo largo de este documento, se usará el término Publindex para hacer referencia a la solución final a la que se llegó tras completar el proceso de desarrollo.

## 1.1 Objetivos

Los objetivos funcionales del proyecto son los siguientes:

- Se ofrecerá un buscador accesible a través de la Web que permita obtener las valoraciones de calidad de una publicación, de una lista de publicaciones o de un investigador.
- Englobará en una sola aplicación las diversas fuentes a través de las cuales se suele valorar la calidad de las publicaciones de investigación, con especial énfasis en aquellas pertenecientes al área de Informática.
- La información de las publicaciones extraída de la Web será almacenada y/o extraída de manera automática para garantizar la disponibilidad de dicha información en el futuro, además de disminuir significativamente el tiempo de acceso a la misma.
- Cada uno de los índices será encapsulado de la manera más independiente posible, abstrayéndonos de los aspectos sintácticos particulares de cada índice Web, para conseguir una aplicación robusta y adaptable fácilmente frente a cambios en la información disponible *online*.
- La información disponible de cada índice será adecuada de manera que la aplicación sirva también como fuente de búsqueda de otros datos relativos a las publicaciones (categorías, país al que pertenecen, lenguaje,...).
- Se añadirá una valoración mucho más genérica de la calidad de las publicaciones, dando una presentación independiente del criterio de puntuación seguido de cada índice.

- Permitirá obtener de manera automática los datos relativos a las referencias bibliográficas de un investigador.

Todos estos objetivos serán identificados y desglosados como requisitos funcionales a analizar, diseñar e implementar durante el proceso de desarrollo.

## 1.2 Terminología empleada

Durante esta memoria se utilizan una serie de términos que son necesarios definir para poner en contexto al lector no familiarizado con esta terminología.

- Investigador: persona que desea analizar la calidad de una o varias publicaciones a través de la aplicación.
- Índice de calidad: fuente existente en la web que asigna valoraciones de calidad a las publicaciones y de la que se importan datos al sistema. Índices de calidad son por ejemplo JCR, SJR, CORE y CSCR.
- Publicación: evento o revista científica a la que se le asigna una valoración de calidad.
- Revista: publicación tipo artículo de carácter serio y de divulgación científica. Es uno de los tipos de publicación que pueden ser buscados en el sistema. Estas revistas también reciben el nombre de *journals* de investigación.
- Conferencia: evento público sobre un asunto científico, filosófico, literario o empresarial. En una conferencia se presentan trabajos y avances realizados por investigadores con objeto de su divulgación y reconocimiento. De estas conferencias se elaboran actas que es el otro tipo posible de publicación de la cual analizar la calidad en el sistema.
- Categoría: permite englobar en un conjunto una serie de publicaciones que poseen una temática en común. Ejemplos de categorías serían inteligencia artificial, robótica, biblioteca, ciencias naturales, sociología...
- Macro categoría: categoría que engloba una serie de categorías. Es utilizada para definir el contexto donde realizar la búsqueda. Por ejemplo, las categorías de inteligencia artificial y robótica de la definición anterior pertenecen a la macro categoría de informática.
- Factor de impacto (FI): valoración de tipo numérico asignada por JCR para indicar la calidad de una publicación. SJR también presenta una especie de factor de impacto.

## 1.3 Herramientas y tecnologías utilizadas

Parte de las herramientas y tecnologías utilizadas han sido seleccionadas por su mayor compatibilidad con el lenguaje Java en el que se implementa la aplicación. Sin embargo, otras herramientas y tecnologías utilizadas fueron elegidas tras realizar una comparativa de varias posibilidades existentes; sus ventajas frente a las otras o su idoneidad para el proyecto en sí llevaron a que fueran seleccionadas.

Las tecnologías y herramientas utilizadas en el proyecto han sido:

- Java como lenguaje de programación de la aplicación web.
- Struts para Java, como estructura y herramienta de desarrollo de la aplicación.
- JSP como herramienta de desarrollo de las interfaces web.
- Hibernate para mapear los objetos de la aplicación con los datos almacenados en BBDD.
- MySQL como Sistema Gestor de la Base de Datos.
- SQLyog como herramienta de gestión del sistema gestor MySQL.
- Eclipse versión Ganymede 3.4.2 como entorno de programación y depuración de código del proyecto.
- JUnit para realizar las pruebas unitarias del sistema.
- Apache Tomcat 6.0 como servidor de la aplicación.
- JFlex como analizador léxico para reconocer y procesar texto.
- CUP como analizador sintáctico para reconocer y procesar texto.
- Microsoft Internet Explorer 7.0, 8.0, Mozilla Firefox y Google Chrome como navegadores donde la aplicación ha sido lanzada y probado su funcionamiento.
- Microsoft Office 2007 para la generación de la documentación.
- Adobe Acrobat 7.0 Professional para la generación de documentos en formato PDF.
- Star UML 5.0.2 para la generación de diagramas de clases y UML.
- Edge Diagrammer 4.1 para la generación de esquemas y diagramas Entidad-Relación de base de datos.

Los principales motivos que llevaron al uso de las tecnologías citadas anteriormente se incluyen en la sección número cinco del prototipo de la aplicación.

## **1.4 Resumen**

Además de la introducción que se presenta en esta sección, se incluyen una serie de secciones que aportarán al usuario una visión global de cada parte concreta del proceso de desarrollo de Publindex.

La segunda sección facilita información general del proyecto y que es tomada en cuenta a lo largo de todo el proceso de desarrollo.: metodología seguida, fases del proyecto...

La tercera sección da una visión general de qué hace el sistema a través del análisis. Dentro de este apartado se presentan los requisitos funcionales y no funcionales identificados, casos de uso, trabajo previo, análisis de las fuentes de calidad a utilizar...

La cuarta sección describe todo el proceso de diseño, mostrando la arquitectura del sistema, el modelo de base de datos y recursos software que modelan el sistema a implementar.

La quinta sección muestra el prototipo de la aplicación comentando todos aquellos aspectos importantes de la implementación como decisiones tomadas, tecnologías utilizadas, versiones de la aplicación, manual de usuario, estadísticas, etc.

La sexta sección contiene las conclusiones finales del proyecto. Este apartado da una descripción resumida y valoración crítica del conjunto del trabajo desarrollado, como cumplimiento de los objetivos, posibles ampliaciones, mejoras del trabajo u opinión personal.

Finalmente, se presenta una última sección con las referencias bibliográficas que han sido soporte durante el proceso de desarrollo y que pueden servir de apoyo al lector para consultar todo aquello que no haya sido descrito de manera detallada y de lo que se tenga una vaga idea.

Dado que este documento tiene como objetivo mostrar una visión global y resumida de todo el proceso de desarrollo de Publindex, se realizan continuas referencias a otros documentos que presentan un desglose mucho más concreto y extendido de cada fase del proyecto que ha tenido una participación relevante en el trabajo realizado.

Los documentos anexos a los que se hará referencia serán los siguientes:

- Anexo A: Especificación de requisitos
- Anexo B: Caso de estudio del proyecto
- Anexo C: Implementación
- Anexo D: Fase de pruebas
- Anexo E: Manual de usuario



## 2 Información general del proyecto

Este apartado tiene como propósito exponer aquellos aspectos más generales del proyecto y que definen la línea de trabajo seguida durante el proceso de desarrollo. Dentro de esta sección se describe la metodología seguida y el trabajo realizado durante todas las fases que componen el proyecto.

### 2.1 Metodología

La metodología que se ha seguido durante el proceso de desarrollo ha sido el Proceso Unificado. Esta metodología está basada en un modelo iterativo e incremental que va definiendo nuevas funcionalidades o refinando las ya implementadas. El proceso de desarrollo es dividido en cuatro fases: ideación, elaboración, construcción y transición (ver figura 1).

Esta metodología fue seleccionada debido a que el sistema presenta requisitos o funcionalidades no muy definidas inicialmente y dependientes de otras a desarrollar. Por ejemplo, parte del análisis, diseño e implementación del analizador de un listado de publicaciones depende directamente de cómo se desarrolle el buscador de una publicación.

Una de las principales características del Proceso Unificado es que el desarrollo está dirigido por los casos de uso, con objeto de capturar los requisitos funcionales y definir los objetivos de las iteraciones. El modelado del sistema software (análisis, diseño...) se realiza utilizando UML. Para más información, consultar referencia [PU].

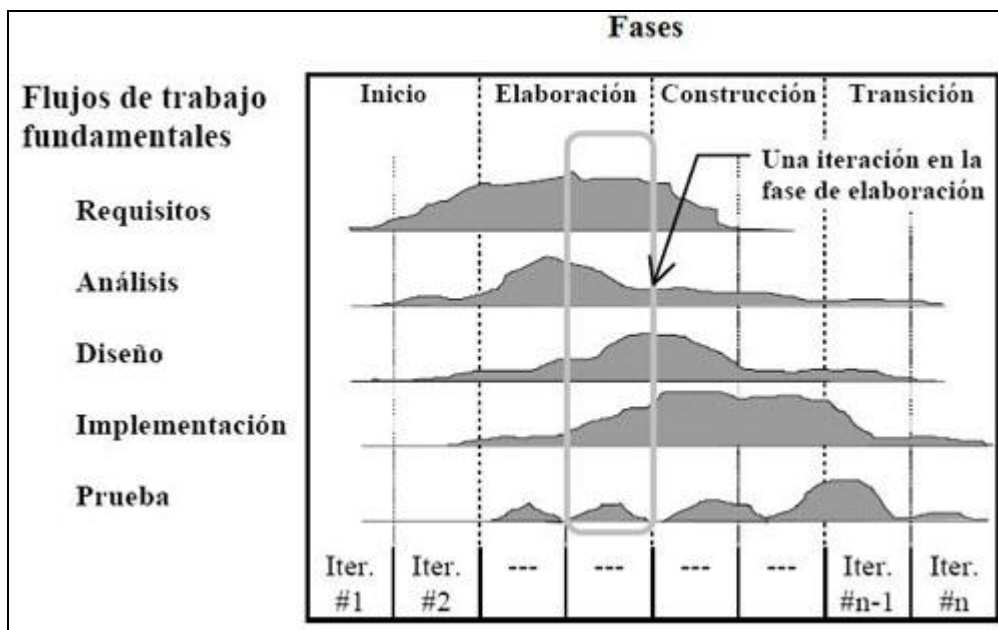


Figura 1: Fases del Proceso Unificado

## **2.2 Fases del proyecto**

El proyecto se divide en las cuatro fases definidas por el Proceso Unificado: fase de ideación, de elaboración, construcción y transición [PU]. Los siguientes párrafos describen el trabajo realizado en cada una de ellas para dar una visión más concreta del proceso de desarrollo y de la metodología. Para una descripción más detallada, consultar el anexo B de Caso de Estudio.

La fase de ideación pretende definir en torno al 80% de la funcionalidad del sistema y así determinar su viabilidad. Esta fase se compone de la identificación y definición de la mayor parte de los casos de uso. Durante la ideación se realizaron dos iteraciones. La primera de ellas identificó las tres principales funcionalidades de búsqueda; la segunda iteración comprendió la definición de la funcionalidad del mantenimiento de base de datos.

La fase de elaboración recoge las labores de análisis y diseño de las funcionalidades más importantes y complejas de la aplicación. Durante la elaboración también se realizaron dos iteraciones, la primera desarrollando la funcionalidad del buscador de una publicación y la segunda desarrollando los otros dos subsistemas buscadores y el de mantenimiento. Esta fase también comprendió las labores de implementación de la arquitectura del sistema y las decisiones relativas a cómo realizar la carga de las fuentes de calidad seleccionadas para almacenarse.

La fase de construcción comprendió las labores de implementación que se realizaron para ir refinando la aplicación, componiéndose de varias iteraciones que llevaron a la versión definitiva del sistema. Durante esta fase ya se comenzaron a realizar pruebas de los usuarios sobre las versiones que iban siendo generadas.

La fase de transición comenzó una vez que se tuvo una versión final y estable de la aplicación y superó todas las pruebas de validez, tanto a nivel de sistema como de usuario. El trabajo realizado en esta fase comprendió las labores de finalización, despliegue y documentación del proyecto.

## 3 Análisis

---

Este apartado tiene como propósito exponer aquellos aspectos más importantes de las labores de análisis del sistema. Dentro de esta sección se incluyen los siguientes apartados: una descripción del sistema a construir, un breve estudio de los índices de calidad considerados como posibles fuentes, el trabajo previo al análisis, requisitos, casos de uso y modelo de dominio seguido.

### 3.1 Índices de calidad

Esta sección detalla las principales características de los índices de calidad de publicaciones que serán considerados a utilizar por su prestigio, versiones actuales y fiabilidad. Las fuentes descritas son JCR, SJR, CORE y CSCR. Para un mayor detalle, consultar el anexo de caso de estudio (Anexo B).

#### 3.1.1 Journal Citation Reports (JCR)

JCR es un índice de calidad de revistas de divulgación científica y social. Dentro de cada uno de estos grupos clasifica las revistas en categorías según la temática a la que pertenezcan [JCR].

La calidad se presenta anualmente a través de un factor de impacto numérico que indica el número de referencias a la revista en los últimos dos años dividido por el número de artículos aceptados en dicha revista. Este factor de impacto es único para una publicación en un año concreto y podría considerarse como una especie de media de citas recientes por artículo aceptado en la revista.

Debido a que el número de categorías es elevado, Publindex sólo considerará la carga de datos de aquellas categorías científicas pertenecientes al área de informática. Cada categoría presenta las revistas que la componen ordenadas por factor de impacto, indicando para cada publicación datos importantes como son su título, ISSN, país o lenguaje. Una revista puede incluirse en más de una categoría dentro de un mismo año.

El proceso de consulta de JCR resulta no escaso de pasos. Los pasos a dar por un usuario de JCR son los siguientes:

1. Acceder a la página web de *ISI Web of Knowledge*.
2. Introducir usuario y contraseña para acceder a la base de datos.
3. Seleccionar la base de datos de JCR.
4. Elegir el año en el cual buscar la revista y el nombre concreto.
5. Del listado de resultados, elegir la revista deseada (en esta parte ya puede consultarse el factor de impacto).

6. Para saber su clasificación en una categoría, acceder al enlace '*Journal rank in categories*'.

Esta manera de consultar la calidad de una revista implica tener conocimientos de en qué años exactos ha sido valorada. Además, la información de la posición que ocupa en cada categoría sólo está disponible para el año consultado resultando muy pesado consultar varios años en una categoría teniéndose que hacer estos cinco pasos para cada año.

### 3.1.2 Computing Research and Education (CORE)

Este índice da valoraciones de la calidad de revistas y conferencias que han sido referenciadas por investigadores australianos [CORE]. CORE asigna anualmente valoraciones únicas para cada publicación basadas en letras con el siguiente orden decreciente: A\*/A+, A, B, C, L, U.

Aunque sólo se facilitan dos rankings, uno para revistas y otro para conferencias, las publicaciones se agrupan por macrocategorías y categorías a través de su código FoR. Cada macrocategoría tiene un código FoR de dos dígitos y puede estar compuesta por varias categorías. Por ejemplo, la macrocategoría *Information and Computing Sciences (FoR 08)* contiene categorías como *Data Format (FoR 0804)* o *Computer Software (FoR 0803)*. Las macrocategorías más importantes y que contienen la mayor parte de las publicaciones hacen referencia al área de informática, ingeniería y tecnología (códigos 08, 09 y 10). Dentro de una categoría las publicaciones con una misma valoración no tienen orden, pudiendo una publicación pertenecer a varias categorías diferentes.

Para realizar la consulta de la calidad a través de CORE deben darse los siguientes pasos:

1. Acceder al sitio web de COnputing Research & Education (<http://core.edu.au/>).
2. Seleccionar la opción del menú *Journal Rankings* o *Conference Rankings*.
3. Los listados de valoraciones pueden ordenarse según varios criterios: ordenados por valoración, alfabéticamente, por código FoR, por valoración y código FoR...
4. Una vez se seleccione el criterio de ordenación, se descarga el fichero PDF que contiene la información.
5. La revista o conferencia debe buscarse en el fichero "a mano".

CORE es un índice poco práctico para consultar la calidad de publicaciones. En primer lugar, cuando actualiza la información en su página web se pierde la antigua, de modo que si se desea buscar la calidad de otros años es conveniente tener un histórico de cada versión que va siendo publicada. Por otro lado, no puede verse la información relativa a una categoría, además de que es necesario conocer a qué categoría hace referencia cada código FoR.

Este índice también presenta una problemática relativa a la consistencia de la información. De un año a otro los nombres de las publicaciones pueden variar, ocasionando que información de una publicación se cargue como información de publicaciones diferentes en distintos años en el sistema a desarrollar.

### 3.1.3 Scientific Journal Rankings (SJR)

SJR es un portal web que recopila los indicadores de calidad de la base de datos *Scopus* para revistas. Al igual que JCR agrupa las publicaciones por categoría, pudiendo también organizarlos por país al que pertenecen [SJR].

Este índice también asigna anualmente un factor numérico de la calidad único para cada revista. La medida es el número de referencias a la publicación dividido por el número de artículos aceptados en los últimos tres años.

SJR también presenta muchas clasificaciones por categorías, por lo que el sistema sólo recopilará la información de aquellas relacionadas con el área de informática. Los pasos a dar para consultar la información de calidad de SJR son los siguientes:

1. Acceder a la página web de *Scimago Journal & Country Rank* (<http://www.scimagojr.com/index.php>).
2. Seleccionar la opción del menú *Journal Search*.
3. Introducir el nombre o ISSN de la revista de la que encontrar la calidad.
4. Seleccionar del resultado de la búsqueda la revista deseada.
5. Para saber su clasificación en las categorías, debe accederse al enlace de cada una de ellas.

Esta web muestra mucha más información que JCR en una sola búsqueda. Por ejemplo, puede consultarse la evolución del factor SJR de calidad y las citas por año para todos los años desde 1999 a 2008 a través de una gráfica. Sin embargo, al igual que ocurría para JCR, en el caso de que se quiera consultar la posición de una revista en una categoría para varios años no puede hacerse con una sola búsqueda. Para ello, es necesario consultar la información de cada categoría año a año.

### 3.1.4 Computer Science Conference Ranking (CSCR)

CSCR es un índice de calidad para conferencias pertenecientes al área de informática. Al igual que SJR y JCR, asigna una valoración numérica a cada evento entre 0.00 y 0.99 agrupándolos por categoría.

Una conferencia sólo es incluida en una sola categoría, incluyendo sólo aquellas conferencias con valoración mayor o igual a 0.5. Cuanta más alta es la valoración mayor es el prestigio de la conferencia.

Las conferencias aparecen ordenadas dentro de una categoría por valoración, indicando para cada una de ellas su título, siglas y valoración. Una categoría engloba varias temáticas, como por ejemplo la categoría '*Applications / Education / Software / Theory / Communications / Graphics / Bioinformatics*', que contiene las conferencias pertenecientes a cualquiera de las áreas citadas en el nombre de la categoría.

Para realizar la consulta de la calidad de una conferencia deben realizarse las siguientes acciones:

1. Acceder al sitio web de *Computer Science Conference Ranking* (<http://www.cs-conference-ranking.org/conferencerankings/alltopics.html>).
2. Buscar en el listado de categorías aquella a la que se sepa que pertenece la conferencia.
3. Buscar dentro de esa categoría la conferencia a través de su nombre.

CSCR parece estar desfasado. Desde finales del 2009 su página web no está disponible, no pudiendo consultar información actualizada. Al igual que CORE es necesario guardar un historial de las versiones antiguas y sólo puede consultarse la valoración de aquellas publicaciones con un factor mayor que 0.5. No obstante, sería aconsejable tener más de un índice de valoraciones para conferencias.

## **3.2 Descripción del sistema**

El usuario podrá analizar la calidad de tres maneras diferentes:

- Analizar una publicación, bien sean revistas o conferencias.
- Analizar un listado de publicaciones a las que hacen referencia una serie de citas bibliográficas.
- Analizar la calidad de un autor a través de las publicaciones en las que ha publicado sus trabajos, referenciados a través de citas bibliográficas.

### **3.2.1 Análisis de calidad de una publicación**

Las búsquedas en el sistema darán como resultado las valoraciones que las publicaciones han registrado según varias fuentes de calidad. Estos índices agrupan las publicaciones por categoría, pudiendo una publicación pertenecer a más de una. Sin embargo, la valoración asignada por cada índice es única para un año en concreto, algo que resulta un baremo poco significativo de la calidad. Por ejemplo, decir que una publicación tiene valoración 'A' cuando el resto de publicaciones de una categoría tiene valoración 'C' es bueno, pero si pertenece a otra categoría cuyo resto de publicaciones tuvieran valoración 'A+' sería negativo.

Como consecuencia de esta problemática, el sistema asignará a cada publicación una valoración en cada categoría a través de un porcentaje en función de la posición de la publicación en la categoría respecto del total de publicaciones de ésta.

El sistema también será utilizado como fuente de información, consultando datos propios de las revistas y conferencias, como título, el lenguaje en el que está escrito, el país o ISSN.

Además, también podrá consultarse información relativa a las categorías de un índice de calidad, visualizando las publicaciones que la componen y la posición de éstas para un año en concreto.

### 3.2.2 Análisis de un listado de publicaciones

El análisis de citas bibliográficas permitirá al usuario dado un conjunto de citas, con un cierto formato, consultar la calidad de los eventos y revistas donde han sido publicados los artículos y así poder determinar el prestigio de éstos.

Esta búsqueda estará relacionada directamente con el analizador individual de una publicación para analizar cada publicación del listado al que hacen referencia las citas.

### 3.2.3 Análisis de la calidad de un autor

La calidad de un autor viene determinada por cuánto prestigio tiene la revista o la conferencia en la que aparezcan sus trabajos. Esta búsqueda de la calidad obtendrá el listado de referencias de un autor y seguidamente se realizará el análisis del listado de publicaciones.

El siguiente esquema muestra visualmente la relación que guarda este mecanismo de análisis con los otros dos explicados anteriormente:

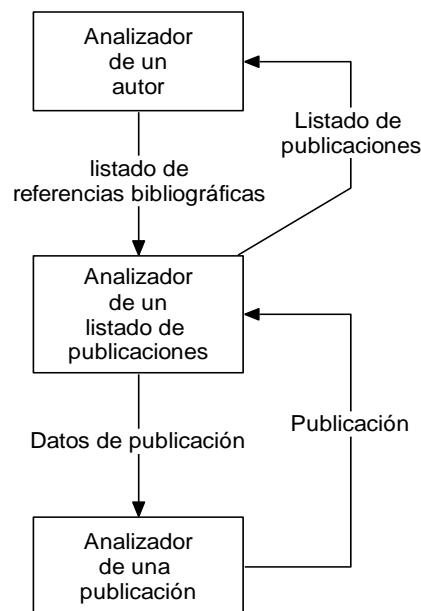


Figura 2: Relación ente los tres analizadores de la aplicación

### 3.2.4 Mantenimiento de datos

Al cargar los datos de los índices de calidad comentados anteriormente, puede surgir la problemática de haber almacenado datos relativos a varias publicaciones que en realidad corresponden a una sola. Esto puede ser debido a que cada índice le asigna un título diferente a una misma publicación o que en un mismo índice el título de un evento varía de un año a otro.

Ante esta situación, la aplicación proporcionará también un mecanismo de mantenimiento de la base de datos, para eliminar duplicados de publicaciones y almacenar en un solo registro de base de datos toda la información relativa a las publicaciones que han sido fusionadas.

### **3.3 Trabajo previo**

Una vez analizados y estudiados los posibles índices de calidad, se tomarán como fuentes de datos *JCR (Journal Citation Reports)*, *CORE (COMputing Research & Education)* y *SJR (Scientific Journal Rankings)*. Las principales características que reúnen estas fuentes son la fiabilidad de la información y que los datos son actualizados de manera periódica (anualmente, trimestralmente,...) de modo que siempre se dispondrá de una información relativamente actual. El índice CSCR se dejará aparcado para una futura incorporación en el caso de que se vayan cumpliendo los plazos de tiempo. En futuras ampliaciones también podrán añadirse nuevos índices de calidad.

Otra de las principales razones para elegir estas fuentes es que los datos se facilitan a través de ficheros en formato Excel y PDF. Estos formatos pueden ser procesados más rápidamente para extraer la información y almacenarla.

### **3.4 Captura de requisitos**

Los requisitos han sido divididos en funcionales y no funcionales. Los requisitos funcionales permiten dar una visión de todas las funcionalidades que presenta el sistema, mientras que los requisitos no funcionales especifican ciertas exigencias a nivel de rendimiento, nivel tecnológico, entorno de desarrollo, de consistencia, etc.

#### **3.4.1 Requisitos funcionales**

Las principales funcionalidades del sistema están englobadas dentro de las tres funciones de análisis de calidad que debe proporcionar la aplicación. Los requisitos identificados dentro de este grupo son los siguientes:

- Analizar una publicación por nombre/siglas o ISSN.
- Analizar un listado de publicaciones a través de referencias bibliográficas.
- Analizar la calidad de un autor.
- Mostrar la información de una categoría de índice.
- Asignar una valoración independiente del índice.
- Ofrecer un servicio de mantenimiento de datos.

Estos requisitos serán englobados en los casos de uso identificados en el sistema. Para ver en más detalle estos requisitos consultar en anexo A de captura de requisitos.



### 3.4.2 Requisitos no funcionales

Los requisitos no funcionales identificados aparecen en el siguiente listado junto a la categoría de requisito a la que pertenecen:

- Entorno tecnológico: La aplicación web debe estar implementada en Java versión 6.0 en adelante.
- Disponibilidad: El sistema debe ser accesible las 24h del día y ser usado por varios usuarios de manera simultánea sin que se vea afectado el tiempo de respuesta.
- Accesibilidad: El sistema tiene limitado el acceso, siendo necesario autenticarse al acceder a cualquier URL relativa a la página de Publindex.
- Compatibilidad de navegadores: compatible con los exploradores Internet Explorer 7.0 en adelante, Mozilla Firefox y Google Chrome.
- Rendimiento: El tiempo de espera del usuario no debe sobrepasar los 15 segundos.
- Otros requisitos: El sistema debe mostrar estadísticas relativas al sitio web (accesos diarios, estimaciones de accesos mensuales, etc.).

Estos requisitos no funcionales aparecen con mayor detalle en el anexo A de captura de requisitos.

### 3.5 Casos de uso

El siguiente diagrama de contexto muestra el escenario en el que se ejecuta la aplicación, identificando los posibles actores y principales acciones a realizar.

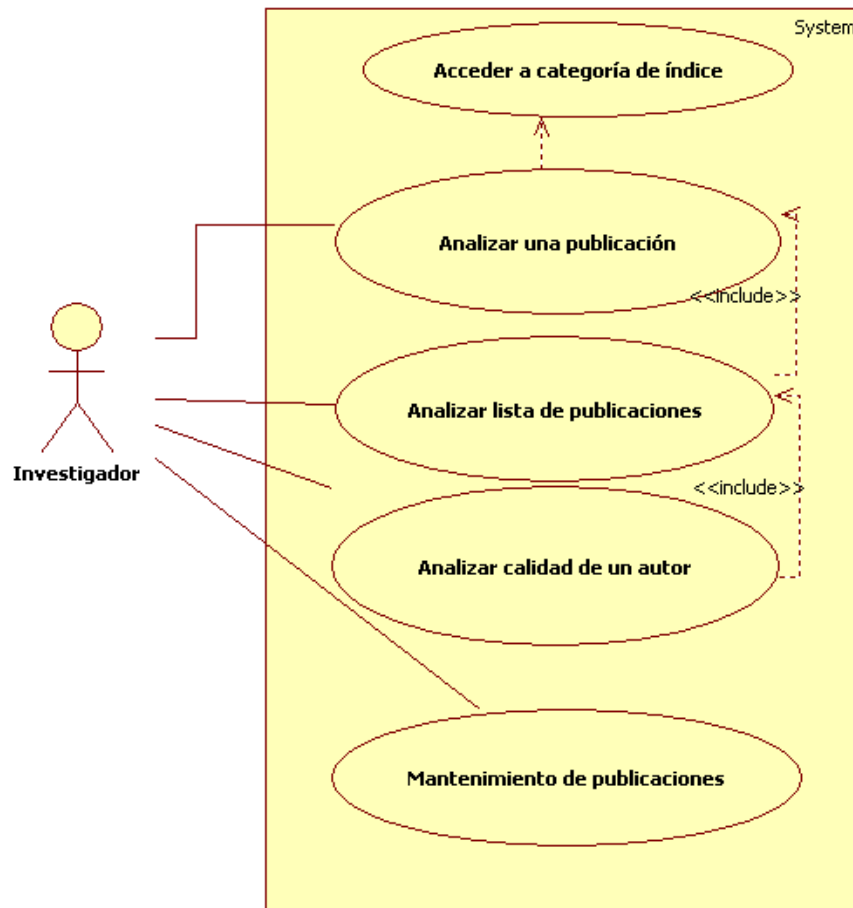


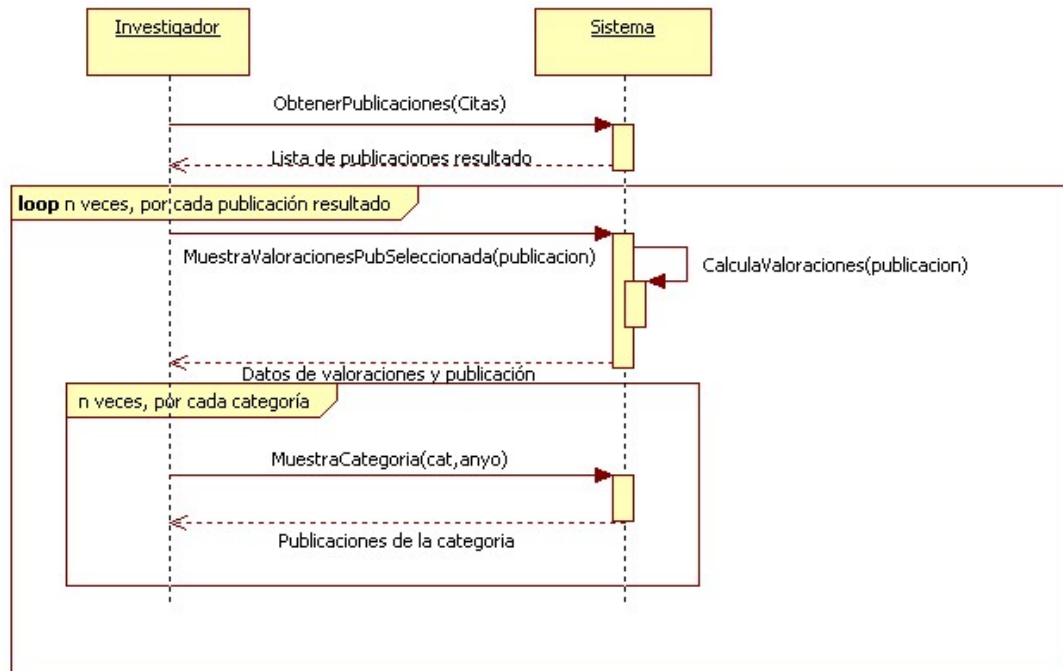
Figura 3: Diagrama de contexto del problema

El único actor existente es el investigador, que utiliza la aplicación para realizar principalmente búsquedas de la calidad de publicaciones.

Las acciones a realizar por el actor permiten identificar las principales entidades y paquetes para analizar del sistema. Estas entidades se corresponden con las funcionalidades identificadas en el apartado 3.2:

- Buscar una publicación y sus valoraciones de calidad de manera individual (Analizador de una publicación).
- Buscar la calidad de un listado de publicaciones a través de referencias bibliográficas (Analizador de un listado de publicaciones).
- Analizar la calidad de un autor.
- Mantenimiento de publicaciones.

Una vez se han definido las acciones e identificado sus funcionalidades se procede a la elaboración de diagramas de secuencia. Estos diagramas permiten representar los eventos de entrada y salida de cada caso de uso, tratando al sistema como una caja negra. Cada caso de uso y los diagramas aparecen descritos de manera detallada en el Anexo B del caso de estudio del proyecto. Un ejemplo de diagrama de secuencia es el siguiente:



*Figura 4: Diagrama de secuencia para el analizador de una lista de publicaciones a través de citas bibliográficas*

### 3.6 Modelo de dominio

Una vez que se ha identificado el problema (ver sección 3.2) es necesario identificar las clases reales del problema y las asociaciones que existen entre ellas. Este modelo sirve como base para elaborar el diagrama de clases software en la fase de diseño. Se identificaron las siguientes entidades y atributos conceptuales:

- Publicación: Nombre, País, Lenguaje, Siglas
- Revista: ISSN
- Conferencia
- Índice de calidad: Nombre
- Categoría: Nombre, Año
- Valoración de calidad: Valoración, Posición
- Autor: Nombre
- Cita: Título, Año
- Macrocategoría: Nombre

Las asociaciones entre ellas pueden verse en el siguiente gráfico:

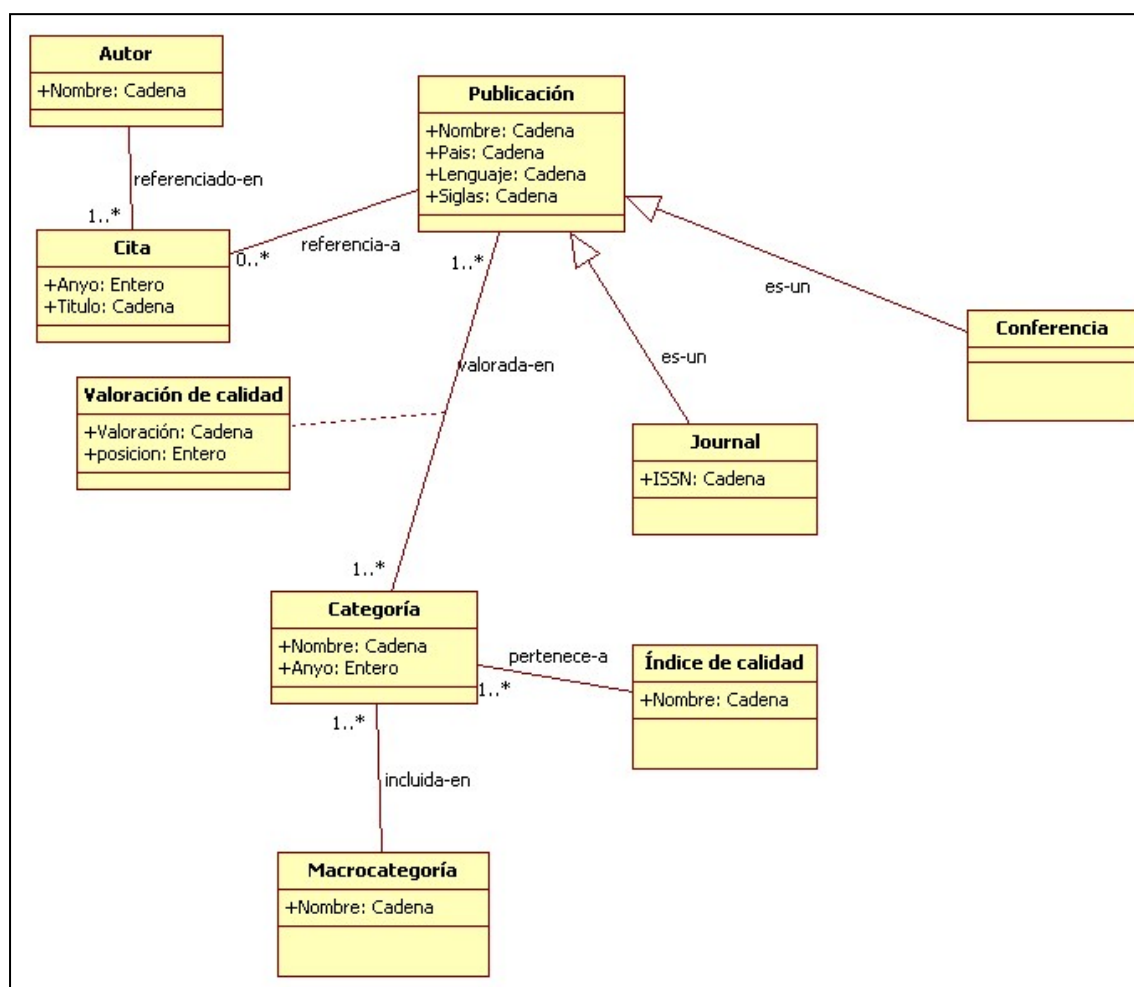


Figura 5: Modelo de dominio del análisis

Las asociaciones entre las entidades permiten representar el contexto del problema. Una publicación puede ser una revista (journal) o un acta de conferencia. Cada publicación tiene una valoración y ocupa una posición en una categoría de índice. Cada categoría está incluida en una macrocategoría y pertenece a un índice de calidad concreto. El trabajo de un autor está presentado en un conjunto de citas bibliográficas. Cada una de estas citas hace referencia a una publicación en concreto.

## 4 Diseño del sistema

---

Es en la fase de diseño cuando se toman las decisiones que llevarán a una homogeneización de la aplicación para tratar varias fuentes de calidad, uso de una base de datos para almacenar la información que consulta la aplicación y cómo estará estructurado el sistema. Además de estos puntos también se presentan los resultados de la tarea de diseño referente al diseño de la base de datos.

### 4.1 Homogeneización de la aplicación

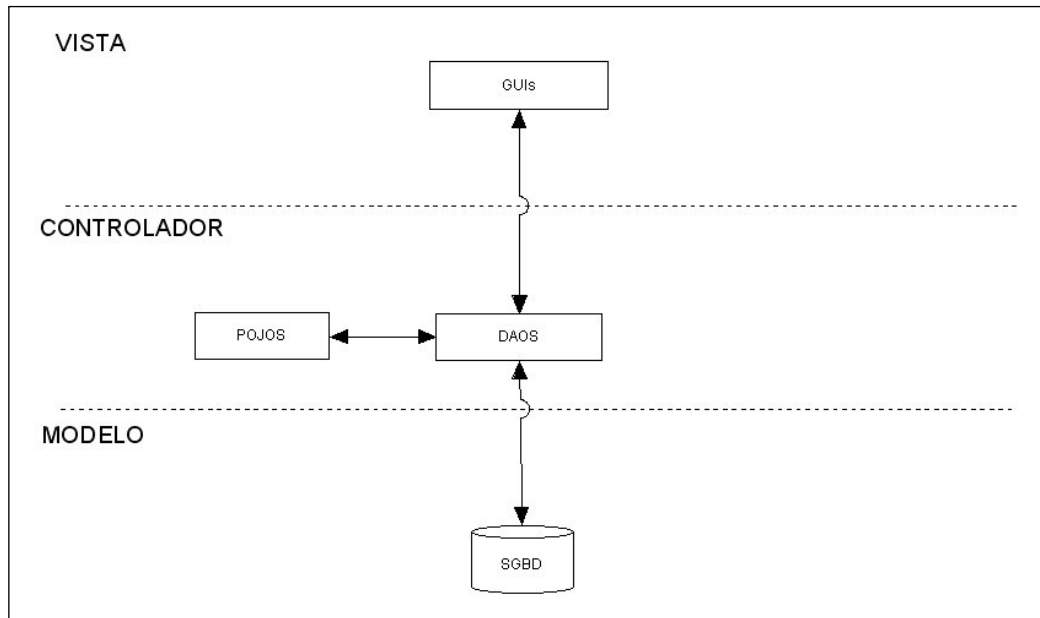
La aplicación almacena varios índices de calidad, teniendo que ser necesario diseñar un sistema que no sea específico para un índice en concreto. Las decisiones tomadas para cumplir esta premisa son las siguientes:

- La información será almacenada en base de datos. De esta manera, puede realizarse un diseño que almacene miles de tuplas de datos sin tratarlas diferente si pertenecen a un índice u a otro. Otra ventaja de la base de datos es que sólo es necesario realizar la carga de los índices de calidad una sola vez, además de un mantenimiento de los datos de una manera más sencilla.
- Realizar un modelado de clases software adecuado, de modo que a través de clases genéricas puedan representarse las publicaciones, categorías, valoraciones...independientemente del índice del que procedan. Un buen punto de partida es el modelo de dominio elaborado en el análisis.
- Será necesario guardar información de qué índices han sido cargados en el sistema con objeto de poder saber de qué fuente de datos procede una valoración y categoría.
- Considerar algún método de valoración genérico independiente de la fuente de calidad, como por ejemplo una valoración por categoría.

Estas decisiones quedan plasmadas en el diseño de la base de datos, modelado de clases y arquitectura del sistema que se presenta en detalle en el anexo B.

### 4.2 Arquitectura del sistema

La arquitectura se ha desarrollado utilizando un diseño multicapa, resultado de haber aplicado ciertos patrones de diseño, como el patrón MVC [UMLPatterns]. Esta estructura en capas facilita el mantenimiento y escalabilidad de la aplicación, haciendo que el desarrollo se lleve a cabo en varios niveles.



*Figura 6: Diseño de la estructura del sistema*

La capa de vista contendrá la parte visual de la aplicación que interactuará con el usuario. En esta capa no sólo vendrán incluidas las interfaces gráficas sino también aquellos ficheros o recursos que éstas utilicen (JavaScript, CSS, ficheros de recursos, etc.).

La capa de lógica permite abstraer a la vista del modelo físico de datos. Esta capa contendrá los controladores de acceso a datos (DAOS) y las entidades de negocio (POJOS), que constituyen los objetos a intercambiar entre las capas definidas (publicaciones, categorías, valoraciones...).

La capa de modelo contiene la representación física de los datos, con las tablas y relaciones y estará controlado por un Sistema Gestor de Base de Datos. El SGBD utilizado será elegido durante la implementación del sistema.

### **4.3 Diseño de la base de datos**

El diseño de base de datos consiste en identificar las entidades físicas de las cuales es necesario guardar información, utilizando para ello el modelo de clases definido. Además de identificar las entidades, es necesario definir sus campos y las relaciones entre ellas.

La base de datos fue evolucionando a la vez que era refinado el problema y era necesario almacenar información de nuevas entidades. El modelo final de diseño se presenta en la siguiente imagen, pudiendo comprobar su evolución más detalladamente en el anexo B del caso de estudio

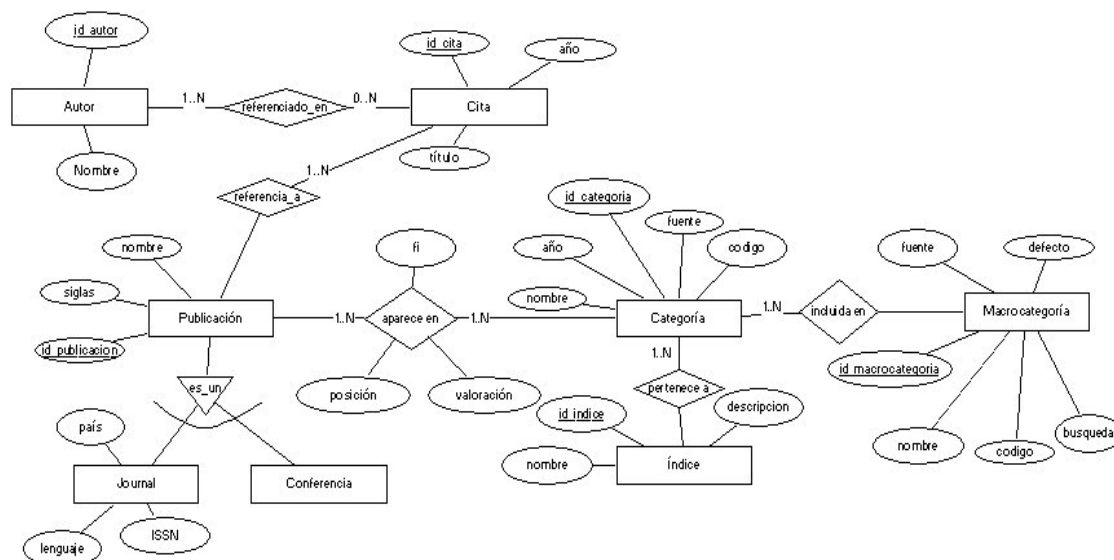


Figura 7: Esquema ER de la base de datos

Una publicación se especializa en revista (journal) o conferencia. Cada publicación se asocia con una categoría de índice, guardando información de la valoración, posición y factor de impacto que posee la publicación en esa categoría. Dentro de un índice existirá una instancia de categoría por cada año. Cada categoría estará incluida en una macrocategoría, que es a través de la cual se definirá el conjunto de categorías en el cual realizar búsquedas de publicaciones y valoraciones.

Deberá guardarse información de los autores, para poder acceder a la calidad a través de las publicaciones a las que referencian sus citas bibliográficas. De cada cita debe guardarse a qué año corresponde y el nombre del artículo publicado. A continuación se presentan las entidades y el tipo de dato de cada campo:

- Autor:
  - Id\_autor: entero (clave primaria)
  - nombre: cadena
- Cita:
  - Id\_cita: entero (clave primaria)
  - año: entero
  - título: cadena
- Publicación
  - Id\_publicación: entero (clave primaria)
  - Siglas: cadena
  - nombre: cadena
- Journal
  - Atributos y clave primaria de publicación
  - ISSN: cadena
  - lenguaje: cadena
  - país: cadena

- Conferencia
  - Atributos de publicación
- Categoría
  - Id\_categoria: entero (clave primaria)
  - fuente: cadena
  - código: cadena
  - año: entero
  - nombre: cadena
- Índice
  - Id\_indice: entero (clave primaria)
  - nombre: cadena
  - descripción: cadena
- Macrocategoría
  - Id\_macro categoría: entero (clave primaria)
  - nombre: cadena
  - código: cadena
  - fuente: cadena
  - búsqueda: booleano
  - defecto: booleano



## 5 Prototipo (implementación)

---

Este apartado contiene todas aquellas decisiones y aspectos que pueden ser cambiados sin invalidar el análisis y diseño anteriormente comentados. Dentro del prototipo se incluyen las decisiones de implementación tomadas, una justificación de las tecnologías utilizadas, la evolución del prototipo y las pruebas realizadas. Existen otros aspectos de implementación que pueden verse en más detalle en el anexo C de este documento.

### 5.1 Decisiones de implementación

Durante la implementación del sistema se tomaron una serie de decisiones que influyeron en el modelo final de clases y en la estructura del sistema:

- Se realizó la carga de datos de JCR, SJR y CORE a través de ficheros en formato CSV, XML o texto. Para ello tuvieron que implementarse algunos reconocedores de dichos formatos para obtener la información y almacenarla.
- Las citas bibliográficas poseen formato texto o BibTex, siendo necesario que se implementara un reconocedor para procesar la información y obtener los datos de las publicaciones a las que hacen referencia.
- Se implementó un mecanismo de búsqueda de la publicación exacta a la que hace referencia la cita bibliográfica para su posterior análisis. Este mecanismo consistía en encontrar aquella publicación de las coincidencias de la búsqueda cuyo título se asemeje más al extraído de la cita. Para calcular la semejanza entre dos cadenas, se recurrió a una librería de Java, *simmetrics*, cuya función *getSimmilarity* determinaba el porcentaje de similitud entre dos cadenas.
- La búsqueda del trabajo de un autor se realizó a través de la web, descargando la información del listado de su trabajo desde la página web que contiene dicha información. Para ello, se reconocía el código fuente de dicha página por un analizador de código HTML recuperando el listado de referencias del autor.
- Los cambios realizados en la base de datos durante el mantenimiento de publicaciones era registrado a través de un fichero de log.
- Se utilizó como *framework* de la aplicación la herramienta Struts.
- Uso de JSP, JavaScript y CSS para la generación de páginas web dinámicas.
- El SGBD utilizado será MySQL y se utilizará Hibernate para realizar el mapeo entre clases del sistema y su representación física en base de datos.
- Uso de un contador web gratuito para mostrar las estadísticas de la página web. Este contador web es facilitado por *Motigo Webstats*, a través del cual puede consultarse información como media diaria de acceso, últimas visitas, predicciones de visitas diarias...

- Los analizadores a utilizar fueron reconocedores léxicos y sintácticos implementados a través de las herramientas JFlex y CUP.

Todas estas decisiones pueden verse con más detalle en el anexo C de implementación, junto con las clases y base de datos que finalmente fueron implementados.

## **5.2 Tecnologías utilizadas**

Esta sección describe en detalle el por qué de la elección de las tecnologías que fueron utilizadas durante la implementación del sistema. Estas tecnologías son mostradas en orden de mayor a menor relevancia en la implementación de la aplicación.

### **5.2.1 Java**

Java es el lenguaje de programación que se ha utilizado para implementar Publindex. Es un lenguaje orientado a objetos, por lo que posee las principales ventajas y principios del paradigma orientado a objetos: herencia, encapsulación, polimorfismo o abstracción entre otros. Se decidió utilizar Java por las siguientes ventajas:

- Su independencia de la plataforma, de manera que el mismo programa puede correr en diferentes sistemas operativos.
- Es una fuente abierta y gratuita.
- Su potente API de consulta de clases, librerías, métodos...
- Permite implementar aplicaciones modulares y códigos reutilizables.
- Permite generar aplicaciones web dinámicas, que es lo que se pretende implementar precisamente en Publindex.
- Gestiona el ciclo de vida de los objetos a través del recolector de basura con el fin de reutilizar el espacio de memoria eficientemente.

El uso de Java llevó a que otras tecnologías aquí comentadas fueran seleccionadas por su compatibilidad con el lenguaje. Ejemplo de estas tecnologías son Hibernate o Struts. Para más información, consultar las referencias [Java] [Java2].

### **5.2.2 MySQL**

MySQL es el sistema seleccionado como gestor y administrador de la base de datos. Bastante utilizado junto con PHP o Drupal, dada la rapidez que posee para realizar las consultas de datos, es idóneo como SGBD para aplicaciones web.

Algunas de las principales ventajas de MySQL es que posee muchas de las funcionalidades de SQL, al igual que un gran subconjunto de su lenguaje. Por otra parte, MySQL es de código abierto y está disponible en una gran cantidad de plataformas y sistemas [MySQL].

### 5.2.3 Struts y Apache Tomcat

Struts, diseñado por Apache Software Foundation, es una herramienta de desarrollo de aplicaciones web para Java J2EE y de uso libre que funciona en todas plataformas que soporten *Java Enterprise*. Struts fue elegido como *framework* de la aplicación por las siguientes características:

- Struts está basado en el patrón MVC que se decidió utilizar en la fase de diseño para modelar la arquitectura del sistema.
- Intuitivo y centralizado control del flujo de la aplicación en un fichero de configuración, de manera que los cambios en la navegación sólo deben realizarse en él.
- Posibilidad de implementar una aplicación en varios idiomas a través de ficheros de recursos de manera muy sencilla.
- Gestión sencilla y personalizada de errores en formularios (campos vacíos, tipos de datos incorrectos, etc.) a través de ficheros de configuración y unas clases auxiliares.
- Distribución gratuita.

Struts requiere el uso de ciertas clases en la aplicación para el control de flujo y errores de formulario descritos en detalle en el anexo C de implementación. Para más información consultar las referencias [StrutsWeb] [Struts].

El servidor en el cual se ejecutará la aplicación será Apache Tomcat. Este servidor, también desarrollado por Apache Soft. Found., utiliza el protocolo HTTP y fue seleccionado por ser de código abierto, distribución gratuita y multiplataforma [Apache].

### 5.2.4 Hibernate

La motivación que llevó a utilizar Hibernate fue que es una herramienta de mapeo objeto-relacional (ORM) para Java que permite asociar las entidades del modelo de datos físico (BBDD) con la representación de clases de la aplicación. Para lograr esta asociación, se realiza un mapeo de los atributos de las entidades de la base de datos con los objetos a través de ficheros declarativos XML.

La principal ventaja de Hibernate es que las consultas y sentencias lanzadas hacia base de datos se orientan hacia el modelo de objetos, que al ser de más alto nivel, permite mucha más abstracción de la representación física y por tanto una lógica mucho más intuitiva.

Otra de sus ventajas es que las consultas usando Hibernate están escritas en HQL, además de permitir también la escritura de consultas en SQL y el uso de la interfaz Criteria, que dispone de operaciones para realizar consultas aplicando ciertas restricciones sin tener que escribir la consulta HQL directamente. Para más información consultar la referencia [Hibernate].

### 5.2.5 JSP

Los primeros prototipos de las interfaces de la aplicación ya hicieron decidir que deberían ser páginas web dinámicas, como por ejemplo la que muestra los resultados de las valoraciones de una publicación.

JSP es la tecnología disponible en Java con la que se implementaron estas páginas web dinámicas. Otras razones por las que fue seleccionada son:

- Es multiplataforma e independiente del servidor.
- Es de distribución libre.
- Permite el uso de etiquetas creadas por el propio usuario.
- Permite la inclusión de código Java en el código de la página.

Todas las páginas web de la aplicación fueron implementadas con JSP. Además, también se utilizaron en ocasiones JavaScript y CSS para mejorar la apariencia de las páginas [JSP].

### 5.2.6 JUnit

JUnit es el conjunto de bibliotecas que fue utilizada para realizar las pruebas unitarias en la aplicación ya que permite realizar una ejecución controlada de clases Java. A través de este control, se permite comprobar si los métodos de estas clases devuelven los resultados esperados ante unas entradas concretas y así determinar su validez [JUnit].

La otra razón para usar JUnit es que el entorno de desarrollo utilizado, Eclipse, incluye una extensión de JUnit para crear plantillas de ficheros de test de las clases que se seleccionen.

### 5.2.7 JFlex y CUP

JFlex, y CUP fueron las tecnologías seleccionadas para implementar los reconocedores de citas y código HTML. Las razones de su uso son las siguientes:

- A través de un conjunto de expresiones léxicas y reglas sintácticas se pueden reconocer una gran combinación de formatos de texto y poder procesar la información reconocida.
- Al seleccionar Java como lenguaje de programación, se precisa de dos herramientas que estén desarrolladas en dicho lenguaje y que también generen código en Java.
- A excepción de pequeñas diferencias en la terminología y palabras reservadas utilizadas, el comportamiento y apariencia de JFlex y CUP es similar a Flex y Yacc respectivamente, de los cuales ya se poseían nociones y experiencia.
- Ampliar estos analizadores sólo requiere modificar los ficheros que contienen las expresiones regulares y la gramática que define la estructura a reconocer.

El uso de JFlex y CUP requirió unas primeras pruebas con objeto de familiarizarse con las tecnologías y estudiar cómo poder incorporarlas a la aplicación. Para más información consultar referencia web [CUP].

### **5.3 Evolución del prototipo**

La aplicación evolucionó a lo largo del proceso de desarrollo conforme se iban completando las iteraciones. Los hitos más importantes del prototipo fueron registrados en las siguientes fechas:

- 7 de Diciembre de 2010: inicio de la fase de construcción.
- 17 de Enero de 2010: carga completa de JCR.
- 2 de Marzo de 2010: primera versión estable del buscador de una publicación a través de su nombre/siglas o ISSN con los datos de JCR cargados. Esta versión beta se pone a disposición del usuario.
- 1 de Mayo de 2010: carga completa de CORE y nueva versión del buscador de una aplicación.
- 30 de Mayo de 2010: versión estable del subsistema de mantenimiento de publicaciones.
- 6 de Junio de 2010: carga completa de SJR.
- 25 de Junio: primera versión estable del buscador de una lista de publicaciones a través de una colección de citas bibliográficas.
- 10 de Julio de 2010: versiones estables de los buscadores de publicaciones por citas y autor.
- 1 de Agosto de 2010: versión definitiva de la aplicación Publindex.

A continuación se describen las evoluciones que sufrió el prototipo a nivel de implementación de la propia aplicación y a nivel de base de datos.

#### **5.3.1 Evolución de la aplicación**

La carga de JCR supuso un hito importante, ya que se disponía de un conjunto de datos estable con el cual comenzar a trabajar en la aplicación. Sin embargo, la primera implementación estuvo muy ligada a que sólo un índice estuviera almacenado en base de datos. Por ejemplo, las publicaciones eran tratadas como revistas, los datos de las valoraciones se consideraban en formato numérico, los top en las categorías se calculaban considerando que toda publicación tenía una posición absoluta en ella...La siguiente imagen muestra una de las primeras versiones de cómo se mostraban los datos en Publindex:

## Resultado de la búsqueda

ACM COMPUTING SURVEYS

ISSN: 0360-0300  
País: UNITED STATES  
Lenguaje: ENGLISH

**Datos procedentes de JCR**

Factor de impacto: 9,92  
Citas: 3273

**Rankings en categorías**

- En categoría [Computer Science, Theory & Metho](#) localizado en la primera posición de 1

Figura 8: Primeras versiones de la búsqueda de resultados (versión 1.6)

Esta perspectiva de la aplicación cambió una vez que el usuario realizó las primeras pruebas. La apariencia de la página de resultados cambió al formato tabla actual y la página de inicio y su apariencia fue refinándose.

Una vez que CORE fue cargado en el sistema, fue necesario realizar una homogeneización de la aplicación para tratar múltiples fuentes de datos. Las primeras versiones tras cargar CORE presentaban una problemática: se habían ignorado sus códigos FoR y las dos únicas categorías existentes eran el CORE de revistas y de conferencias. Esto implicó volver a cargar los datos de CORE considerando sus categorías además de cómo realizar el cálculo de los top en la categoría. Una versión de los resultados tras cargar CORE es la siguiente:

Resultado de la búsqueda									
ACM COMPUTING SURVEYS									
ISSN: 0360-0300									
País: UNITED STATES									
Lenguaje: ENGLISH									
JCR	2002	2003	2004	2005	2006	2007	2008	2009	2010
Factor impacto	2,769	7,15	10,037	7,4	4,13	5,25	9,92	-	-
Categorías									
Computer Science, Theory & Method	1/69(top 1%)	-	1/70(top 1%)	1/71(top 1%)	1/75(top 1%)	1/79(top 1%)	1/85(top 1%)	-	-
Unknown Category	-	4/459(top 0%)	-	-	-	-	-	-	-
CORE	2002	2003	2004	2005	2006	2007	2008	2009	2010
Factor impacto	-	-	-	-	-	-	A*	A*	A*
Categorías	-	-	-	-	-	-	-	-	-

Figura 9: Primeras versiones de la aplicación con CORE incluido (versión 2.4)

La manera de presentar los resultados de las búsquedas también fue mejorada e inicialmente presentaba el siguiente aspecto:



*Figura 10: Página del listado de coincidencias (versión 2.4)*

Las mejoras realizadas consistieron en agrupar las coincidencias por revistas y conferencias, mostrar en cada coincidencia las últimas valoraciones asignadas por cada índice e incorporar el mantenimiento de publicaciones. El resultado de estos cambios es la versión final de la que dispone la aplicación y que puede ser consultada en el anexo E del manual de usuario.

La carga de SJR y la inclusión de las macrocategorías para delimitar las búsquedas completaron el buscador de una publicación, cuyo resultado también puede verse en el anexo E de manual de usuario.

El siguiente paso fue la implementación de los analizadores del listado de publicaciones, a través de un conjunto de referencias bibliográficas y de autor. El incorporar estas funcionalidades hizo que la página de inicio de la aplicación estuviera compuesta de un menú que redirigiera a un analizador u a otro. El resultado de añadir estas funcionalidades es la aplicación final, cuya versión es la 3.7.

### 5.3.2 Evolución de la base de datos

A medida que los índices de calidad eran cargados en la base de datos, el tamaño de ésta comenzó a crecer considerablemente. Veamos en la siguiente tabla el crecimiento del número de registros por tabla en base de datos al añadir cada una de las fuentes de datos:

	Categoría	Macrocategoría	Publicación	Revista	Conferencia	Valoración
Carga JCR	71	5	578	578	0	4012
Carga CORE	307	29	24147	21463	2684	38147
Carga SJR	402	48	24438	21754	2684	45839

*Tabla 1: Número de registros en base de datos al cargar los índices*

Notar que la carga de JCR y SJR no supone un incremento tan grande en proporción a CORE. Esto es debido a que de JCR y SJR sólo se carga la información de las categorías

pertenecientes al área de informática, tal y cómo se comentó en la sección de análisis. Como puede comprobarse, la carga de CORE supuso un crecimiento bastante importante en el tamaño de los datos. Este crecimiento hizo que aparecieran problemas que hasta ese momento no se habían encontrado por no disponer de un juego de datos suficientemente grande:

- La búsqueda de coincidencias por nombre resultaba bastante lenta cuando existía un número de coincidencias considerable. El fallo residía en las consultas, en las cuales se utilizaban *'select \*'* que provocaba mucha penalización de tiempo al devolver los datos. Su resolución hizo que el tiempo de respuesta se acortara considerablemente, aunque el presentar un número grande de coincidencias por pantalla todavía consume un tiempo elevado (ver futuras ampliaciones del apartado 6).
- La aplicación daba errores de memoria al llevar un tiempo en ejecución. Esto sucedía al almacenar en sesión el listado de objetos publicación resultado de la búsqueda para transferirlos a la página JSP que los mostraba. Este problema se solucionó utilizando el objeto *request* para transferir los datos desde negocio a la página que muestra las coincidencias.
- La carga de categorías CORE que no pertenecían al área de informática hizo que las búsquedas devolvieran muchos resultados que no eran interesantes para los usuarios que sólo estuvieran buscando información de un área en concreto. Esta problemática hizo que se decidiera dar al usuario la opción de poder seleccionar las macrocategorías donde buscar los resultados al igual que puede ser seleccionado el criterio de búsqueda en la página de inicio del analizador de una publicación.
- El usuario necesitaba saber si cuando no aparecían datos en un año en concreto, era porque el índice no estaba cargado o no porque no existía información en ese año para esa categoría o publicación concreta. Esto se solucionó guardando qué índices estaban almacenados en cada año. En la página de resultados y de coincidencias, se presenta una pequeña leyenda para que el usuario pueda consultar esta información.

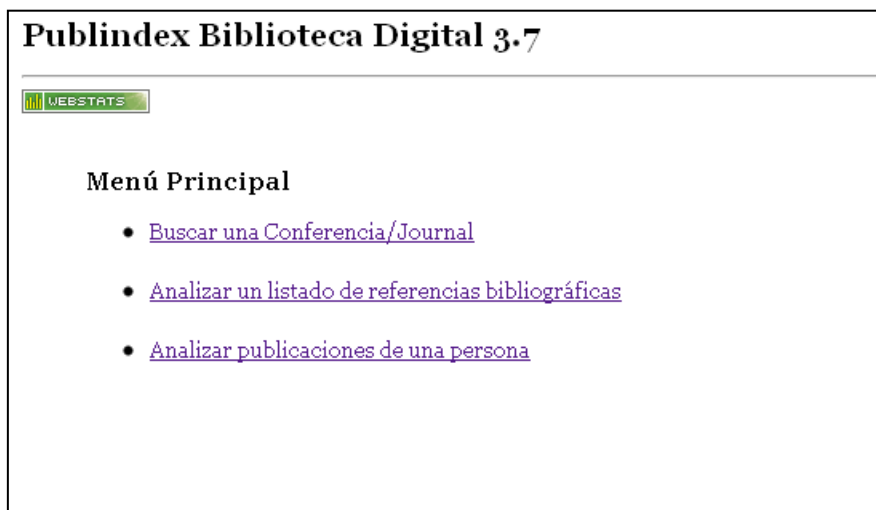
La carga de SJR no provocó ningún problema a nivel de base de datos, al estar ya preparada la aplicación para almacenar y manejar varias fuentes de calidad y un número de datos considerable de una manera eficiente.

### 5.3.3 Diseño final

A continuación se realiza una muestra de los pantallazos más relevantes de la aplicación y que permiten mostrar las principales funcionalidades: la página de inicio con el menú principal y la página que muestra los resultados de valoraciones de una publicación. El anexo E de manual de usuario contiene una guía muy precisa de la aplicación y que muestra en detalle todas las funcionalidades y la apariencia de Publindex.



La página de inicio contiene el menú principal con las tres opciones de análisis posibles: búsqueda de una publicación y sus valoraciones, analizar un listado de publicaciones por citas bibliográficas y por autor. Cada una de estas opciones es un enlace que lleva a la página de inicio de cada analizador. La dirección URL donde ha sido desplegado Publindex y a través de la que se accede a la página de inicio es la siguiente: <http://nacho.cps.unizar.es:8080/Publindex/jsps/index.jsp>.



*Figura 11: Página de inicio de la aplicación Publindex*

En esta página se muestran los datos de una publicación tales como el nombre, lenguaje o país y los datos de las valoraciones asignadas en cada año por un índice concreto. Estas valoraciones se muestran en formato tabla mostrando los datos de cada índice en las filas y los años en los que existen valoraciones en las columnas.

Para cada índice, las valoraciones mostradas son la asignada por el propio índice y además el top % que ocupa la publicación en una categoría en un año concreto. Este top se presenta en modo de enlace que redirige a mostrar los datos de la categoría en el año que hace referencia.

Journal IEEE PERVASIVE COMPUTING									
ISSN: 1536-1268									
País: UNITED STATES									
Lenguaje: ENGLISH									
Resultados desde año 2002 a año 2010 <input type="button" value="Buscar"/>									
JCR	2002	2003	2004	2005	2006	2007	2008	2009	2010
Valoracion	?	?	?	FI 2,036	FI 2,062	FI 2,293	FI 2,615	?	?
Categorias									
Computer Science, Information Systems	?	?	?	14/83 (top 17%)	14/87 (top 16%)	9/92 (top 10%)	12/99 (top 12%)	?	?
Telecommunications	?	?	?	5/59 (top 8%)	4/59 (top 7%)	2/66 (top 3%)	7/67 (top 10%)	?	?
CORE	2002	2003	2004	2005	2006	2007	2008	2009	2010
Valoracion	?	?	?	?	?	?	B	?	A
Categorias									
Distributed Computing	?	?	?	?	?	?	38/70 (top 54%)	?	20/83 (top 24%)
SJR	2002	2003	2004	2005	2006	2007	2008	2009	2010
Valoracion	?	?	0,091	0,087	0,106	0,115	0,15	?	?
Categorias									
Computer Networks and Communications	?	?	15/116 (top 13%)	16/116 (top 14%)	11/116 (top 9%)	12/116 (top 10%)	5/116 (top 4%)	?	?

Figura 12: Página que muestra las valoraciones y datos de una revista

Los top asignados para las categorías en JCR y SJR es calculado en función de la posición que ocupa la publicación en la categoría con respecto al número total de publicaciones. Por ejemplo, un top 5/58 (top 8%) significa que la publicación ocupa la quinta de 58 posiciones en la categoría o lo que es lo mismo, se encuentra entre el primer 8% de publicaciones de la categoría.

Sin embargo, calcular el top de CORE no es tan sencillo, ya que las publicaciones con una misma valoración no tienen un orden. El top se calcula considerando el número de publicaciones con valoración mayor o igual que la de la publicación dividido entre el número de publicaciones. Por ejemplo, si tenemos una categoría con 4 publicaciones de valoración 'A', 10 publicaciones de valoración 'B' y 5 de valoración 'C', el top de una publicación de valoración B es igual a  $14 / 19$  ( $4 A + 10 B$ ) /  $19$  ( $4 A + 10 B + 5 C$ ).

### 5.3.4 Evaluación de la implementación del extractor de publicaciones a través de citas bibliográficas

Como ya se ha comentado a lo largo del documento, la aplicación utiliza una serie de herramientas para la obtención de resultados que no es del todo precisa. Estas herramientas corresponden con los reconocedores sintácticos y léxicos de referencias bibliográficas.

Se realizó un estudio para evaluar la calidad de la implementación de estos analizadores léxicos y sintácticos de citas bibliográficas y código HTML. El estudio consistió en probar la precisión de las búsquedas del trabajo de los investigadores pertenecientes al SID Carlos Bobed, Sergio Ilarri, Raquel Trillo y Jorge Gracia. Algunos valores importantes de estas búsquedas son:

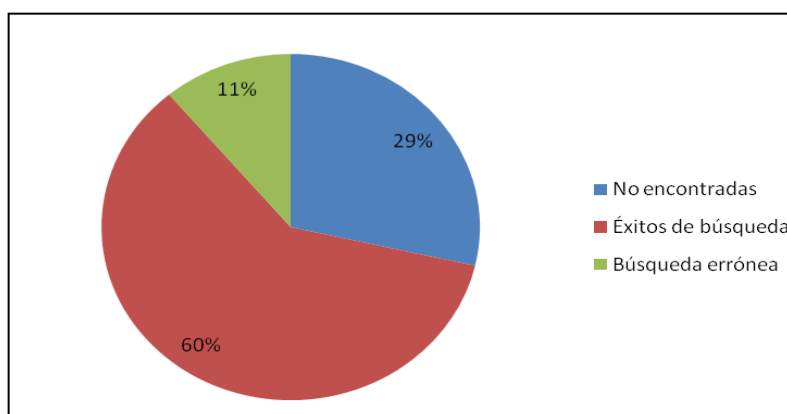
Autor	Publicaciones	Conf. /Revistas	Analizadas	No encontradas	Éxitos de búsqueda	Búsqueda errónea
Carlos Bobed	8	7	7	2	5	0
Sergio Ilarri	54	46	34	12	19	3
Raquel Trillo	18	17	11	2	7	2
Jorge Gracia	15	14	11	2	7	2
<b>Total</b>	<b>95</b>	<b>84</b>	<b>63</b>	<b>18</b>	<b>38</b>	<b>7</b>

*Tabla 2: Mediciones relativas a las pruebas sobre el buscador de publicaciones por autor*

De estas mediciones se llegó a los siguientes parámetros que sirven como índice de la calidad de los analizadores:

- Número de publicaciones encontradas: 45 (38 éxitos + 7 erróneas)
- Número de citas referentes a revistas y conferencias: 84
- Tasa de reconocimiento de citas: 75% (63 analizadas/ 84 publicaciones válidas)
- Tasa de acierto de búsqueda de publicaciones: 84% (38 búsquedas válidas/ 45 pubs, encontradas)

De este estudio se identificaron tres tipos de búsqueda: no encontrada (cuando la publicación no se encuentra almacenada en el sistema), éxito (publicación encontrada y que corresponde con la referenciada en la cita) y errónea (publicación encontrada pero que no se corresponde con la indicada en la cita). A continuación se muestra un gráfico circular que muestra qué porcentaje del total de publicaciones analizadas corresponde a cada tipo de búsqueda:



*Figura 13: Porcentaje de las publicaciones analizadas que pertenecen a cada tipo de búsqueda*

El análisis de los resultados de llevó a la conclusión de que los analizadores tienen un porcentaje de acierto elevado y que por tanto pueden ser considerados como herramientas aptas para implementar la funcionalidad definida en la aplicación.

## **5.4 Pruebas realizadas**

Una vez que una funcionalidad era implementada era necesario realizar las pruebas oportunas que demostraran su validez y la de la aplicación. Estas pruebas se dividieron en tres grupos en función del nivel al que sean realizadas: pruebas unitarias, de sistema y de usuario.

Las pruebas unitarias tenían el propósito de validar las clases del sistema realizando para ello pruebas de sus métodos. A través de JUnit, se ejecutaron una serie de ficheros de test que prueban el correcto funcionamiento de los métodos analizando dadas unas entradas concretas, los resultados obtenidos.

Una vez finalizadas las pruebas unitarias, debió comprobarse que las clases funcionaran correctamente en su conjunto, realizando para ello pruebas en la aplicación validando que ésta ofrecieran los resultados y funcionalidad definidos en los requisitos.

También es necesario que el usuario realice una serie de pruebas para poder detectar errores que hasta ahora el desarrollador hubiera pasado por alto o para informar de comportamientos no deseados en la aplicación. A partir de las primeras versiones válidas de la aplicación el usuario ya comenzó a probarla reportando errores o mejoras en el sistema. Para ver en detalle las pruebas realizadas, consultar el anexo D de pruebas [Test].

## 6 Conclusiones

---

Publindex es el resultado del desarrollo de una aplicación web que permite al investigador mejorar los mecanismos de búsqueda que actualmente utiliza para consultar y valorar la calidad de revistas y conferencias de investigación. Para ello, se recopiló la información de una serie de fuentes de calidad y se implementaron varios tipos de análisis de publicaciones. El sistema se caracteriza principalmente por lo siguiente:

- La aplicación permite el análisis individual de una publicación, de un listado de publicaciones por citas bibliográficas y de un autor.
- La aplicación reúne la información de los índices de calidad JCR, SJR y CORE, disponiendo de más de 24000 publicaciones (revistas y conferencias), 400 categorías y 45000 valoraciones de calidad.
- Maneja la información de una publicación de una manera homogénea sin depender así de qué tipo de índice de calidad proceda.
- Posee un novedoso sistema de reconocimiento online de la información de un autor.
- Su usabilidad permite al usuario realizar consultas y búsquedas de una manera simple y centralizada, evitándole así el consultar múltiples sitios web e invertir un tiempo considerable.
- El sistema permite futuras actualizaciones de los índices de calidad para que la aplicación siga siendo una fuente de referencia con el paso del tiempo.
- Sus posibles ampliaciones funcionales hacen que el sistema pueda ser mejorado en un futuro.

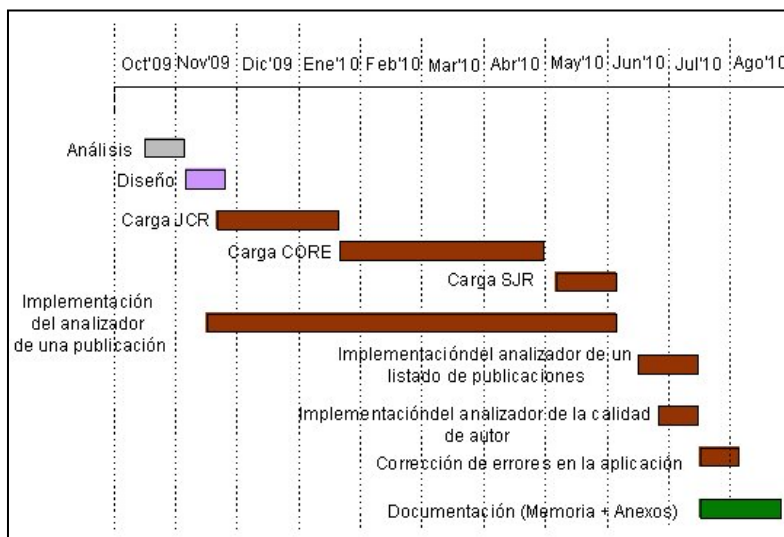
Las siguientes secciones muestran aquellos aspectos a destacar del proyecto una vez finalizado todo el proceso de desarrollo, como son el cronograma que muestra la evolución temporal, seguimiento realizado durante el desarrollo, los principales problemas encontrados, posibles ampliaciones y mejoras del sistema y la opinión personal del proyecto.

### 6.1 Cronograma

Esta sección muestra el cronograma que representa la evolución temporal del proyecto y el tiempo que se invirtió en cada tarea importante del desarrollo. Las tareas importantes que fueron identificadas junto con su leyenda de colores son las siguientes:

- Análisis (Gris): requisitos, identificación de casos de uso y análisis de los índices de calidad disponibles en la Web. Comprende desde 04/10/09 al 03/11/09.
- Diseño (Morado): arquitectura del sistema, BBDD, modelo de clases. Comprende desde el 08/11/09 a 21/11/09.
- Implementación (Marrón): debido a la importancia de esta fase, los periodos se presentan desglosados por tareas:

- Carga de JCR: del 28/11/09 a 17/01/10.
- Carga de CORE: del 19/01/10 a 01/05/10.
- Carga de SJR: del 02/05/10 a 06/06/10.
- Analizador de una publicación: de 15/11/09 al 06/06/10.
- Analizador de listado de publicaciones por citas: de 13/06/10 al 10/07/10.
- Analizador de la calidad de un autor: de 28/06/10 a 10/07/10.
- Corrección de errores en la aplicación: 10/07/10 a 01/08/10.
- Documentación (Verde): memoria y anexos. Desde 12/07/10 a 23/08/10.



*Figura 14: Evolución temporal del proyecto*

Como puede observarse, existen tareas que se solapan. Por ejemplo, a la vez que se comenzó a implementar la aplicación se iba ampliando el catálogo de datos añadiendo información de los índices de calidad.

El periodo de implementación del analizador de una publicación fue el más largo, debido a que era la funcionalidad que más dificultad e importancia tenía y que dependía de la carga de los índices de calidad. Los analizadores de un listado de publicaciones y de autor fueron implementados una vez que el primer analizador estuvo terminado, ya que su implementación dependía directamente de que la búsqueda de la calidad de una sola publicación estuviera terminada.

La implementación finalizó a mediados de Julio, pero hasta principios de Agosto la aplicación se estuvo probando y corrigiendo los errores que se iban detectando. A la vez que se realizaban estas correcciones, se comenzó a elaborar la documentación, ya que estas correcciones sólo implicaron refinar la aplicación y no añadir o modificar ninguna funcionalidad existente.

## 6.2 Seguimiento del proyecto

A lo largo de todo el proceso de desarrollo se realizó un seguimiento del proyecto por parte del director. Este seguimiento se realizó a través de correo electrónico y de reuniones.

El correo electrónico fue una herramienta que se utilizaba para resolver pequeñas dudas que no requerían invertir mucho tiempo, enviar las nuevas versiones de la aplicación y concretar las próximas reuniones. En total se enviaron más de 120 correos electrónicos.

Las reuniones se realizaban cuando debían ser tratados temas importantes como por ejemplo el haber añadido una nueva funcionalidad, alguna duda que impedía continuar con el desarrollo, revisar el trabajo realizado hasta la fecha, realizar controles de tiempo.... Aunque la mayor parte de las reuniones se fijaron para hacerse en persona, también hubo reuniones telefónicas por celebrarse a horas en las que proyectando y director no podían desplazarse hasta un mismo lugar. La siguiente tabla muestra las reuniones que se celebraron durante el proyecto:

Fecha	Motivo	Tipo de reunión	Duración
21/09/2009	Primer contacto con el director del proyecto e introducción al problema.	Telefónica	1h
29/09/2009	Fijar primeras ideas del proyecto de análisis de las fuentes de calidad.	En persona	1h 45min
08/10/2009	Estudiar los mecanismos de tratamiento de la información de las fuentes de calidad	En persona	1h
04/11/2009	Presentación de la primera versión del buscador de una publicación con JCR cargado como índice.	En persona	1h 30min
25/11/2009	Comprobación del primer prototipo corregido, mejoras a realizar y revisión de JCR.	En persona	1h
11/12/2009	Homogeneizar la aplicación y mejoras a realizar.	En persona	1h 15min
22/01/2010	Presentación y revisión de la primera versión de la aplicación con algunas valoraciones CORE cargadas.	En persona	1h 30min
26/02/2010	Revisión de la primera versión estable de la aplicación para ser probada por el usuario.	Telefónica	1h 15min
29/03/2010	Revisión de la aplicación.	En persona	1h 30min
12/04/2010	Revisión de la aplicación y de la propuesta del PFC.	En persona	2h
17/05/2010	Resolución de dudas y revisión de la aplicación.	Telefónica	1h

10/06/2010	Refinar la definición de los buscadores por listado de publicaciones.	En persona	1h 30min
29/06/2010	Revisar la aplicación y las nuevas versiones de los buscadores por citas y autor.	En persona	1h 30min
12/08/2010	Revisar documentación del proyecto.	En persona	2h 30min
23/08/2010	Revisar documentación del proyecto.	En persona	2h

*Tabla 3: Reuniones realizadas con el director del proyecto*

### 6.3 Problemas encontrados

Los principales problemas encontrados a lo largo del proyecto pueden ser clasificados en problemas de desarrollo y problemas de planificación. En cuanto a problemas de desarrollo, las principales dificultades residieron en recopilar la información de los índices de calidad y poderla almacenar.

Guardar la información de JCR y SJR fue relativamente fácil, ya que se disponía de ficheros Excel que podían ser procesados a través de una biblioteca Java que ofrecía métodos de lectura del formato Excel CSV. Los ficheros Excel que recopilaban la información de JCR y SJR facilitaron la manera de importar este índice de calidad (ver subsección 3.1). Guardar la información de CORE supuso mayores problemas, ya que los rankings se presentaban en formato PDF y para poderlos procesar era necesario guardarlos en formato XML y a través de un reconocedor sintáctico obtener la información y almacenarla en base de datos. Sin embargo, existían ficheros que no pudieron ser convertidos a XML y tuvo que ser necesario extraer el texto del PDF a través de una aplicación, completar a mano la información que no se extrajo correctamente y finalmente a través de un analizador léxico y sintáctico almacenar la información. Todo esto requirió invertir mucho tiempo en el desarrollo y provocó retrasos en la implementación de la aplicación.

El cuanto a estimaciones temporales, inicialmente se estimó entregar el proyecto en la convocatoria de Julio de 2010 que, debido a un desfase en la fase de construcción la entrega, se retrasó hasta Septiembre de 2010. Este retraso fue debido a que a la vez que se estuvo realizando el PFC también se estuvo trabajando desde Septiembre de 2009 hasta finales de Junio de 2010, no pudiendo dedicar muchas horas a la semana, aproximadamente unas 14, a la implementación de la aplicación. En la fase de transición, que comprendió las labores de documentación, se estuvo trabajando a tiempo completo, dedicando una media de 36 horas semanales.

### 6.4 Posibles Ampliaciones

Algunas posibles ampliaciones funcionales podrían ser:

- Hacer que sea una aplicación multi idioma. Para ello sólo es necesario añadir ficheros de recursos para cada idioma, añadir un nuevo *ActionMapping* en el fichero



de configuraciones de Struts para redirigir el flujo y una nueva clase *Action* que realice el cambio de idioma.

- Ampliar la búsqueda de autores añadiendo nuevas fuentes, como DBLP. Esto precisaría de realizar búsquedas online en la página web de DBLP, modificando el análisis léxico y sintáctico (ficheros .lex y .cup) de código HTML para poder reconocerla y procesar las referencias bibliográficas. Además, sería necesario comprobar el formato de estas citas y en el caso de que no fuera reconocido con la implementación actual se deberían modificar el analizador de citas (ficheros .lex y .cup).
- Añadir nuevas fuentes de calidad como por ejemplo CS Conference Ranking para tener más valoraciones de calidad de conferencias. En el anexo E del manual de usuario se describe cómo realizar este proceso.

En cuanto a mejoras técnicas a realizar podrían ser:

- Hacer que el tiempo de respuesta de las búsquedas se reduzca. Este exceso de tiempo se concentra principalmente cuando se muestra por pantalla un número elevado de coincidencias (por ejemplo 300 revistas). Una solución sería la de paginar los resultados de la búsqueda, mostrando sólo una pequeña cantidad cada vez.
- Mejorar la precisión del analizador de citas, pudiendo analizarse formatos hasta ahora no reconocidos (por ejemplo, citas en formato BibTex con '{,}' en sus campos de datos, otros formatos de tipo texto...). Para ello sólo es necesario modificar el fichero que contiene el análisis sintáctico de las citas.
- Uso de procedimientos almacenados que contengan las consultas de base de datos, así se evita escribir las consultas en el código Java y añadir más abstracción.
- Uso de AJAX para mejorar la interacción con el usuario, permitiendo la comunicación asíncrona con el servidor.
- Almacenar en base de datos los autores y sus citas bibliográficas, para realizar una búsqueda evitando el analizador de código HTML.
- Añadir una apariencia gráfica más agradable utilizando por ejemplo algún color y nuevos estilos CSS.

## 6.5 Opinión personal

Desde un punto de vista técnico, el proyecto me ha hecho adquirir nuevos conocimientos en tecnologías hasta ahora no dominadas, como son Java e Hibernate. El poseer nociones en estas dos tecnologías es muy importante en el mundo laboral actual, ampliando mi currículum y compensando así la falta de experiencia de los primeros años laborales de un recién titulado.

Aunque en ocasiones el proceso se hizo bastante tedioso (sobre todo la carga de los índices de calidad que requería editar ficheros Excel o XML a mano) y nivel de exigencia de mi

director ha sido grande estoy muy orgullosa del resultado. Gracias a dicho nivel exigencia se ha conseguido llegar a unos resultados de esta calidad.

Desde lo personal, el proyecto me ha hecho madurar y adquirir organización y responsabilidad. Sin lugar a dudas, el principal problema encontrado fue el tiempo que se pudo emplear a la semana en el PFC. Dada mi situación de estar trabajando y haciendo el proyecto a la vez, tuve que tener una enorme fuerza de voluntad para llegar a casa después de trabajar y ponerme con el proyecto o sacrificar más de un fin de semana trabajando en vez de descansar después de la agotadora semana. Estas circunstancias hacen que tengas que aprender a organizarte, porque el riesgo de trabajar y terminar la carrera es ir dejando poco a poco abandonado el PFC.

En definitiva, aunque ha supuesto un esfuerzo grande y se ha alargado más de lo esperado, considero que este proyecto me ha aportado muchas más cosas positivas que negativas, mejorando mi experiencia en tecnologías de las que hasta ahora sabía lo básico y haciéndome crecer como persona. Espero que esta aplicación sea verdaderamente útil y que se utilice durante mucho tiempo.

## 7 Bibliografía

---

### 7.1 Referencias Web

[JCR]: [http://en.wikipedia.org/wiki/Revista\\_Citation\\_Reports](http://en.wikipedia.org/wiki/Revista_Citation_Reports)

[CORE]: <http://www.core.edu.au/>

[SJR]: <http://www.scimagojr.com/index.php>

[CSCR]: <http://www.cs-conference-ranking.org/conferencerankings/alltopics.html>

[StrutsWeb]: <http://struts.apache.org/>

[HibWeb]: <http://www.hibernate.org/>

[WikiSQL]: <http://es.wikipedia.org/wiki/MySQL>

[SID]: <http://sid.cps.unizar.es/>

*Todas las referencias web han sido probadas a fecha de 25/08/2010.*

### 7.2 Referencias bibliográficas

[Java]: David Flanagan (2005). *Java In A Nutshell*. O'Reilly.

[Java2]: Patrick Niemeyer, Jonathan Knudsen (2005). *Learning Java*. O'Reilly.

[Apache]: Ben Laurie, Peter Laurie (2002). *Apache: The Definitive Guide*. O'Reilly.

[MySQL]: Paul DuBois (2008). *MySQL*. Addison-Wesley.

[PU]: I. Jacobson, G. Booch, J. Rumbaugh (2000). *El proceso Unificado de desarrollo de Software*. Pearson Education.

[UMLPatterns]: Larman C. (2001). *Applying UML and patterns: An introduction to O-O design and iterative development*. Prentice Hall.

[Struts]: James Holmes (2004). *Struts: The Complete Reference*. McGraw-Hill y Osborne Media.

[JSP]: Jayson Falkner, Ben Galbraith, Romin Irani (2002). *Desarrollo web con JSP*. Anaya Multimedia.

[Hibernate]: Christian Bauer and Gavin King (2004). *Hibernate in action*. Manning Publications.

[Test]: Elfriede Dustin (2002). *Effective Software Testing*. Addison Wesley.

[JUnit]: Petar Tahchiev, Felipe Leme, Vincent Massol, Gary Gregory (2010). *JUnit in Action*. Manning.

[CUP]: Sergio Gálvez Rojas, Miguel Ángel Mora Mata (2005). *Java a Tope: Traductores Y Compiladores Con Lex/yacc, Jflex/cup Y Javacc*. Universidad de Málaga.

## Anexos

---

Este apartado contiene la documentación anexa en la que se apoya la memoria del proyecto. Los anexos que se presentan a continuación son los siguientes:

- Anexo A de Captura de requisitos. Describe de manera detallada todos los requisitos funcionales del sistema que fueron definidos durante la fase de ideación. Además, se incluyen aquellos requisitos no funcionales que también debe cumplir la aplicación.
- Anexo B de Caso de estudio del proyecto. Incluye todas decisiones importantes de análisis y diseño.
- Anexo C de Implementación. Detalla las decisiones más importantes a este nivel. Dentro de este anexo se incluyen cuestiones tan importantes como por ejemplo cómo se cargaron los índices de calidad en base de datos, cómo realizar las búsquedas de la calidad por referencias bibliográficas o estadísticas importantes de la implementación
- Anexo D de Fase de pruebas. Detalla el proceso que fue llevado a cabo para validar la aplicación. Varios apartados enumerarán todas las pruebas a distinto nivel que fueron realizadas.
- Anexo E de Manual de usuario. Contiene el manual que servirá de referente al usuario para poder acceder a todas las opciones de la aplicación.

Cada anexo amplía unas secciones concretas de la memoria. La sección de información general del proyecto y de análisis (secciones 2 y 3) son ampliadas con los anexos A y B; la sección de diseño (sección 3) es ampliada con el anexo de caso de estudio; la sección del prototipo (sección 5) es ampliada a través de los anexos de implementación, pruebas y manual de usuario.

