



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMATICAS**



*“Sistema para la integración de datos
bibliográficos almacenados en diversos
formatos.”*

Autor: Anel Pérez Acosta

Tutor: Vladimir Milián Núñez

La Habana

Junio de 2013

“La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica.”

Aristóteles.

Declaración de Autoría

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo a la Facultad 2 de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2013.

AUTOR

Anel Pérez Acosta

TUTOR

Ing. Vladimir Milián Núñez

Dedicatoria

Le dedico esta tesis a una mujer que amo con la vida, que ha sido todo para mí y por quien me levanto cada mañana. A esa mujer que creyó en mí siempre, que ha sido mi tutora y maestra de la vida, mi amiga incondicional y mi todo. A esa mujer que ha vivido este sueño cada segundo junto conmigo, y a quien admiro y respeto. Con todo el amor del mundo esto es para ti mamá.

A mi papá, el hombre más extraordinario que conocí en este mundo. Aunque la vida me lo arrancó de los brazos y no puede estar aquí conmigo, sé que donde quiera que esté está muy orgulloso de mí. A él, que me mostró el camino a seguir y no vaciló ni un segundo en reprenderme en los momentos en que flaqueo. A él le debo todo y cuanto soy hoy.

A mi hermano, mi ejemplo a seguir en el mundo de la informática, "cuando sea grande quiero ser como tú". Te adoro.

A toda mi familia, en especial a mi familia materna por tanto amor.

Agradecimientos

A la Revolución y a la UCI por darme la oportunidad de materializar mi sueño y formarme profesionalmente.

A Purisel por brindarme su amor, su amistad y su comprensión, por estar ahí en todos los momentos difíciles de la vida, sin ti hubiese sido muy difícil superarlos todos. Gracias por darme tu apoyo incondicional, por darme fuerzas para llegar al final de cada batalla, por reír conmigo cada victoria y llorar cada derrota, por haberme convertido en la mujer que soy hoy.

A mi tutor Vladimir por confiar en mí y darme esta tesis, gracias por tu apoyo y ayuda en todo momento.

A mi otra gran familia. A Dania, "La Margó", que ha sido como una segunda madre para mí, gracias por cada consejo, cada regaño, y por estar ahí siempre. A todos los hermanos de la vida, Danay, "la cuqui", que es la hermana hembra que la naturaleza no me dio, gracias por estar en las buenas, malas y regulares, gracias por todo. A Pablo, "el chuly chuly", que es otro hermano, amigo y confidente que quiero muchísimo. A Iris mi hermana mayor, aunque estés lejos sabes que te quiero, gracias por tu apoyo.

A mi mejor amiga Claudia "La Klau", que aunque es insoportable la quiero mucho, siempre ha estado ahí para mí. Entre nosotras nada es tuyo, ni mío, todo es nuestro (sobre todo los problemas), gracias por todo.

A todos mis amigos de la universidad. En especial a Wendy que me ayudó mucho en la recta final, que aun cuando ella también estaba en el proceso de llegar a la meta tuvo tiempo para mí, gracias. A Sachie, Jora, Rogelio, Josué, Pedro, Damaysis, Lili, Alexei, Fander, Selles, a todos gracias, fue un placer compartir estos años con ustedes.

A todas las personas que de una forma u otra contribuyeron a mi formación profesional y a que este sueño se hiciera realidad.

A todos GRACIAS.

Resumen

El proceso de integración y extracción de información de datos semi-estructurados es uno de los retos a los que se enfrentan los sistemas que utilizan este tipo de datos como fuente de información. Las bases de datos bibliográficas, son un ejemplo de fuente de dato semi-estructurado, que son muy utilizadas por investigaciones cuyo objetivo es la vigilancia y medición de la actividad científica. Estas bases de datos ponen a disposición del público, para su descarga y uso, toda su información en forma de archivos con diferentes formatos, siendo los más comunes RIS, TXT, BIB y BibTeXXML.

Cuando se analizan los datos provenientes de diferentes bases de datos bibliográficas, nos tropezamos con la dificultad de que no todas las referencias bibliográficas contienen la misma cantidad de atributos, presentan variaciones en la organización de dichos atributos en función del formato en que se descarguen las referencias bibliográficas, no todas siguen un estándar común, etc.

El presente trabajo aborda el desarrollo de la herramienta INDABIB, la cual permitirá la integración de datos provenientes de bases de datos bibliográficas con extensión RIS, TXT, BIB o BibTeXXML, en un archivo único en formato XML para su posterior uso. El sistema fue desarrollado utilizando tecnología .NET y usa el estándar BibTEX como referencia para la creación del archivo de salida.

Palabras clave: base de datos bibliográfica, integración de datos, XML, C#.

Índice

Introducción	1
CAPÍTULO 1. Fundamentación Teórica	5
1.1. Introducción	5
1.2. Conceptos	5
1.2.1. Referencia Bibliográfica	5
1.2.2. Gestor Referencia Bibliográfica	5
1.2.1. Base de Datos Bibliográfica	5
1.2.2. Extensión BIB	6
1.2.3. Extensión RIS	7
1.2.4. Extensión BibTeXML	8
1.2.5. Extensión TXT	9
1.2.6. BibTeX	11
1.3. Tendencias Actuales de los Sistemas de Integración de Datos Bibliográficos. ...	13
1.4. Metodología, Lenguajes y Herramientas de Desarrollo	13
1.4.1. Metodología de desarrollo	14
1.4.2. Lenguajes de Modelado UML	17
1.4.3. Herramienta Case: Visual Paradigm for UML Enterprise Edition v8.0	17
1.4.4. Lenguaje de Programación C#	17
1.4.5. Entorno Integrado de Desarrollo (IDE) Visual Studio 2010	19
1.4.6. FRAMEWORK .NET v 4.0	19
CAPÍTULO 2. Características del Sistema	21
2.1. Introducción	21
2.2. Objeto de Automatización	21
2.3. Propuesta del Sistema	21
2.4. Modelo de Dominio	21
2.4.1. Descripción del Modelo conceptual	22
2.5. Requerimientos Funcionales del sistema	23
2.6. Requerimientos no Funcionales del Sistema.	24
2.7. Modelo de Casos de Uso del Sistema.	24
2.8. Especificación de los Casos de Uso del Sistema	25

Introducción

2.8.1. Descripción CU Cargar Base de Datos Bibliográfica.....	25
Conclusiones.....	29
CAPÍTULO 3. Modelado del Sistema.....	30
3.1. Introducción.....	30
3.2. Arquitectura del Sistema.....	30
3.2.1. Capas del Sistema	30
3.2.2. Patrones y Estilo Arquitectónico	31
3.3. Diseño del Sistema.....	33
3.3.1. Modelo de Diseño.....	33
3.3.2. Patrones de Diseño	35
Conclusiones.....	37
CAPÍTULO 4. Implementación y Prueba.....	38
4.1. Introducción.....	38
4.2. Modelo de Implementación.....	38
4.2.1. Diagrama de Componentes.....	38
4.2.2. Diagrama de Despliegue	39
4.3. Modelo de Prueba	39
4.3.1. Técnicas de Evaluación Dinámica	40
4.3.2. Entorno de las Pruebas	43
4.3.3. Resultados de las Pruebas.....	43
Conclusiones.....	44
Conclusiones Generales.....	45
Recomendaciones.....	46
Referencias Bibliográficas.....	47
Anexos	48

Índice de Ilustraciones

Figura 1: Ejemplo del formato en que exporta la extensión BIB.....	6
Figura 2: Ejemplo del formato en que exporta la extensión RIS.....	7
Figura 3: Ejemplo del formato en que exporta la extensión BibTeXML	9
Figura 4: Ejemplo del formato en que exporta la extensión TXT	10
Figura 5: Representación de la estrella de BOEHM y Turner para el desarrollo del sistema INDABIB.	14
Figura 6: Proceso Unificado Ágil.....	15
Figura 7: Modelo de Dominio.....	22
Figura 8: Diagrama de casos de uso	25
Figura 9: Modelo Vista Presentador (expandido).....	32
Figura 10: Diagrama de Paquetes del sistema INDABIB	33
Figura 11: Diagrama de Clases del Diseño del sistema INDABIB.....	34
Figura 12: Ejemplo de aplicación del patrón Experto	35
Figura 13: Ejemplo de aplicación del patrón Controlador.....	36
Figura 14: Ejemplo de aplicación del patrón Creador	36
Figura 15: Ejemplo de aplicación del patrón Singleton.	37
Figura 16: Diagrama de Componentes del sistema INDABIB	38
Figura 17: Diagrama de despliegue.....	39
Figura 18: Fragmento de código de las pruebas unitarias en la capa Presentador.....	41
Figura 19: Resultados de la pruebas unitarias del sistema INDABIB.....	41
Figura 20: Resultado de la segunda iteración de las pruebas unitarias en el sistema INDABIB.....	42
Figura 21: Resultado de las pruebas del sistema INDABIB	44

Índice de Tablas

Índice de Tablas

Tabla 1: Tipo de registro de la extensión RIS.....	8
Tabla 2: Tipos de campos de la extensión RIS	8
Tabla 3: Tipos de Registros de la extensión TXT	10
Tabla 4: Tipos de campos de la extensión TXT	11



Introducción

Los cambios acelerados y el desarrollo en el ámbito de las tecnologías de la información y las comunicaciones (TICs) en la actualidad han introducido grandes reestructuraciones en los espacios de trabajo del hombre, donde la producción masiva de documentación, la necesidad de acceder y gestionar la misma son planes fundamentales en el desarrollo de la ciencia y la tecnología.

Cuba a pesar de ser un país subdesarrollado y bloqueado no está ajena al progresivo y constante cambio de la era actual, como consecuencia del desarrollo vertiginoso de las TICs. Por ello el gobierno cubano ha decidido incorporar avances tecnológicos para manejar toda la información que se genera en cualquier sector.

El Instituto de Cibernética, Matemática y Física (ICIMAF) es un centro de investigación cubano. Cuenta con una consolidada presencia nacional e internacional y es reconocido por el prestigioso colectivo científico que posee, ya que ha obtenido buenos resultados en las investigaciones, así como en los servicios científicos y tecnológicos de alto valor agregado. Su misión es gestionar y ejecutar proyectos de investigación, desarrollo e innovación (I+D+i) con personal motivado y de competencia reconocida. Además de realizar actividades de formación posgraduada, asesorías y brindar servicios científicos y tecnológicos que aportan soluciones de alto valor agregado. (1)

Para el desarrollo de proyectos, procesos y sistemas en el marco I+D+i la vigilancia tecnológica es una herramienta fundamental, puesto que a través de ella se generan ideas utilizables que concluirán en el desarrollo de un nuevo producto, servicio o proceso para la organización. La vigilancia tecnológica permite entre otros aspectos:

- ❖ Alertar sobre las innovaciones científicas o técnicas susceptibles de crear oportunidades o amenazas.
- ❖ Reducir las decisiones erróneas en el inicio de un proyecto de investigación y desarrollo, así como en las fases de lanzamiento previo al mercado.
- ❖ Investigar los hallazgos realizados para el desarrollo de nuevos productos, servicios y procesos.
- ❖ Buscar soluciones tecnológicas para que sean abordadas en los nuevos proyectos.

Introducción

- ❖ Realizar de manera sistemática la observación y búsqueda de señales de cambio y novedades enfocadas a la captura de información, la selección y análisis, la difusión y comunicación para convertirla en conocimiento, que permita la toma de decisiones, y el seguimiento de la explotación de los resultados.
- ❖ Identificar una serie de ideas para satisfacer nuevas necesidades de mercado o mejora de productos y/o procesos ya existentes, al aplicar nuevas tecnologías; crear nuevos productos.
- ❖ Evaluar el posible impacto de un hecho o cambio en el entorno.
- ❖ Facilitar la observación del entorno, necesario para establecer y apoyar el desarrollo de proyectos y sistemas de gestión de la I+D+i, contribuyendo a definir la posición estratégica que debe tomar la organización así como sus objetivos en esa materia.

Actualmente el ICIMAF se encuentra inmerso en el proyecto de gran envergadura: “Vigilancia Tecnológica para la Investigación y la Innovación Científica”, el cual es desarrollado para el Centro de Gestión de la Información y Desarrollo de la Energía (CUBAENERGÍA). Con el objetivo de garantizar y facilitar el desarrollo de este proyecto se decidió crear cuatro grupos de trabajo:

1. Grupo de trabajo de tratamiento de datos estructurados (numéricos).
2. Grupo de trabajo de tratamientos de datos no estructurados o semi-estructurados (patentes, referencias de artículos, documentos a textos completos).
3. Grupo de trabajo de datos semi-estructurados (procedentes de internet e intranet).
4. Grupo de trabajo de desarrollo de biblioteca digital y repositorios de la organización.

El segundo grupo de trabajo de los antes mencionados, tiene como principal misión crear un “Sistema para el Tratamiento de Datos No Estructurados y Semi-Estructurados con Técnicas de Minería de Datos” (Sistrada NOYSE), para ello es necesario automatizar las partes que conforman el ciclo de vida de la vigilancia tecnológica y que se relacionan a continuación: búsqueda y organización, procesamiento, análisis, presentación y difusión de información. Sistrada NOYSE utiliza la información contenida en bases de datos bibliográficas con soporte digital (CD-ROM, catálogo en línea, generados por gestores bibliográficos, etc.), las cuales almacenan los datos en diferentes formatos sin seguir un estándar entre ellas y donde la cantidad de atributos varía en dependencia del formato y estándar utilizado.

Introducción

Las bases de datos bibliográficas que son de interés para Sistrada NOYSE se clasifican como bases de datos documentales, las mismas ponen a disposición de las personas que las utilizan toda la información contenida en ellas para su descarga en archivos con extensión RIS, TXT, BibTeX o BIB.

Planteada la **situación problemática** se puede enunciar el **problema a resolver** a partir de la siguiente interrogante: ¿Cómo garantizar que el sistema Sistrada NOYSE sea capaz de reconocer las bases de datos bibliográficas con extensión RIS, TXT, BibTeX o BIB?

El **objeto de estudio** determinado para la presente investigación se basa en el proceso de gestión de los datos bibliográficos y el **campo de acción** se enmarca en la integración y estandarización de datos bibliográficos.

La **idea a defender** se plantea de la siguiente forma: “Con la implementación del Sistema de Integración de Datos Bibliográficos (INDABIB) se garantizará la integración y la conversión de las bases de datos bibliográficas con extensión RIS, TXT, BibTeX o BIB a un formato estándar.”

La presente investigación tiene como **objetivo general** desarrollar un sistema que sea capaz de exportar y convertir a un formato estándar la información de las bases de datos de referencias bibliográficas de diferentes extensiones.

Para complementar el objetivo general se determinaron los siguientes **objetivos específicos**:

- ❖ Elaborar el marco teórico orientado al desarrollo de los sistemas de integración de datos bibliográficos.
- ❖ Diseñar la propuesta de solución.
- ❖ Implementar el proceso de leer, transformar y exportar los datos bibliográficos.
- ❖ Realizar pruebas a la propuesta de solución.

Para cumplir con los objetivos propuestos en esta investigación se han trazado las siguientes tareas investigativas:

1. Análisis de soluciones similares a las que se quiere implementar.
2. Estudio y análisis del formato BibTeX.
3. Descripción de las herramientas y metodología a utilizar para el desarrollo del sistema.

Introducción

4. Identificación de los requisitos del sistema.
5. Implementación del proceso de importación de los datos bibliográficos en diferentes formatos.
6. Implementación del proceso de transformación de los datos bibliográficos en formato estándar.
7. Implementación del proceso de exportación de los datos bibliográficos transformados.
8. Selección de las técnicas de pruebas a realizar al sistema.
9. Aplicación de las pruebas.
10. Recolección de las no conformidades.
11. Resolución de no conformidades y aplicación de las pruebas.

La investigación está estructurada en 4 capítulos que agruparán los contenidos de la siguiente manera:

Capítulo 1. Fundamentación Teórica: Se tratan los elementos teóricos de la investigación tales como: las extensiones de las bases de datos bibliográficas, metodología, lenguaje y herramientas de desarrollo que serán utilizadas para implementar la solución.

Capítulo 2. Características del Sistema: Se describe detalladamente la propuesta de solución y se muestran algunos artefactos como el modelo de dominio y el diagrama de casos de usos del sistema.

Capítulo 3. Modelado del Sistema: Se muestran los artefactos generados en la etapa de modelado como el modelo de diseño, también se define la arquitectura y los patrones de diseños a utilizar en el desarrollo de la solución.

Capítulo 4. Implementación y Prueba: En este capítulo se realiza el modelo de implementación y se presentarán los resultados de las diferentes pruebas realizadas al sistema implementado, para verificar que cumple con los requisitos funcionales y no funcionales obtenidos en la captura de requisitos.

CAPÍTULO 1. Fundamentación Teórica

1.1. Introducción

El objetivo principal del capítulo es explicar el marco teórico en el que se fundamenta el desarrollo del sistema informático propuesto. Para ello, se realiza un estudio de BibTeX lo que posibilita el entendimiento del formato de salida. También se realiza un estudio del marco de trabajo, ofreciendo una descripción de las principales características de los lenguajes, herramientas y metodología de desarrollo de software que se utilizarán en la implementación del sistema INDABIB.

1.2. Conceptos

1.2.1. Referencia Bibliográfica

Una referencia bibliográfica: *“Es un conjunto de datos precisos y detallados con los que un autor facilita la remisión a fuentes documentales, o a sus partes y a sus características editoriales”*. (2)

Incluyen elementos sin los cuales no se podría identificar un documento como autor, título, etc. También poseen otros elementos que aunque no son considerados primordiales brindan información acerca de la fuente documental de donde se obtiene la referencia como son: el número de las páginas, los números de tomos, las series, entre otros.

1.2.2. Gestor Referencia Bibliográfica

Es una aplicación que ayuda a crear, organizar y manejar bases de datos de referencias bibliográficas obtenidas a partir de cualquier fuente de información. (3)

Un gestor de referencia puede exportar una base de datos bibliográficas en una o varias extensiones. En cada una de las extensiones el formato utilizado a la hora de exportar los datos es distinto.

1.2.1. Base de Datos Bibliográfica

Una base de datos bibliográfica es un conjunto de información estructurada en registros bibliográficos. Cada registro constituye una unidad autónoma de información que puede a su vez estar estructurada por campos o tipos de datos. Según el contenido de sus registros se pueden dividir en tres grupos:

- ❖ Bases de datos con información factual.
- ❖ Directorios.
- ❖ Base de datos documentales.

Las bases de datos bibliográficas que son de interés para el sistema informático propuesto son las bases de datos documentales, en las cuales cada registro se corresponde con un documento, sea éste de cualquier tipo: una publicación, un documento audiovisual, de archivo, electrónico, etc.

Los registros de las bases de datos documentales pueden incluir o no el contenido completo de los documentos que describen, según lo cual se distinguen tres categorías:

- ❖ Bases de datos de texto completo.
- ❖ Archivos electrónicos de imágenes.
- ❖ Bases de datos referenciales.

Atendiendo a los registros que componen las bases de datos documentales que son de interés para el desarrollo del sistema informático propuesto se pueden clasificar como bases de datos referenciales. Sus registros no contienen el texto original sino tan sólo la información fundamental para describir y permitir la localización de documentos impresos, sonoros, iconográficos, audiovisuales o electrónicos.

1.2.2. Extensión BIB

Es un formato de archivo que contienen una lista de referencias que haya sido formateada por BibTeX ([ver definición](#)). La extensión de archivo BIB también se utiliza como archivos de bibliografía compuestos de texto y se almacenan con un formato en ASCII. Estos sólo se pueden ver con un editor de texto. (4)

```
@ARTICLE{LampportIndex,  
  AUTHOR = {Leslie Lampport},  
  YEAR = 1987,  
  TITLE = {Makeindex, {A}n Index Processor for {\LaTeX\ }},  
  NOTE = {Documentaci{\'}o{n} contenida en makeindex.doc}}
```

Figura 1: Ejemplo del formato en que exporta la extensión BIB

1.2.3. Extensión RIS

Es un formato de archivo para representar las citas bibliográficas. Contiene una serie de líneas, donde cada una comienza con un código de dos caracteres y un valor correspondiente, que proporciona información sobre alguno de los campos, así como el empezar y el terminar de la referencia. (5)

Los archivos con extensión RIS fueron creados para estandarizar el intercambio de información de citas que comúnmente se hace entre las bibliotecas digitales. Se guardan en un formato de texto ASCII. (5) En la siguiente figura se muestra el formato utilizado en la extensión RIS:

```

TY  - JOUR
AU  - van Eck, Nees Jan
T1  - Citation analysis may severely ...
JO  - PLoS One
Y1  - 2013
VL  - 8
IS  - 4
SP  - e62395
EP  - e62395
UR  - http://dx.doi.org/10.1371/journal.pone.0062395
M3  - http://dx.doi.org/10.1371/journal.pone.0062395
N2  - Citation analysis has become an important ...
ER  -
    
```

Figura 2: Ejemplo del formato en que exporta la extensión RIS

Esta extensión posee 55 tipos de registros, representado por la etiqueta TY. A continuación se relacionan los tipos de registro existentes en esta extensión:

Tipos de Registro	
ADVS: Material Audiovisual.	GRANT: Subvención.
AGGR: Base de Datos Agregadas.	HEAR: Audiencia.
ANCIENT: Texto Antiguo.	ICOMM: Comunicación por Internet.
ART: Obra.	INPR: En Prensa.
BILL: Proyecto de Ley.	JFULL: Nombre de Revista Completo.
BLOG: Blog.	JOUR: Nombre de Revista.
BOOK: Todo el Libro.	LEGAL: Norma Legal o en el Reglamento.
CASE: Caso.	MANSCPT: Manuscrito.
CHAP: Capítulo de Libro.	MAP: Mapa.
CHAR: Gráfico.	MGZN: Artículo de la Revista.
CLSWK: Obra Clásica.	MPCT: Película
COMP: Programa Informático.	MULTI: Multimedia en línea.

Introducción

CONF: Conferencia.	MUSIC: Cuenta de la música.
CPAPER: Documento de Conferencia.	NEWS: Periódico.
CTLG: Catálogo.	PAMP: Folleto.
DATA: Archivo de Datos.	PAT: Patentes.
DBASE: Base de datos en línea.	PCOMM: Comunicación Personal.
DICT: Diccionario.	RPRT: Informe.
EBOOK: Libro Electrónico.	SER: Publicación Seriada.
ECHAP: Sección de Libros Electrónicos.	SLIDE: Pancarta.
EDBOOK: Libro Editado.	SOUND: Grabación de Sonido.
EJOUR: Artículo de Electrónica.	STAND: Estándar.
ELEC: Página Web.	STAT: Estatuto.
ENCYC: Enciclopedia.	THES: Tesis / Disertación.
EQUA: Ecuación.	UNPB: Los Trabajos no Publicados.
FIGURE: Figura.	VÍDEO: Grabación de vídeo.
GEN: Genérico.	GOVDOC: Documento Gobierno.
AUS: Resumen.	

Tabla 1: Tipo de registro de la extensión RIS

Esta extensión posee 55 tipos de campos, representado por las etiquetas que a continuación se relacionan:

Tipos de Campos		
TY: Tipo de referencia.	L1: Archivos Adjuntos.	T2: Título Secundario.
A2: Autor Secundario.	L4: Figura.	T3: Título Terciario.
A3: Autor Terciario.	LA: Idioma.	TA: Autor Traducido.
A4: Autor Filial.	LB: Label.	TI: Título.
AB: Resumen.	M1: Número.	TT: Título Traducido.
AD: Dirección del Autor.	M3: Tipo de Trabajo.	UR: URL.
AN: Número de Acceso.	N1: Notas.	VL: Volumen.
AU: Autor.	NV: Número de Volúmenes.	Y2: Acceso Fecha.
C1: Personalizado 1.	OP: Publicación Original.	CY: Lugar de Publicación.
C2: Personalizado 2.	PB: Editorial.	DA: Fecha.
C3: Personalizado 3.	PY: Año.	DB: Nombre de la base de datos.
C4: Personalizado 4.	RI: Opinión de Artículo.	DO: DOI.
C5: Personalizado 5.	RN: Apuntes de Investigación.	DP: Base de Datos de Proveedores.
C6: Personalizado 6.	RP: Reimpresión Edición.	EP: Página Final.
C7: Personalizado 7.	SE: Sección.	ET: Edición.
C8: Personalizado 8.	SN: ISBN / ISSN.	ES: Número.
CA: Leyenda.	SP: Página de Inicio.	J2: Título alternativo.
CN: Signatura Topográfica.	ST: Título Corto.	ER: Fin de Referencia.
KW: Palabras clave.		

Tabla 2: Tipos de campos de la extensión RIS

1.2.4. Extensión BibTeXML

BibTeXML es una representación XML de los datos BibTeX. La ventaja de BibTeXML sobre la sintaxis nativa de BibTeX es que se puede controlar fácilmente utilizando herramientas XML estándar, por ejemplo las hojas de estilo, mientras que en BibTeX nativo los datos sólo pueden ser manipulados mediante herramientas especializadas. El objetivo de BibTeXML es desarrollar un entorno XML de representación y estructuración de la bibliografía BibTeX, lo que hace que sea más fácil la gestión de los datos bibliográficos, y la construcción de una base de datos en línea que permita la carga y la descarga de los registros bibliográficos.(6)

Utiliza los mismos campos y tipos de registros que el formato BibTeX. En la siguiente figura se muestra cual es el formato utilizado en la extensión BibTeXML.

```
<?xml version="1.0" ?>
<bibtex:file xmlns:bibtex="http://bibtexml.sf.net/">
  <!-- This file was exported from JabRef jabref.sf.net -->
  <!-- 2006-01-17 added DOI XML character formatting, because DOI can contain &lt; and &gt; -->

  <bibtex:entry id="2013">
    <bibtex:article>
      <bibtex:author>van Eck&#44; Nees Jan and Waltman&#44;</bibtex:author>

      <bibtex:title>Citation analysis may severely underestimate ...</bibtex:title>

      <bibtex:journal>PLoS One</bibtex:journal>

      <bibtex:year>2013</bibtex:year>
      <bibtex:volume>8</bibtex:volume>

      <bibtex:pages>e62395</bibtex:pages>
      <bibtex:number>4</bibtex:number>

      <bibtex:institution>Centre for Science and Technology..</bibtex:institution>

      <bibtex:abstract>Citation analysis has become an important tool ...</bibtex:abstract>
      <bibtex:url>http://dx.doi.org/10.1371/journal.pone.0062395</bibtex:url>
      <bibtex:doi>10.1371/journal.pone.0062395</bibtex:doi>

    </bibtex:article>
  </bibtex:entry>
</bibtex:file>
```

Figura 3: Ejemplo del formato en que exporta la extensión BibTeXML

1.2.5. Extensión TXT

Es uno de los formatos de archivo que utilizan los gestores de referencias bibliográficas para representar las citas bibliográficas. Contiene una serie de líneas, que comienzan con dos caracteres. El primero es % y viene acompañado de otro carácter que juntos brindan información sobre el tipo de registro o campos que se encuentra en la línea. (7)

Introducción

En la siguiente figura se muestra el formato utilizado en la extensión TXT.

```
%0 Journal Article
%A van Eck,, Nees Jan; Waltman, Ludo; van Raan,, Anthony F J.
%D 2013
%T Citation analysis may severely underestimate ...
%B PLoS One
%V 8
%N 4
%P e62395
%X Citation analysis has become an important ...
%U http://dx.doi.org/10.1371/journal.pone.0062395
```

Figura 4: Ejemplo del formato en que exporta la extensión TXT

Esta extensión posee 38 tipos de registros, representado por la etiqueta %0. A continuación se relacionan dichos tipos de registro:

Tipos de Registros	
Generic: Genérico.	Hearing: Audiencia.
Artwork: Obra.	Journal Article : Artículo de Revista.
Audiovisual Material: Material Audiovisual.	Legal Rule/Regulation: Regla Legal / Regulación.
Bill: Proyecto de Ley.	Magazine Article. Artículo de Revista.
Book: Libro.	Manuscript: Manuscrito.
Book Section: Sección Libro.	Map: Mapa.
Case: Caso.	Newspaper Article Artículo de periódico
Char or Table: Gráfico o Tabla.	Online Database: Base de Datos en Línea.
Classical Work: Obra Clásica.	Online Multimedia: Multimedia en Línea.
Computer Program: Programa de Computación.	Patent: Patente.
Conference Paper: Artículo de Evento.	Personal Communication: Comunicación Personal.
Conference Proceeding: Memoria de Eventos.	Report: Informe.
Edited Book: Libro Editado.	Statute: Estatuto.
Equation: Ecuación.	Thesis: Tesis.
Electronic Article: Artículo Electrónico.	Unpublished Work: Trabajo no Publicado.
Electronic Book: Libro Electrónico.	Unused 1: Sin Usar 1.
Electronic Source: Fuente Electrónica.	Unused 2: Sin Usar 2.
Figure: Figura.	Unused 3: Sin Usar 3.
Film or Broadcast: Transmisión o Película.	Government Document: Documento de Gobierno.

Tabla 3: Tipos de Registros de la extensión TXT

Esta extensión posee 52 tipos de campos representados por las etiquetas que a continuación se relacionan:

Tipos de Campos		
%A: Autor.	%0: Tipo de Registro	%S: Título Terciario.
%B: Título Secundario.	%1: Personalizado 1.	%T: Título.
%C: Lugar de Publicación.	%2: Personalizado 2.	%U: URL.
%D: Año.	%3: Personalizado 3.	%V: Volumen.
%E: Editor.	%4: Personalizado 4.	%X: Resumen.
%F: Label.	%<: Fecha de Acceso.	%Y: Autor Terciario
%G: Lenguaje.	%!: Título Corto.	%Z: Notas.
%H Autor Traducido.	%^: Enlace a PDF.	%#: Personalizado 5.
%I: Nombre del Publicador.	%>: Notas de Investigación.	%\$: Personalizado 6.
%J: Nombre de la Revista.	%6: Número de Volumen.	%]: Personalizado 7.
%K: Palabras Claves.	%7: Edición.	%&: Publicación Original.
%L: Número de Contacto.	%8: Fecha.	%(: Reimpresión de Edición.
%M: Número de Acceso.	%9: Tipo de Trabajo.	%) : Comentario.
%N: Número.	%?: Autor Subsidiario.	%*: Dirección del Autor.
%P: Páginas.	%@: ISBN/ISSN.	%+: Subtítulo
%R: Número del Recurso Electrónico.	%~: Nombre de la Base de Datos	%W: Proveedor de la Base de Datos.
%Q: Título Traducido.	%=: Última Fecha de Modificación.	

Tabla 4: Tipos de campos de la extensión TXT

1.2.6. BibTeX

Es una poderosa herramienta diseñada para el apoyo de la elaboración de la bibliografía. Es también un formato de archivo que se utiliza para describir y procesar listas de referencias, sobre todo en relación con los documentos LaTeX¹. Esta herramienta se recomienda para bibliografías muy extensas o cuando algunos recursos se citan con frecuencia, ya que el trabajo de capturar una referencia bibliográfica se hace sólo una vez y se puede aprovechar tantas veces como sea necesario en todos los documentos que se desee. Lo único que se necesita para emplear BibTeX es una biblioteca virtual, esto es un archivo con extensión BIB ([ver definición](#)) en el que se encuentra la bibliografía que requiere el documento. Las grandes ventajas que ofrece su uso se incrementan cuando se maneja una lista bibliográfica amplia o cuando con la misma bibliografía hay que escribir varios documentos. (7)

Este formato posee 14 tipos de registros que a continuación se relacionan:

- ❖ **article**: Un artículo de un periódico o revista.
- ❖ **book**: Un libro con una editorial que se indica en forma explícita.

¹ LaTeX es el estándar para la comunicación y publicación de documentos científicos.

Introducción

- ❖ **booklet:** Una obra que está impresa y encuadernada, pero sin una editorial o institución patrocinadora.
- ❖ **conference:** Un artículo en las actas de sesiones de una conferencia.
- ❖ **inbook:** Una parte de un libro, que puede ser un capítulo (o sección).
- ❖ **incollection:** Una parte de un libro que tiene su propio título.
- ❖ **inproceedings:** Un artículo en las actas de sesiones de una conferencia.
- ❖ **manual:** Documentación técnica.
- ❖ **mastersthesis:** Una tesis de maestría o proyecto de fin de carrera.
- ❖ **misc:** Para uso cuando los demás tipos no corresponden.
- ❖ **phdthesis:** Una tesis de doctorado.
- ❖ **proceedings:** Las actas de sesiones de una conferencia.
- ❖ **techreport:** Un reporte publicado por una escuela u otra institución.
- ❖ **unpublished:** Un documento que tiene un autor y título, pero que no fue formalmente publicado.

Este formato posee 20 tipos de campos que se relacionan a continuación:

- ❖ **address:** Dirección del publicador o la institución.
- ❖ **autor:** Nombre de los autores.
- ❖ **booktitle:** Título de un libro o parte que estamos citando.
- ❖ **chapter:** Número de capítulo, sección, etc.
- ❖ **edition:** Edición de un libro.
- ❖ **editor:** Nombre de los editores.
- ❖ **intitution:** Institución que financia un informe.
- ❖ **journal:** Nombre de la revista.
- ❖ **month:** Mes de publicación.
- ❖ **number:** Número de la revista.
- ❖ **note:** Permite introducir cualquier información adicional.
- ❖ **organization:** Organización que financia una conferencia o publica un manual.
- ❖ **pages:** Página o intervalo de páginas.
- ❖ **publisher:** Nombre del que publica el documento referenciado.
- ❖ **school:** Escuela, instituto o facultad donde se ha escrito una tesis.
- ❖ **series:** Nombre de una serie o conjunto de libros.
- ❖ **title:** Título del documento referenciado.
- ❖ **type:** Tipo de informe técnico.
- ❖ **volume:** Volumen de revista o libro de varios volúmenes.

- ❖ **year**: Año de publicación o escritura.

Estos campos son considerados necesarios u opcionales dependiendo del tipo de registro en el que se vayan a utilizar.

El formato determinado para el sistema INDABIB utilizará BibTeX para la organización y presentación de los datos en el fichero de salida. En el caso de la utilización de los tipos de registros y campos se respetará los existentes en la extensión que se quiera cargar con el objetivo de evitar la pérdida de datos que pudieran ser valiosos.

1.3. Tendencias Actuales de los Sistemas de Integración de Datos Bibliográficos.

Los Sistemas de Integración de Datos Bibliográficos se utilizan con el objetivo de integrar y convertir las bases de datos de referencias bibliográficas a un formato específico determinado para cada sistema. Como ejemplos de estos sistemas se pueden mencionar los gestores de referencia bibliográfica: EndNote, JabRef, ProCite y Zotero, entre otros, ya que cada uno de ellos cuenta entre sus funcionalidades con la de importar y exportar una base de datos de referencias bibliográficas en más de un formato. Sin embargo este no es su principal objetivo por lo que estas funcionalidades que poseen no están implementadas con profundidad. Esto significa que el proceso para hacer dichas operaciones se hace más largo, ya que consiste en crear una nueva base de datos bibliográfica, importar el archivo a la base de datos bibliográfica creada y después exportarlo al formato deseado, pero el archivo tipo XML creado por dichos sistemas no sigue el estándar deseado.

En la investigación realizada no se encontró ninguna aplicación que se dedique específicamente a integrar y estandarizar las bases de datos de referencias bibliográficas, de ahí la necesidad de crear una.

1.4. Metodología, Lenguajes y Herramientas de Desarrollo

El entorno de desarrollo definido para la construcción de INDABIB fue producto del estudio realizado de las características de la solución que se desea implementar. Se expondrán los argumentos que determinaron esta selección así como una breve descripción de cada herramienta a utilizar, enfatizando en las características que tienen impacto en el desarrollo de la solución.

1.4.1. Metodología de desarrollo

Se define la metodología como: “El conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.” (9)

Para producir un software de manera eficiente, que cumpla con los requisitos del cliente, es necesaria una metodología que sirva como guía para todos los participantes en el proceso de desarrollo. A partir de esta necesidad se aplicó el método de Boehm y Turner (23), el cual a través del análisis de sus cinco criterios que son: tamaño del equipo, criticidad del producto, dinamismo de los cambios, cultura del equipo y personal con que se cuenta, permitió determinar que la metodología más adecuada para el desarrollo del sistema INDABIB debería ser una metodología ágil.

En la siguiente imagen se representa cómo se comportan los 5 criterios que propone este método en el entorno de desarrollo del sistema INDABIB, así como también cuál es el terreno más adecuado en el cual se debe encontrar la metodología de desarrollo de software a utilizar.

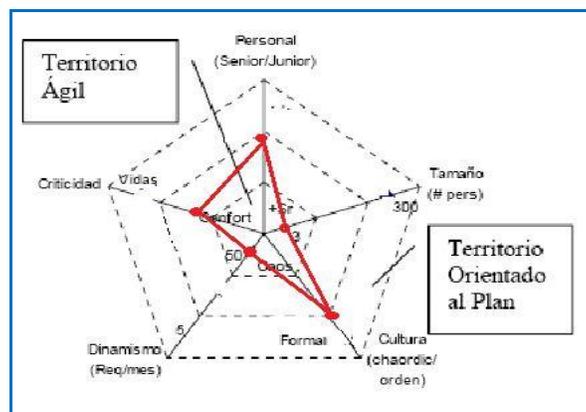


Figura 5: Representación de la estrella de BOEHM y Turner para el desarrollo del sistema INDABIB.

El Proceso Unificado Ágil (AUP) es una versión simplificada del Proceso Unificado de Desarrollo (RUP). Describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software, usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. AUP puede verse más simple que las versiones más clásicas del proceso unificado, mientras que desde el punto de vista ágil, puede entenderse como una versión más pesada que las metodologías ágiles. (10).

Introducción

La utilización de AUP ofrece numerosas ventajas al equipo de desarrollo, entre ellas se pueden mencionar: (10)

- ❖ **El personal sabe lo que están haciendo.** Las personas no van a leer documentación de procesos detalladamente, pero quieren alguna guía de alto nivel y/o capacitaciones de vez en cuando. Este producto proporciona vínculos a muchos de los detalles, si está interesado, pero no los obliga ni los impone.
- ❖ **Simplicidad.** Todo se describe brevemente utilizando pocas páginas y no miles de ellas.
- ❖ **Agilidad.** Permite responder de manera rápida y apropiada a los cambios.
- ❖ **Enfocar las actividades de alto nivel.** Se centra en las actividades que realmente cuentan, no en cada cosa posible que podría pasarle al proyecto.
- ❖ **Independencia de Herramientas.** Se puede usar el conjunto de herramientas que se desee con AUP.

La siguiente figura representa a AUP en sus dos dimensiones:

- ❖ El eje horizontal representa el tiempo en las cuatro fases en que se descompone el proceso.
- ❖ El eje vertical representa la serie de flujos de trabajo que lo construyen gradualmente.

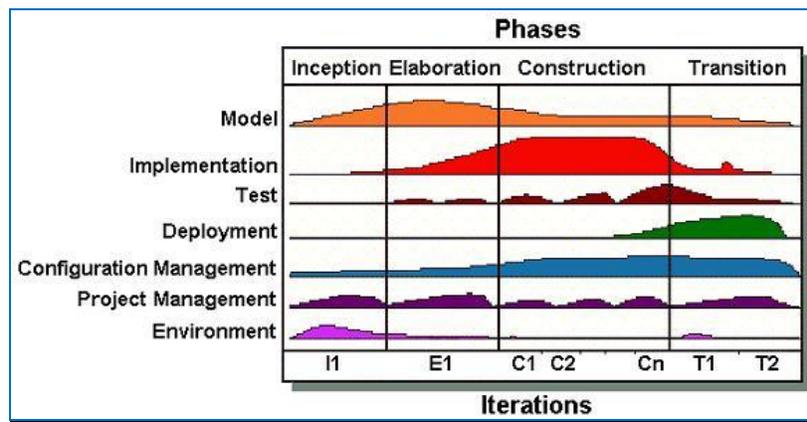


Figura 6: Proceso Unificado Ágil

Como se aprecia en la Figura 6, AUP está compuesto por cuatro Fases que son: (10)

- ❖ **Inicio:** El objetivo es identificar el alcance inicial del proyecto, una arquitectura base del sistema, obtener una primera propuesta de presupuesto del proyecto así como una aceptación de los involucrados.
- ❖ **Elaboración:** El objetivo es establecer un plan de proyecto y una arquitectura correcta del sistema.

Introducción

- ❖ **Construcción:** El objetivo es construir un software funcional sobre una base regular e incremental, las cuales cumplan con las prioridades más importantes para los involucrados o usuarios del proyecto.
- ❖ **Transición:** El objetivo es validar y desplegar el sistema en su ambiente de producción.

Como se aprecia en la Figura 6, AUP está compuesto por siete Disciplinas o Flujos de Trabajos que son: (10)

- ❖ **Modelado:** El objetivo de esta disciplina es entender el negocio de la organización e identificar una solución viable para abordar el dominio del problema.
- ❖ **Implementación:** El objetivo de esta disciplina es transformar sus modelos en código ejecutable y realizar pruebas en un nivel básico.
- ❖ **Prueba:** El objetivo de esta disciplina consiste ejecutar una evaluación de los objetivos para asegurar la calidad. Esto incluye encontrar defectos, validar que el sistema función como fue diseñado y verificar que los requerimientos están completos.
- ❖ **Despliegue:** El objetivo de esta disciplina es planificar la entrega del sistema y ejecutar el plan para que el sistema esté a disposición de los usuarios finales.
- ❖ **Gestión de configuración:** El objetivo de esta disciplina es administrar el acceso a los entregables o productos del proyecto. Esto incluye no sólo el rastreo de versiones del producto en el tiempo, sino que también incluye controlar y administra los cambios que ocurran.
- ❖ **Gestión de proyectos:** El objetivo de esta disciplina es dirigir las actividades que se llevan a cabo en el proyecto. Esto incluye administración del riesgo, la dirección de personas (asignar tareas, seguimiento de los procesos, etc.), y coordinar con los sistemas y personas fuera del alcance del proyecto para que este termine a tiempo y dentro del presupuesto.
- ❖ **Entorno:** El objetivo de esta disciplina es apoyar el resto de los esfuerzos por garantizar que, el proceso adecuado, la orientación (normas y directrices) y herramientas (hardware, software, etc.) estén disponibles para el equipo según sea necesario.

1.4.2. Lenguajes de Modelado UML

Como lenguaje de modelado se escogió el Lenguaje Unificado de Modelado (UML), que es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. (11)

La especificación de UML no define un proceso estándar, pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. UML también contiene construcciones organizativas para agrupar los modelos en paquetes, lo que permite a los equipos de software dividir grandes sistemas en piezas de trabajo, para entender y controlar las dependencias entre paquetes, y para gestionar las versiones de las unidades del modelo, en un entorno de desarrollo complejo. (11)

1.4.3. Herramienta Case: Visual Paradigm for UML Enterprise Edition v8.0

La Ingeniería de Software Asistida por Computadora (CASE por sus siglas en inglés), constituye la aplicación de métodos y técnicas utilizadas para ayudar a las actividades del proceso del software, como el análisis de requerimientos, el modelo de sistemas, la depuración y las pruebas. Las herramientas CASE representan una forma que permite modelar los procesos de negocios de las empresas y desarrollar los sistemas de información.

Visual Paradigm es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Utiliza UML como lenguaje de modelado. Es una herramienta que genera la documentación del proyecto automáticamente en varios formatos como Web o PDF, soporta la realización de ingeniería tanto directa como inversa. Presenta una interfaz de uso intuitiva y con muchas facilidades a la hora de modelar los diagramas que soportan la Ingeniería de Requerimientos. Dentro sus características están su robustez, usabilidad y portabilidad. (12)

1.4.4. Lenguaje de Programación C#

Los lenguajes de programación son un medio de expresar un algoritmo o solucionar el problema de forma tal que la máquina lo entienda. Se pueden clasificar atendiendo a

Introducción

varios criterios, uno de los criterios es el nivel de abstracción y el otro según la forma de ejecución (13).

Según su nivel de abstracción:

- ❖ Lenguajes máquina.
- ❖ Lenguajes de bajo nivel.
- ❖ Lenguajes de medio nivel.
- ❖ Lenguajes de alto nivel.

Según la forma de ejecución:

- ❖ Lenguajes compilados.
- ❖ Lenguajes interpretados.

C# es el lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo. La sintaxis y estructuración es muy parecida a la de C++ o Java, puesto que la intención de Microsoft es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Es un lenguaje compilado y de alto nivel según su forma de ejecución y su nivel de abstracción respectivamente. Entre sus principales características están:(14)

- ❖ **Modernidad:** C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones, como un tipo básico decimal, la inclusión de una instrucción **foreach**, la inclusión de un tipo básico **string** o la distinción de un tipo **bool** específico para representar valores lógicos.
- ❖ **Orientación a objetos:** C# es un lenguaje orientado a objetos. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro. No admite ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código. Soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo.

Introducción

- ❖ **Sencillez:** C# elimina elementos que otros lenguajes incluyen y que son innecesarios en .NET.

1.4.5. Entorno Integrado de Desarrollo (IDE) Visual Studio 2010

Los IDEs son un conjunto de herramientas para el programador, que suelen incluir en una misma suite, un buen editor de código, administrador de proyectos, enlace transparente a compiladores y **debuggers** e integración con sistemas controladores de versiones o repositorios. Hoy en día los entornos de desarrollo proporcionan un marco de trabajo para la mayoría de los lenguajes de programación existentes en el mercado (por ejemplo C, C++, C#, Java, Python y Visual Basic entre otros). Además es posible que un mismo entorno de desarrollo tenga la posibilidad de utilizar varios lenguajes de programación. (15)

Visual Studio 2010 es un entorno integrado que simplifica las tareas básicas de creación, depuración e implementación de aplicaciones. Posee eficaces herramientas de elaboración de prototipos, arquitectura y desarrollo que le permiten dar vida a las ideas y dirigir las a un gran número de plataformas y tecnologías, incluida la informática de nube y en paralelo. Utiliza características de colaboración avanzadas y posee herramientas de pruebas y depuración integradas para detectar y corregir errores de un modo rápido y sencillo, lo que permite crear soluciones de alta calidad y reducir el coste del desarrollo de soluciones.

Toda la interfaz ha sido rediseñada para simplificar el flujo de trabajo y brindar fácil acceso a las herramientas. Las barras de herramientas están simplificadas, se redujo el desorden de pestañas. El Visual Studio 2010 describe las nuevas características del lenguaje C# y del editor de código. Algunas de estas características son el tipo dynamic, los argumentos opcionales y con nombre, la programación de Office mejorada y la varianza. (16)

1.4.6. FRAMEWORK .NET v 4.0

El Framework .NET v 4.0 es un modelo de programación completo y coherente para la creación de aplicaciones que tengan experiencias de usuario visualmente impresionantes y comunicación segura y sin problemas. El perfil de cliente está diseñado para ejecutar aplicaciones cliente y permitir el despliegue rápido posible para Windows Presentation Foundation (WPF) y la tecnología de Windows Forms. (17)

Introducción

El Framework .NET v 4.0 ofrece nuevas características y mejoras: (17)

- ❖ Mejoras en Common Language Runtime (CLR) y Base Class Library (BCL)
 - Mejora del rendimiento incluyendo mejor soporte multi-núcleo y recolección de basura de fondo.
 - Nueva memoria mapeada de archivos y tipos numéricos.
 - Más fácil la depuración, incluida la depuración de volcado, mini volcados Watson, depuración en modo mixto para 64 bits y contratos de código.
- ❖ Las innovaciones en el lenguajes de Visual Basic y C, por ejemplo, continuaciones de línea implícitas, distribución dinámica y parámetros con nombre/opcional.
- ❖ Las mejoras en el acceso a datos y modelado.
- ❖ Mejoras en Windows Presentation Foundation (WPF)
 - Añadido soporte para 7 multi-touch, controles de cinta de Windows, y las características de extensibilidad de la barra de tareas.
 - Nuevos controles de línea de negocio, incluyendo control de gráficos, edición inteligente, cuadrícula de datos, y otros que mejoran la experiencia de los desarrolladores que crean aplicaciones centradas en datos.
 - Las mejoras en el rendimiento y la escalabilidad.
 - Mejoras visuales en la claridad del texto, ajuste de píxeles de diseño, localización e interoperabilidad.

Conclusiones

En este capítulo se realizó un estudio del marco teórico en el que se fundamenta el desarrollo del sistema informático propuesto, así como de las herramientas que se utilizarán, quedando escogido el C# como lenguaje de programación a utilizar, el Visual Studio 2010 como IDE de desarrollo y la metodología AUP es la encargada de guiar todo el proceso de elaboración del sistema informático propuesto.

CAPÍTULO 2. Características del Sistema.

2.1. Introducción

En el presente capítulo se detallan las características que debe tener el sistema a desarrollar, a partir de la descripción general de la propuesta de solución y de los procesos que serán objeto de automatización, se crea el modelo de dominio e identifican los requisitos funcionales y no funcionales del sistema.

2.2. Objeto de Automatización

Para el desarrollo del sistema Sistrada NOYSE se hace necesario automatizar los procesos de búsqueda, organización, procesamiento, análisis, presentación y difusión de la información, que son las fases que conforman el ciclo de vida de la vigilancia tecnológica. Su ejecución sin la ayuda de un sistema informático puede resultar engorrosa y propensa a errores. El sistema INDABIB se encargará de automatizar la fase de organización de la información, computarizando el proceso de importar, formatear y exportar los datos bibliográficos con el objetivo de que el sistema sea capaz de reconocerlas para su posterior análisis.

2.3. Propuesta del Sistema

Atendiendo al estudio antes realizado se define la siguiente propuesta de solución: INDABIB será una aplicación de escritorio, que pone a disposición del sistema Sistrada NOYSE las bases de datos bibliográficas con extensión RIS, TXT, BibTeX o BIB estandarizadas para su análisis. El sistema debe ser capaz de integrar y convertir las bases de datos de referencias bibliográficas con extensión RIS, TXT, BibTeX o BIB a un formato estándar, así como exportar las mismas a un fichero con extensión .XML. Se utilizará BibTeX para la organización y presentación de los datos en el fichero de salida, en el caso de la utilización de los tipos de registros y campos se respetarán los existentes en la extensión que se quiera cargar con el objetivo de evitar la pérdida de información que podría ser valiosa. La aplicación propuesta podrá ser utilizada como módulo inicial o de entrada del sistema Sistrada NOYSE.

2.4. Modelo de Dominio

Un modelo de dominio es: *“Una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés.”*(21)

Capítulo 2. Características del Sistema

En el modelo de dominio están representados los principales conceptos y las relaciones entre ellos, pudiendo permitir la comprensión de los conceptos fundamentales y el contexto en el que se desarrolla el sistema.

Un gestor de referencias bibliográficas genera bases de datos de referencias bibliográficas que pueden tener extensión RIS, TXT, BibTeX o BIB. El investigador de forma manual tiene que realizar el proceso de transformación necesario para que las bases de datos bibliográficas contengan el formato BibTeX y la extensión XML necesarias para ser reconocidas por el sistema Sistrada NOYSE.

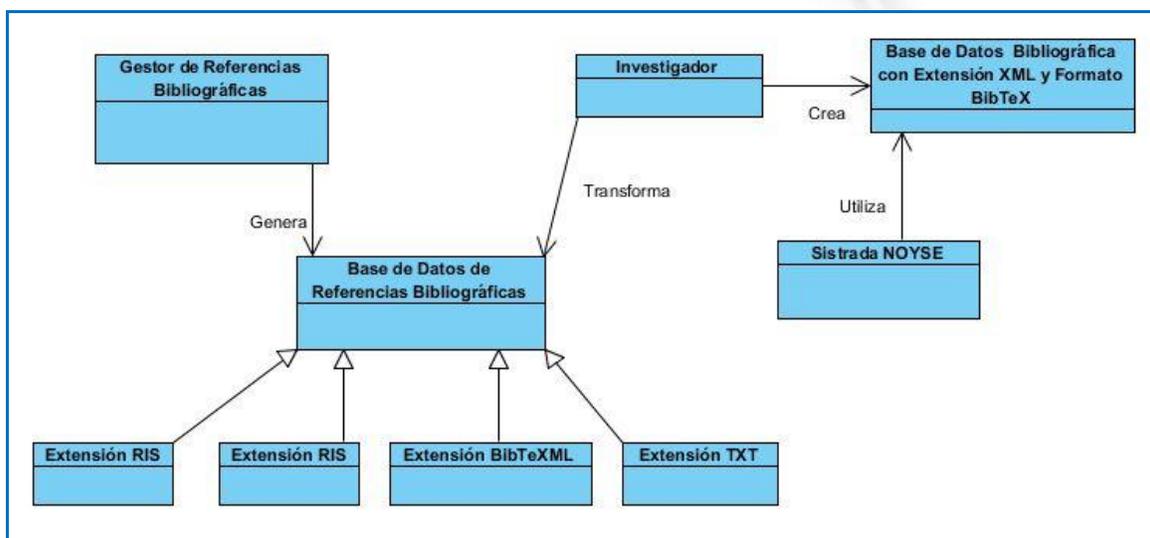


Figura 7: Modelo de Dominio

2.4.1. Descripción del Modelo conceptual

Gestor de Referencias Bibliográficas: Herramienta encargada de generar las bases de datos bibliográficas para ser utilizadas por el investigador.

Base de Datos de Referencias Bibliográficas: Son los archivos generados por los gestores de referencias bibliográficas, que poseen un formato y extensión específico para cada una de ellos.

Extensión BIB: Es una de las extensiones en que se puede exportar una base de datos bibliográficas y posee un formato específico.

Extensión RIS: Es una de las extensiones en que se puede exportar una base de datos bibliográficas y posee un formato específico.

Capítulo 2. Características del Sistema

Extensión BibTeXML: Es una de las extensiones en que se puede exportar una base de datos bibliográficas y posee un formato específico.

Extensión TXT: Es una de las extensiones en que se puede exportar una base de datos bibliográficas y posee un formato específico.

Investigador: Persona encargada de transformar las bases de datos bibliográficas de forma manual y crear una nueva base de datos bibliográfica con extensión XML y formato BibTeX.

Base de Datos de Referencias Bibliográficas con Extensión XML y Formato BibTeX: Son las bases de datos bibliográficas que poseen un formato basado en BibTeX y extensión XML.

2.5. *Requerimientos Funcionales del sistema.*

Los requisitos funcionales constituyen las capacidades o condiciones que el sistema debe cumplir. Estos deben ser comprensibles por los clientes, usuarios y desarrolladores, deben tener una sola interpretación y estar definidos en forma medible y verificable. (18)

Los requisitos funcionales que el sistema INDABIB debe cumplir son:

- ❖ **RF 1.** Cargar base de datos bibliográfica.
 - 1.1. Cargar base de datos bibliográfica con extensión RIS.
 - 1.2. Cargar base de datos bibliográfica con extensión BIB.
 - 1.3. Cargar base de datos bibliográfica con extensión TXT.
 - 1.4. Cargar base de datos bibliográfica con extensión BibTeXML.

- ❖ **RF 2.** Preparar referencias bibliográficas.
 - 2.1. Preparar referencia bibliográfica con extensión RIS.
 - 2.2. Preparar referencia bibliográfica con extensión BIB.
 - 2.3. Preparar referencia bibliográfica con extensión TXT.
 - 2.4. Preparar referencia bibliográfica con extensión BibTeXML.

- ❖ **RF 3.** Formatear referencia bibliográfica.
 - 3.1. Formatear referencia bibliográfica con extensión RIS.
 - 3.2. Formatear referencia bibliográfica con extensión BIB.
 - 3.3. Formatear referencia bibliográfica con extensión TXT.
 - 3.4. Formatear referencia bibliográfica con extensión BibTeXML.

Capítulo 2. Características del Sistema

- ❖ **RF 4.** Exportar base de datos bibliográfica en archivo de tipo XML.

2.6. *Requerimientos no Funcionales del Sistema.*

Los requisitos no funcionales son propiedades o cualidades que el sistema debe tener. Estas propiedades o cualidades son las características que hacen al sistema atractivo, usable y confiable. (18)

Los requisitos no funcionales que el sistema INDABIB debe cumplir son:

Apariencia o Interfaz Externa

- **RNF 1.** Los colores utilizados en el diseño de las interfaces deben ser: negro, blanco, escala de azul y gris.
- **RNF 2.** Todos los textos y mensajes en pantalla aparecerán en idioma español.

Usabilidad

- **RNF 3.** El nivel de funcionamiento del sistema deberá corresponder el nivel básico de conocimiento informático de los usuarios.

Software

- **RNF 4.** En las estaciones de trabajo se requiere Framework .NET v 4.0 o superior.
- **RNF 5.** En las estaciones de trabajo se requiere que el sistema operativo sea Windows XP o superior.

Hardware

- **RNF 6.** En las estaciones de trabajo se requiere que el disco duro contenga 10 MB libres o más.
- **RNF 7.** En las estaciones de trabajo se requiere 256 MB de memoria RAM.
- **RNF 8.** En las estaciones de trabajo se requiere un procesador a 1GHz.

2.7. *Modelo de Casos de Uso del Sistema.*

El modelo de casos de uso se define como el conjunto de todos los casos de uso; es un modelo de la funcionalidad y entorno del sistema. (21)

Capítulo 2. Características del Sistema

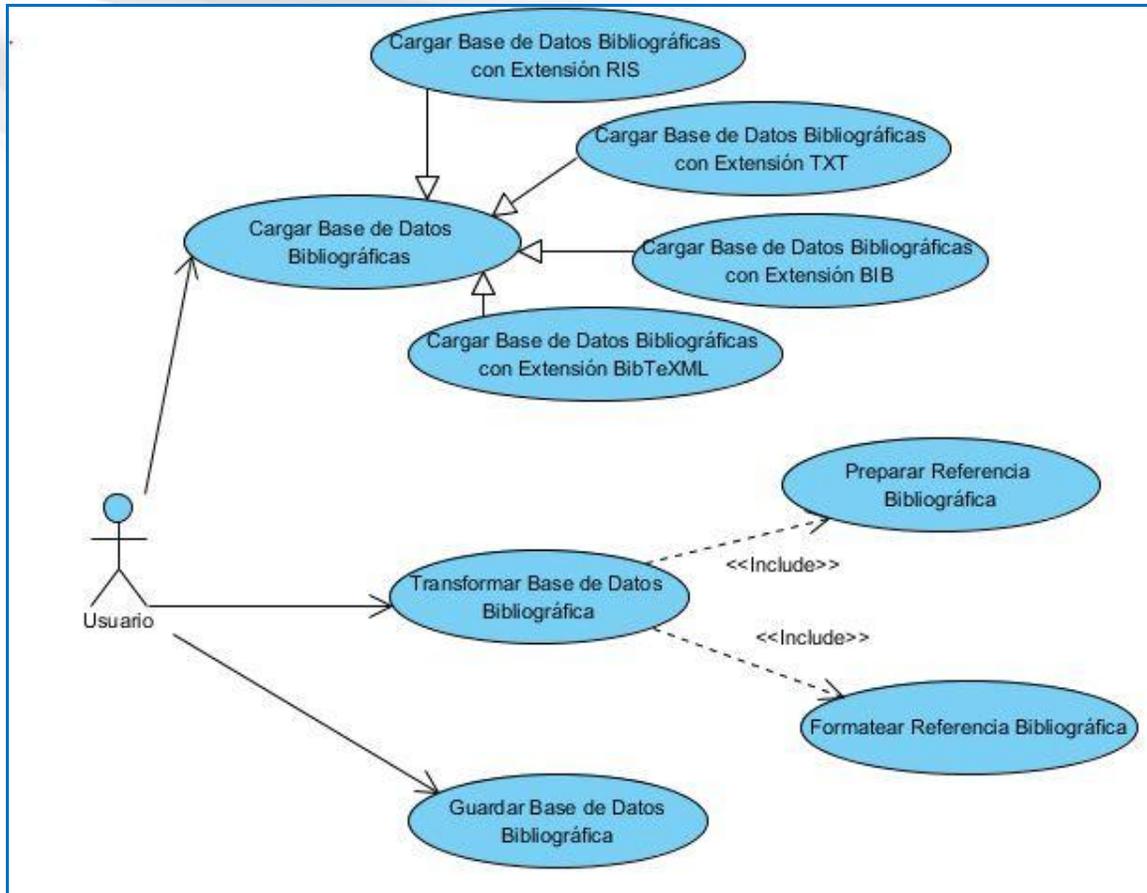


Figura 8: Diagrama de casos de uso

2.8. Especificación de los Casos de Uso del Sistema.

Mediante la especificación de los casos de uso se puede describir paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. A continuación se presenta la descripción del caso de uso Cargar Base de Datos Bibliográfica que es el encargado de iniciar el flujo principal del sistema.

2.8.1. Descripción CU Cargar Base de Datos Bibliográfica.

Objetivo	Cargar una base de datos de referencia bibliografía en el sistema.
Actores	Usuario
Resumen	El CU comienza cuando el usuario desea cargar una base de datos de referencias bibliográficas en el sistema. Este es dependiente de la extensión que posea la base de datos bibliográfica lee sus datos y los separa por referencias bibliográficas. Cuando la base de datos bibliográfica haya sido descompuesta completamente en referencias bibliográficas termina el CU.
Complejidad	Baja

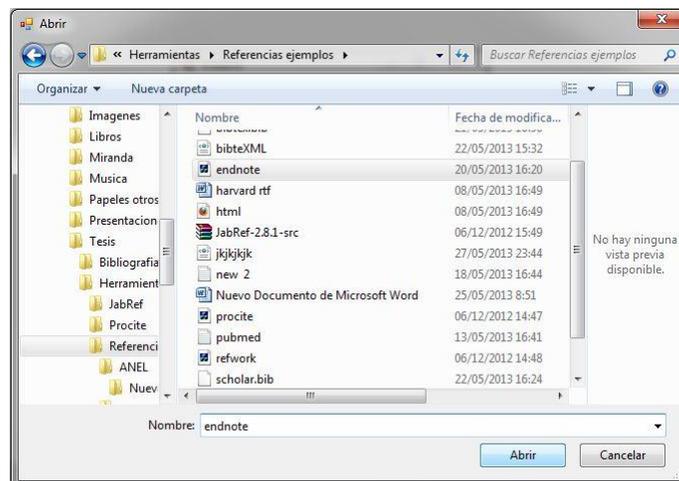
Capítulo 2. Características del Sistema

Precondiciones	La base de datos bibliográfica debe tener extensión RIS, TXT, BibTeXML o BIB.
Postcondiciones	
Flujo de eventos	
Flujo Básico	
Actor	Sistema
	1. El sistema muestra una interfaz de bienvenida. Y permite: <ul style="list-style-type: none"> • Aceptar la operación
2. El usuario oprime el botón "Aceptar".	3. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Transformar Base de Datos Bibliográfica (Deshabilitada). • Guardar Base de Datos Bibliográfica (Deshabilitada). • Salir (Permite salir del sistema).
4. El usuario selecciona la opción "Cargar Base de Datos Bibliográfica".	5. El sistema muestra la interfaz para escoger el archivo que se desea cargar. Y permite: <ul style="list-style-type: none"> • Aceptar la operación • Cancelar la operación
6. El usuario selecciona el archivo deseado y oprime el botón "Aceptar"	7. El sistema verifica que el archivo seleccionado este en alguno de las siguientes extensiones: RIS, TXT, BibTeXML o BIB.
	8. El sistema muestra una interfaz que contiene la información de la base de datos bibliográfica cargada. Y permite: <ul style="list-style-type: none"> • Aceptar la operación
9. El usuario oprime el botón "Aceptar"	10. El sistema identifica la extensión de la base de datos bibliográfica. Si la extensión es RIS invoca <u>CU. Cargar Base de Datos Bibliográficas con Extensión RIS</u> . Si la extensión es TXT invoca <u>"CU. Cargar Base de Datos Bibliográficas con Extensión TXT"</u> .

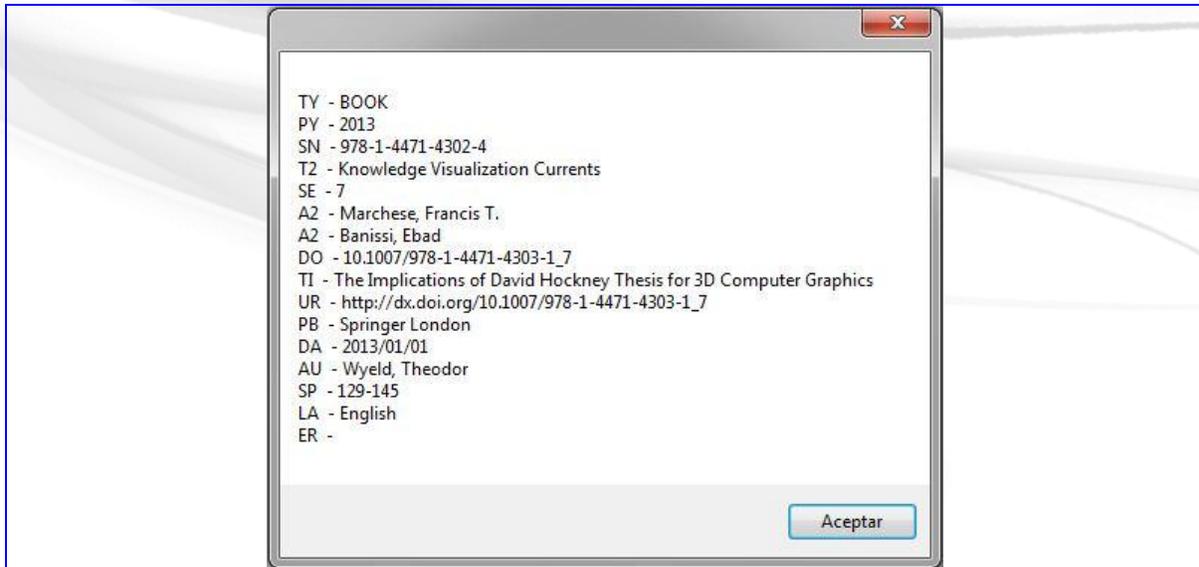
Capítulo 2. Características del Sistema

	<p>Si la extensión es BIB invoca “<u>CU. Cargar Base de Datos Bibliográficas con Extensión BIB</u>”.</p> <p>Si la extensión es BibTeXML invoca “<u>CU. Cargar Base de Datos Bibliográficas con Extensión BibTeXML</u>”.</p>	
	<p>11. El sistema muestra el siguiente mensaje de información: “La base de datos bibliográfica ha sido cargada satisfactoriamente.”</p> <p>Y permite:</p> <ul style="list-style-type: none"> • Aceptar la operación 	
11. El usuario oprime el botón “Aceptar”.	<p>12. El sistema habilita en el menú de la interfaz principal la opción:</p> <p>Transformar Base de Datos Bibliográfica (Permite formatear la base datos bibliográfica).</p>	
Flujos Alternos		
Flujo Alternativo 4b. Seleccionar opción Salir		
Actor	Sistema	
	4b.1. El sistema finaliza.	
Flujo Alternativo 6b Oprime el botón Cancelar		
Actor	Sistema	
	6b.1 El sistema regresa al paso 3 del flujo básico.	
Flujo Alternativo 7b Extensión incorrecta		
Actor	Sistema	
	7b.1 El sistema muestra el siguiente mensaje: “La Base de Datos Bibliográfica no se encuentra en uno de los siguientes formatos RIS, TXT, BibTeXML o BIB.”.	
7b.2 El usuario oprime el botón Aceptar.	7b.3 El sistema regresa al paso 3 del flujo básico.	
Relaciones	CU Incluidos	No
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		

Capítulo 2. Características del Sistema



Capítulo 2. Características del Sistema



Para acceder al resto de las descripciones de los casos de uso de sistema INDABIB (ver Anexo 1).

Conclusiones

En este capítulo se hizo referencia a las principales características que tendrá el sistema a desarrollar, quedando definida una propuesta de solución. Se concretaron también los requisitos funcionales y no funcionales que cumplirá el software, así como los prototipos de interfaz que tendrá la propuesta de solución.

CAPÍTULO 3. Modelado del Sistema.

3.1. Introducción

En el presente capítulo se tratan los temas relacionados con el diseño de la propuesta de solución, se describe la arquitectura base y se muestran algunos de los artefactos generados en la etapa del modelado que guiarán la implementación del sistema.

El objetivo del diseño es lograr una representación arquitectónica de un sistema, que a su vez pueda servir como marco de trabajo para llevar a cabo las actividades del diseño más detalladas. (20)

3.2. Arquitectura del Sistema

“La arquitectura del software alude a la estructura global del software y a las formas en que la estructura proporciona la integridad conceptual de un sistema. En su forma más simple, la arquitectura es la estructura jerárquica de los componentes del programa (módulos), la manera en que los componentes interactúan y la estructura de datos que van a utilizar los componentes. Sin embargo, en un sentido más amplio, los componentes se pueden generalizar para representar los elementos principales del sistema y sus interacciones.” (20)

Las arquitecturas en capas son utilizadas para el desarrollo de aplicaciones en la actualidad por las ventajas que proveen. Su principal objetivo es reducir dependencias entre artefactos, situándolos en capas lógicas, donde cada capa depende del servicio prestado por la inferior y presta un servicio a la superior, proporcionando a los desarrolladores ventajas en cuanto al mantenimiento y reutilización de componentes o artefactos. Cada capa se ocupa de un nivel del problema y debe tener poco acoplamiento con las demás, de manera tal que el cambio en una capa, no provoque grandes cambios en otra. Además, el estilo admite optimizaciones, refinamientos y proporciona amplia reutilización.

3.2.1. Capas del Sistema

El sistema INDABIB se construirá sobre una arquitectura en capas lo que permitirá realizar un desacople tanto lógico como físico de todos sus componentes. La arquitectura a utilizar constará de 3 capas que a continuación se detallan:

Capítulo 3. Modelado del Sistema

- ❖ **Capa de presentación:** Esta es la capa encargada de la interacción con los usuarios, se encarga de modelar la forma en que se cargarán y mostrarán los datos a los usuarios, así como de la apariencia visual que tendrá la aplicación. Se comunica con la capa de lógica de negocio a la cual envía todas las solicitudes de operaciones que los usuarios desean hacer y recibe la respuesta después de que hayan sido procesadas. En la aplicación esta capa está compuesta por las interfaces **V_Bienvenida** e **V_Indabib**.
- ❖ **Capa de lógica de negocio:** La capa de lógica de negocio representa las clases controladoras del negocio. Es la encargada de modelar la lógica de negocio que dará solución a los casos de usos de la aplicación, es en esta donde se establecen las reglas o restricciones que debe cumplir el sistema. Se comunica con la capa de presentación para recibir las solicitudes de operaciones que los usuarios desean hacer y enviar las respuestas después del procesamiento. En la aplicación esta capa está compuesta por las clases **P_Indabib**, **P_BDBibliografica**, **P_Extension_RIS**, **P_Extension_TXT**, **P_Extension_BIB**, **P_Extension_BibTeXML**.
- ❖ **Capa de acceso a datos:** La capa de acceso a datos contiene las clases que almacenan la información del sistema y es la encargada del manejo de datos persistentes dentro del mismo. Esta capa se comunicará con la capa de lógica de negocio desde donde recibe las solicitudes de almacenar u obtener información. Actúa como intermediaria entre el gestor de base de datos y los demás componentes de la arquitectura. En la aplicación esta capa está compuesta por las clases **AC_BDBibliografica**, **AC_Extension_RIS**, **AC_Extension_TXT**, **AC_Extension_BIB**, **AC_Extension_BibTeXML**.

3.2.2. Patrones y Estilo Arquitectónico

Un patrón de arquitectura de software se selecciona con base en objetivos y restricciones. Los patrones expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos; así como ayudan a especificar la estructura fundamental de una aplicación. (21)

El patrón de arquitectura Modelo Vista Presentador (MVP) divide una aplicación interactiva en tres partes: el modelo, las vistas y el presentador (22).

Capítulo 3. Modelado del Sistema

- ❖ **El Modelo:** Contiene los datos y las funcionalidades esenciales. Es la representación en objetos del negocio y será manipulado desde y hacia la vista mediante el presentador. El modelo puede estar dado directamente por la estructura de base de datos o puede estar dado por objetos del tipo entidad que representan al negocio y que serán persistidos en la base de datos. En el sistema INDABIB el modelo está conformado por las clases que contienen la información perdurable y que será persistida en un archivo tipo XML. Las clases que lo componen son: **AC_BDBibliografica**, **AC_Extension_RIS**, **AC_Extension_TXT**, **AC_Extension_BIB**, **AC_Extension_BibTeXML**.
- ❖ **Las Vistas:** Despliegan la información al usuario. Son las interfaces de usuario propiamente dichas. Cada vista es una clase que implementara una interfaz, y será totalmente pasiva, esto significa que todo el comportamiento (básicamente métodos) será implementado por la clase presentador y quitados de la vista. En el sistema INDABIB las vistas que son: **V_Bienvenida** e **V_Indabib** solo se encargan de la presentación y recogida de los datos.
- ❖ **El Presentador:** Manejan las entradas de datos. Funciona como intermediario entre la vista y el modelo de datos. Tiene la responsabilidad de implementar el comportamiento que antes estaba implementado en las vistas. En el sistema INDABIB las clases que lo conforman son: **P_Indabib**, **P_BDBibliografica**, **P_Extension_RIS**, **P_Extension_TXT**, **P_Extension_BIB**, **P_Extension_BibTeXML** y se encargan de todo el comportamiento lógico del sistema.

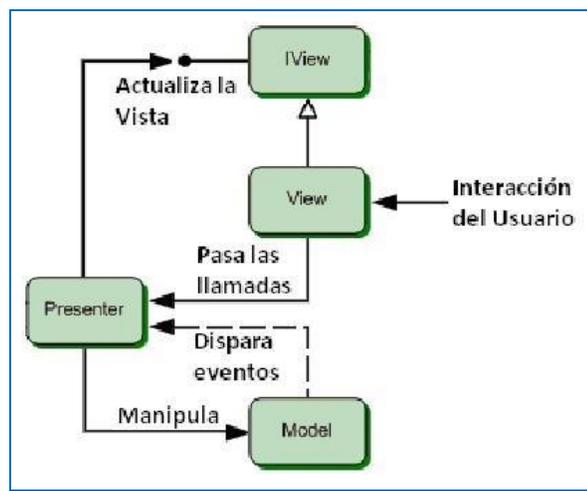


Figura 9: Modelo Vista Presentador (expandido).

3.3. Diseño del Sistema

En el diseño se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, tanto funcionales como no funcionales y las restricciones que se le imponen. Es usado como entrada esencial en las actividades relacionadas a la implementación y representa a los casos de uso en el dominio de la solución.

3.3.1. Modelo de Diseño

“El modelo de diseño es un modelo de objetos que describe la realización de los casos de uso y se utiliza como abstracción del modelo de implementación y el código fuente.”(20)

Está conformado por otros modelos. La metodología AUP utilizada en el desarrollo de la solución, plantea que no es necesario crear todos los modelos que conforman al modelo de diseño, solamente los que sean apropiados al contexto donde se aplica. Los modelos creados para la implementación del sistema INDABIB se relacionan a continuación.

3.3.1.1. Diagrama de Paquetes

Los diagramas de paquetes muestran la descomposición jerárquica lógica de un sistema, así como las dependencias entre las partes lógicas que lo conforman. A continuación se muestra el diagrama de paquetes del sistema INDABIB, en el cual se puede apreciar que la estructura organizativa de los paquetes responde a la arquitectura definida para el sistema.

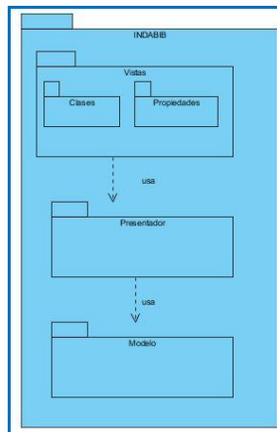


Figura 10: Diagrama de Paquetes del sistema INDABIB

3.3.2. Patrones de Diseño

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características, una de ellas es que se debe haber comprobado su efectividad resolviendo problemas similares; otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. Entre los elementos que componen a un patrón con el objetivo de que sea más fácil su utilización se encuentran el nombre, el problema, la solución, y las consecuencias. (21)

Entre los patrones de diseño se encuentran los Patrones Generales de Software de Asignación de Responsabilidades (GRAPS) que describen los principios fundamentales de la asignación de responsabilidades a objetos y los *Gans of Four* (GoF) que enmarcan los llamados patrones de diseño Estructurales, Creacionales y de Comportamiento. A continuación se listan los patrones de diseño utilizados: (21)

❖ GRASP

- **Patrón Experto:** Asignar una responsabilidad al experto en la información, es decir a la clase que contiene la información necesaria para realizar la responsabilidad. Es un principio básico que suele utilizarse cuando se programa orientado a objetos. (21)

La aplicación del patrón Experto permitió la correcta distribución del comportamiento de las clases encargadas de manejar la información del sistema, lo que a su vez contribuye a una alta cohesión y bajo acoplamiento de los componentes del modelo de clases.

En la siguiente figura se muestra un ejemplo del uso de este patrón en el sistema informático propuesto, evidenciando como la clase **P_Extensión_BibTeXML** es la encargada de realizar las operaciones relacionadas a este formato:

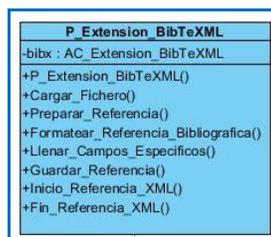


Figura 12: Ejemplo de aplicación del patrón Experto

Capítulo 3. Modelado del Sistema

- **Patrón Controlador:** Aumenta el potencial de reutilización, y asegura que la lógica de la aplicación no se maneja en la capa de interfaz. Un Controlador es un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema y define el método para la operación del sistema. (21)

En la siguiente figura se muestra un ejemplo del uso de este patrón en el sistema informático propuesto, siendo la clase **P_Indabib** la encargada de controlar el flujo del sistema.

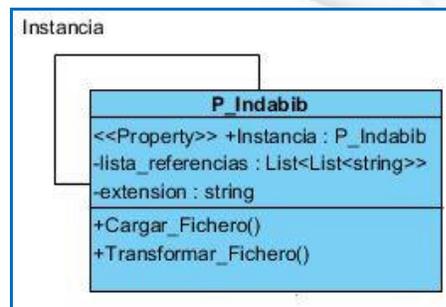


Figura 13: Ejemplo de aplicación del patrón Controlador

- **Patrón Creador:** Es el encargado de guiar la asignación de responsabilidades relacionadas con la creación de los objetos, encontrando un creador para cada objeto, que al ser seleccionado de manera adecuada da soporte al bajo acoplamiento. (21)

A continuación se muestra una figura ejemplificando la aplicación de este patrón en la solución propuesta:

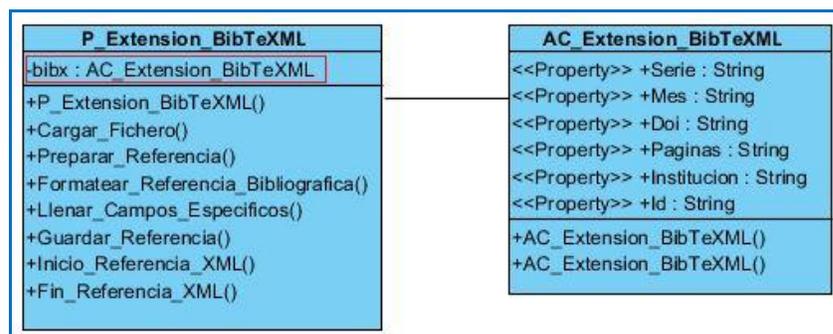


Figura 14: Ejemplo de aplicación del patrón Creador

- **Patrón Alta Cohesión:** Es un principio evaluativo que aplica un diseñador mientras evalúa todas las decisiones de diseño. Se puede medir el nivel de cohesión en una clase cuanto más enfocado sea su comportamiento. Cada

Capítulo 3. Modelado del Sistema

elemento del diseño debe realizar una labor única dentro del sistema, que no se desempeñe en ninguno de los demás elementos del sistema. (21)

En el sistema INDABIB se evidencia el patrón Alta Cohesión debido a que la asignación de responsabilidades se realizó de modo que la cohesión siguiese siendo alta; posibilitando una mayor capacidad de reutilización.

❖ GOF

- **Singleton:** Este patrón permite controlar la instanciación de una clase, proporcionando un punto de acceso global a esta. No permite que existan múltiples instancias de una clase, dando un único punto de acceso a la misma. (21) En la siguiente figura se muestra un ejemplo del uso de este patrón en el sistema informático propuesto, garantizando que la clase **P_Indabib** sea instanciada una única vez.

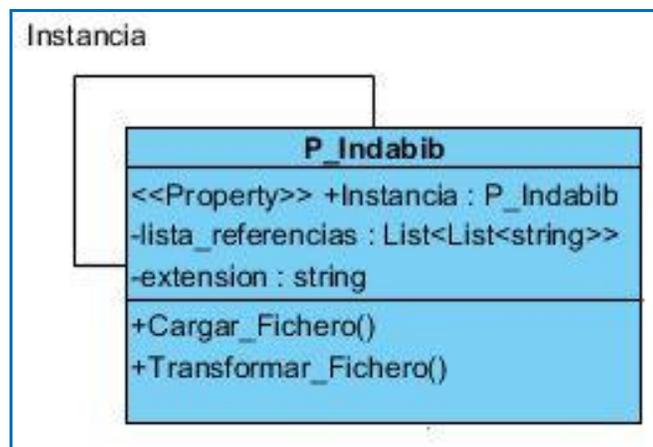


Figura 15: Ejemplo de aplicación del patrón Singleton.

Conclusiones

En este capítulo se definió una arquitectura base flexible, adaptada a las funcionalidades del sistema, que permitió realizar un diseño robusto y conforme a las buenas prácticas. Se estudiaron y aplicaron patrones de diseño para asignar responsabilidades a las clases, necesarios para la implementación. Además, se ilustraron los modelos que integran el modelo de diseño necesario para la construcción del sistema INDABIB.

CAPÍTULO 4. Implementación y Prueba.

4.1. Introducción

En el presente capítulo a partir de los resultados obtenidos del Modelado, se realiza el modelo de implementación. Además se aplican diversas pruebas para comprobar la calidad y funcionalidad del sistema informático propuesto.

4.2. Modelo de Implementación

El modelo de implementación describe como los elementos del modelo de diseño, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc. El modelo de implementación describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizado, y como dependen los componentes entre ellos. (18)

4.2.1. Diagrama de Componentes

“Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño”. (18)

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. A continuación se muestra el diagrama de componentes del sistema INDABIB:

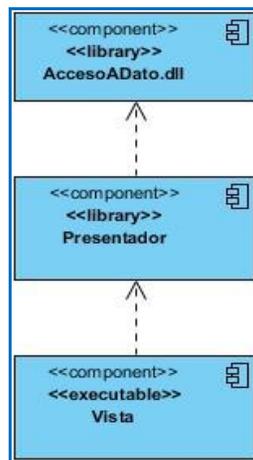


Figura 16: Diagrama de Componentes del sistema INDABIB

Capítulo 4. Implementación y Prueba

4.2.1.1. Descripción de los Componentes

- ❖ AccesoADatos.dll: Es una librería que contiene las clases en la cuales se almacena la información perdurable.
- ❖ Presentador.dll: Es una librería que contiene las clases en las cuales se almacena todo el comportamiento lógico del sistema.
- ❖ Vistas: Este componente contiene las interfaces encargadas de la interacción con el usuario.

4.2.2. Diagrama de Despliegue

Un diagrama de despliegue modela la topología del hardware sobre el que se ejecuta un sistema. Muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Un nodo es un elemento físico que existe en tiempo de ejecución. Representa un recurso computacional que, por lo general, tiene memoria y capacidad de almacenamiento. Además, representa el despliegue físico de un componente. (18)

A continuación se muestra el diagrama de despliegue del sistema, el cual está compuesto por un nodo PC_Cliente que representa las estaciones de trabajo de los usuarios que utilizarán la aplicación. Dicho nodo necesita tener instalado el FrameWork .NET v 4.0, así como tener copiada la aplicación.

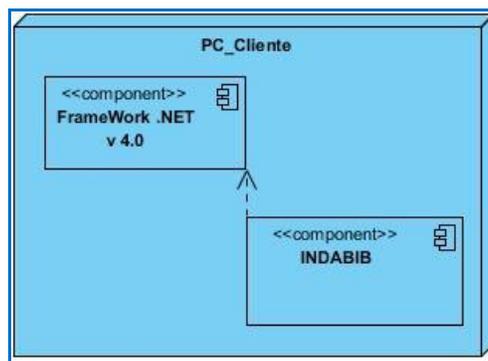


Figura 17: Diagrama de despliegue

4.3. Modelo de Prueba

Las pruebas de software son actividades con las cuales el sistema o parte de él es ejecutado bajo condiciones específicas. Los resultados obtenidos permiten realizar una

Capítulo 4. Implementación y Prueba

evaluación del mismo con el objetivo de valorar su calidad. Las pruebas constituyen un elemento fundamental para garantizar la calidad del software. (13)

4.3.1. Técnicas de Evaluación Dinámica

La aplicación de estas técnicas se conoce como pruebas de software. Estas pruebas de software se pueden agrupar en:

- ❖ Caja blanca o estructural: Se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa. Requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas de diseño o el código.
- ❖ Caja negra o funcional: Son las pruebas que se llevan a cabo sobre la interfaz de usuario, se centra principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Examina aspectos del modelo principalmente del sistema sin tener mucho en cuenta la estructura interna del software.

4.3.1.1. Pruebas de Caja Blanca

Pruebas de Unidad

Con el objetivo de comprobar el funcionamiento del sistema INDABIB se le aplicaron pruebas de unidad, enfocadas a los elementos más pequeños del sistema que brindan una funcionalidad, para comprobar que actúen correctamente. Esta prueba se realizó a los componentes del sistema comprobando que desempeñaran las funciones esperadas, además de las pruebas hechas por el equipo de desarrollo durante el proceso de implementación de las diferentes funcionalidades.

Para la aplicación de las pruebas unitarias se utilizó la funcionalidad Test que trae integrada el IDE de desarrollo utilizado en la implementación del sistema.

En la siguiente figura se muestra un ejemplo del código de las pruebas realizadas a la capa Presentador que es la encargada de modelar toda la lógica del negocio:

Capítulo 4. Implementación y Prueba

```
[TestMethod()]
public void Cargar_FicheroTest()
{
    P_Indabib_Accessor target = new P_Indabib_Accessor();
    StreamReader fichero = null;
    string tipo_Archivo = string.Empty;
    target.Cargar_Fichero(fichero, tipo_Archivo);
}

[TestMethod()]
public void Exportar_FicheroTest()
{
    P_Indabib_Accessor target = new P_Indabib_Accessor();
    string expected = string.Empty;
    string actual;
    actual = target.Exportar_Fichero();
    Assert.AreEqual(expected, actual);
}

public void Fin_Referencia_XMLTest()
{
    P_Indabib_Accessor target = new P_Indabib_Accessor();
    string line = string.Empty;
    bool expected = false;
    bool actual;
    actual = target.Fin_Referencia_XML(line);
    Assert.AreEqual(expected, actual);
}
```

Figura 18: Fragmento de código de las pruebas unitarias en la capa Presentador

Resultados de las Pruebas de Caja Blanca

Al sistema INDABIB se le realizaron dos iteraciones de pruebas unitarias. En la primera iteración se pudo validar el 83% de las funcionalidades, se encontraron 20 no conformidades que fueron resueltas. Una vez resuelta las no conformidades encontradas se aplicó una segunda iteración de pruebas en la cual no se detectaron no conformidades. La siguiente gráfica muestra la relación de estos datos:



Figura 19: Resultados de la pruebas unitarias del sistema INDABIB

Capítulo 4. Implementación y Prueba

En la siguiente figura se muestra como en la segunda iteración el 100% de las funcionalidades del sistema INDABIB devolvieron resultados satisfactorios:

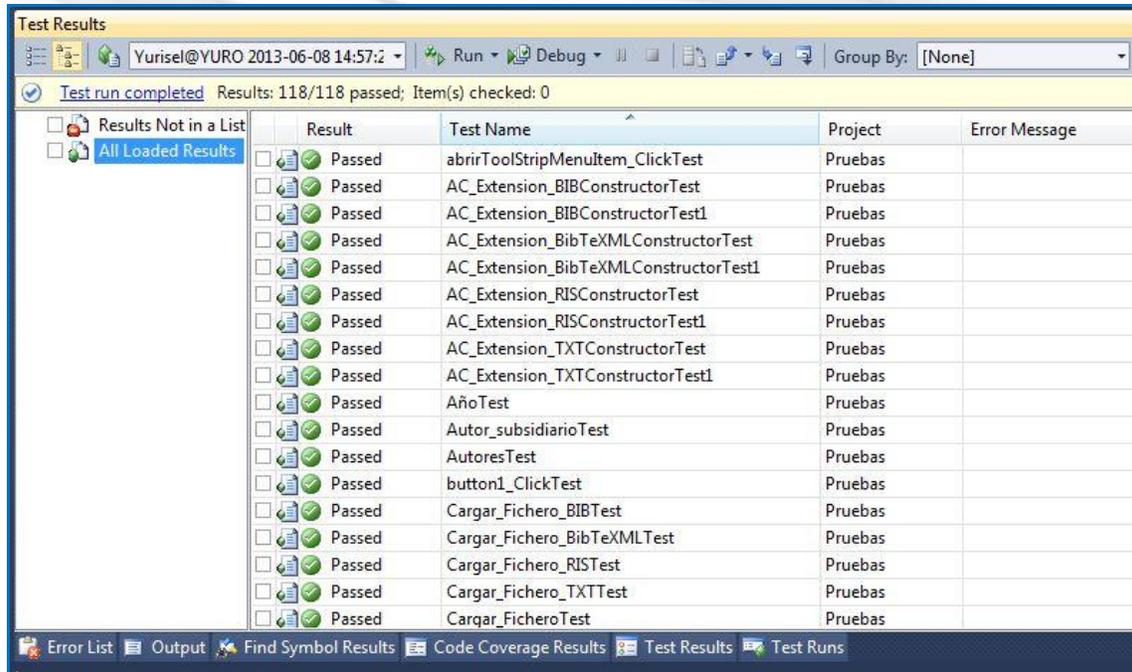


Figura 20: Resultado de la segunda iteración de las pruebas unitarias en el sistema INDABIB

4.3.1.2. Pruebas de Caja Negra

Descripción de los Casos de Pruebas

Las descripciones de casos de prueba que se utilizarán para las pruebas de la aplicación están diseñadas con el objetivo de verificar los requerimientos del usuario. A continuación se muestra la descripción del caso de prueba cargar base de datos bibliográfica.

Escenario	Descripción	Fichero	Respuesta del sistema	Flujo central
EC 1.1 Cargar exitosamente una base de datos bibliográficas.	Se carga un fichero.	V	El sistema muestra el mensaje: "La base de datos bibliográfica ha sido cargada satisfactoriamente."	<ol style="list-style-type: none"> 1. Seleccionar la opción "Cargar Base de Datos Bibliográficas" en el Menú Principal. 2. Introducir la base de datos bibliográfica. 3. Oprimir el botón "Aceptar".
EC 1.2 Cancelar	Se cancela la operación de cargar una base de datos bibliográfica.	NA	El sistema no carga el fichero y vuelve a la vista principal.	<ol style="list-style-type: none"> 1. Seleccionar la opción "Cargar Base de Datos Bibliográficas" en el Menú Principal. 2. Introducir la base de datos bibliográfica. 3. Oprimir el botón "Cancelar".

Capítulo 4. Implementación y Prueba

<p><i>EC 1.3 Extensión de las bases de datos bibliográficas incorrecta.</i></p>	<p><i>Se intenta cargar una base datos bibliográfica que posee una extensión que no está reconocida en el sistema.</i></p>	<p><i>I</i></p>	<p><i>El sistema muestra un mensaje “No se ha podido cargar el fichero porque no se encuentra en alguna de las siguientes extensiones: .RIS, TXT, BibTeX o BIB”. Posteriormente el sistema vuelve a la vista principal.</i></p>	<p><i>1. Seleccionar la opción “Cargar Base de Datos Bibliográficas” en el Menú Principal. 2. Introducir la base de datos bibliográfica. 3. Oprimir el botón “Aceptar”.</i></p>
<p><i>EC 1.4 Fichero inválido.</i></p>	<p><i>Se intenta cargar un fichero que posee alguna de las extensiones que están reconocidas en el sistema pero que no es una base de datos bibliográfica.</i></p>	<p><i>I</i></p>	<p><i>El sistema muestra un mensaje “El fichero que intenta cargar no es una base de datos bibliográfica o está dañado”. Posteriormente el sistema vuelve a la vista principal.</i></p>	<p><i>1. Seleccionar la opción “Cargar Base de Datos Bibliográficas” en el Menú Principal. 2. Introducir la base de datos bibliográfica. 3. Oprimir el botón “Aceptar”.</i></p>

Resultados de las Pruebas de Caja Negra

Al sistema INDABIB se le realizaron dos iteraciones de pruebas de caja negra. En la primera iteración se pudo validar un 90% de las funcionalidades que se desea que el sistema realice, se determinaron tres no conformidades que fueron resueltas. En la segunda iteración el 100% de las funcionalidades arrojaron resultados satisfactorios.

4.3.2. Entorno de las Pruebas

Las pruebas de software realizadas se llevaron a cabo en un entorno de pruebas con las siguientes características:

- ❖ Procesador: AMD C-50 CPU 1.00GH_z.
- ❖ Memoria RAM: 2GB.
- ❖ Sistema Operativo: Windows 7 Ultimate SP-1.

4.3.3. Resultados de las Pruebas

De forma general al sistema INDABIB se le realizaron dos iteraciones de pruebas. La primera iteración devolvió que en el sistema funcionaba correctamente en un 86.5% de las funcionalidades. Entre las no conformidades que se encontraron se pueden mencionar:

- ❖ Validación omitida de los datos nulos en algunas funcionalidades.
- ❖ Validación omitida del contenido del fichero cargado.

Capítulo 4. Implementación y Prueba

- ❖ Faltas de ortografía en algunos mensajes del sistema (tildes y mayúsculas incorrectas).
- ❖ Colores inapropiados en el diseño de las interfaces.

Las no conformidades encontradas en la primera iteración fueron resueltas satisfactoriamente. La segunda iteración de las pruebas arrojó que el sistema funciona correctamente en un 100% de sus funcionalidades. La siguiente gráfica muestra la relación entre los dos tipos de pruebas aplicados:



Figura 21: Resultado de las pruebas del sistema INDABIB

Conclusiones

Con la realización de este capítulo quedó definido el modelo de implementación del sistema INDABIB a través del diagrama de componentes. Se aplicaron dos iteraciones de pruebas y se solucionaron las no conformidades, resultando que la aplicación trabaja correctamente y cumple con los requisitos funcionales y no funcionales.

Conclusiones Generales

El presente trabajo de diploma concluye con el cumplimiento del objetivo general propuesto, obteniéndose un sistema capaz de exportar y convertir a un formato estándar la información de las bases de datos bibliográficas de diferentes extensiones.

Para ello:

- ❖ Se realizó un estudio del estado del arte de las tendencias actuales de los sistemas de integración de datos bibliográficos, así como de los conceptos fundamentales que fueron necesarios para la concepción del sistema INDABIB entre ellos el formato BibTeX.
- ❖ Se realizó la selección y descripción de las herramientas y metodología con las cuales se elaboraron los diversos diagramas y modelos que recogen la estructura lógica de los componentes utilizados en el desarrollo de la aplicación.
- ❖ Se realizó el modelado del sistema obteniendo los diferentes modelos y diagramas que guiaron la implementación del sistema INDABIB.
- ❖ Se implementaron los procesos de leer, transformar y exportar los datos bibliográficos aplicando los patrones de diseño seleccionados y respetando la arquitectura escogida en el modelado del sistema que permitió reducir dependencias entre artefactos situándolos en capas lógicas.
- ❖ Se le realizaron pruebas al sistema, se recolectaron y solucionaron las no conformidades.

Con el diseño y desarrollo de INDABIB se resolvió la siguiente problemática: ¿Cómo garantizar que el sistema Sistrada NOYSE sea capaz de reconocer las bases de datos bibliográficas con extensión RIS, TXT, BibTeXXML o BIB? Por lo que se puede concluir que se dio cumplimiento al objetivo trazado: Desarrollar un sistema que sea capaz de exportar y convertir a un formato estándar la información de las bases de datos de referencias bibliográficas de diferentes extensiones.

Recomendaciones

Para mejorar en pos de alcanzar la más alta calidad en el desarrollo de la presente solución informática se recomienda:

- ❖ Agregarle el soporte de nuevas extensiones con el objetivo de obtener una herramienta que sea capaz de procesar la mayor cantidad de formatos de almacenamiento de bases de datos bibliográficas.
- ❖ La posibilidad de que el sistema pueda obtener información de bases de datos online como DBLP, Google Scholar, MedLine, Citeseerx, entre otros.
- ❖ Adicionar la posibilidad de que el sistema pueda guardar los registros en una base de datos tipo SQL.
- ❖ Brindarle al sistema la posibilidad de editar los campos o registros de las referencias de la base de datos bibliográfica que se esté procesando.

Referencias Bibliográficas

Referencias Bibliográficas

1. ICIMAF - Instituto de Cibernética, Matemática y Física [En línea] [Citado: enero , 2013.] <http://www.icimaf.cu/>.
2. **Torres, Silvia, Boronino, Adina González y Irina Vavilova.** *Refererencias Bibliográficas: Guía basada en la norma APA.* Buenos Aires : s.n., 2012.
3. Biblioteca Central Rubén Matínez Villena. [En línea]. <http://biblioteca.dict.uh.cu/?q=node/858>.
4. Filetonic [En línea] <http://filetonic.com/file-extension/results/bib>.
5. Fileinfo.com [En línea] <http://www.fileinfo.com/extension/ris>.
6. BIBTEXML [En línea] <http://bibtexml.sourceforge.net>.
7. Fileinfo.com [En línea] <http://www.fileinfo.com/extension/txt>.
8. BibTex.org [En línea] <http://www.bibtex.org/>.
9. Real Academia Española - 22a edición. [En línea] <http://lema.rae.es/drae/>.
10. AUP The Agile Unified Process. [En línea] <http://www.cc.una.ac.cr/AUP/html/philosophies.html>.
11. **Jacobson, Ivar, Rumbaugh, James and Booch, Grady.** *El lenguaje Unificado de Modelado. Manual de Referencia.* Madrid : Addison Wesley, 2007.
12. Free Download Manager. *Free Download Manager.* [En línea] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UM_L_%28M%C3%8D%29_14720_p/.
13. UDA Universidad de Alcalá. [En línea] [Citado: diciembre , 2012.] <http://www2.uah.es/jcaceres/uploaded/teoria/LenguajesProgramacion/tema1.pdf>.
14. **Seco, Jose Antoni Gonzalez.** *El Lenguaje de Programación C#.*
15. LuaUf. [En línea] <http://luauf.com/2008/05/13/entornos-de-desarrollo/>.
16. Microsoft. [En línea] <http://www.microsoft.com/es-es/download/details.aspx?id=12752>.
17. Microsoft. [En línea] <http://www.microsoft.com/NET>.
18. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** *El Proceso Unificado de Desarrollo del Software.* [trad.] Salvador Sánchez, y otros. Español. Madrid: Addison Wesley, 2000. pág. 110, pág. 257-258, pág. 218, pág. 288. y pág. 260 ISBN: 84-7829-036-2.
19. **Kruchten, Philippe. November 1995.** "Architectural Blueprints--The 4+1 View Model of Software Architecture". S.I.: IEEE Software, Institute of Electrical and Electronics Engineers, November 1995. pág 42-50.
20. **Pressman, Roger S.** *Ingeniería del Software Un enfoque práctico.* Quinta Edición.
21. **Larman, Craig.** *UML y Patrones. Segunda Edición.*
22. **Potel, Mike. 1996.** "MVP:Model - View - Presenter The Taligent Programming Model for C++ and Java".
23. **Boeras Velázquez, M** "Aplicando el Método de Boehm y Turner". [En línea] <http://publicaciones.uci.cu/index.php/SC/article/view/891>.

Anexos

Anexo 1. Descripciones de CU del sistema INDABIB

Cargar Base de Datos Bibliográfica con Extensión RIS.

Objetivo	Cargar una base de datos de referencia bibliografía en el sistema con extensión RIS.
Actores	Sistema
Resumen	El CU comienza cuando el sistema identifica la extensión. Este lee sus datos y los separa por referencias bibliográficas. Cuando la base de datos bibliográfica haya sido descompuesta completamente en referencias bibliográficas termina el CU.
Complejidad	Baja
Precondiciones	La base de datos bibliográfica debe tener extensión RIS.
Postcondiciones	Se lee la base de datos bibliográfica.
Flujo de eventos	
Flujo Básico	
Actor	Sistema
	1. El sistema valida que el fichero sea una base de datos bibliográfica correspondiente a la extensión que posee.
	2. El sistema identifica las referencias bibliográfica contenidas en la una base de datos bibliográfica.
	3. El sistema crea una lista de referencias bibliográfica y almacena las referencias identificadas en la base de datos bibliográfica.
	4. El sistema verifica que la lista de referencias bibliográfica contenga al menos una referencia bibliográfica.
Flujos Alternos	
Flujo Alternativo 1b. Fichero inválido.	
Actor	Sistema
	1b.1. El sistema muestra un mensaje "El fichero que intenta cargar no es una base de datos bibliográfica o está dañado".

		<p>1b.2. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones:</p> <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Guardar Base de Datos Bibliográfica (deshabilitada). • Salir (Permite salir del sistema).
Flujo Alternativo 4b. Base de datos bibliográfica vacía.		
Actor	Sistema	
	4b.1. El sistema muestra un mensaje “La base de datos bibliográfica está vacía o dañada”.	
	<p>4b.2. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones:</p> <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Guardar Base de Datos Bibliográfica (deshabilitada). • Salir (Permite salir del sistema). 	
Relaciones	CU Incluidos	No
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		
No.		
Formatos de Entrada/Salida	No procede.	

Cargar Base de Datos Bibliográfica con Extensión TXT

Objetivo	Cargar una base de datos de referencia bibliografía en el sistema con extensión TXT.
Actores	Sistema
Resumen	El CU comienza cuando el sistema identifica la extensión. Este lee sus datos y los separa por referencias bibliográficas. Cuando la

	base de datos bibliográfica haya sido descompuesta completamente en referencias bibliográficas termina el CU.
Complejidad	Baja
Precondiciones	La base de datos bibliográfica debe tener extensión TXT.
Flujo de eventos	
Flujo Básico	
Actor	Sistema
	1. El sistema valida que el fichero sea una base de datos bibliográfica correspondiente a la extensión que posee.
	2. El sistema identifica las referencias bibliográfica contenidas en la una base de datos bibliográfica.
	3. El sistema crea una lista de referencias bibliográfica y almacena las referencias identificadas en la base de datos bibliográfica.
	4. El sistema verifica que la lista de referencias bibliográfica contenga al menos una referencia bibliográfica.
Flujos Alternos	
Flujo Alterno 1b. Fichero inválido.	
Actor	Sistema
	1b.1. El sistema muestra un mensaje “El fichero que intenta cargar no es una base de datos bibliográfica o está dañado”.
	1b.2. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Guardar Base de Datos Bibliográfica (deshabilitada). • Salir (Permite salir del sistema).

Flujo Alternativo 4b. Base de datos bibliográfica vacía.		
Actor	Sistema	
	4b.1. El sistema muestra un mensaje “La base de datos bibliográfica está vacía o está dañada”.	
	4b.2. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Guardar Base de Datos Bibliográfica (deshabilitada). • Salir (Permite salir del sistema). 	
Relaciones	CU Incluidos	No
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		
No.		
Formatos de Entrada/Salida	No procede.	

Cargar Base de Datos Bibliográfica con Extensión BIB

Objetivo	Cargar una base de datos de referencia bibliografía en el sistema con extensión BIB.
Actores	Sistema
Resumen	El CU comienza cuando el sistema identifica la extensión. Este lee sus datos y los separa por referencias bibliográficas. Cuando la base de datos bibliográfica haya sido descompuesta completamente en referencias bibliográficas termina el CU.
Complejidad	Baja
Precondiciones	La base de datos bibliográfica debe tener extensión BIB.
Flujo de eventos	
Flujo Básico	
Actor	Sistema
	1. El sistema valida que el fichero sea una base de datos bibliográfica correspondiente a la extensión que posee.
	2. El sistema identifica el inicio y el fin de las

	referencias bibliográfica contenidas en la una base de datos bibliográfica.
	3. El sistema lee las referencias bibliográfica contenidas en la una base de datos bibliográfica.
	4. El sistema crea una lista de referencias bibliográfica y almacena las referencias identificadas en la base de datos bibliográfica.
	5. El sistema verifica que la lista de referencias bibliográfica contenga al menos una referencia bibliográfica.
Flujos Alternos	
Flujo Alterno 1b. Fichero inválido.	
Actor	Sistema
	1b.1. El sistema muestra un mensaje “El fichero que intenta cargar no es una base de datos bibliográfica o está dañado”.
	1b.2. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Guardar Base de Datos Bibliográfica (deshabilitada). • Salir (Permite salir del sistema).
Flujo Alterno 5b. Base de datos bibliográfica vacía.	
Actor	Sistema
	5b.1. El sistema muestra un mensaje “La base de datos bibliográfica está vacía o está dañada”.
	5b.2. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de

	datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). <ul style="list-style-type: none"> • Guardar Base de Datos Bibliográfica (deshabilitada). • Salir (Permite salir del sistema). 	
Relaciones	CU Incluidos	No
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		
No.		
Formatos de Entrada/Salida	No procede.	

Cargar Base de Datos Bibliográfica con Extensión BibTeXML

Objetivo	Cargar una base de datos de referencia bibliografía en el sistema con extensión BibTeXML.
Actores	Sistema
Resumen	El CU comienza cuando el sistema identifica la extensión. Este lee sus datos y los separa por referencias bibliográficas. Cuando la base de datos bibliográfica haya sido descompuesta completamente en referencias bibliográficas termina el CU.
Complejidad	Baja
Precondiciones	La base de datos bibliográfica debe tener extensión BibTeXML.
Flujo de eventos	
Flujo Básico	
Actor	Sistema
	1. El sistema valida que el fichero sea una base de datos bibliográfica correspondiente a la extensión que posee.
	2. El sistema identifica el inicio y el fin de las referencias bibliográfica contenidas en la una base de datos bibliográfica.
	3. El sistema lee las referencias bibliográfica contenidas en la una base de datos bibliográfica.
	4. El sistema crea una lista de referencias bibliográfica y almacena las referencias identificadas en la base de datos bibliográfica.

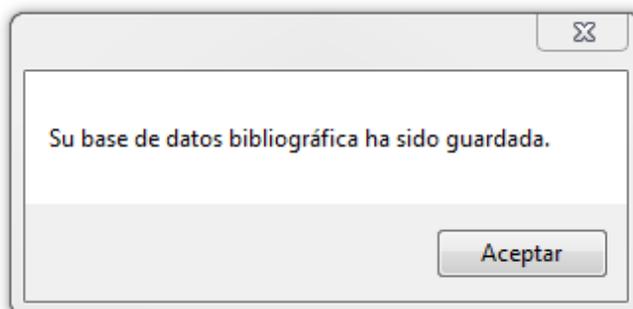
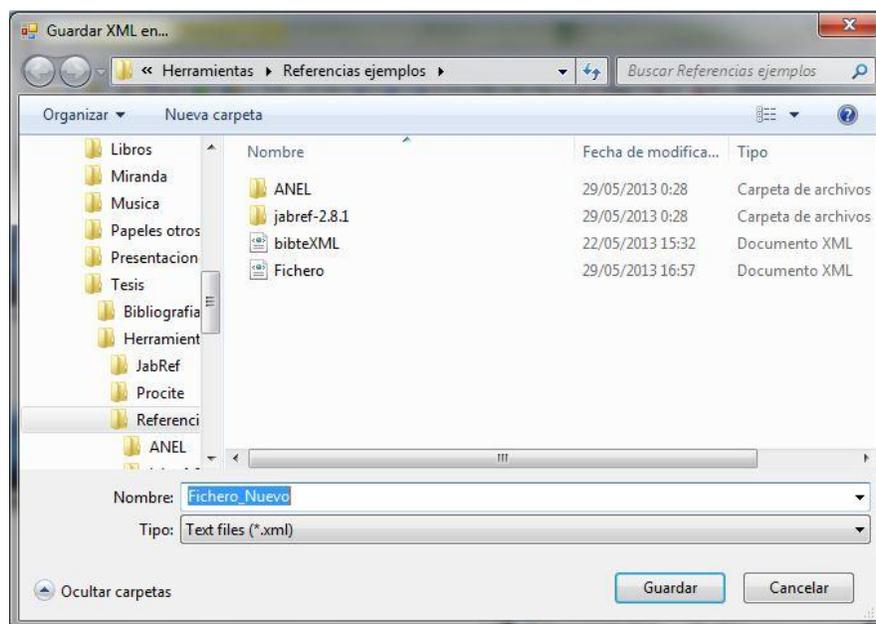
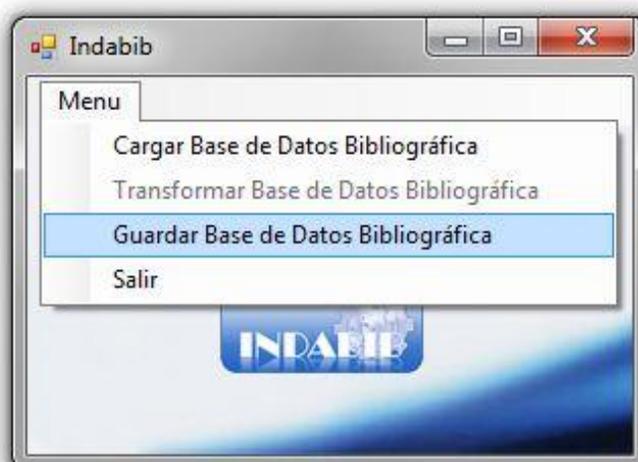
	5. El sistema verifica que la lista de referencias bibliográfica contenga al menos una referencia bibliográfica.	
Flujos Alternos		
Flujo Alterno 1b. Fichero inválido.		
Actor	Sistema	
	1b.1. El sistema muestra un mensaje “El fichero que intenta cargar no es una base de datos bibliográfica o está dañado”.	
	1b.2. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Guardar Base de Datos Bibliográfica (deshabilitada). • Salir (Permite salir del sistema). 	
Flujo Alterno 5b. Base de datos bibliográfica vacía.		
Actor	Sistema	
	5b.1. El sistema muestra un mensaje “La base de datos bibliográfica está vacía o está dañada”.	
	5b.2. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Guardar Base de Datos Bibliográfica (deshabilitada). • Salir (Permite salir del sistema). 	
Relaciones	CU Incluidos	No
	CU Extendidos	No

Prototipo elemental de interfaz gráfica de usuario	
No.	
Formatos de Entrada/Salida	No procede.

Guardar Base de Datos Bibliográfica

Objetivo	Guardar una base de datos de referencia bibliografía fuera del sistema.
Actores	Usuario
Resumen	El CU comienza cuando el usuario desea guardar una base de datos de referencias bibliográficas fuera del sistema. Este en dependencia de la extensión que posea la base de datos bibliografía exporta los datos bibliográficos. Cuando la base de datos bibliográfica haya sido guardada fuera del sistema termina el CU.
Complejidad	Media
Precondiciones	El sistema debe tener una base de datos bibliográfica cargada.
Postcondiciones	Se guarda un fichero xml fuera del sistema con la información de la base de datos bibliográfica cargada formateada.
Flujo de eventos	
Flujo Básico	
Actor	Sistema
	<p>1. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones:</p> <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Guardar Base de Datos Bibliográfica (Permite guardar una base de datos bibliográfica formateada con extensión XML). • Salir (Permite salir del sistema).
2. El usuario selecciona la opción "Guardar Base de Datos Bibliográfica".	<p>3. El sistema muestra la interfaz para escoger la dirección donde desea guardar el archivo. Y permite:</p>

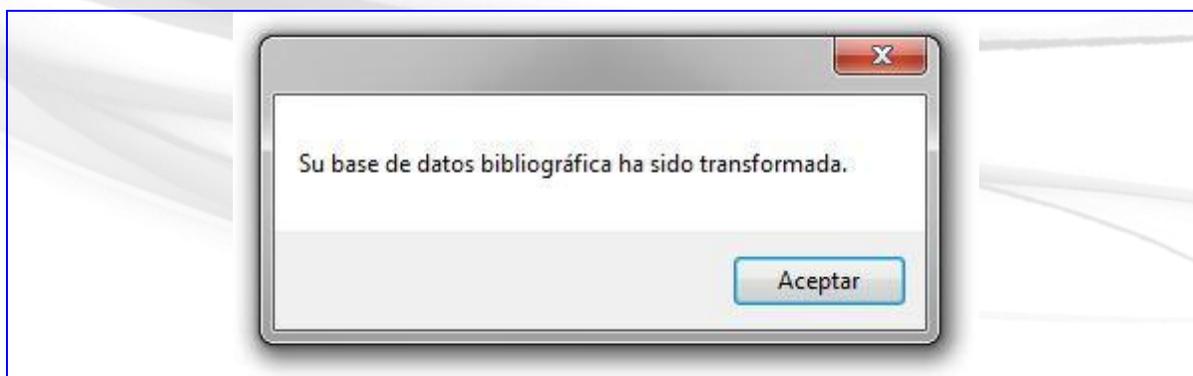
	<ul style="list-style-type: none"> • Introducir el nombre • Guardar • Cancelar la operación 	
4. El usuario introduce el nombre de la base de datos bibliográfica y oprime el botón "Guardar"	5. El sistema crea el fichero con extensión XML en la dirección seleccionada.	
	6. El sistema guarda las referencias formateadas en el fichero con extensión xml.	
	7. El sistema deshabilita en el menú de la interfaz principal la opción: Guardar Base de Datos Bibliográfica (Permite exportar una base datos bibliográfica formateada con extensión XML).	
Flujos Alternos		
Flujo Alternativo 2b. Seleccionar opción Salir		
Actor	Sistema	
	2b.1. El sistema finaliza.	
Flujo Alternativo 2c. Seleccionar opción Cargar Base de Datos Bibliográfica		
Actor	Sistema	
	2c.1. El sistema invoca <u>CU. Cargar Base de Datos Bibliográficas</u> .	
Flujo Alternativo 4b Oprime el botón Cancelar		
Actor	Sistema	
	4b.1 El sistema regresa al paso 1 del flujo básico.	
Flujo Alternativo 4c No introduce el nombre		
Actor	Sistema	
4c.1 Oprime el botón Guardar	4c.2 El sistema regresa al paso 3 del flujo básico.	
Relaciones	CU Incluidos	Exportar Base de Datos Bibliográfica
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		



Transformar Base de Datos Bibliográfica

Objetivo	Crear una base de datos bibliografía con las referencias bibliográficas formateada.
Actores	Sistema
Resumen	El CU comienza cuando el investigador desea transformar una base de datos de referencias bibliográficas. Este prepara las referencias, las transforma y las guarda en una nueva base de datos. Cuando la base de datos bibliográfica haya sido llenada completamente termina el CU.
Complejidad	Alta
Precondiciones	El sistema debe tener una base de datos bibliográfica cargada.
Flujo de eventos	
Flujo Básico	
Actor	Sistema
	1. El sistema verifica que la base de datos bibliográfica contenga alguna referencia para prepararla y formatearla.
	2. El sistema verifica el tipo de extensión que posee la base de datos bibliográfica.
	3. El sistema prepara cada una de las referencias bibliográficas de la lista de referencias para su transformación. Invocar <u>CU. Preparar Referencia Bibliográfica</u> ".
	4. El sistema formatea cada una de las referencias bibliográficas de la lista de referencias. Invocar <u>CU. Formatear Referencia Bibliográfica</u> ".
	5. El sistema crea una base de datos bibliográfica y guarda las referencias bibliográficas transformadas.

Flujos Alternos		
Flujo Alterno 1b. Base de datos bibliográfica vacía.		
Actor	Sistema	
	<p>1b.1. El sistema muestra un mensaje “El fichero que intenta cargar no es una base de datos bibliográfica o está dañado”.</p> <p>Y Permite:</p> <ul style="list-style-type: none"> • Aceptar la operación. 	
1b.2. El usuario oprime el botón Aceptar.	<p>1b.3. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones:</p> <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Salir (Permite cerrar la aplicación) 	
Relaciones	CU Incluidos	Preparar Referencia Bibliográfica. Formatear Referencia Bibliográfica.
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		
		



Preparar Referencia Bibliográfica

Objetivo	Identificar los campos y el valor de los mismos de una referencia bibliográfica.
Actores	Sistema
Resumen	El CU comienza cuando el sistema verifica que la base de datos bibliográfica contiene alguna referencia bibliográfica para ser preparada para la transformación. Este identifica los campos que posee, el valor de los mismos y los guarda en el sistema. Cuando de la referencia bibliográfica se hayan preparado todos los campos que posee termina el CU.
Complejidad	Media
Precondiciones	La base de datos bibliográfica debe tener referencias bibliográficas.
Flujo Normal de Eventos	
Sección "Preparar Referencia Bibliográfica con extensión RIS"	
Flujo Básico	
Actor	Sistema
	1. El sistema verifica que la base de datos bibliográfica contenga alguna referencia para ser preparada.
	2. El sistema crea una lista para los tipos de campos y otra con el valor de los mismos.
	3. El sistema separa los tipos de campos y valor de los mismos y los almacena.
	4. El sistema compara que los tipos de campos sean campos de la extensión RIS.

		5. El sistema guarda la información de los campos que pertenecen a la extensión RIS.
Flujos Alternos		
Flujo Alterno 1b. Base de datos bibliográfica vacía.		
Actor	Sistema	
	<p>1b.1. El sistema muestra un mensaje “El fichero que intenta cargar no es una base de datos bibliográfica o está dañado”.</p> <p>Y Permite:</p> <ul style="list-style-type: none"> • Aceptar la operación. 	
1b.2. El usuario oprime el botón Aceptar.	<p>1b.3. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones:</p> <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Salir (Permite cerrar la aplicación). 	
Relaciones	CU Incluidos	No.
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		
No		
Sección “Preparar Referencia Bibliográfica con extensión TXT”		
Flujo Básico		
Actor	Sistema	
	1. El sistema verifica que la base de datos bibliográfica contenga alguna referencia para ser preparada.	
	2. El sistema crea una lista para los tipos de campos y otra para en valor de los mismos.	
	3. El sistema identifica los tipos de campos.	
	4. El sistema verifica que los campos no estén vacíos.	
	5. El sistema identifica los tipos de campos y el valor de los mismos y los almacena.	

	6. El sistema compara que los tipos de registro sean campos de la extensión TXT.	
	6. El sistema guarda la información de los tipos de que pertenecen a la extensión TXT.	
Flujos Alternos		
Flujo Alterno 1b. Base de datos bibliográfica vacía.		
Actor	Sistema	
	1b.1. El sistema muestra un mensaje “El fichero que intenta cargar no es una base de datos bibliográfica o está dañado”. Y Permite: <ul style="list-style-type: none"> • Aceptar la operación. 	
1b.2. El usuario oprime el botón Aceptar.	1b.3. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Salir (Permite cerrar la aplicación). 	
Flujo Alterno 4b. Campos vacíos.		
Actor	Sistema	
	4b.1. El sistema le asigna nulo al valor del campo.	
	4b.2. El sistema regresa al paso 4 del flujo básico.	
Relaciones	CU Incluidos	No.
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		
No		
Sección “Preparar Referencia Bibliográfica con extensión BIB”		
Flujo Básico		
Actor	Sistema	
	1. El sistema verifica que la base de datos bibliográfica contenga alguna referencia para ser preparada.	
	2. El sistema crea una lista para los tipos de	

	campos y otra para en valor de los mismos.	
	3. El sistema identifica el tipo de registro y el valor del mismo y lo almacena.	
	4. El sistema identifica los tipos de campos y el valor de los mismos y los almacena.	
	5. El sistema compara que los tipos de registro sean campos de la extensión BIB.	
	6. El sistema guarda la información de los tipos de que pertenecen a la extensión BIB.	
Flujos Alternos		
Flujo Alterno 1b. Base de datos bibliográfica vacía.		
Actor	Sistema	
	1b.1. El sistema muestra un mensaje “El fichero que intenta cargar no es una base de datos bibliográfica o está dañado”. Y Permite: <ul style="list-style-type: none"> • Aceptar la operación. 	
1b.2. El usuario oprime el botón Aceptar.	1b.3. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Salir (Permite cerrar la aplicación). 	
Relaciones	CU Incluidos	No.
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		
No		
Sección “Preparar Referencia Bibliográfica con extensión BibTeXML”		
Flujo Básico		
Actor	Sistema	
	1. El sistema verifica que la base de datos bibliográfica contenga alguna referencia para ser preparada.	

Anexos

	2. El sistema crea una lista para los tipos de campos y otra para en valor de los mismos.	
	3. El sistema determina el identificador de l referencia bibliográfica y lo almacena.	
	4. El sistema identifica los tipos de campos y el valor de los mismos y los almacena.	
	5. El sistema compara que los tipos de registro sean campos de la extensión BIB.	
	6. El sistema guarda la información de los tipos de que pertenecen a la extensión BIB.	
Flujos Alternos		
Flujo Alterno 1b. Base de datos bibliográfica vacía.		
Actor	Sistema	
	1b.1. El sistema muestra un mensaje “El fichero que intenta cargar no es una base de datos bibliográfica o está dañado”. Y Permite: <ul style="list-style-type: none"> • Aceptar la operación. 	
1b.2. El usuario oprime el botón Aceptar.	1b.3. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Salir (Permite cerrar la aplicación). 	
Relaciones	CU Incluidos	No.
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		
No		

Formatear Referencia Bibliográfica

Objetivo	Darle el formato requerido para la salida a la referencia bibliográfica.
Actores	Sistema

Resumen	El CU comienza cuando el sistema termina de preparar una referencia bibliográfica. Este formatea los campos de la referencia bibliográfica. Cuando de la referencia bibliográfica se haya formateado todos los campos que posee termina el CU.
Complejidad	Media
Precondiciones	- El sistema debe tener una referencia bibliográfica preparada. - El tipo de registro de la referencia bibliográfica preparada debe ser válido para la extensión que posee la base de datos bibliográfica.
Flujo Normal de Eventos	
Sección “Formatear Referencia Bibliográfica con extensión RIS”	
Flujo Básico	
Actor	Sistema
	1. El sistema verifica que el tipo de registro no sea nulo.
	2. El sistema verifica que el tipo de registro sea válido para la extensión RIS.
	3. El sistema le asigna un identificador a la referencia bibliográfica.
	4. El sistema verifica que los campos no sean nulos y les da formato.
Flujos Alternos	
Flujo Alternativo 1b. Tipo de registro nulo.	
Actor	Sistema
	1b.1. El sistema muestra un mensaje “La base de datos bibliográfica está dañado”. Y Permite: <ul style="list-style-type: none"> • Aceptar la operación.

1b.2. El usuario oprime el botón Aceptar.	1b.3. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Salir (Permite cerrar la aplicación). 	
Flujo Alterno 2b. Tipo de registro inválido.		
Actor	Sistema	
	2b.1. El sistema no transforma la referencia bibliográfica.	
Relaciones	CU Incluidos	No.
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		
No		
Sección “Formatear Referencia Bibliográfica con extensión TXT”		
Flujo Básico		
Actor	Sistema	
	1. El sistema verifica que el tipo de registro no sea nulo.	
	2. El sistema verifica que el tipo de registro sea válido para la extensión TXT.	
	3. El sistema le asigna un identificador a la referencia bibliográfica.	
	4. El sistema verifica que los campos no sean nulos y les da formato.	
Flujos Alternos		
Flujo Alterno 1b. Tipo de registro nulo.		
Actor	Sistema	
	1b.1. El sistema muestra un mensaje “La base de datos bibliográfica está dañado”. Y Permite: <ul style="list-style-type: none"> • Aceptar la operación. 	

1b.2. El usuario oprime el botón Aceptar.		1b.3. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Salir (Permite cerrar la aplicación). 	
Flujo Alterno 2b. Tipo de registro inválido.			
Actor		Sistema	
		2b.1. El sistema no transforma la referencia bibliográfica.	
Relaciones	CU Incluidos	No.	
	CU Extendidos	No	
Prototipo elemental de interfaz gráfica de usuario			
No			
Sección “Formatear Referencia Bibliográfica con extensión BIB”			
Flujo Básico			
Actor		Sistema	
		1. El sistema verifica que el tipo de registro no sea nulo.	
		2. El sistema verifica que el tipo de registro sea válido para la extensión BIB.	
		3. El sistema le asigna un identificador a la referencia bibliográfica.	
		4. El sistema verifica que los campos no sean nulos y les da formato.	
Flujos Alternos			
Flujo Alterno 1b. Tipo de registro nulo.			
Actor		Sistema	
		1b.1. El sistema muestra un mensaje “La base de datos bibliográfica está dañado”. Y Permite: <ul style="list-style-type: none"> • Aceptar la operación. 	

1b.2. El usuario oprime el botón Aceptar.		1b.3. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeXML o BIB). • Salir (Permite cerrar la aplicación). 	
Flujo Alterno 2b. Tipo de registro inválido.			
Actor		Sistema	
		2b.1. El sistema no transforma la referencia bibliográfica.	
Relaciones	CU Incluidos	No.	
	CU Extendidos	No	
Prototipo elemental de interfaz gráfica de usuario			
No			
Sección “Formatear Referencia Bibliográfica con extensión BibTeXML”			
Flujo Básico			
Actor		Sistema	
		1. El sistema verifica que el tipo de registro no sea nulo.	
		2. El sistema verifica que el tipo de registro sea válido para la extensión BibTeXML.	
		3. El sistema verifica que los campos no sean nulos y les da formato.	
Flujos Alternos			
Flujo Alterno 1b. Tipo de registro nulo.			
Actor		Sistema	
		1b.1. El sistema muestra un mensaje “La base de datos bibliográfica está dañado”. Y Permite: <ul style="list-style-type: none"> • Aceptar la operación. 	

1b.2. El usuario oprime el botón Aceptar.	1b.3. El sistema muestra una interfaz con un Menú que contiene las siguientes opciones: <ul style="list-style-type: none"> • Cargar Base de Datos Bibliográfica (Permite cargar en el sistema una base de datos bibliográfica con extensión RIS, TXT, BibTeX o BIB). • Salir (Permite cerrar la aplicación). 	
Flujo Alternativo 2b. Tipo de registro inválido.		
Actor	Sistema	
	2b.1. El sistema no transforma la referencia bibliográfica.	
Relaciones	CU Incluidos	No.
	CU Extendidos	No
Prototipo elemental de interfaz gráfica de usuario		
No		

Anexo 2: Caso de Prueba del sistema INDABIB

Guardar Base de Datos Bibliográfica.

Escenario	Descripción	Dirección	Nombre	Respuesta del sistema	Flujo central
EC 1.1 Guardar exitosamente una base de datos bibliográfica.	Se exportan a un xml una base de datos bibliográfica cargada y formateada en el sistema.	V	V	El sistema muestra un mensaje: "La base de datos bibliográfica ha sido guardada."	<ol style="list-style-type: none"> 1. Seleccionar la opción "Guardar Base de Datos Bibliográficas" en el Menú Principal. 2. Seleccionar la dirección donde se desea guardar el fichero. 3. Insertar nombre con el que va a guardar el fichero. 4. Oprimir el botón "Aceptar".
EC 1.2 Cancelar	Se cancela la operación de guardar una base de datos bibliográfica.	NA	NA	El sistema vuelve a la vista principal.	<ol style="list-style-type: none"> 1. Seleccionar la opción "Guardar Base de Datos Bibliográficas" en el Menú Principal. 2. Seleccionar la dirección donde desea guardar el fichero. 3. Insertar nombre con el que va a guardar el fichero. 4. Oprimir el botón "Cancelar".

Transformar Base de Datos Bibliográfica.

Escenario	Descripción	Respuesta del sistema	Flujo central
<i>EC 1.1 Transformar exitosamente una base de datos bibliográfica.</i>	<i>Se transforma una base de datos bibliográfica cargada al formato determinado.</i>	<i>El sistema muestra un mensaje: “La base de datos bibliográfica ha sido transformada.”</i>	<i>1. Seleccionar la opción “Transformar Base de Datos Bibliográficas” en el Menú Principal.</i>