

# Universidad de las Ciencias Informáticas

Facultad 4



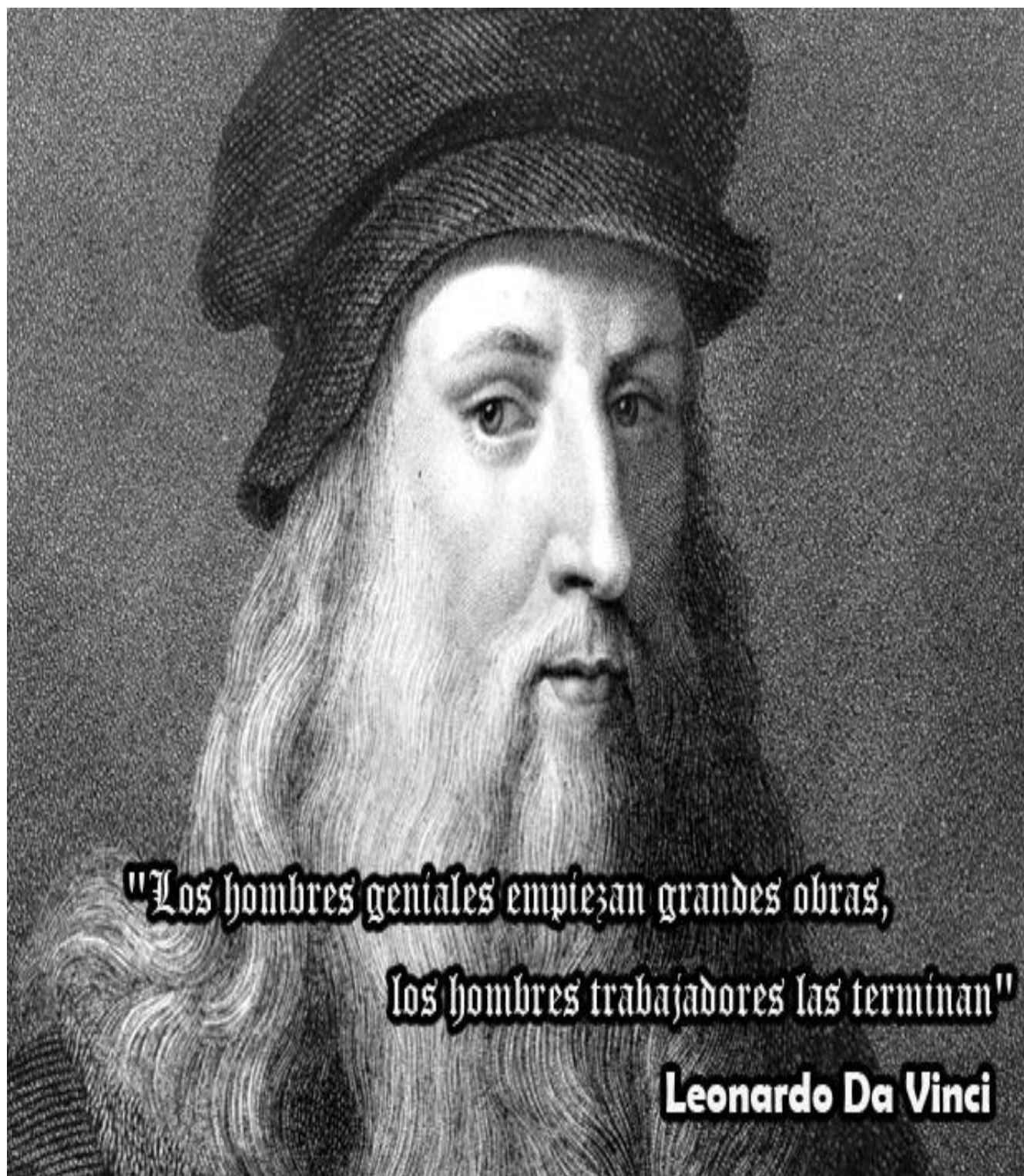
**Título:** *Portal del Centro de Recursos para el Aprendizaje y la Investigación de la Facultad de Economía de la Universidad de La Habana*

**Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas**

**Autores:** *Orlando Grabiél Toledano López  
Elias Bello Camps*

**Tutores:** *MSc. Mailin Carballosa Infante  
Ing. Yordankis Matos López*

*“Año 57 de la Revolución”  
La Habana, junio 2015*



**"Los hombres geniales empiezan grandes obras,  
los hombres trabajadores las terminan"**

**Leonardo Da Vinci**

## **Declaración de Autoría**

Declaramos ser autores del presente trabajo y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas de hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Orlando Grabiél Toledano López

Elias Bello Camps

---

Firma del autor

---

Firma del autor

MSc. Mailin Carballosa Infante

Ing. Yordankis Matos López

---

Firma del tutor

---

Firma del tutor

## **Dedicatoria**

### *De Orlando:*

*En la vida hay personas que siempre están presentes en los momentos más importantes y críticos de nuestra vida, seríamos ingratos si olvidáramos solo un segundo de sus acciones desinteresadas.*

*A mi mamá, por ser una mujer excepcional y por inspirarme confianza.*

*A mi papá, por transmitirme la fuerza que necesito día a día, por ser luz, ejemplo y guía.*

*A mi hermano Ismael y mis hermanas.*

*A mi novia, por su apoyo y amor incondicional.*

### *De Elías:*

*A mi tía Adís, por dedicarme sus últimos años, por haber convertido mi sueño en el suyo, y por brindarme todo el apoyo que necesitaba cuando me encontraba lejos de mis padres.*

## **Agradecimientos**

### *De Orlando:*

*A mi mamá, que me dio la vida y me disciplinó con amor para que hoy fuera el hombre que soy, gracias también por enseñarme el valor del trabajo y el esfuerzo por alcanzar una meta. Por siempre levantarte temprano para que todo esté en orden, por cuidarme siempre y darme lo que no tenías en momentos muy difíciles que vivimos.*

*A mi papá, por su sabiduría, comprensión y guía en los momentos difíciles de mi vida, gracias por creer en mí y espero nunca defraudarte, eres de los hombres que enseñan a ver la vida de una forma que pocas personas la ven, y eso me lo has sabido enseñar. Gracias por ser el mejor padre y el mejor ejemplo para mí y mis hermanos.*

*A mi hermano Ismael, por siempre inspirarme a hacer mejor las cosas, por transmitirme la diligencia y la capacidad de adaptarme a cualquier circunstancia, por ser el hermano que siempre estuvo a mi lado y que siempre quiso lo mejor para mí.*

*A mi novia Yohana, por darme amor verdadero, ese que hoy en día escasea mucho y vale más que el propio oro, por ser mi confianza, mi apoyo y siempre estar a mi lado a pesar de los problemas.*

*A mi compañero de tesis, por tener confianza en mí, por la amistad que me ha brindado y por su paciencia, sé que trabajar con una persona como yo es todo un reto.*

*A mi amigo Osmar, por ser mi compañero de las mil pruebas o las mil batallas, por siempre acompañarme para estudiar y brindarme su amistad sincera aceptando mis defectos y virtudes.*

*A mis tutores Mailín y Yordankís, por su guía e incomparable ayuda, sin ustedes esto no hubiese sido posible, gracias por todo el tiempo que dedicaron y por su disponibilidad en todo momento.*

*A mis compañeros de aula, sépase que en cualquier lugar del mundo donde estén y aun cuando pasen muchos años, siempre les daré mi mano amiga.*

*A todas las personas que de una forma u otra dieron un poco de su tiempo y ayuda para lograr culminar esta obra.*

## **Agradecimientos**

### *De Elías:*

*A mi padre por ser mi ejemplo a seguir, en mi carrera y en la vida, por su apoyo incondicional.*

*A mi madre, por su amor, dedicación y sacrificio, por estar siempre a mi lado en los momentos difíciles.*

*A todas esas personas que forman parte de mi familia, que me quieren, y que han estado a mi lado, personas como mi tío Jose, mi tía Marta, Paula, Ana, Migdalia, Vivian, quienes de una forma u otra, me han dado todo lo que han podido, y que sé que hoy disfrutarán tanto como yo este logro. En especial agradecer, a mi tía Adis, por ser la persona más cercana estos últimos años, por darme su cariño, por convertir mi sueño en el suyo, por quererme tanto, gracias. En fin a toda mi familia que hoy siente satisfacción con este logro. También a los amigos que me han apoyado y han estado a mi lado, amigos como Arcel, por alentarme y ayudarme a transitar momentos duros, a otros como David, Alejandro, Osmar, Dennis, por apreciarme a pesar de mis defectos y por compartir conmigo su amistad. Por último a mis tutores y a mi compañero de tesis y amigo Orlando, quien ha batallado junto a mí esta última etapa de la carrera, la cual ha sido dura, pero ha valido la pena. Muchas gracias.*

## Resumen

El desarrollo vertiginoso de la ciencia y la técnica junto a la globalización del uso de las Tecnologías de la Información y las Comunicaciones, ha incrementado el proceso de generación del conocimiento, permitiendo que un gran número de bibliotecas universitarias en el mundo evolucionen hacia los Centros de Recursos para el Aprendizaje y la Investigación. Estos centros, brindan un grupo de servicios mediante un portal web, que se encarga de ser la primera línea de información, donde el usuario puede interactuar por primera vez con su universidad y el conocimiento. A raíz de esto, la Facultad de Economía de la Universidad de La Habana ha encaminado un proyecto para poner en marcha dicha transformación, y requiere de un sistema que permita la gestión de los servicios que se brindarán en línea, al carecer de un espacio donde se publique información sobre la infraestructura de la universidad, donde se gestionen, recomienden y soliciten materiales para llevar a cabo el estudio y las investigaciones. El presente trabajo describe el desarrollo de un portal web que proveerá los servicios requeridos, el sistema fue desarrollado mediante la metodología AUP, se basa en el *framework* Vaadin y el lenguaje de programación Java. Se realizaron los flujos de Modelado, Implementación y Prueba, indicados por la metodología elegida, obteniéndose un sistema que cumple con los requerimientos funcionales y no funcionales. Con el *software* desarrollado, se permite la gestión de contenidos, materiales y servicios, facilitando un acceso a estos de forma remota, rápida y sencilla.

**Palabras clave:** Centro de Recursos para el Aprendizaje y la Investigación, portal web.



## Índice

Introducción .....	1
Capítulo 1. Fundamentación teórica .....	5
1.1    Conceptos relacionados con el dominio del problema .....	5
1.2    Área CRAI en Cuba y el mundo .....	7
1.2.1    Ejemplos de soluciones similares en el mundo .....	7
1.2.2    Situación de los CRAI en Cuba .....	8
1.3    Descripción las tecnologías, herramientas y metodologías a utilizar .....	9
1.3.1    Aplicación web .....	9
1.3.2    Metodología de desarrollo .....	10
1.3.3    Gestor de base de datos .....	13
1.3.4    Lenguaje de programación .....	16
1.3.5    Framework de programación .....	17
1.3.6    Entorno de desarrollo .....	18
1.3.7    Servidores de aplicaciones web .....	19
1.3.8    Lenguaje de modelado .....	19
1.4    Patrón de arquitectura .....	20
1.5    Conclusiones Parciales .....	21
Capítulo 2. Propuesta de solución .....	23
2.1    Modelo de dominio .....	23
2.1.1    Conceptos del dominio .....	23
2.1.2    Diagrama del modelo de dominio .....	24
2.2    Requerimientos del sistema .....	25
2.3    Requerimientos no funcionales .....	27

2.4	Modelo de casos de uso del sistema .....	28
2.4.1	Descripción de los actores del sistema .....	29
2.4.2	Diagrama de casos de uso del sistema .....	29
2.4.3	Patrones de casos de uso .....	30
2.2.4	Descripción textual de los casos de uso del sistema .....	31
2.5	Conclusiones parciales .....	37
Capítulo 3. Desarrollo de la propuesta de solución .....		38
3.1	Patrones de diseño .....	38
3.1.1	Patrones estructurales .....	40
3.2	Modelado de diseño .....	40
3.2.1	Diagramas de clases del diseño .....	40
3.2.2	Diagramas de secuencia .....	42
3.2.3	Vista de despliegue .....	43
3.2.4	Diagramas de componentes .....	43
3.3	Pruebas de software .....	44
3.3.1	Niveles de prueba .....	44
3.3.2	Métodos de prueba .....	46
3.3.3	Partición equivalente .....	46
3.3.4	Diseño de casos de prueba .....	46
3.3.5	Resultados de las pruebas de caja negra .....	51
3.3.6	Pruebas de rendimiento .....	51
3.4	Conclusiones parciales .....	54
Conclusiones generales .....		55
Recomendaciones .....		56

Referencias Bibliográficas.....	57
Bibliografía.....	62
Glosario de términos.....	64

## Introducción

El uso de las modernas tecnologías ha provocado un gran impacto en el desarrollo de la sociedad, permitiendo que los avances científicos tengan un crecimiento más acelerado que en épocas anteriores, todo esto se ha conseguido gracias a la incorporación de las Tecnologías de la Información y las Comunicaciones (TIC) en el quehacer innovador del hombre. Hoy en día está muy difundido el uso del *Internet* a nivel mundial, donde se posibilita la comunicación y el intercambio de información entre personas sin importar la distancia donde estas se encuentren, haciendo posible explotar al máximo las ventajas de las nuevas tecnologías.

La red de redes, como muchos le llaman hoy al *Internet*, posee un potencial enorme de servicios y oportunidades provistas mediante aplicaciones web, portales, blogs, foros, wikis, motores de búsqueda, entre otros sistemas que se encargan de satisfacer la creciente demanda informacional que tienen los usuarios en la era digital. Esta a su vez brinda nuevos instrumentos de apoyo al proceso docente educativo, incrementando los conocimientos y habilidades de los estudiantes, eliminando los monopolios de información que se imponen con los métodos tradicionales y eleva la actividad investigativa, imponiendo la creatividad sobre la reproducción, sin sustituir la necesaria labor del educador.

Con el avance logrado por las nuevas tecnologías, las bibliotecas han tenido que transformarse, al evidenciarse la necesidad de ofrecer nuevos servicios y nuevas formas de información para la sociedad de hoy, la cual es muy distinta a la de los tiempos donde surgieron estas instituciones, pues desde sus inicios constituían el centro del saber de las universidades, y se centraban más en la protección de sus materiales y menos en los servicios, hasta el siglo XIII donde se desechó este viejo modelo, transformándose en el núcleo intelectual de la universidad. (1)

Los Centros de Recursos para el Aprendizaje y la Investigación (CRAI), constituyen una excelente innovación para las bibliotecas universitarias donde estas dejan de ser un simple lugar para la consulta y préstamos de materiales para dar apoyo a la docencia, sino que se convierten en un centro activo para el aprendizaje con el fin de incrementar las competencias de los usuarios y sus necesidades potenciales en múltiples aspectos.

Cuba hace grandes esfuerzos en la búsqueda de la informatización de la sociedad, este proceso incluye varios renglones y esferas, incluyendo la educación superior. Se han creado diferentes centros productivos

a lo largo del país los cuales buscan llevar estas potencialidades a las principales instituciones productivas y centros de enseñanza. Con la creación de la Universidad de las Ciencias Informáticas (UCI) en el año 2002, se han impulsado muchos de estos proyectos, consiguiéndose resultados favorables para la sociedad. (2)

En este contexto, la Facultad de Economía de la Universidad de La Habana está desarrollando una investigación encaminada a convertir su biblioteca en un CRAI, al evidenciarse deficiencias en los servicios que prestan en línea:

- Existe total ausencia de instrumentos que sirvan de apoyo al aprendizaje autónomo; estos instrumentos constituyen guías, tutoriales y materiales que permiten el desarrollo de competencias informativas en los usuarios preparándolo para la producción de conocimientos (3).
- No existe información referente a la infraestructura universitaria, que sirva de guía a personas interesadas en llevar a cabo alguna investigación.
- No se brinda un servicio de recomendación de materiales a los usuarios según sus temáticas de interés.
- Se carece de un espacio en línea donde los usuarios puedan solicitar fuentes bibliográficas, que no existan en la biblioteca.

A partir de lo planteado como situación problemática se define el siguiente **problema resolver**: ¿Cómo mejorar la gestión de los servicios en línea para los usuarios de la biblioteca de la Facultad de Economía de la Universidad de La Habana, mediante el uso de las tecnologías y a través de la web?

El diseño de la investigación permite declarar como **objeto de estudio** los procesos y servicios que brindan los CRAI, como **campo de acción** los portales web para CRAI en línea.

Se define como **objetivo general**: Desarrollar un portal web que brinde los servicios de un CRAI en línea en la biblioteca de la Facultad de Economía de la Universidad de La Habana, mediante el cual los usuarios puedan acceder a materiales de diferentes temáticas de interés.

Como **idea a defender** se plantea que si se desarrolla un portal web que brinde los servicios de un CRAI en línea, se incrementarán las competencias informativas de los usuarios y se elevará la capacidad de la biblioteca universitaria para atender las necesidades de los mismos.

Para dar cumplimiento al **objetivo general** se plantean las siguientes **objetivos específicos**:

- Analizar elementos teóricos y principales tendencias del desarrollo web en la actualidad.
- Analizar las diferentes funcionalidades que brindan los CRAI en línea, implantados en el mundo.
- Definir las tecnologías y herramientas necesarias para el desarrollo de la propuesta de solución.
- Implementar las funcionalidades de la propuesta de solución.

Para el desarrollo de la investigación se emplearon los siguientes métodos de la investigación científica:

### **Métodos teóricos:**

- Analítico-Sintético: El uso del mismo permitió analizar los principales servicios que debe brindar un CRAI en línea, y mediante su análisis se pudieron determinar las funcionalidades esenciales a implementar.
- Modelación: Es utilizado para realizar los diagramas necesarios que modelarán la solución.

### **Métodos empíricos:**

- Observación: Se utilizó para obtener información de las necesidades existentes a lo largo del desarrollo del proyecto.

La presente investigación está conformada por tres capítulos, quedando de la siguiente manera:

**Capítulo 1. Fundamentación Teórica:** En este capítulo se exponen los elementos teóricos utilizados en la investigación, describiendo además las tecnologías, metodologías, herramientas y el lenguaje de programación utilizado en el desarrollo de la solución, así como los principales conceptos involucrados, para una mejor comprensión.

**Capítulo 2. Propuesta de solución:** En este capítulo se exponen los elementos que permiten describir la propuesta de solución, tales como: modelo de dominio, requerimientos funcionales y no funcionales, diagramas de casos de uso y la descripción textual de cada uno de ellos incluyendo los prototipos de interfaz de usuario para lograr un entendimiento claro del sistema a desarrollar.

**Capítulo 3. Construcción de la propuesta de solución:** En este apartado se realiza el modelado de diseño de las funcionalidades y la implementación de la propuesta de solución, donde se obtiene un portal web que

## INTRODUCCIÓN

brinda los servicios requeridos por el cliente. Se describen además las pruebas realizadas al sistema, una vez que concluye la implementación, para asegurar que este cumple con las especificaciones requeridas.

## CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

El presente capítulo expone los elementos teóricos y conceptuales asociados al problema a resolver, se explican las principales terminologías relacionadas con el área CRAI, se caracterizan los sistemas similares implantados en universidades de Cuba y el mundo, incluyendo sus principales servicios. También se describen las diferentes herramientas de software, metodologías y lenguaje de programación que se emplearán en el proceso de diseño y desarrollo de la solución planteada, así como el uso de patrones de arquitectura.

### 1.1 Conceptos relacionados con el dominio del problema

Para la comprensión total del trabajo se debe tener en cuenta una serie de conceptos y definiciones asociadas al dominio del problema.

**Alfabetización informacional (ALFIN):** Se basa en promover el uso responsable, crítico, creativo y autónomo de la información para la generación del conocimiento individual y colectivo, brindando herramientas para aprender, enseñar e investigar. Es la respuesta de la bibliotecología al nuevo escenario que suponen las TIC, para buscar el desarrollo de competencias informativas en los usuarios. (4)

El desarrollo de estas competencias requiere que las escuelas se conviertan en un espacio flexible, donde se formen personas con capacidad de respuesta en un mundo cambiante y donde la herramienta principal sea el conocimiento, siendo necesario su constante actualización. (5)

Existe otro concepto que guarda una estrecha relación con el analizado anteriormente, pero está enfocado en mejorar las competencias de los usuarios en cuanto al uso de las TIC. A este concepto se le conoce como alfabetización digital.

**Alfabetización digital:** Es una parte de la alfabetización informacional, que muestra el manejo de las infraestructuras técnicas, habilidades para usar las aplicaciones informáticas, conceptos fundamentales sobre redes e información, habilidades intelectuales para manejar la tecnología de la información como paso previo para manejar la información a la que estas herramientas permiten el acceso. (3)



# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

**Biblioteca universitaria:** "Un centro de recursos para el aprendizaje, la docencia, la investigación y las actividades relacionadas con el funcionamiento y la gestión de la Universidad/institución en su conjunto", según García Recha, Candil Gutiérrez y Bustamante Rodríguez (2004) (6). Su función está pensada en satisfacer las necesidades de información a los estudiantes y profesores que radican en una universidad.

**Biblioteca virtual:** Es una biblioteca que no tiene existencia física y la totalidad de la información se encuentra en formato electrónico. El usuario tiene la sensación de estar físicamente en la biblioteca, ilusión que se obtiene a partir de una simulación de la realidad. Las ventajas que ofrece se basan en la premisa de que todos los usuarios tienen las mismas posibilidades de acceso a los recursos de las bibliotecas, ya que poseen recursos permanentes que pueden ser accedidos desde cualquier parte y en cualquier momento. (7)

**Catálogo electrónico:** Este concepto constituye un servicio que permite el acceso en línea a todos los materiales en diferentes formatos de la biblioteca y contiene información sobre las colecciones disponibles, permite el acceso en muchos casos a bases de datos propias de la biblioteca o de otras instituciones incluyendo diferentes formas de búsqueda. (8)

**Recursos electrónicos:** Un conjunto de materiales compuestos por datos (información numérica, textual, gráficos, imágenes y sonido o una combinación de ellos); programas (instrucciones o rutinas para la ejecución de determinadas tareas, como el procesamiento de los datos) y/o una combinación de ambos, datos y programas electrónicos, servicios en línea y multimedia interactivos, codificados para ser manipulados por ordenador. (9)

**Libro electrónico:** El libro electrónico es la versión digitalizada de un libro que se publicará en la web o en cualquier otro tipo de formato electrónico. Se le puede denominar además como libro digital, ciberlibro, *e-book* entre otros términos. (10)

**Centro de Recursos para el Aprendizaje y la Investigación (CRAI):** Un CRAI o Centro de Recursos para el Aprendizaje y la Investigación, es un entorno dinámico en el que se integran todos los recursos que dan soporte al aprendizaje y la investigación en la universidad, y en los modelos españoles lo definen como un servicio universitario que tiene como objetivo ayudar a los profesores y a los estudiantes a facilitar las actividades de aprendizaje, de formación, de gestión y de resolución de problemas sean técnicos, metodológicos y de conocimiento en el acceso y uso de la información. (11)

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

Se debe tener en cuenta que este concepto es un nombre con el que REBUIN (Red de Bibliotecas Universitarias Españolas) bautizó a los **LRC (*Learning Reseach Centres*)**. (11) Estos se entienden como un modelo de universidades occidentales donde la biblioteca tiene un proceso central y neurálgico en la actividad educativa.

## 1.2 Área CRAI en Cuba y el mundo

### 1.2.1 Ejemplos de soluciones similares en el mundo

En el mundo existen varios portales que proveen servicios informáticos a las bibliotecas universitarias convertidas en CRAI. Estos se encuentran principalmente en instituciones prestigiosas de la región europea y norteamericana, debido al alto índice de desarrollo que existe en las investigaciones sobre este campo. Se pueden citar los siguientes ejemplos de instituciones que ya poseen dicha solución: Universidad de Bath, Universidad de Bournemouth, Universidad de Cambridge y la Universidad Carlos III.

**CRAI de la Universidad de Bath:** Es una biblioteca y centro para el aprendizaje, pertenece al Reino Unido y ofrece los siguientes servicios: Catálogo electrónico, bases de datos, libros electrónicos, salas de conferencias, directorios de expertos, oferta de postgrado, cursos de habilidades informacionales, servicio para personas con capacidades diferentes, asesoría sobre los recursos electrónicos, información de materiales docentes existentes en la biblioteca, últimas noticias, Programa de conferencias públicas, Instalaciones deportivas y guía completa de universidades. (5)

**CRAI de la Universidad de Bournemouth:** Portal que ofrece servicios de biblioteca y centro de aprendizaje para la Universidad de Bournemouth en el Reino Unido. Ofrece los siguientes servicios: Catálogo electrónico, libros electrónicos, bases de datos, noticias y eventos, lista de investigadores, catálogo, sala para las colecciones especiales, red inalámbrica, puntos de conexión para el acceso a los recursos electrónicos desde ordenadores portátiles personales, salas de estudio, zona de estudio de postgrado, zonas para el estudio en silencio y servicios para personas con necesidades especiales. (5)

**CRAI de la Universidad de Cambridge:** Portal para el aprendizaje y la investigación de la prestigiosa Universidad de Cambridge en el Reino Unido, se destaca por ofrecer servicio de imágenes, área de asistencia tecnológica, préstamos interbibliotecario, catálogo electrónico, recursos electrónicos, aulas, salas de conferencia, biblioteca digital y acceso abierto. Posee una extensa colección de libros, revistas, mapas, fotografías y grabaciones sonoras. (5)

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

**CRAI de la Universidad Carlos III:** Centro de recursos para el Aprendizaje de la Universidad Carlos III en España. Trabaja las 24 horas del día de lunes a viernes y ofrece los siguientes servicios: direcciones y teléfonos, red inalámbrica, acceso remoto, préstamo interbibliotecario, préstamo intercampus, salas de trabajo en grupo, gestores bibliográfico (*Refworks*), préstamo de portátiles, aulas CRAI, auto préstamo, puestos para personas con discapacidad, correo electrónico, atención telefónica, guías y tutoriales, propiedad intelectual y derechos de autor, préstamo domiciliario, equipamiento para el aprendizaje de idiomas, equipos audiovisuales, autoservicio de fotocopias, impresión y escáner, bases de datos, revistas electrónicas, libros electrónicos, referencia digital y prensa diaria. (5)

A partir del estudio de estas soluciones, se puede constatar que las mismas persiguen el objetivo común de proporcionarle al usuario el acceso a materiales y otros recursos digitales de la propia biblioteca, por lo que estos sistemas permiten la mejora de los servicios que se prestan en la institución, al extender su disponibilidad en la web.

## 1.2.2 Situación de los CRAI en Cuba

Debido a los cambios que están atravesando las bibliotecas universitarias de Cuba desde el año 1959, el Consejo de Estado, en el año 2010, ha establecido el Decreto ley 271 al evidenciarse la necesidad de su ordenamiento y replanteo para el futuro. Este decreto establece que las bibliotecas de la República de Cuba deben constituir centros para el aprendizaje, la docencia, la investigación y las actividades relacionadas con el funcionamiento y la gestión de las universidades, constituido por los fondos bibliográficos, documentales, audiovisuales y digitales adquiridos por los diversos departamentos, centros y servicios, cualquiera que sea la forma de su adquisición, incluidos los legados, y aquellos donados a favor de la Universidad por otras instituciones. (12)

El establecimiento de este decreto ley muestra el interés del estado cubano de perfeccionar las bibliotecas universitarias y adaptarlas al nuevo contexto y refleja su necesaria transformación hacia un modelo de CRAI propio como futuro de estas instituciones. (12)

El desarrollo de este campo en Cuba está experimentando un proceso de surgimiento y evolución, se dice de esta manera debido a que solo existe una implementación de esta solución en la Universidad “Carlos Rafael Rodríguez” de Cienfuegos, donde se cuenta actualmente con un portal y una aplicación multimedia. La solución se logró implantar en dicha universidad mediante el uso de una metodología que se desarrolló

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

con el objetivo de implementar los CRAI en el sistema de enseñanza superior cubano. Esta metodología se sometió a evaluación por varios especialistas del tema, catalogándola como adecuada por las encuestas realizadas. (13)

**CRAI de la Universidad “Carlos Rafael Rodríguez”:** A partir de su implementación los estudiantes y profesores se beneficiaron de los siguientes servicios: Buró de información, Biblioteca virtual, Laboratorio de ALFIN, reprografía de documentos y tecnología educativa.

Con el estudio de soluciones similares en Cuba, se puede apreciar la baja aplicación práctica de estos sistemas en las instituciones universitarias, al existir en una única institución en el país. (13) Existe además la fortaleza de haberse creado una metodología que permite aplicarlos en el sistema educacional superior cubano, por lo que debe aprovechar este conocimiento en aras de extender su uso y poner en práctica lo planteado por el gobierno cubano en su Decreto ley 271.

## **1.3 Descripción las tecnologías, herramientas y metodologías a utilizar**

El desarrollo de las redes telemáticas y la internet hoy en día es muy amplio, la tendencia actual del mundo gira alrededor del empleo de aplicaciones web sobre las aplicaciones de escritorio, debido a que estas poseen una gran comunidad de desarrolladores, existe un incremento en la calidad y robustez de las herramientas que se utilizan para construirlas y están disponibles desde cualquier parte del mundo, además de ser el principal soporte informático para brindar servicios que utilizan los CRAI.

### **1.3.1 Aplicación web**

El uso de aplicaciones web ofrece a los usuarios finales grandes ventajas, estas le ahorran tiempo evitando la instalación de las mismas, ahorran espacio al estar alojadas en un servidor remoto destinado a atender múltiples peticiones, evitan los problemas de compatibilidad con los sistemas operativos, pues solo requieren de un navegador web para su ejecución y poseen en muchos casos servicios y funcionalidades que las hacen altamente colaborativas.

En los servidores, el despliegue de estas aplicaciones es relativamente sencillo, los clientes pueden accederlas mediante el empleo de un navegador web estándar, aunque en algunos casos se requiere de la instalación de complementos adicionales que son provistos por los desarrolladores de estos navegadores a través de los sitios de soporte. Existe un gran número de tecnologías y lenguajes de programación para su construcción; aunque su principal desventaja es que requieren de una conexión permanente a la red,

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

cada día se incrementan las necesidades de los usuarios, no solo buscan información, también desean tener a su disposición sistemas para satisfacer rápidamente alguna necesidad específica. Por otra parte las empresas obtienen grandes beneficios brindando estas soluciones a los clientes, por lo que su uso es altamente difundido. (14)

Por las características descritas anteriormente y las ventajas que traen consigo el uso de las aplicaciones web en entornos empresariales y otras instituciones, se hace necesario que la propuesta de solución sea un sistema de este tipo, debido a que la misma posibilitará una mayor colaboración e intercambio entre los usuarios y se extenderá el alcance de los servicios de la biblioteca universitaria, permitiendo atender múltiples necesidades de los usuarios de forma simultánea.

## **1.3.2 Metodología de desarrollo**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. (15) Esta le indica al equipo de desarrollo cómo hay que obtener los distintos productos parciales y finales. Con su empleo se realiza el seguimiento de uno o varios modelos del ciclo de vida del proceso de desarrollo de los sistemas, facilitando en gran medida el éxito en las entregas y la calidad del producto. Se elige una determinada metodología en dependencia de las características del entorno o proyecto en cuestión, pues no existe una metodología universal para todos los casos.

Para la elección de la metodología de desarrollo es necesario en primer orden analizar qué tipo de enfoque será utilizado en el proyecto, en dependencia de variables y elementos que predominan en el entorno. Existen dos enfoques: el prescriptivo o tradicional y el ágil. Luego de tener identificado uno de estos enfoques de forma correcta para el proyecto, ya se logra un paso de avance para la elección de la metodología más adecuada.

En la actualidad, existe un ambiente de negocio muy variable, donde las peticiones del cliente varían con facilidad y se hace necesario minimizar el tiempo en las entregas de los productos, por lo que el enfoque ágil prevalece sobre el tradicional, al ser este más dinámico.

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

Para la elección del enfoque se utilizó como punto de partida el modelo de *Barry Boehm* y *Richard Turner* debido a su fiabilidad a la hora de definir el enfoque del proyecto tomando en cuenta sus características, además de ser un método que solo necesita cinco criterios para lograr este objetivo. (16)

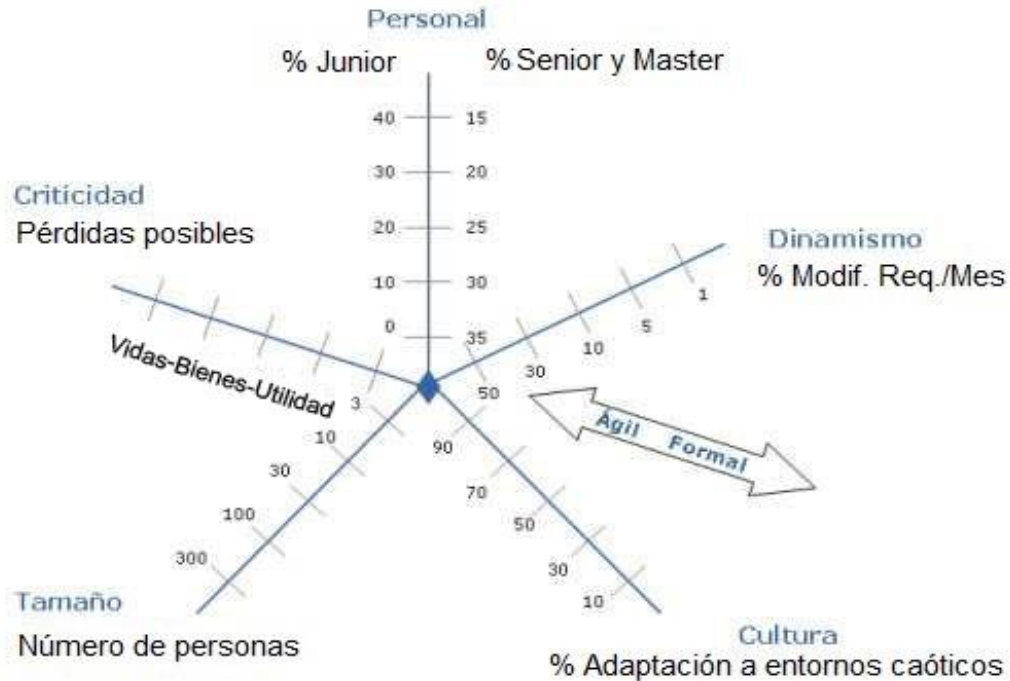


Figura 1. Estrella de Boehm-Turner. (16)

A continuación se explican cada una de las aristas que conforman la estrella:

**Tamaño:** Este criterio se utiliza para representar el número de personas involucradas en el proyecto. Pueden tenerse en cuenta el nivel de complejidad que pueda presentarse en la comunicación entre los miembros del proyecto y los costos que pueden provocar cambios esperados.

**Criticidad:** Se utiliza para evaluar la naturaleza del daño ocasionado por defectos que no hayan sido detectados al producto. Su evaluación puede ser cualitativa.

**Dinamismo:** Representa la rapidez con la que pueden estar cambiando los requerimientos del proyecto.

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

**Personal:** Representa la proporción del personal con experiencia alta, media y baja. Los métodos orientados al plan no se ven afectados negativamente por este factor pues no interesa el nivel de experiencia con la que cuenten los miembros del equipo.

**Cultura:** Las organizaciones y las personas que relaciona el proyecto pueden depender de la confianza o de la relación contractual. Esto refleja el nivel de ceremonia necesario y aceptado: documentación, control, formalismo en las comunicaciones. (16)

Para el desarrollo del proyecto se cuenta con un equipo constituido por dos personas no expertas con conocimientos básicos de las tecnologías a emplear en la solución, estos intervendrán en un ambiente dinámico donde las especificaciones y las necesidades del cliente varían, en cuanto a la criticidad o impacto social del proyecto este se encuentra dentro de la zona de confort donde no existen posibilidades de pérdidas económicas ni de vidas humanas por el uso futuro del sistema, por tanto se decide utilizar un enfoque ágil.

Existen varias metodologías con enfoque ágil, se han analizado las más representativas y utilizadas en el mundo para el proceso de desarrollo de software, a continuación se exponen alguna de estas.

**Programación extrema (XP):** Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. (17)

Esta metodología es eficaz en un ambiente dinámico donde persisten altos riesgos técnicos, su principal artefacto son las historias de usuarios, aunque estas pueden no estar completamente definidas, no traen consigo grandes dificultades, pues se pueden modificar o sustituir por otras.

**SCRUM:** Desarrollada por *Ken Schwaber, Jeff Sutherland* y *Mike Beedle*. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración máxima de 30 días. (17)

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

Esta metodología tiene la particularidad de poder combinarse con XP, esto es favorable en algunos contextos donde se deseen aplicar las potencialidades de ambas.

**Crystal Methodologies:** Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. (17)

La familia de este tipo de metodología requiere del uso de pocos recursos y está enfatizada en la mejora de las competencias del personal, enmarca sus políticas según el tamaño del equipo de proyecto.

**Proceso Unificado Ágil (AUP):** Es una versión simplificada de *Rational Unified Process* (RUP). Este define un flujo de trabajo con disciplinas diferentes a las de RUP, aunque conserva sus fases en cada una. La disciplina de modelación incluye la modelación del negocio, requisitos, análisis y diseño. Por otra parte, se integran además la Gestión de Cambios y Gestión de Configuración en una sola disciplina. (18)

La metodología está compuesta por cuatro disciplinas que sirven de apoyo a la ingeniería (Modelación, Implementación, Prueba y Despliegue) y cuenta con tres destinadas al soporte (Gestión de configuración, Gestión de Proyecto y Ambiente). Estas son ejecutadas y guiadas de forma iterativa.

De las metodologías ágiles anteriormente mencionadas el equipo de trabajo selecciona AUP, debido a la familiarización que posee con los artefactos que son manejados en las diferentes fases de la misma, al ser los mismos que genera RUP. Entre las ventajas que brinda está en su apoyo a un proceso formalizado evitando las improvisaciones, tiene bien definido los roles y fases, es ágil estando basada en un proceso iterativo evolutivo, incrementa la productividad y facilita el trabajo en proyectos de pequeña envergadura.

### 1.3.3 Gestor de base de datos

Se define un Sistema Gestor de Bases de Datos o SGBD, también llamado DBMS (*Data Base Management System*) como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos o BD, (*DB Data Base*). (19)

Para la elección del SGBD más adecuado a emplear en el desarrollo de la propuesta de solución, se procede a realizar un análisis de algunos de estos sistemas, donde se tendrá en consideración sus características, ventajas y desventajas.



# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

**Microsoft SQL Server:** Es un gestor creado por la gran transnacional informática Microsoft, posee licencia EULA (Licencia de usuario final, por sus siglas en inglés), la cual es propietaria y utiliza como lenguaje de consultas T-SQL y ANSI-SQL. (20)

## Principales aspectos

1. Diseñado para trabajar en entornos híbridos, tanto de forma local o en la nube.
2. Fue creado para soportar bases de datos relacionales.
3. Posee un cliente gráfico que soporta sentencias para lenguaje DML (Lenguaje de Manipulación de Datos) y DDL (Lenguaje de Definición de Datos).
4. Administra información de otros servidores de datos.

## Desventajas

- En comparación con otros gestores, posee una limitada configuración de la memoria compartida para sistemas operativos de 32 bits, pues solo permite hasta 64 GB.
- No es multiplataforma, solo funciona sobre Microsoft Windows.

Por los aspectos mencionados, no puede ser elegido para el desarrollo de la solución por las políticas de seguridad que sigue la universidad, al necesitarse un software con licencia libre y que se pueda ejecutar sobre GNU Linux.

**MySQL:** Es un software libre patrocinado por ORACLE y desarrollado en el lenguaje C/C++. Se utiliza comúnmente por los desarrolladores de aplicaciones web, se puede manejar remotamente desde un cliente web y utiliza para las consultas lenguaje SQL (Lenguaje Estructurado de Consultas). (21)

## Principales aspectos

- 1 Servidores multicapas con diferentes módulos.
- 2 Implementa funciones SQL usando librerías de clase altamente optimizadas.
- 3 Para el manejo de los datos utiliza tablas hash, las cuales son usadas como tablas temporales.
- 4 Soporte B-Tree para el manejo de índices.

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

- 5 Posee motores transaccionales y mono transaccionales.
- 6 Replicación y conectividad segura.

Actualmente es uno de los gestores más robustos, posee pocas desventajas y se han registrado pocos errores. (22) Algunas de las desventajas es que el planificador no actualiza los privilegios cuando se elimina una tabla. Esto puede no parecer peligroso, si el desarrollador se percata y actualiza los privilegios manualmente, pero puede traer errores futuros. Por otra parte, existe una escasa documentación y bibliografía sobre el tema, por lo que sería muy costosa la capacitación en dicha herramienta, además de que la universidad no cuenta con una comunidad dedicada a la misma.

**PostgreSQL:** Gestor de base de datos desarrollado por PostgreSQL *Global Development Group* (PGDG), es multiplataforma y posee licencia BSD, la cual es una licencia libre permisiva. Permite no solo la implementación de bases de datos relacionales, también simula las multidimensionales. Se puede manejar mediante un el cliente de entorno gráfico PGAdmin o mediante un cliente en forma de terminal de texto llamado PSQL, el cual es muy efectivo para servidores con pocos recursos de hardware. (23)

## Principales aspectos

- 1 Números de precisión larga.
- 2 Para tipos de datos en forma de cadena, permite cadenas de longitud ilimitada.
- 3 Soporta IP versión 6.
- 4 Incluye varios métodos para el manejo de índices, utilizando por defecto B-Tree.
- 5 Permite a los usuarios definir su propio tipo de datos.
- 6 Utiliza nativamente el lenguaje procedural PL/PGSQL para implementar funciones, además de soportar otros, tales como: C, C++, Java PL, Java web, PL/Perl, PL/PHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl y PL/Scheme.

## Ventajas

- Seguridad en términos generales.

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

- Integridad referencial y afirmaciones (*Assertions*) y restricciones de dominio.
- Disparadores (*Triggers*), autorizaciones, transacciones y respaldos.

## Desventajas

- La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva, aunque es necesario destacar que es uno de los gestores más utilizados por la universidad, y existe una comunidad dedicada a ello, la cual se encarga en muchos casos de ofrecer ayuda sobre ciertos temas.

Para el desarrollo del trabajo de diploma será necesario el uso del gestor **PostgreSQL**, debido a que es un software con licencia libre, se dispone de una amplia documentación tanto por parte del equipo de trabajo y la universidad, además de que el equipo está bien adiestrado en el uso de dicha herramienta, al estar incluida en su programa de estudio de la disciplina de Bases de Datos, y haber desarrollado varios sistemas de gestión haciendo uso de la misma.

### 1.3.4 Lenguaje de programación

Para el desarrollo de aplicaciones web existe un gran número de lenguajes de programación, divididos en dos grupos, lenguajes del lado del cliente y lenguajes del lado del servidor. Para el primer grupo son muy utilizados en la capa de presentación o las vistas de los usuarios los lenguajes: XHTML, CSS y JavaScript. (24) El segundo grupo es utilizado en la capa de negocio, la cual se encarga de la implementación de los servicios o funcionalidades que el sistema proveerá al usuario. Los más utilizados son: Python, Ruby, PHP, ASP.NET, Java entre otros. (25)

Para el desarrollo de la solución se utilizará **Java** como lenguaje de programación del lado del servidor, es un lenguaje fácil de utilizar, práctico y con una larga trayectoria, llegando a realizar grandes proyectos en un tiempo relativamente corto. (26) Posee un paradigma puramente orientado a objetos, al ser un lenguaje ya maduro dispone de una amplia comunidad de desarrollo donde se pueden adquirir tutoriales y libros para su aprendizaje. Por otra parte, el equipo de trabajo cuenta con conocimientos sobre el mismo y ha desarrollado ya varias aplicaciones escritas en este lenguaje.

Para lograr que el sistema tenga un aspecto uniforme se construirán un grupo de plantillas utilizando los lenguajes **XHTML** y **CSS**. Estas plantillas constituyen temas de diseño que permiten definir la estructura, formateo de las páginas y establecen el maquetado de las vistas de la aplicación. Cabe destacar que los

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

componentes creados en Java pueden integrarse con facilidad sobre estas plantillas, dándole una mejor organización a la aplicación web y creando vistas más agradables al usuario. (27)

## 1.3.5 Framework de programación

**Framework (plataforma, entorno, marco de trabajo):** Desde el punto de vista del desarrollo de software, un *framework* es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. (28)

Estos incluyen las mejores prácticas ya probadas para problemas comunes que surgen en la construcción de sistemas, centrando el desarrollo en lo que realmente le importa al equipo. (28)

Para Java existen varios marcos de trabajo, entre los que se destacan Spring MVC, JSF, Vaadin, GWT (*Google Web Toolkit*), Grails, Play 2, entre otros. (29)

**Vaadin 7.1.7:** Marco de trabajo para el desarrollo de aplicaciones web escritas en Java, está diseñado para la creación y mantenimiento de sistemas de alta calidad y con interfaces de usuario fáciles de construir. Soporta los modelos de programación del lado del cliente y del servidor, siendo este último su principal fortaleza. Se encarga desde el lado del servidor de la administración de las interfaces de usuario y la comunicación AJAX (JavaScript asíncrono y XML) entre el navegador y el servidor.

Vaadin es un *framework* puramente RIA (Aplicaciones de Internet Enriquecidas, por sus siglas en inglés), elemento que lo hace popular hoy en día. Posee una arquitectura por capas, donde se separan los elementos del navegador, el servidor y el *Back-end* (Acceso a datos, Persistencia, Lógica de negocios y Servicios web), permitiendo así una programación segura. (27)

**Bootstrap 2.3.2:** Es el *framework* para CSS más popular que existe para el desarrollo web, posee una amplia documentación, haciéndose extensible su uso. Provee de forma fácil las mejores prácticas en el diseño de plantillas, donde con este se generan vistas muy agradables y ligeras para el cliente. Es de código abierto, con una comunidad que lo mantiene a través de GitHub. (30)

### **Frameworks para la capa de acceso a datos**

Para el acceso a los datos, Java cuenta con varios marcos de trabajo que así lo permiten. Estas tecnologías están centradas en la conversión de las entidades del sistema hacia tablas de las bases de datos y también funciona de forma inversa. A este proceso se le denomina mapeo ORM (Mapeo Relacional de Objetos, por

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

sus siglas en inglés). Entre los marcos de trabajo más destacados para este lenguaje se encuentran *Hybernate* y JPA (*Java Persistence API*). (31) Para el desarrollo de la solución será utilizado este último debido a su facilidad de uso, su alta compatibilidad con otras herramientas, es la API (*Application Program Interface*) construida para Java EE y está incluida en el estándar EJB3 (*Enterprise JavaBean*, estándar de aplicaciones empresariales). (32)

**JPA 2.0:** Construida por *Sun Microsystems* para la plataforma Java EE, se utiliza como ORM. Permite elegir cualquier proveedor de persistencia: sea *Hybernate*, Eclipse Link, entre otros según las necesidades, pero todos trabajarán bajo el mismo API que este provee. Se realiza el mapeo de las entidades definiendo anotaciones sobre sus atributos, indicando las restricciones que estos tendrán a la hora de almacenarse en las tablas correspondientes de la base de datos o por un archivo de configuración en formato XML. Tiene definido un gran número de anotaciones para distintos tipos de datos e incluye soporte para expresiones regulares. (31)

## 1.3.6 Entorno de desarrollo

Para el trabajo con el *framework* Vaadin se pueden utilizar tres entornos de desarrollo que lo soportan: Netbeans, IntelliJ IDEA y Eclipse. Estos sistemas son los recomendados para trabajar con el *framework* y existe amplia documentación para su configuración y uso. (27) Poseen varios sitios en línea para el soporte y la ayuda de sus distintas versiones, pero fue elegido el último a petición del cliente.

**Eclipse Kepler 4.3:** Es un IDE desarrollado por la prestigiosa compañía ORACLE basado en código abierto, permite la creación de múltiples proyectos, herramientas en Java y el desarrollo de complementos (*plug-ins*), posee una plataforma de herramientas web y se integra con Maven. (33)

La integración de este IDE con Vaadin se puede realizar mediante la instalación de un *plug-in* o mediante Maven, este último es una opción factible debido a que dota al proyecto de un sistema que permite la gestión de las librerías y objetos mediante la instalación del repositorio Maven local, luego este se configura para que se conecte a un repositorio remoto en la web. La universidad cuenta con este tipo de servicio a través de la intranet local.

**Apache Maven 3.1.0:** Repositorio local que se encarga de gestionar librerías, complementos y drivers para proyectos de Vaadin 7, permite desde la web incorporar dichos elementos a utilizar mediante la inserción

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

de dependencias al archivo de configuración en formato XML del proyecto (pom.xml), una vez que está configurado al repositorio institucional. (34)

## 1.3.7 Servidores de aplicaciones web

Un servidor web permite procesar aplicaciones de este tipo del lado del servidor, donde se establece una conexión bidireccional entre el cliente que realiza peticiones y el servidor que se encarga de dar respuestas o ejecutarlas. Para Java existen varios servidores, tales como: GlassFish, JBoss y Apache Tomcat.

**GlassFish:** Es un servidor de aplicaciones web basado en código abierto y asistencia gratuita, fue creado inicialmente por *Sun Microsystems* y luego adquirido por ORACLE. Posee arquitectura modular basada en OSGi (*Open Services Gateway Initiative*), es multiplataforma y posee un certificado de compatibilidad con Java EE (*Enterprise Edition*). (35)

**JBoss:** Este servidor actualmente es el que lidera el mercado y al igual que GlassFish posee un certificado de compatibilidad de Java EE, permite el desarrollo de aplicación web para sistemas empresariales y servicios de SOA (Arquitectura Orientada a Servicios) y está pensado para la obtención del máximo rendimiento y productividad. (36)

**Apache Tomcat 7:** Es un servidor de aplicaciones web basado en código abierto para tecnologías Java Servlet 3.0 y Java Server Pages 2.2 (JSP). Está publicado bajo licencia Apache versión 2. Posee una comunidad de desarrollo muy selectiva, utiliza Catalina como contenedor de *servlets* a partir de la versión 4. Es multiplataforma, siendo compatible con todo sistema operativo que posea instalada la máquina virtual de Java.

Para el desarrollo de la solución planteada el equipo de trabajo decide optar por **Apache Tomcat** debido que está recomendado para Vaadin 7, permite el monitoreo y la administración remota de las aplicaciones mediante una conexión segura, servicios de *virtual hosting*, *clustering* y equilibrado de cargas en las aplicaciones. (37)

## 1.3.8 Lenguaje de modelado

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño, el proceso indica los pasos que se deben seguir para llegar a este. (38) Al estar estandarizado permite a los desarrolladores documentar el software en cuanto a funcionalidades, procesos de negocios y

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

conceptos asociados. Para la modelación de la solución se propone el uso de UML (Lenguaje de Modelado Unificado) en su versión 2.0.

**UML 2.0:** Es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de 1980 y principios de 1990 (38). Posee notaciones estandarizadas que permiten la construcción de diagramas y artefactos tanto en el modelado de análisis y diseño del sistema. Esta versión actual simplifica en gran medida la representación de los conceptos, facilitando su uso y la comunicación entre los involucrados en el proyecto.

## Herramienta CASE

Para el uso de un lenguaje de modelado existen varias herramientas CASE (Ingeniería de Software Asistida por Computadoras), estas son aplicaciones informáticas que traen grandes impactos al proyecto al reducir de forma significativa costes y tiempo de producción al trabajar directamente en cada fase del ciclo de vida de desarrollo del software. Para el desarrollo de la solución se propone **Visual Paradigm** en su versión 8.

**Visual Paradigm 8.0:** Herramienta CASE que utiliza UML como lenguaje de modelado. Provee una estructura de proyecto jerarquizada donde se pueden modelar las diferentes perspectivas del sistema y sus artefactos en cada fase. Incluye herramientas de animación para simular el funcionamiento del modelado que se construye. Permite por otra parte: el análisis de impacto, administración de tareas, diseño colaborativo y generación de código en múltiples lenguajes a partir de clases. (39)

## 1.4 Patrón de arquitectura

Antes de hablar de patrones primero se debe centrar el análisis en el concepto de arquitectura de software, a partir de este se tendrá una visión más clara, para llegar a la definición de patrón de arquitectura y cómo emplearla en el desarrollo de la solución.

La **arquitectura de software** de un sistema de programa o computación es la estructura de las estructuras del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente y las relaciones entre ellos. (40)

Estas se definen de forma temprana el proceso de desarrollo del sistema debido a que permite la comunicación entre las partes interesadas, decisiones del diseño y muestra cómo está estructurado el mismo.

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

**Patrón de arquitectura:** Descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución. El esquema de solución se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma de cómo estos colaboran entre sí. (41)

Para la elección de un determinado patrón de arquitectura se deben tener en cuenta los diferentes atributos de calidad, cada estilo o patrón propicia atributos de calidad en el sistema, pero con la diferencia de que cada uno se enfoca en grupos de atributos diferentes mejorando o disminuyendo algunos de estos.

**Arquitectura de tres capas:** Es una arquitectura para la construcción de aplicaciones basadas en un entorno cliente-servidor. Se encarga de separar de manera lógica la interfaz de usuario (capa de presentación), el negocio (capa de negocio) y las entidades (capa de datos).

La comunicación entre dichas capas se realiza de forma lineal, comenzando desde la capa de presentación donde el usuario realiza las peticiones, estas son procesadas en la capa de negocio y luego los datos son almacenados, consultados y actualizados los datos en esta última capa. (42)

## Descripción de cada una de las capas

**Capa de presentación:** Se encarga de mostrar los datos mediante una interfaz gráfica o terminal de texto al usuario, incluyendo además notificaciones de error, formateo de datos, ordenación y validación de los mismos.

**Capa de lógica de negocio:** Se encarga de la publicación de los servicios del negocio, control de acceso, implementación de las funcionalidades e invoca la persistencia.

**Capa de datos:** Asegura la integridad, disponibilidad de los datos y gestiona la persistencia.

## 1.5 Conclusiones Parciales

Con el estudio realizado se puede constatar que la biblioteca de la Facultad de Economía de la Universidad de La Habana en su estado actual no supe las necesidades informacionales de sus usuarios a través de la web y requiere una adaptación al nuevo contexto, evidenciándose la necesidad de convertirla en CRAI. Para llevar a cabo la solución del problema se propone el desarrollo de un portal web, con el objetivo de proveer los servicios requeridos. Para desarrollar dicha propuesta se definió el uso de la metodología AUP la cual guiará el proceso de desarrollo de software y para su construcción se ha elegido el *framework* Vaadin 7.1.7



# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

sobre el entorno de desarrollo Eclipse Kepler 4.3, el cual trabajará en paralelo con el servidor Apache Tomcat 7, Apache Maven 3.1.0 y JPA 2.0 como *framework* para la capa de datos. Como gestor de base de datos se empleará PostgreSQL en su versión 9.3 y como herramienta de modelado Visual Paradigm 8.0 utilizando el lenguaje de modelado UML en su versión 2.0.

# CAPÍTULO 2 PROPUESTA DE SOLUCIÓN

La metodología AUP define en cada una de sus fases un grupo de artefactos que brindan una comprensión clara del negocio y el sistema a desarrollar, al tener la misma un enfoque ágil permite que el equipo elabore la documentación necesaria que facilite la comunicación entre los desarrolladores y las partes interesadas. En este capítulo se presentarán los siguientes artefactos: Modelo de dominio, la especificación de requisitos funcionales y no funcionales que tendrá la propuesta de solución, se modelarán los casos de uso y la interacción de los actores con el sistema, así como una descripción detallada de los mismos.

## 2.1 Modelo de dominio

Un modelo de dominio o modelo conceptual es una representación de conceptos del dominio de un problema determinado. Mediante el uso de UML este se puede modelar utilizando un grupo de diagramas de estructura estática que no representa ninguna operación. Permite representar elementos del mundo real y no componentes del software. Provee conocimiento de la nomenclatura del dominio al comunicarle a las partes interesadas los términos importantes y como se relacionan entre sí, siendo un artefacto fundamental para realizar el análisis orientado a objetos. (43)

### 2.1.1 Conceptos del dominio

Para un mayor entendimiento del problema se procede a esclarecer las terminologías asociadas al dominio, definiendo los términos más importantes que le serán de utilidad al equipo de trabajo para la posterior confección del modelo.

**Contenido:** Constituye una publicación de información en el portal web y se clasifica por categorías, el administrador del sistema se encarga de gestionarlos con el objetivo de publicar información que este desee en función de las necesidades del usuario y la institución, luego estos pueden ser accedidos por los usuarios que visiten el portal.

**Material:** Los materiales pueden ser de cuatro tipos: artículos, audios, audiovisuales o libros, estos constituyen la base del conocimiento almacenado en el CRAI, sirviendo de apoyo al aprendizaje y la investigación.

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

**Usuario:** Existen cuatro tipos de usuarios: estudiantes, profesores, investigadores y administradores. Los tres primeros consumen recursos y servicios del CRAI y los administradores son los encargados de publicar las temáticas y brindar los diferentes servicios de gestión a los restantes usuarios.

**Biblioteca personal:** Es un espacio virtual que posee un usuario registrado en el sistema, le permite a este elegir un número limitado de materiales disponibles a su biblioteca, facilitando el acceso rápido a estos materiales, así como la búsqueda de los mismos, siendo útil cuando el número de materiales registrados en el sistema sea muy grande.

**Temática:** Concepto que permite organizar los materiales registrados en el sistema. Cada usuario puede elegir una o varias temáticas en su perfil de forma opcional, permitiendo que el sistema pueda recomendar materiales a los usuarios según las temáticas elegidas.

### 2.1.2 Diagrama del modelo de dominio

El siguiente diagrama muestra la relación entre los conceptos identificados del problema y anteriormente descritos.

# CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

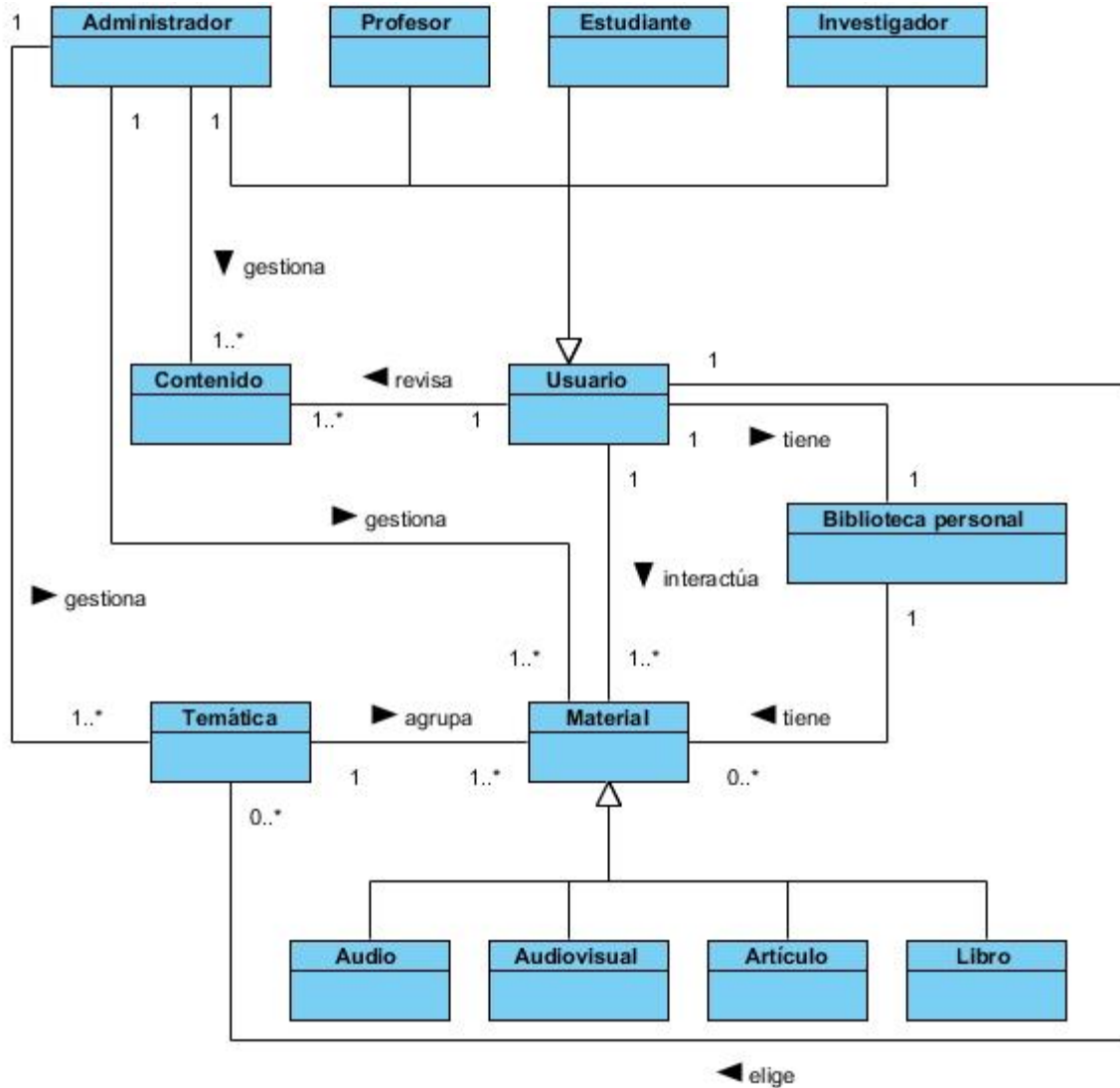


Figura 2. Modelo de dominio.

## 2.2 Requerimientos del sistema

La definición de requisitos del sistema permite establecer las capacidades del mismo y establecen con detalle las funciones, servicios y restricciones operativas que el sistema debe cumplir. En algunos casos los requerimientos funcionales de los sistemas pueden declarar explícitamente lo que este no debe hacer. (44) Los mismos son planteados con el fin de lograr un entendimiento claro entre el cliente y el equipo de desarrollo, en aras de satisfacer sus verdaderas necesidades.

## **CAPÍTULO 2. PROPUESTA DE SOLUCIÓN**

A continuación se presentan los requisitos funcionales del sistema.

**RF1** Gestionar usuario:

**RF1.1** Eliminar usuario.

**RF1.2** Ver los datos de un usuario.

**RF2** Autenticar usuario.

**RF3** Registrar usuario.

**RF4** Editar perfil.

**RF5** Gestionar material:

**RF5.1** Registrar material.

**RF5.2** Eliminar material.

**RF5.3** Editar datos de un material.

**RF6** Descargar material.

**RF7** Gestionar categoría:

**RF7.1** Registrar categoría.

**RF7.2** Eliminar categoría.

**RF7.3** Ver datos de una categoría.

**RF7.4** Editar datos de una categoría.

**RF8** Gestionar contenido:

**RF8.1** Registrar contenido.

**RF8.2** Eliminar contenido.

**RF8.3** Editar datos de un contenido.

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

**RF9** Ver contenido.

**RF10** Buscar material.

**RF11** Gestionar solicitud de materiales:

**RF11.1** Ver datos de una solicitud.

**RF11.2** Atender solicitud.

**RF11.3** Eliminar solicitud.

**RF12** Solicitar material.

**RF13** Gestionar biblioteca personal:

**RF13.1** Añadir material a biblioteca personal.

**RF13.2** Quitar material de la biblioteca personal.

**RF14** Ver materiales recomendados.

**RF15** Gestionar temática:

**RF15.1** Crear temática.

**RF15.2** Ver datos de una temática.

**RF15.3** Eliminar temática.

**RF15.4** Editar datos de una temática.

### **2.3 Requerimientos no funcionales**

Los siguientes requisitos no funcionales definen cualidades que complementan el funcionamiento del sistema.

#### **Usabilidad**

- El sistema debe poseer una interfaz fácil de utilizar para cualquier tipo de usuario con conocimientos básicos de informática y en el manejo de ordenadores.

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

### Seguridad

- El acceso a la información debe estar restringido por usuario, contraseña y rol, siendo el rol Administrador, el que tendrá control total sobre la información del sistema.
- Cuando un usuario se autentique o registre en el sistema se cifrará la contraseña del usuario mediante el algoritmo MD5, tanto para almacenarla en la base de datos, cuando se produzca un registro nuevo y para compararla cuando se inicie sesión.

### Rendimiento

- Los tiempos de respuestas del sistema a cada acción que ejecute el usuario no deben tomar un tiempo superior a los 6 segundos.

### Accesibilidad

- El sistema debe estar disponible desde cualquier estación de trabajo conectada a la red y permitir su administración de forma remota.

### Apariencia

- El sistema debe tener una interfaz con colores adecuados al contenido, predominando un estilo de diseño sencillo que logre un buen contraste entre los elementos.
- Debe informar al usuario sobre cualquier cambio que se realice o esté a punto de efectuar, informándole mediante notificaciones. Estas deben poseer un mismo formato según el tipo de notificación, tanto si es error, alerta o éxito.
- Debe existir un menú principal que comprenda todas las opciones disponibles al usuario y este debe estar accesible desde todas las vistas.

### 2.4 Modelo de casos de uso del sistema

El modelado de casos de uso permite documentar el comportamiento del sistema desde el punto de vista del usuario, posibilita representar los requisitos funcionales que debe ejecutar el software y los usuarios que interactúan con este, resultan muy útiles en la comunicación con el cliente, debido a su fácil interpretación. Sirven de base para la ejecución de futuras fases en el proceso de desarrollo de software. (45)

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

### 2.4.1 Descripción de los actores del sistema

Tabla 1. Actores del sistema y su descripción

Actor	Descripción
Usuario	Actor que podrá acceder a las funcionalidades que brinda el sistema, tales como: autenticarse, registrarse, leer contenido, editar perfil, descargar material, añadir material a la biblioteca personal, buscar materiales con opciones de filtrado y solicitar información bibliográfica de un tema.
Administrador	Actor que tendrá acceso total a todas las funcionalidades del sistema, para realizar operaciones de gestión sobre la información que se maneja en el mismo.

### 2.4.2 Diagrama de casos de uso del sistema

Este artefacto permite mostrar la relación entre los actores y los casos de uso del sistema, siendo el actor el que inicializa estos casos de uso.



## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

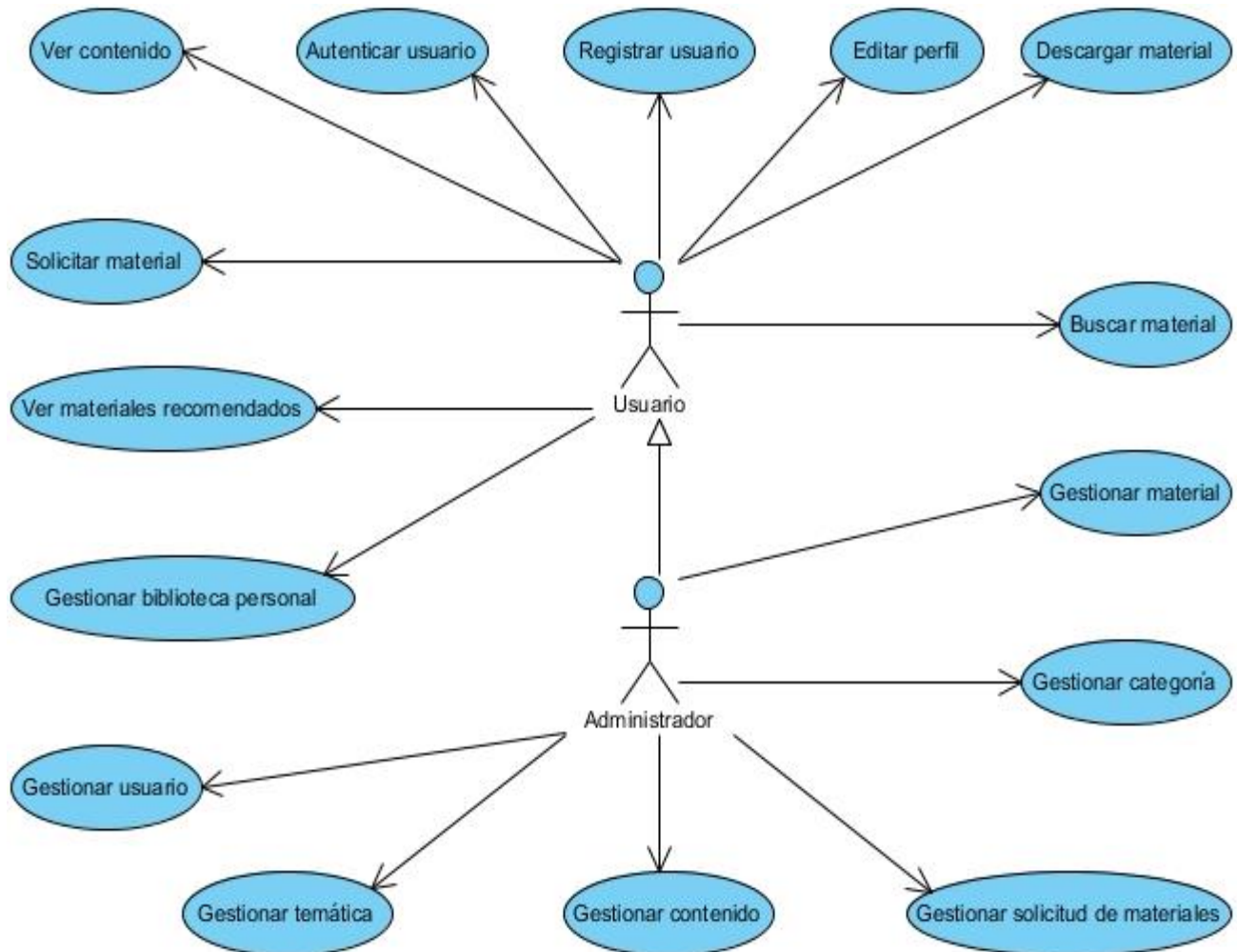


Figura 3. Diagrama de casos de uso del sistema.

### 2.4.3 Patrones de casos de uso

Al modelado por casos de uso se le aplican un grupo de patrones que permiten simplificar el diagrama y mejorar su comprensión, los mismos describen diferentes soluciones para el modelado de problemas específicos, atendiéndolos por separado. (46)

**Patrón CRUD:** Es un patrón que permite agrupar las operaciones de crear, leer, actualizar y eliminar piezas de información de un tipo o entidad. El mismo puede ser completo o parcial, dependiendo si se agrupan o no todas estas operaciones en un mismo caso de uso. (46)

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

- **CRUD completo:** Se puede observar su aplicación en el caso de uso Gestionar categoría y Gestionar temática, donde este incluye todas las operaciones descritas anteriormente sobre los datos de una categoría y de una temática.
- **CRUD parcial:** Se puede observar su aplicación en el caso de uso Gestionar contenido, donde este solo incluye las operaciones de eliminar, editar y crear para el actor Administrador sobre los datos de la entidad contenido, separando la operación ver contenido para el actor Usuario. Este patrón también es aplicado en el caso de uso Gestionar material.

### Patrón múltiples actores:

- **Rol común:** Permite que dos actores desempeñen el mismo rol, para que a través de este puedan acceder a un determinado caso de uso. (46) Se puede apreciar este patrón al crearse el rol Usuario para acceder a un grupo de casos de uso, que pueden ser accedidos por el actor Administrador, mientras que este último puede iniciar otros escenarios que no pueden ser accedidos por el rol Usuario.

### 2.2.4 Descripción textual de los casos de uso del sistema

A continuación se muestran las descripciones textuales de los casos de uso Solicitar material y Gestionar solicitud de materiales. Las restantes descripciones textuales están presentes en el Anexo 1.

Tabla 2. Descripción textual del caso de uso Solicitar material

Caso de uso	Solicitar material
Actores	Usuarios
Resumen	El caso de uso inicia cuando el actor decide solicitar un material al sistema. Para ello el actor selecciona la opción Solicitar material del menú Servicios. El sistema muestra un formulario con los campos correspondientes a una solicitud. Permite crear una nueva solicitud en el sistema o cancelar la operación, finalizando el caso de uso.
Precondiciones	El Usuario debe estar autenticado.
Postcondición	Se solicitó un material

## CAPÍTULO 2.PROPUESTA DE SOLUCIÓN

Referencias	RF12
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
Acciones del Actor.	Respuesta del sistema.
1. El caso de uso inicia cuando el actor selecciona la opción Solicitar material en el menú Servicios.	
	<p>2. El sistema brinda la posibilidad de insertar los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Título del material</li> <li>• Autor del material</li> <li>• Descripción del material</li> </ul> <p>Permite seleccionar:</p> <ul style="list-style-type: none"> <li>• Categoría, con los valores: <ul style="list-style-type: none"> <li>-Libro</li> <li>-Audiovisual</li> <li>-Artículo</li> </ul> </li> </ul> <p>Y las opciones:</p> <ul style="list-style-type: none"> <li>• Aceptar</li> <li>• Cancelar</li> </ul>
3. Llena los campos requeridos en el formulario y selecciona la opción Aceptar.	
	4. Valida los datos introducidos.
	5. Registra una nueva solicitud de material en el sistema.
	5. Muestra un mensaje: "Solicitud registrada correctamente".
	6. El caso de uso termina.
<b>Flujos Alternos</b>	
<b>*.a Selecciona la opción Cancelar</b>	

## CAPÍTULO 2.PROPUESTA DE SOLUCIÓN

Acciones del Actor	Respuesta del sistema
	4.a.1 El sistema regresa a la vista anterior.
	4.a.2 El caso de uso termina.
<b>4.a Existen campos vacíos</b>	
Acciones del Actor	Respuesta del sistema
	4.a.1 El sistema muestra un mensaje “Verifique que no existan campos vacíos”.
	4.a.2 Muestra el indicador de campos que no pueden estar vacíos.
	4.a.3 Regresa al paso 2.
<b>4.b No se ha seleccionado una categoría</b>	
Acciones del Actor	Respuesta del sistema
	4.b.1 El sistema muestra un mensaje “Debe elegir una categoría”.
	4.b.2 Regresa al paso 2.
<b>Prototipos de interfaz</b>	

## CAPÍTULO 2.PROPUESTA DE SOLUCIÓN

### Realizar Solicitud

Título del material\*

Autor del material

Descripción del material\*

Categoría\*

Tabla 3. Descripción textual del caso de uso Gestionar solicitud de materiales

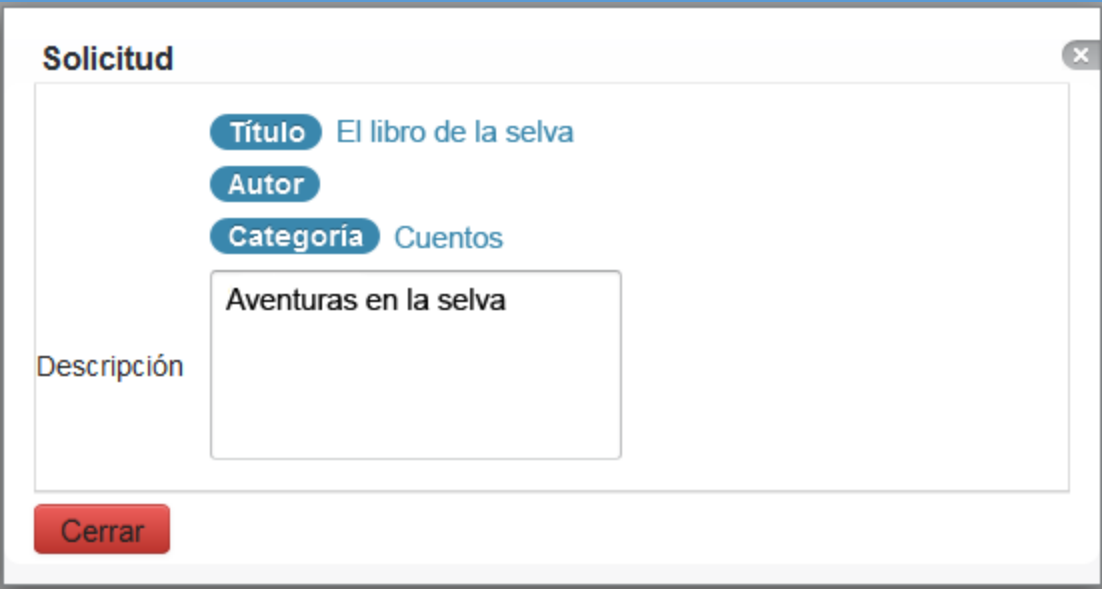
Caso de uso	Gestionar solicitud de materiales
Actores	Administrador
Resumen	El caso de uso inicia cuando el actor accede a la ventana inicial de administración del sistema y accede a la opción Lista de solicitudes del menú Atender solicitud. Muestra un listado con las solicitudes registradas en el sistema y por cada una permite: ver, atender y eliminar solicitud. El actor visualiza, atiende o elimina una solicitud finalizando así el caso de uso.
Precondiciones	El administrador debe estar autenticado.
Postcondición	Se visualizó, se atendió o se eliminó una solicitud.
Referencias	RF11, RF11.1, RF11.2, RF11.3
<b>Flujo de eventos</b>	

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Flujo básico	
Acciones del Actor	Respuesta del sistema.
1. El caso de uso inicia cuando el actor accede al sistema y selecciona la opción Lista de solicitudes del menú Atender solicitud.	
	2. El sistema muestra una lista cuyos elementos representan los datos de cada solicitud almacenada en el sistema. Y permite: <ul style="list-style-type: none"> <li>• Visualizar solicitud.</li> <li>• Atender solicitud. Ver sección 1: “Atender solicitud”.</li> <li>• Eliminar solicitud. Ver sección 2: “Eliminar solicitud”.</li> </ul>
3. Selecciona la opción Visualizar solicitud.	
	4. El sistema muestra una ventana auxiliar con los datos de la solicitud, sin poder modificar los mismos. Brinda la opción: <ul style="list-style-type: none"> <li>• Cerrar</li> </ul>
5. Revisa la solicitud y selecciona la opción Cerrar.	
	6. Cierra la ventana actual.
	7. El caso de uso termina.
Sección 1: Atender solicitud	
Flujo Básico	
Acciones del Actor	Respuesta del sistema
1. Selecciona la opción Atender solicitud.	
	2. Envía un mensaje de correo electrónico al usuario solicitante indicando que la petición ya fue atendida por el administrador del sistema.

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

	3. Muestra el mensaje: “Solicitud atendida correctamente”.			
	3. El caso de uso termina.			
<b>Sección 2: Eliminar solicitud</b>				
<b>Flujo Básico</b>				
Acciones del Actor	Respuesta del sistema			
1. Selección la opción Eliminar solicitud.				
	2. Muestra un mensaje solicitando la confirmación del usuario de dicha acción, permitiendo elegir: <ul style="list-style-type: none"> <li>• Aceptar</li> <li>• Cancelar</li> </ul>			
3. Selecciona la opción Aceptar				
	5. Elimina la solicitud del sistema			
	6. Muestra el mensaje: “Solicitud eliminada correctamente”.			
	6. Termina el caso de uso.			
<b>Flujo alternativo</b>				
<b>*.a Selecciona la opción cancelar</b>				
Acciones del Actor	Respuesta del sistema			
	*.a.1 Cierra el mensaje de confirmación.			
	*.a.2 Retorna a la vista anterior.			
<b>Prototipos de Interfaz</b>				
<b>Lista de Solicitudes</b>				
IDENTIFICADOR	NOMBRE DE USUARIO	VISUALIZAR SOLICITUD	ATENDER SOLICITUD	ELIMINAR SOLICITUD
31	ogtoledano	✓ <a href="#">Visualizar solicitud</a>	✓ <a href="#">Atender Solicitud</a>	✗ <a href="#">Eliminar</a>
32	ogtoledano	✓ <a href="#">Visualizar solicitud</a>	✓ <a href="#">Atender Solicitud</a>	✗ <a href="#">Eliminar</a>
34	isla	✓ <a href="#">Visualizar solicitud</a>	✓ <a href="#">Atender Solicitud</a>	✗ <a href="#">Eliminar</a>



The image shows a web form titled "Solicitud" (Request) with a close button (X) in the top right corner. The form contains the following fields:

- Título** (Title): El libro de la selva
- Autor** (Author): (empty field)
- Categoría** (Category): Cuentos
- Descripción** (Description): Aventuras en la selva

At the bottom left of the form is a red button labeled "Cerrar" (Close).

### 2.5 Conclusiones parciales

Con el estudio de los principales conceptos asociados al dominio, se modeló su representación para facilitar su comprensión y la forma en que se vincularán las funcionalidades con el entorno. La definición de los requisitos funcionales y no funcionales del sistema establecieron las capacidades que debe cumplir la propuesta de solución así como los diferentes servicios que brindará el portal web. Se construyó el diagrama de casos de uso con el fin de representar las relaciones entre los actores del sistema y sus diferentes escenarios. Para facilitar el entendimiento entre los desarrolladores y las partes interesadas se realizaron las descripciones textuales de los casos de uso del sistema donde se detallan las acciones del actor y la respuesta del sistema a estos estímulos, incluyendo los prototipos de interfaz de usuario. El desarrollo de estos artefactos servirá como punto de partida para la ejecución de la etapa de modelado del sistema.



# DESARROLLO DE LA PROPUESTA DE SOLUCIÓN **3**

La disciplina de modelado de la metodología AUP comprende las fases de modelado del negocio, requisitos, análisis y diseño. Este modelado es colaborativo, generando los artefactos necesarios para la comprensión del sistema con un mayor nivel de abstracción y una vez obtenidos dichos elementos se da paso a la implementación y las pruebas. En el presente capítulo se desarrollará la propuesta de solución, donde se elaborará el modelado de diseño del sistema debido a que el equipo de trabajo y el cliente poseen una total claridad de los requerimientos, por lo que se generarán los diagramas de clases, diagramas de secuencia y diagrama de despliegue. Para la ejecución de la implementación y las pruebas será necesario la obtención de los diagramas de componentes y el diseño de los casos de prueba, con esto se implementará el sistema y luego se someterá a evaluación para validarlo.

### 3.1 Patrones de diseño

Hoy en día no se concibe el desarrollo de un sistema robusto, sin que exista en su diseño el uso de patrones, fundamentalmente en el diseño orientado a objetos donde existe un amplio repertorio y expresiones que guían el proceso de desarrollo de los sistemas. Estos constituyen la solución de un problema determinado y se pueden aplicar en diferentes contextos, su finalidad no es expresar nuevas ideas en el diseño sino resolver los problemas mediante una solución ya probada y fiable. (47)

De acuerdo al patrón de arquitectura antes elegido en la fundamentación teórica, por las tecnologías y herramientas de desarrollo seleccionadas, se decide emplear en el diseño del sistema patrones GRASP (Patrones Generales de Software para Asignar Responsabilidades, siglas en inglés), los cuales serán muy útiles para facilitar la implementación del sistema, posibilitar la reutilización de código y lograr un software con alto grado de mantenibilidad.

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones, define un grupo de principios fundamentales para diseñar eficazmente un software. (43) Para la propuesta de solución se pretenden utilizar los siguientes.

**Creador:** Se emplea cuando se le aplica la responsabilidad a una clase determinada de crear una o más instancias de otra. Esto sucede en caso de que la clase creadora contenga, agregue, registre, utilice o posea

## CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

datos de inicialización de objetos de alguna clase determinada. (43) El empleo de este patrón será muy útil para aquellas clases encargadas de registrar datos de entidades que manejará el sistema, y este se aplica en las clases: **ControlMaterialImpl**, **ControlCategorialImpl**, **ControlTematicalImpl**, **ControlContenidoImpl**, **ServicelImpl** y **PaginadorImpl**; por otra parte existen dos clases encargadas de crear instancias de cada componente visual o clase interfaz del sistema para ubicarlas en las localizaciones definidas, para la vista del usuario se encarga la clase **VistaPrincipal** y para la administración la clase **Administrar**.

**Experto:** Se basa en asignar la responsabilidad a una clase que se encargará de ser experto de la información, la misma cuenta con la información necesaria para cumplir con la responsabilidad. (43) Al aplicarlo al sistema, este patrón permitirá identificar las clases que poseen la información requerida, para que luego estas puedan ser manipuladas por otras. Se evidencia en las clases que extienden de la clase **Entidad**, las cuales son expertas en su propia información, tales como: **Contenido**, **Temática**, **Usuario**, **Solicitud**, **Material** y **Categoría**.

**Alta cohesión:** Se basa en asignar una responsabilidad de modo que la cohesión siga siendo alta, resolviendo el problema de mantener la complejidad en las clase dentro de los límites manejables, permitiendo definir en las clases las funcionalidades relacionadas para que estas no realicen un trabajo enorme. (43) Este patrón permitirá al equipo de desarrollo que la implementación de las clases encargadas de ofrecer los servicios del sistema sean fáciles de comprender, reutilizar, mantener y poco susceptibles a cambios. Para ello el equipo de desarrollo determinó crear varias clases controladoras y otros componentes según la similitud de los requisitos funcionales, de forma tal que cada una de estas solo contenga los métodos correspondientes a su área funcional.

**Controlador:** Asigna la responsabilidad del manejo de mensajes de los eventos de un sistema a una clase que represente el sistema global, organización, elemento activo del mundo real o manejador artificial de los eventos del sistema. (43) Este patrón se evidencia en las clases controladoras creadas para realizar las operaciones de gestión sobre las entidades y en la clase **ControladorJPAGenerico**, la cual posibilita el acceso a la información almacenada en la base de datos.

# CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

## 3.1.1 Patrones estructurales

Estos patrones de diseño fueron concebidos para resolver problemas de agregación y composición de las relaciones de clases. (48)

**Patrón Fachada:** Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema, se define una interfaz de alto nivel para facilitar el uso del subsistema, sirviendo de intermediaria entre el cliente y un grupo de interfaces más complejas. (49) Será aplicado en la implementación del sistema dicho patrón, para facilitar el uso de las clases controladoras de la capa de negocio y los controladores genéricos para JPA, concentrando el acceso a sus métodos hacia la clase **FacadeImpl**, permitiendo un mejor entendimiento a los programadores sobre el código que usan las librerías y las clases, haciendo más flexible el desarrollo.

## 3.2 Modelado de diseño

El modelo de diseño crea un modelo de software enfocado en la representación de los datos, las funciones y el comportamiento requerido. Permite al ingeniero de software modelar el sistema o producto que se va a construir, posibilitando evaluar su calidad y efectuar mejoras antes de generar el código. Con este se obtiene una representación arquitectónica, de interfaz y despliegue del sistema. (40)

### 3.2.1 Diagramas de clases del diseño

Estos diagramas representan la vista estática del sistema y modelan los conceptos asociados al dominio de la aplicación así como los conceptos internos definidos para la parte de la implementación. No se describe el comportamiento del sistema dependiente del tiempo y se representa mediante clases y sus relaciones. (50)

Se presenta a continuación el diagrama de clases del diseño del caso de uso Solicitar material, donde se observan los paquetes que agrupan las clases participantes y sus respectivas relaciones. El resto de los diagramas se encuentran en el Anexo 14, se han diseñado los diagramas más representativos según la necesidad del equipo de desarrollo.

# CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

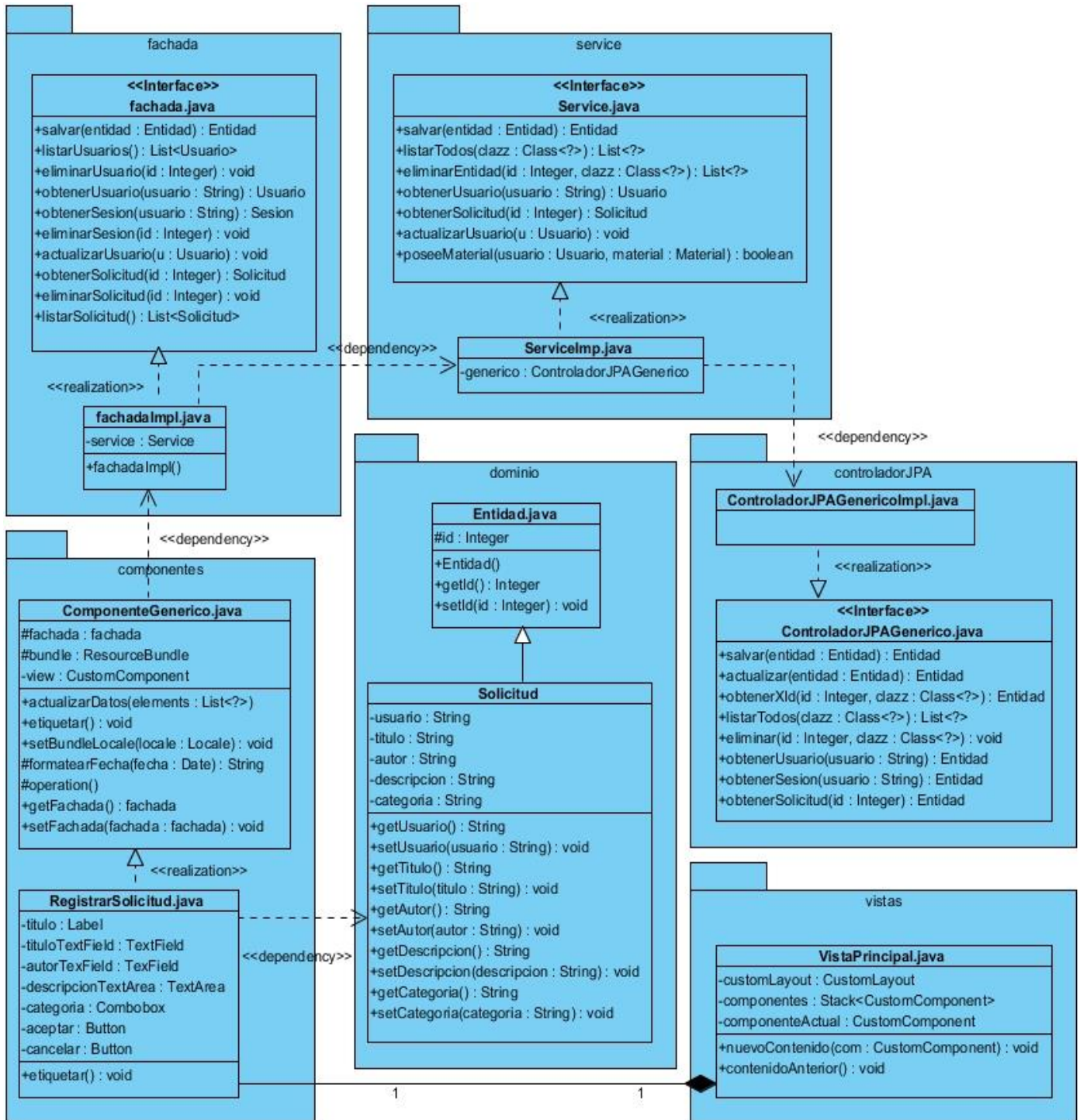


Figura 4. Diagrama de clases del diseño del caso de uso Solicitar material.

# CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

## 3.2.2 Diagramas de secuencia

Un diagrama de secuencia muestra un conjunto de mensajes, dispuestos en una secuencia temporal, donde cada rol en la secuencia se muestra como una línea de vida que representa el rol durante cierto plazo del tiempo, con la interacción completa. Los mensajes se muestran como flechas entre las líneas de vida. Un diagrama de secuencia puede mostrar un escenario, es decir una historia individual de una transacción. (50)

Este artefacto permitirá representar la interacción de los objetos que intervienen en una funcionalidad, mediante la transferencia de mensajes o métodos; le indicarán además al equipo de desarrollo en qué momento un objeto determinado toma el control del sistema mediante su línea de vida.

A continuación se presenta el diagrama de secuencia del caso de uso Solicitar material, el resto de los diagramas se encuentran en el Anexo 16.

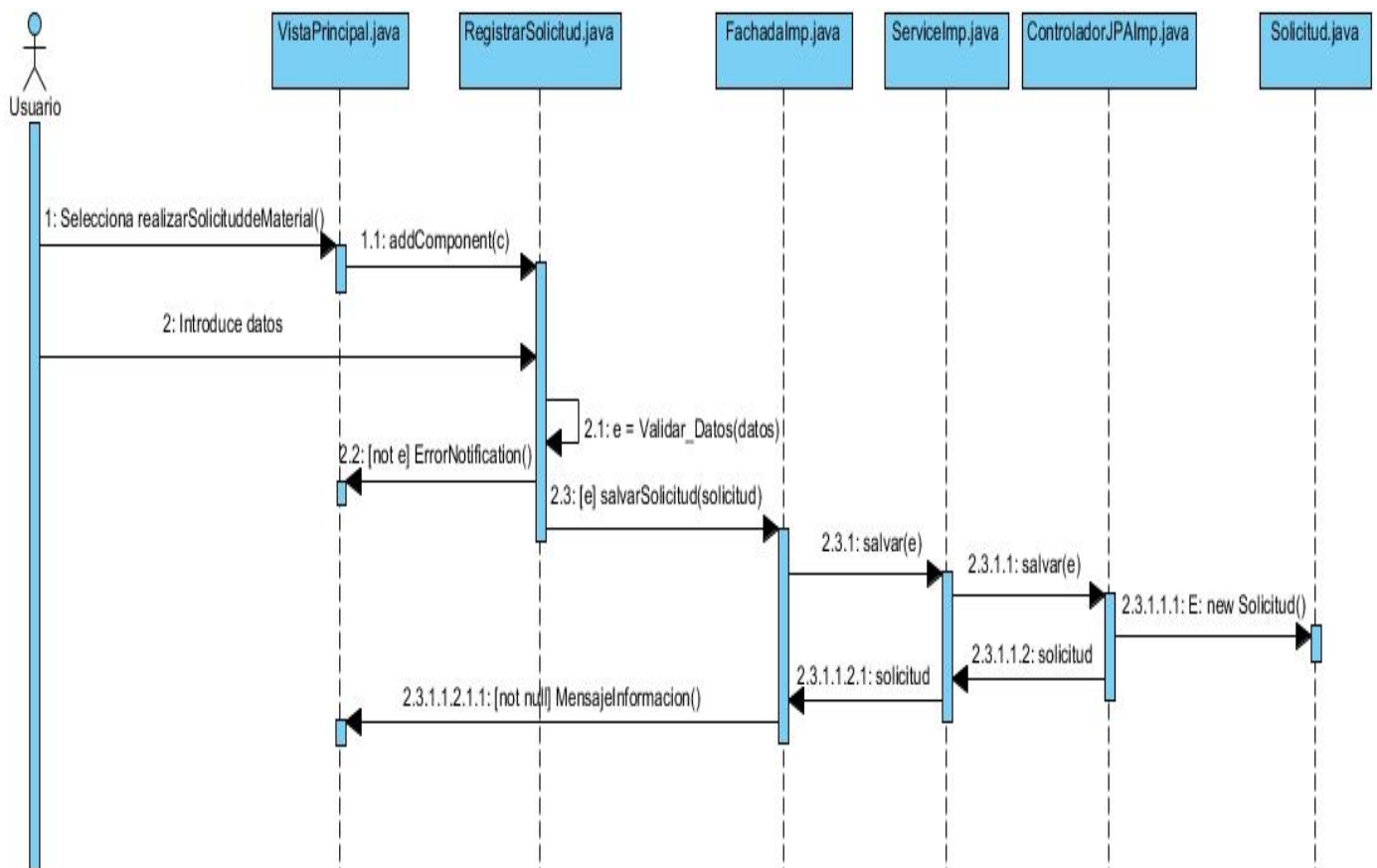


Figura 5. Diagrama de secuencia del caso de uso Solicitar material.

# CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

## 3.2.3 Vista de despliegue

La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos. Un nodo es un recurso de ejecución, tal como una computadora, un dispositivo o una memoria. Esta vista permite determinar la secuencia de distribución y asignación de recursos. Para su representación se emplea el artefacto diagrama de despliegue. (50)

A continuación se muestra el diagrama de despliegue del sistema:

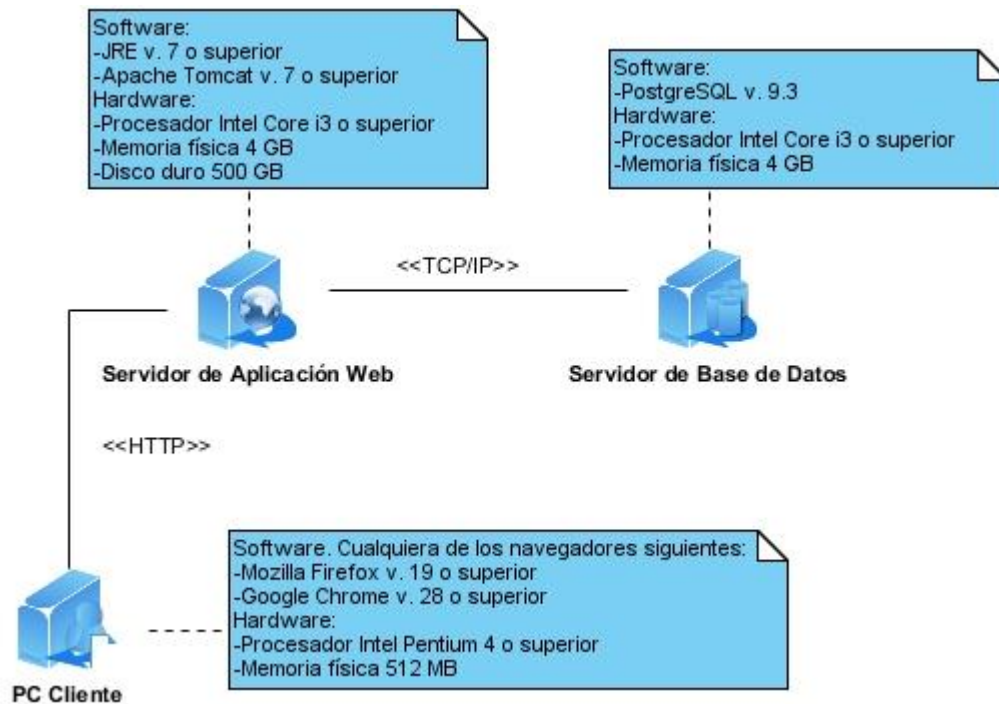


Figura 6. Diagrama de despliegue del sistema.

## 3.2.4 Diagramas de componentes

Los diagramas de componentes permiten modelar la vista de implementación del sistema a partir del cual se construye la aplicación, se representan las dependencias entre los componentes para poder determinar el impacto de un cambio y modela además la asignación de clases y otros elementos del modelo a los componentes. (50)

La siguiente figura muestra el diagrama de componentes del caso de uso Solicitar material, el resto de los diagramas de este tipo se encuentran en el Anexo 22.



## CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

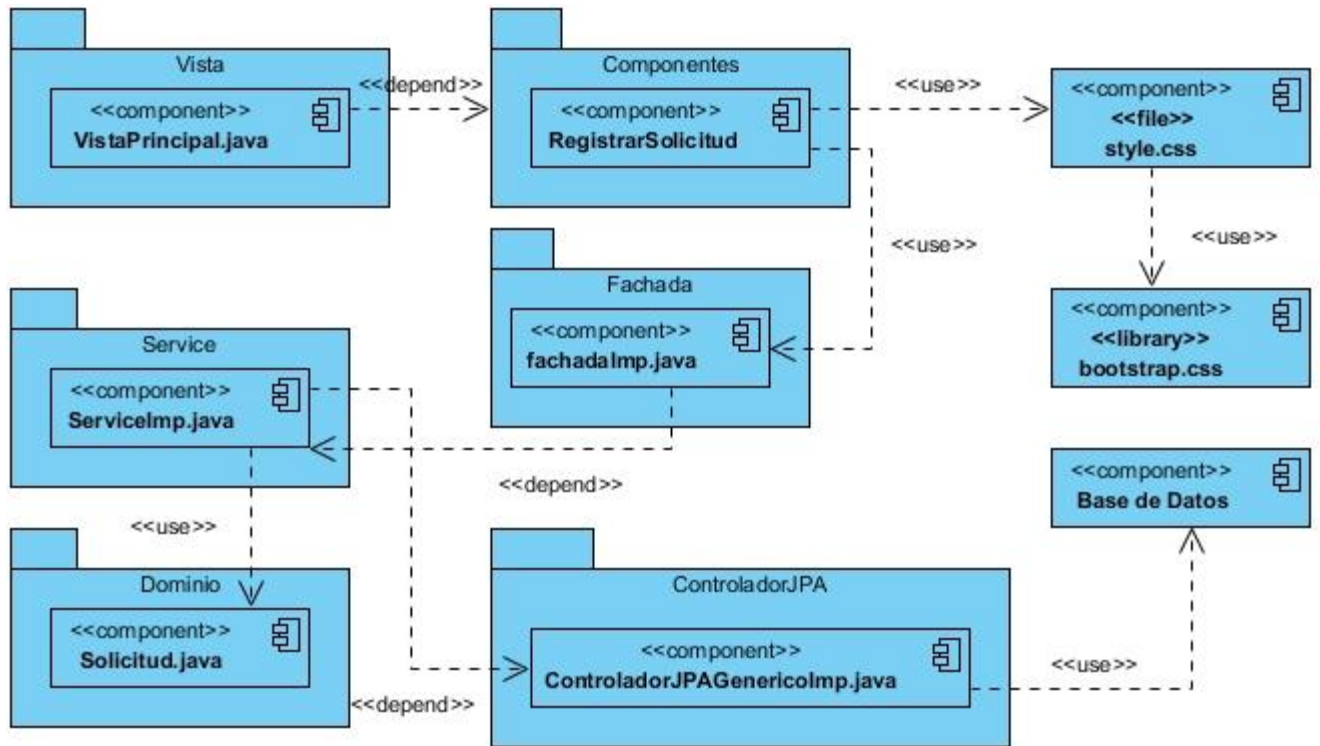


Figura 7. Diagrama de componentes del caso de uso Solicitar material.

### 3.3 Pruebas de software

Las pruebas del software son una técnica dinámica de validación y verificación (V&V), las cuales implican ejecutar una implementación del software con datos de prueba, se examinan las salidas del software y su entorno operacional para comprobar que funciona tal y como se requiere. (44)

Al ejecutarse esta actividad se persigue descubrir defectos en el sistema que hagan que este tenga un comportamiento incorrecto o no deseable y para verificar que cumple con los requerimientos del cliente, con el fin de suplir sus necesidades. Con esto no se obtiene un sistema totalmente libre de errores, pero si apto para ser usado por el usuario final.

#### 3.3.1 Niveles de prueba

El proceso de pruebas se realiza en varios niveles, estos se enfocan a determinados objetivos y están estrechamente relacionados con los tipos de pruebas. Se muestran a continuación algunos niveles de pruebas para valorar cuales serán utilizados:

## CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

**Pruebas unitarias:** En este nivel se verifican por separados las piezas de software en un funcionamiento aislado, estas piezas pueden ser módulos individuales, subprogramas y componentes. Para realizarlas pueden llevarse a cabo con herramientas de depuración, acceso al código fuente y pueden participar en esta de forma opcional los programadores. (51)

**Pruebas de integración:** Se verifica la interacción de los componentes del sistema de software, pueden ser guiadas por la arquitectura donde se integran los componentes según los hilos de funcionalidad, o incremental donde se combina el siguiente módulo a integrar con el resto de los ya integrados. La manera incremental puede realizarse de forma ascendente (*Bottom Up*) o descendente (*Top Down*). (51)

➤ Incremental ascendente

- Se comienza por los módulos hoja (pruebas unitarias)
- Se combinan los módulos según la jerarquía
- Se repite en niveles superiores

➤ Incremental descendente

- Primero en profundidad, completando ramas del árbol
- Primero en anchura, completando niveles de jerarquía

**Pruebas de sistema:** Verifican el funcionamiento del sistema en su conjunto, este nivel permite comprobar requisitos no funcionales así como utilidades, unidades físicas y entornos operativos después que el hardware y el software han sido integrados. Para ejecutarla, se prueba un incremento que será entregado al cliente, verificándose que este satisface los requerimientos y se asegura que el sistema es confiable y apto para su uso final. (51)

Para la ejecución de las actividades de prueba no se aplicarán las pruebas de integración debido a que al sistema no se le incorporarán componentes o módulos desarrollados independientemente de la solución. Serán aplicadas las pruebas unitarias para poder validar las funcionalidades con las que cuenta el sistema mediante el uso de la herramienta de depuración que brinda el IDE Eclipse Kepler y estas serán ejecutadas por el propio equipo de desarrollo. Se ejecutarán las pruebas de sistema una vez que esté concluido, para verificar que cumple con los requerimientos funcionales y no funcionales, dejándolo listo para su uso final.



## CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

### 3.3.2 Métodos de prueba

Se puede probar cualquier producto de ingeniería de dos formas: Conociendo la funcionalidad específica para la cual fue diseñado, demostrando mediante pruebas que cada funcionalidad es operativa o conociendo el funcionamiento del producto, donde cada componente interno trabaja de la forma adecuada. Al primer enfoque se le denomina prueba de caja negra y al segundo prueba de caja blanca. (40) Para la validación de la propuesta de solución será aplicado el método de caja negra, debido a que este permitirá corregir problemas de rendimiento, en la interfaz del usuario y se comprobará que la propuesta de solución realiza las funciones requeridas por el usuario.

**Prueba de caja negra:** Se realizan las pruebas sobre la interfaz del programa a desarrollar, entiéndase por interfaz de entrada y salida del programa, no requieren del conocimiento de la lógica del programa y solamente se debe conocer la funcionalidad que este debe realizar. (52)

### 3.3.3 Partición equivalente

Es una técnica de caja negra que se basa en dividir en subconjuntos equivalentes respecto a una relación específica (clase de equivalencia) el dominio de las entradas. La prueba de un valor representativo de una clase permite suponer que el resultado obtenido será el mismo que para otro valor de la clase. Se realiza un conjunto representativo de casos de prueba para cada clase de equivalencia. (51)

Para aplicar correctamente esta técnica el equipo debe diseñar cada caso de prueba de forma tal que descubra una clase determinada de errores, para así reducir el número de casos a ejecutar, obteniéndose solo los necesarios.

Esta técnica constituye una aproximación sistemática al diseño de casos de prueba, pues se basa en identificar las particiones para un sistema o componente y los casos de prueba se diseñan para que las entradas y salidas pertenezcan a estas particiones. (44)

### 3.3.4 Diseño de casos de prueba

El diseño de casos de prueba es una parte de las pruebas de componentes y sistemas en las que se diseñan las entradas y las salidas esperadas para probar el sistema, el objetivo que persigue es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface los requerimientos. (44)

## CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

A continuación se muestran los diseños de casos de prueba de los casos de uso Solicitar material y Gestionar solicitud de materiales, el resto de los artefactos de este tipo se encuentran en el Anexo 24.

**Tabla 4** Diseño de casos de prueba del caso de uso Solicitar material

<b>CP Solicitar material</b>							
<b>Descripción general</b>							
El caso de uso inicia cuando el actor decide solicitar un material al sistema. Para ello el actor selecciona la opción Solicitar material en el menú Servicios. El sistema abre un formulario con los campos correspondientes a una solicitud. Permite crear una nueva solicitud en el sistema o cancelar la operación, finalizando el caso de uso.							
<b>Condiciones de ejecución</b>							
El usuario debe estar autenticado.							
<b>SC 1 Solicitar material</b>							
Escenario	Descripción	Título	Descripción	Categoría	Respuesta del sistema	del	Flujo Central
<b>EC1.1</b>	Selecciona la opción Solicitar material en el menú Servicios.				El sistema brinda la posibilidad de insertar los siguientes datos <ul style="list-style-type: none"> <li>• Título del Material</li> <li>• Autor del Material</li> <li>• Descripción del Material</li> </ul> Permite seleccionar: <ul style="list-style-type: none"> <li>• Categoría, con los valores:                             <ul style="list-style-type: none"> <li>-Libro</li> <li>-Audiovisual</li> </ul> </li> </ul>		Inicio/Solicitar material

### CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

					-Artículo Y las opciones: <ul style="list-style-type: none"> <li>• Aceptar</li> <li>• Cancelar</li> </ul>	
<b>EC 1.2</b>	Introduce datos y selecciona Aceptar	V	V	V	Valida los datos introducidos. Registra una nueva solicitud de material en el sistema. El sistema muestra un mensaje de éxito.	Inicio/Solicitar material /Aceptar
<b>EC1.3</b>	Datos incompletos.	I	V	N/A	El sistema muestra un mensaje “Existen campos vacíos”.  Muestra el indicador de los campos que no pueden ser vacíos.  Regresa al 1.1	Inicio/Solicitar material /Aceptar
		V	I	N/A		
		V	V	N/A		
<b>EC1.4</b>	No se ha elegido una categoría	N/A	N/A	I	El sistema muestra un mensaje “Debe elegir una categoría”. Regresa al 1.1	Inicio/Solicitar material /Aceptar
<b>EC1.6</b>	Selecciona la opción Cancelar.				Regresa a la vista anterior.	Inicio/Solicitar material /Aceptar

## CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

Tabla 5. Diseño de casos de prueba del caso de uso Gestionar solicitud de materiales

<b>CP Gestionar solicitud de materiales</b>			
<b>Descripción general</b>			
<p>El caso de uso inicia cuando el actor accede a la ventana inicial de administración del sistema y accede a la opción Lista de solicitudes en el menú Atender solicitud. Muestra un listado con las solicitudes registradas en el sistema y por cada una permite: ver, atender y eliminar solicitud. El actor visualiza, atiende o elimina una solicitud finalizando así el caso de uso.</p>			
<b>Condiciones de ejecución</b>			
El administrador debe estar autenticado.			
<b>SC1 Visualizar solicitud</b>			
Escena rio	Descripción	Respuesta del sistema	Flujo Central
<b>EC1.1</b>	Selecciona la opción Lista de solicitudes en el menú Atender solicitud.	<p>El sistema muestra una lista cuyos elementos representan los datos de cada solicitud almacenada en el sistema.</p> <p>Y permite:</p> <ul style="list-style-type: none"> <li>• Visualizar solicitud.</li> <li>• Atender solicitud. Ver SC2: “Atender solicitud”.</li> <li>• Eliminar solicitud. Ver SC3: “Eliminar solicitud”.</li> </ul>	Administración /Lista de solicitudes
<b>EC1.2</b>	Selecciona la opción Visualizar solicitud	<p>El sistema muestra una ventana auxiliar con los datos de la solicitud, sin poder modificar los mismos.</p> <p>Brinda la opción:</p> <ul style="list-style-type: none"> <li>• Cerrar</li> </ul>	Administración /Lista de solicitudes/ Visualizar solicitud

## CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

<b>EC1.3</b>	Selecciona la opción Cerrar.	Cierra la ventana actual.  Retorna al 1.1.	Administración /Lista de solicitudes/ Visualizar solicitud /Cerrar
<b>SC2 Atender solicitud</b>			
<b>Escena</b>	<b>Descripción</b>	<b>Respuesta del sistema</b>	<b>Flujo Central</b>
<b>EC2.1</b>	Selecciona la opción Atender solicitud.	Envía un mensaje de correo electrónico al usuario solicitante indicando que la petición ya fue atendida por el administrador del sistema.  Muestra un mensaje de éxito.	Administración /Lista de solicitudes/Ate nder solicitud
<b>SC3 Eliminar solicitud</b>			
<b>Escena</b>	<b>Descripción</b>	<b>Respuesta del sistema</b>	<b>Flujo Central</b>
<b>EC3.1</b>	Selección la opción Eliminar solicitud.	Muestra un mensaje solicitando la confirmación del usuario de dicha acción, permitiendo elegir: <ul style="list-style-type: none"> <li>• Aceptar</li> <li>• Cancelar</li> </ul>	Administración/ Lista de solicitudes/Elimi nar
<b>EC3.2</b>	Selecciona la opción Aceptar	Elimina la solicitud del sistema  Muestra un mensaje de éxito.	Administración/ Lista de solicitudes

## CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

			/Eliminar/Aceptar
<b>EC3.3</b>	Selecciona la opción Cancelar.	Cierra el diálogo actual.	Administración/Lista de solicitudes/Eliminar/Cancelar

### 3.3.5 Resultados de las pruebas de caja negra

Para la realización de las pruebas del sistema se aplicó el método de caja negra sobre las interfaces gráficas del sistema, con la ejecución de las mismas se pudo constatar el cumplimiento de los requisitos funcionales y no funcionales del software obtenido.

A continuación se presenta el resultado de las pruebas realizadas:

**Tabla 6. Resultados de las pruebas por iteración**

Iteraciones	Cantidad de casos de prueba	No conformidades detectadas			
		Alta	Media	Baja	Total
1	15	4	11	15	30
2	15	1	12	7	20
3	15	1	3	3	7

La tabla anterior muestra la ejecución de tres iteraciones, donde en cada iteración se le da solución a las no conformidades detectadas, mejorando la calidad del software y preparándolo progresivamente para su uso final.

### 3.3.6 Pruebas de rendimiento

Para realizar las pruebas de rendimiento al portal, se seleccionó la herramienta **Apache JMeter 2.13**, debido a que es una aplicación de código abierto, diseñada para medir el comportamiento funcional de aplicaciones

## CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

web, mediante pruebas de carga. Posee un bajo nivel de complejidad y está especialmente diseñada para Java. Con esta herramienta se puede simular una carga pesada en un servidor o grupo de servidores y obtener diferentes métricas. (53)

Para las pruebas al portal se tendrán en cuenta las siguientes métricas:

**Muestra:** Cantidad de usuarios que realizan una petición HTTP (*Hypertext Transfer Protocol*, siglas en inglés).

**Media:** Tiempo de respuesta promedio que demora el servidor en responder una petición en milisegundos.

**Error:** Porcentaje de error del servidor al responder las peticiones HTTP.

En una primera iteración de las pruebas realizadas al portal, se utilizaron tres tipos de muestras, la primera para 100 usuarios, la segunda para 1000 y por último 5000. Estas cifras fueron seleccionadas al evaluar un aproximado del número de usuarios máximos que puedan concurrir en la conexión con el portal.

Tabla 7. Resultados de las pruebas de rendimiento para la primera iteración

Etiqueta	Muestra	Media	Error
Petición HTTP	100	2388	0 %
Total	100	2388	0 %
Etiqueta	Muestra	Media	Error
Petición HTTP	1000	4125	36.6 %
Total	1000	4125	36.6 %
Etiqueta	Muestra	Media	Error
Petición HTTP	5000	5632	48.26 %
Total	5000	5632	48.6 %

## CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

En esta iteración a medida que aumentaba la cantidad de usuarios realizando peticiones a la aplicación, aumentaba el porcentaje de error, debido un fallo que no fue detectado en la etapa de pruebas de caja negra, la aplicación lanzaba una excepción informando la existencia de un objeto nulo, que no impedía el correcto funcionamiento de la misma, pero si influía de forma directa en la respuesta que daba el servidor a las peticiones de conexión de los usuarios, donde en algunos casos estas se perdían o se denegaba el acceso al sitio.

A continuación se muestra la segunda iteración de este tipo de pruebas, después de realizar reajustes en el código para erradicar el error detectado.

**Tabla 8. Resultados de las pruebas de rendimiento para la segunda iteración**

Etiqueta	Muestra	Media	Error
Petición HTTP	100	2112	0 %
Total	100	2112	0 %
Etiqueta	Muestra	Media	Error
Petición HTTP	1000	3821	1.6 %
Total	1000	3821	1.6 %
Etiqueta	Muestra	Media	Error
Petición HTTP	5000	4266	2.2 %
Total	5000	4266	2.2 %

Como se evidencia en los resultados, la disminución del error fue considerable hasta el punto de encontrarse por debajo de un 3 %, el cual no constituye ningún problema, estos valores están dados por fluctuaciones en la red y por encontrarse la aplicación desplegada en un servidor de prueba local.



## CAPÍTULO 3. DESARROLLO DE LA PROPUESTA DE SOLUCIÓN

### 3.4 Conclusiones parciales

El uso de patrones de diseño permitió aplicar buenas prácticas en el proceso de modelado e implementación del software, logrando una mayor reutilización y mantenibilidad del código. La realización del modelado del sistema permitió obtener los diagramas de clases del diseño, secuencia y despliegue, los cuales le indicaron al equipo de desarrollo los objetos que se deben construir, los atributos y comportamientos de las clases, su relación y la forma en la que interactúan las mismas desde que el usuario realiza una acción. El proceso de implementación estuvo guiado por la elaboración de los diagramas de componentes de los casos de uso más representativos, con lo que se representó la vista de implementación del sistema y se obtuvo al finalizar dicha fase el producto final con las funcionalidades requeridas. Para la validación del software se realizaron las pruebas al sistema mediante la técnica de caja negra, con esta se detectaron a tiempo un grupo de no conformidades en cada iteración, las cuales fueron resueltas al final de cada una. Las pruebas de rendimiento fueron realizadas mediante la herramienta Apache JMeter 2.13, donde se detectaron errores en el servidor y se comprobaron los tiempos de respuesta, mejorando progresivamente la calidad del sistema al resolver los mismos.

## Conclusiones generales

Con la realización del presente trabajo se arribaron a las siguientes conclusiones:

- Se realizó un estudio de las características de los CRAI evidenciando la necesidad de crear un portal para mejorar la gestión de los servicios en línea para los usuarios de la biblioteca de la Facultad de Economía de la Universidad de La Habana mediante la utilización de herramientas libres y multiplataforma.
- Se construyó un portal web que brindará los servicios de un Centro de Recursos para el Aprendizaje y la Investigación en línea, mediante el cual los usuarios podrán gestionar materiales, servicios e información referente a su biblioteca y centro de estudio.
- Se realizaron pruebas al sistema mediante la técnica de caja negra, detectando un total de 57 no conformidades en tres iteraciones, las cuales fueron resueltas al finalizar cada iteración, mejorando la calidad del sistema.
- El cliente quedó satisfecho con la solución informática entregada, la misma se implantará en la biblioteca de la Facultad de Economía de la Universidad de La Habana y posibilitará atender las necesidades informacionales de los usuarios de dicha institución.

## Recomendaciones

Teniendo en cuenta los resultados obtenidos en el presente trabajo, se proponen las siguientes recomendaciones:

- Crear un manual de usuario para la administración del sistema.
- Incorporar un módulo estadístico, que incluya gráficas de diferentes tipos y muestre estadísticas relacionadas con la utilización de recursos en línea, la solicitud de materiales y las líneas de investigación de mayor demanda.
- Mejorar el servicio de recomendación de materiales al usuario, mediante el uso de técnicas de inteligencia artificial.

### Referencias Bibliográficas

1. **Martín Gavilán, César.** Bibliotecas universitarias: concepto y función los CRAI. 2008, pág. 3.
2. **Universidad de las Ciencias Informáticas.** Historia. *Portal de la Universidad de las Ciencias Informáticas*. [En línea] [Citado el: 4 de 3 de 2015.] <http://www.uci.cu/?q=historia>.
3. **Quevedo Pacheco, Nelva Liliana.** *Alfabetización informacional en las bibliotecas universitarias : Conceptos y experiencia*. Universidad de Lima : s.n.
4. **Universidad Nacional de Cuyo.** ¿Qué es ALFIN? *Sistema Integrado de Documentación*. [En línea] [Citado el: 1 de 06 de 2015.] <http://sid.uncu.edu.ar/sid/alfin/>.
5. **Zamora Fonseca, MSc. Raquel.** *TESIS DOCTORAL. Propuesta de una metodología para la implementación de los Centros de Recursos para el Aprendizaje y la Investigación en el Sistema de Educación Superior Cubano*. Editorial de la Universidad de Granada. 2013. págs. 29-32. ISBN: 978-84-9028-716-3.
6. **González Guitián, María Virginia y Molina Piñero, Maricela.** Las bibliotecas universitarias: breve aproximación a sus nuevos escenarios y retos. [En línea] 2008. [Citado el: 6 de 12 de 2014.] <http://scielo.sld.cu/pdf/aci/v18n2/aci02808.pdf>.
7. **Sánchez Díaz, Lic. Marlery y Vega Valdés, Dr. Juan Carlos.** *Bibliotecas electrónicas, digitales y virtuales: tres entidades por*. La Habana : ACIMED, 2002. ISSN 1024-9435.
8. **Universidad Austral de Chile.** Catálogos. *Sistemas de bibliotecas UACH*. [En línea] [Citado el: 3 de 06 de 2015.] [http://www.biblioteca.uach.cl/catalogos/otros.htm#c\\_hispano](http://www.biblioteca.uach.cl/catalogos/otros.htm#c_hispano).
9. **EDUKANDA.** Recursos electrónicos. *EDUKANDA*. [En línea] [Citado el: 03 de 06 de 2015.] [http://www.edukanda.es/mediatecaweb/data/zip/627/PID\\_00143755/web/main/m1/v1\\_4.html](http://www.edukanda.es/mediatecaweb/data/zip/627/PID_00143755/web/main/m1/v1_4.html).
10. **Definición ABC.** Definición de Libro electrónico. *Definición ABC*. [En línea] [Citado el: 03 de 06 de 2015.] <http://www.definicionabc.com/tecnologia/libro-electronico.php>.
11. **REBIUN, CRUE.** DE LAS BIBLIOTECAS UNIVERSITARIAS A LOS CENTROS DE RECURSOS PARA EL APRENDIZAJE Y LA INVESTIGACION. [En línea] 2005. [Citado el: 6 de 12 de 2014.] <http://www.rebiun.org/doc/z2.pdf>.

## REFERENCIAS BIBLIOGRÁFICAS

12. **González-Santos, Odalys y Bicet-Álvarez, Edgar.** *La biblioteca universitaria: de la conservación al CRAI.* Isla de la Juventud : s.n., 2013. ISSN: 0864-4659.
13. **Fonseca, M.Sc. Raquel Zamora.** *Cambios operados en la implementación de un Centro de Recursos para el Aprendizaje y la Investigación: experiencia en la Universidad de Cienfuegos.* 2012.
14. **Login Desarrollos.** Login Desarrollos. *Desarrollo de Aplicaciones Web.* [En línea] [Citado el: 7 de 2 de 2015.] <http://www.logindesarrollos.com/es/Servicios-aplicaciones-web-21>.
15. **Barzanallana, Rafael.** Universidad de Murcia. *Metodologías de desarrollo de software.* [En línea] 2 de 6 de 2005. [Citado el: 1 de 12 de 2014.] <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.
16. **Boeras Velázquez, Mairelys, Cabrera Barroso, Laritza y Llano Castro, Eileén.** *Aplicando el método de Boehm y Turner.* La Habana : s.n., 2012.
17. **Penadés, Patricio Letelier y M. Carmen.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* págs. 6-8.
18. **Ambler, Scott.** The Agile Unified Process (AUP). *Ambyssoft.* [En línea] [Citado el: 4 de 11 de 2015.] <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.
19. **McGraw-Hill Interamericana de España, SL.** Sistemas gestores de bases de datos. *McGraw-Hill Interamericana de España, SL.* [En línea] [Citado el: 2 de 12 de 2014.] <http://www.mcgraw-hill.es/bcv/guide/capitulo/8448148797.pdf>.
20. **Microsoft.** SQL Server . *Microsoft.* [En línea] [Citado el: 7 de 2 de 2015.] <http://www.microsoft.com/es-es/server-cloud/products/sql-server/>.
21. **MySQL.** Introduction. *MySQL.* [En línea] [Citado el: 27 de 10 de 2014.] <http://dev.mysql.com/doc/refman/5.7/en/introduction.html>.
22. **Oracle.** Suport. *MySQL. The world's most popular open source database.* [En línea] Oracle. [Citado el: 3 de 4 de 2015.] <http://www.mysql.com/support/>.
23. **PostgreSQL.** Sobre PostgreSQL. *PostgreSQL-es: Portal en español sobre PostgreSQL.* [En línea] 2 de 10 de 2010. [Citado el: 7 de 2 de 2015.] [http://www.postgresql.org.es/sobre\\_postgresql#intro](http://www.postgresql.org.es/sobre_postgresql#intro).

## REFERENCIAS BIBLIOGRÁFICAS

24. **Jojooa Group.** Definición de Lenguajes de Programación del Lado del Cliente - ¿Qué son los lenguajes de programación del lado del cliente? *Jojooa - tecnología, marketing y crm.* [En línea] 2005. [Citado el: 29 de 5 de 2015.] <https://sites.google.com/site/jojooa/informatica-tecnologia/definicion-de-los-lenguajes-de-programacion-del-lado-del-cliente-que-son-los-lenguajes-de-programacion-del-lado-del-cliente>.
25. **Google Inc.** Documentos de Google. *Google.* [En línea] <https://docs.google.com/presentation/d/1ZQglsW4KCylfH8NyiQH9ueanM83Ym7bjitaxyQKS0hs/edit?pli=1#slide=id.i56>.
26. **Zahumenszky, Carlos.** ¿Qué lenguaje de programación debería aprender para empezar? *GIZMODO.* [En línea] 12 de 09 de 2013. [Citado el: 10 de 10 de 2014.] <http://es.gizmodo.com/que-lenguaje-de-programacion-deberia-aprender-para-emp-1479554075>.
27. **Grönroos, Marko.** *Book of Vaadin 7ma Edicion.* s.l. : Creative Commons CC-BY-ND, 2014.
28. **Alegsa.** Diccionario de Informática y Tecnología - Definición de framework. *alegsa.* [En línea] [Citado el: 2 de 12 de 2014.] <http://www.alegsa.com.ar/Dic/framework.php>.
29. **Java Hispano.** Los 4 Framework Web Java más usados. *Java Hispano.* [En línea] 20 de 8 de 2014. [Citado el: 10 de 1 de 2015.] <http://www.javahispano.org/portada/2014/8/20/los-4-framework-web-java-mas-usados.html>.
30. **Bootstrap.** Introducing Bootstrap. *Bootstrap.* [En línea] [Citado el: 7 de 2 de 2015.] <http://getbootstrap.com/2.3.2/>.
31. **Marco, David.** Introducción a JPA 2.0 (I). *DM davidmarco.es.* [En línea] 24 de 1 de 2010. [Citado el: 7 de 2 de 2015.] <http://www.davidmarco.es/articulo/introduccion-a-jpa-2-0-i>.
32. **Universidad Carlos III.** Interfaz de Persistencia Java (JPA) - Entidades y Managers. *EjemplosTIW.* [En línea] [Citado el: 2 de 06 de 2015.] <http://www.lab.inf.uc3m.es/~a0080802/RAI/jpa.html>.
33. **The Eclipse Foundation.** Eclipse Kepler. *Eclipse.* [En línea] 26 de 6 de 2013. [Citado el: 7 de 2 de 2015.] <http://eclipse.org/kepler/>.
34. **Apache Maven Project.** Apache Maven Project. [En línea] [Citado el: 4 de 12 de 2014.] <http://maven.apache.org/>.

## REFERENCIAS BIBLIOGRÁFICAS

35. **GlassFish**. Comparing GlassFish Open Source Edition versions 2.x and 3.0.x. *GlassFish - World's first Java EE 7 Application Server*. [En línea] [Citado el: 4 de 12 de 2014.] [https://glassfish.java.net/public/comparing\\_v2\\_and\\_v3.html](https://glassfish.java.net/public/comparing_v2_and_v3.html).
36. **Red Hat**. JBoss Application Server. *Red Hat*. [En línea] [Citado el: 4 de 12 de 2014.] <http://www.redhat.com/es/technologies/jboss-middleware/application-platform>.
37. **The Apache Software Foundation**. Apache Tomcat 7. [En línea] [Citado el: 4 de 12 de 2014.] <http://tomcat.apache.org/tomcat-7.0-doc/index.html>.
38. **González Cornejo, José Enrique**. *¿Qué es UML? El lenguaje de modelado Unificado*. 2008.
39. **Visual Paradigm**. Features List. [En línea] [Citado el: 5 de 12 de 2014.] <http://www.visual-paradigm.com/features/>.
40. **Pressman, Roger S**. *Ingeniería de software: Un enfoque práctico 5ta edición*. s.l. : McGraw Hill, 2002.
41. **Camacho, Erika y Cardeso, Favio**. *Arquitectura de software*. 2004. pág. 21.
42. **Managinf**. ARQUITECTURA DE TRES CAPAS - Managinf. [En línea] 2006. [www.managinf.com/arquitectura.pdf](http://www.managinf.com/arquitectura.pdf).
43. **Larman, Craig**. *UML y Patrones. Introducción al análisis orientado a objetos*. págs. 85-88.
44. **Sommerville, Ian**. *Ingeniería de software 7ma edición*. s.l. : Pearson-Addison Wesley, 2005.
45. **Cáceres Tello, Jesús**. *Diagramas de Casos de Uso*. Universidad de Alcalá : s.n.
46. **Gunnar Övergaard, Karin Palmkvist**. *Use Cases Patterns and Blueprints*. s.l. : Addison Wesley Professional, 2004.
47. **Olivares Rojas, Juan Carlos**. *Patrones de diseño*. Sistema Nacional de Educación Superior Tecnológica : s.n.
48. **Gamma, Erich y Helm, Richard**. *Design Patterns: Elements of Reusable Object-Oriented Software*. 1995. ISBN 0-201-63361-2.
49. **Mühlrad, Daniel**. *Patrones de diseño*. 2008.

## REFERENCIAS BIBLIOGRÁFICAS

50. **Rumbaugh, James, Jacopson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado. Manual de referencia.* s.l. : Addison Wesley.
51. **Blanco Bueno, Carlos.** Ingeniería de Software - Construcción y pruebas. [En línea] [Citado el: 23 de 4 de 2015.] <http://ocw.unican.es/enseñanzas-tecnicas/ingenieria-del-software-ii/materiales/tema1-pruebasSistemasSoftware.pdf>.
52. **Juristo, Natalia, Moreno, Ana M. y Vegas, Sira.** *Técnicas de Evaluación de Software.* 2005.
53. **The Apache Software Foundation.** Apache JMeter TM. *Apache JMeter TM.* [En línea] [Citado el: 7 de 06 de 2015.] <http://jmeter.apache.org>.



## Bibliografía

1. **Universidad de Alicante.** Arquitectura cliente/servidor: Programación en internet. *Repositorio Institucional de la Universidad de Alicante*. [En línea] 2008. [Citado el: 9 de 12 de 2014.] <http://rua.ua.es/dspace/bitstream/10045/16740/4/03a-ArquitecturaClienteServidor.pdf>.
2. **IntelliJ IDEA.** Developing applications for Android in IntelliJ IDEA. *Wiki jetbrains*. [En línea] [Citado el: 4 de 12 de 2014.] [http://wiki.jetbrains.net/intellij/Developing\\_applications\\_for\\_Android\\_in\\_IntelliJ\\_IDEA](http://wiki.jetbrains.net/intellij/Developing_applications_for_Android_in_IntelliJ_IDEA).
3. **Area Moreira, Manuel.** *Introducción a la tecnología Educativa*. Universidad de La Laguna, España : Creative Commons, 2009.
4. **Universidad Politécnica de Cartagena.** Reproducción documental. *CRAI Universidad Politécnica de Cartagena*. [En línea] [Citado el: 29 de 05 de 2015.] <http://www.bib.upct.es/reproducciondocumental>.
5. **Serrano-Vicente, Rocío.** *Los Learning Centres en el Reino Unido. Estudio de caso de seis universidades medianas*. 2007.
6. **CRUE.** ¿Qué es REBIUN? *REBIUN*. [En línea] [Citado el: 1 de 06 de 2015.] <http://www.rebiun.org/queesrebiun/Paginas/default.aspx>.
7. **Ambyssoft Inc.** La diciplina de pruebas - Agile UP. *Ambyssoft Inc.'s Agile UP Product*. [En línea] 2006. [Citado el: 23 de 4 de 2015.] <http://www.cc.una.ac.cr/AUP/html/test.html>.
8. **BAUER, CHRISTIAN y King, Gavin.** *Java Persistence with Hybernate*. New York : MANNING. ISBN 1-932394-88-5.
9. **Delphiaccess.** El-top-ten-de-popularidad-en-bases-de-datos. [En línea] [Citado el: 27 de 10 de 2014.] <http://elpoli.delphiaccess.com/el-top-ten-de-popularidad-en-bases-de-datos/>.
10. **Deister.** Query\_Planning. *Deister*. [En línea] [Citado el: 27 de 10 de 2014.] [http://kb.deister.net/index.php/Query\\_Planning](http://kb.deister.net/index.php/Query_Planning).
11. **MySQL.** Features. *MySQL*. [En línea] [Citado el: 27 de 10 de 2014.] <http://dev.mysql.com/doc/refman/5.7/en/features.html>.

12. **World Wide Web Consortium.** Guía Breve de XHTML. *W3C España*. [En línea] [Citado el: 2 de 06 de 2015.] <http://www.w3c.es/Divulgacion/GuiasBreves/XHTML>.
13. **Universidad de Cádiz.** Recursos electrónicos. *UCA Universidad de Cádiz*. [En línea] [Citado el: 03 de 06 de 2015.] <http://biblioteca.uca.es/recursosdeinformacion/recursosselectronicos>.

## Glosario de términos

**API:** Interfaz de Programación de Aplicaciones. Representa la capacidad de comunicación entre componentes de software, poseen un conjunto de subrutinas, procedimientos, funciones o métodos ofrecidos por una biblioteca para ser utilizados por otro software.

**CSS:** Hoja de estilo en cascada. Es un mecanismo simple que permite definir y crear la presentación de un documento HTML o XML. La idea inicial de su creación era separar la estructura de un documento de su presentación. Permite a los desarrolladores el control total del estilo y el formato de múltiples páginas web al mismo tiempo.

**Java EE:** Java Empresarial. Es una plataforma de programación desarrollada inicialmente por *Sun Microsystems*, permite crear y ejecutar software escritos en lenguaje Java. Contiene varias API que incluyen servicios con diferentes fines y especificaciones que poseen tecnologías para la web.

**JRE:** Entorno de ejecución de Java, siglas en inglés. Sistema que proporciona las bibliotecas, Máquina Virtual de Java (JMV, siglas en inglés) y otros componentes que permiten ejecutar applets y aplicaciones escritas en Java. Está conformado por las tecnologías de implementación *Java Plug-in* y *Java Web Start*, la primera permite ejecutar applets en un navegador y la segunda despliega aplicaciones independientes a una red.

**JSP:** *Java Server Pages*, siglas en inglés. Es una tecnología que permite a los desarrolladores mediante el uso del lenguaje Java crear documentos web dinámicos basados en HTML y XML. Para su ejecución requieren de servidores para Java web, tales como: *Apache Tomcat*, *JBoss* o *GlassFish*.

**LRC:** Centros de Recursos para el Aprendizaje, siglas en inglés. Son un modelo de las universidades norteamericanas y británicas, donde las bibliotecas tienen una posición central dentro del proceso educativo. Estos integran servicios y recursos bibliotecarios, tecnológicos, audiovisuales, sistemas de información e instalaciones y medios para la edición electrónica y la creación de materiales interactivos.

**REBIUN:** Red de Bibliotecas Universitarias Españolas. Es una comisión sectorial de la Conferencia de Rectores de las Universidades Españolas (CRUE) desde 1998. Desde su creación, a iniciativa de un grupo de directores de bibliotecas en 1988, REBIUN constituye un organismo estable en el que están representadas todas las bibliotecas universitarias y científicas españolas.

**Reprografía de documentos:** Proceso mediante el cual se reproducen documentos mediante el empleo de varias técnicas como la fotocopia, serigrafía o fotografía. Tiene como fin obtener un número determinado de copias de un documento.

**RIA:** Aplicaciones de Internet Enriquecidas. Aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio, estos sistemas pueden ser ejecutados mediante un navegador web tradicional y requieren de complementos adicionales que permitan ejecutar sus características, generalmente máquinas virtuales.

**Tecnología educativa:** Es el modo sistemático de concebir, aplicar y evaluar el conjunto de procesos de enseñanza y aprendizaje, teniendo en cuenta a la vez los recursos técnicos, humanos y las interacciones entre ellos, como forma de obtener una más efectiva educación.

**XHTML:** Lenguaje de Marcadores e Hipertextos Extensible. Es una versión más estricta de HTML creada con el objetivo de reemplazar dicho lenguaje y avanzar hacia la web semántica. Combina la sintaxis de HTML 4.0 con XML, donde el primero está diseñado para mostrar datos y el segundo para describirlos.

**XML:** Lenguaje de Marcas Extensibles. Es un lenguaje de marcas que permite estructurar información en un documento o en cualquier fichero que contenga texto. Es una tecnología sencilla que propone un estándar para la comunicación de información entre diferentes plataformas, mejora la compatibilidad entre aplicaciones y provee soporte a las bases de datos.