

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 4**



**INGENIERÍA DE REQUERIMIENTOS DE UN SOFTWARE  
PARA SOPORTAR EL ENTORNO PERSONAL DE  
APRENDIZAJE DE LOS ESTUDIANTES DE LA UNIVERSIDAD  
DE LAS CIENCIAS INFORMÁTICAS.**

Trabajo de diploma para optar por el título de Ingeniero en Ciencias  
Informáticas

Autora: Taire Faife Rodríguez

Tutores: DrC. Dunia María Colomé Cedeño

Ing. Gustavo García González

La Habana, Cuba

2018

## Pensamiento

*“Esta universidad debe caracterizarse por la gran variedad de formas diferentes de enseñar, de preparar...centro docente experimental, centro docente-productor.”*

*Bichard*



## DECLARACIÓN DE AUTORÍA

Declaro ser la única autora del presente trabajo de diploma titulado "*Ingeniería de requerimientos de un software para soportar el entorno personal de aprendizaje de los estudiantes de la UCI*" y otorgo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los 19 días del mes de junio del año 2018.

---

Firma del Autor

Taire Faife Rodriguez

---

Firma del Tutor

DrC. Dunia María Colomé Cedeño

---

Firma del Tutor

Ing. Gustavo García González

## *Agradecimientos*

*A Fidel Castro Ruz, a la Revolución Cubana y a la Universidad de las Ciencias Informáticas, por haberme dado la oportunidad de estudiar en esta casa de altos estudios, por formarme como profesional, como revolucionaria y joven comunista.*

*A mi tutora Dunia, por ser mi guía, por todo el apoyo, por sus acertados consejos, por el cariño y todas sus enseñanzas. Por ser una madre y una amiga.*

*A mi tutor Gustavo por su paciencia, por enseñarme que el trabajo de diploma requiere sacrificio y dedicación.*

*A mami y papi, por todos los sacrificios, por quererme como a una hija. Porque cada línea de esta tesis fue escrita pensando en la satisfacción y el orgullo que sentirían cuando llegara este día.*

*A mis padres. A mi mamá por su constancia y su amor. Porque sin su apoyo y dedicación este sueño no hubiese sido posible. A mi papá por su ejemplo de vida, por darme alas para volar y saber cuándo debía emprender el vuelo sola.*

*A mi padrastro Jose por ser como un padre y siempre alentarme para que me superara profesionalmente.*

*A mi novio Daniel por tener tanta fe en mí, por ser mi amor, mi apoyo y mi refugio. Este trabajo de diploma es nuestro, porque juntos hemos superado todas las pruebas que nos ha puesto la vida, porque juntos hemos crecido...*

*A mis suegros, por estar siempre atentos al avance de este trabajo de diploma. Por quererme como una hija y cuidarme siempre.*

*A todos mis amigos, gracias por estar, en especial a Aniran, Gesly, Maydalis, Ariel, Elianys, Ernesto, Eli, Hermes, Alexis y Yunier.*

*A los que compartieron conmigo la dirección de la FEU y la UJC de la Facultad 4, en especial a Alvaro, Nay, Drake, Claudia, Daylilis, Luis Miguel, Javier y Hugo.*

*A cada uno de mis compañeros de aula por haber sido una gran familia.*

*A los profesores y trabajadores de la UCI, a los que de una forma u otra han aportado su granito de arena en mi formación personal y profesional.*

## *Dedicatoria*

*A mis hermanos de sangre y de sentimiento Tairí, Robert Julio y Talía. Por ser mi mayor inspiración, para que aprendan que los sueños se alcanzan con constancia y empeño. No es fácil, pero vale la pena...*

*A mami y papi, los mejores abuelos del mundo, por todo el tiempo que no he podido estar con ustedes...*

## **Resumen**

La constante evolución de los entornos personales de aprendizaje es uno de los temas referentes a tecnología educativa que más investigaciones está teniendo en los últimos años, partiendo de la base de la incorporación de elementos de la Web 2.0 en los procesos formativos y el continuo desarrollo de las Tecnologías de la Información y la Comunicación. Todas las personas tienen sus propios entornos de aprendizaje, los que se caracterizan por la variedad de herramientas e instrumentos que los conforman. En la Universidad de las Ciencias Informáticas los estudiantes emplean disímiles sistemas para aprender, aún sin estar conscientes de que conforman su entorno de aprendizaje. Existen variados software que permiten gestionar los entornos personales de aprendizaje, sin embargo, estos poseen deficiencias que dificultan su empleo por los estudiantes de la UCI. En este trabajo se presenta el desarrollo de una ingeniería de requerimiento asociada a la construcción de un software para gestionar el entorno personal de aprendizaje de los estudiantes de la Universidad de las Ciencias Informáticas; como parte de esta ingeniería se presentan los resultados obtenidos en las etapas Estudio de factibilidad, Adquisición y análisis, Especificación y Validación. El desarrollo de estas etapas permitió obtener las funcionalidades primarias del software y establecer las bases para su diseño y construcción. En el desarrollo de la investigación se emplearon métodos de investigación teóricos y empíricos.

**Palabras claves:** entorno personal de aprendizaje, ingeniería de requerimientos, tecnología educativa.

# ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL .....</b>	<b>6</b>
INTRODUCCIÓN .....	6
1.1 CONCEPTUALIZACIÓN DE LOS ENTORNOS PERSONALES DE APRENDIZAJE .....	6
1.2 ESTRUCTURA BÁSICA DE UN ENTORNO PERSONAL DE APRENDIZAJE .....	8
1.3 HERRAMIENTAS PARA LA GESTIÓN DE ENTORNOS PERSONALES DE APRENDIZAJE .....	11
1.4 INGENIERÍA DE REQUERIMIENTOS .....	16
1.4.1 <i>Clasificación de los requerimientos del software</i> .....	16
1.4.2 <i>Definición de Ingeniería de Requerimientos</i> .....	18
1.4.3 <i>Actividades principales en el proceso de Ingeniería de Requerimientos</i> .....	19
1.4.4 <i>Técnicas para la Ingeniería de Requerimientos</i> .....	24
1.5 METODOLOGÍAS Y HERRAMIENTAS PARA EL DESARROLLO DE LA INGENIERÍA DE REQUERIMIENTOS.....	26
1.5.1 <i>La ingeniería de requerimientos en las metodologías de desarrollo de software</i> 26	
1.5.2 <i>Lenguaje de modelado unificado</i> .....	29
1.5.3 <i>Herramienta CASE</i> .....	30
1.5.4 <i>Herramienta para el diseño de los prototipos de baja fidelidad</i> .....	30
1.5.5 <i>Herramientas para el diseño de los prototipos de alta fidelidad</i> .....	31
CONCLUSIONES DEL CAPÍTULO.....	32
<b>CAPÍTULO 2. ADQUISICIÓN Y ESPECIFICACIÓN DE REQUERIMIENTOS.....</b>	<b>33</b>
INTRODUCCIÓN .....	33
2.1 ESTUDIO DE FACTIBILIDAD .....	33
2.2 ADQUISICIÓN Y ANÁLISIS DE REQUERIMIENTOS .....	37
2.2.1 <i>Descripción de requerimientos funcionales</i> .....	38
2.2.2 <i>Descripción de los requerimientos no funcionales</i> .....	41
2.3 ESPECIFICACIÓN DE REQUERIMIENTOS.....	43
CONCLUSIONES DEL CAPÍTULO.....	49
<b>CAPÍTULO 3. VALIDACIÓN DE REQUERIMIENTOS.....</b>	<b>50</b>
INTRODUCCIÓN .....	50
3.1 DISEÑO DEL PROTOTIPO DE INTERFAZ DE USUARIO .....	50
3.2 MEDICIÓN DEL GRADO DE SATISFACCIÓN DEL CLIENTE.....	53

3.3	MÉTRICAS CUANTITATIVAS PARA MEDIR LA CALIDAD PARA LA ESPECIFICACIÓN DE REQUERIMIENTOS.....	59
	CONCLUSIONES DEL CAPÍTULO.....	60
	<b>CONCLUSIONES.....</b>	<b>62</b>
	<b>RECOMENDACIONES.....</b>	<b>63</b>
	<b>REFERENCIAS BIBLIGRÁFICAS.....</b>	<b>64</b>
	<b>ANEXOS.....</b>	<b>66</b>

## ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1. Niveles hacia el PLE Fuente: (Dabbagh y Kitsantas 2012).....</i>	<i>9</i>
<i>Ilustración 2. Partes de un PLE. Fuente: (Adell y Castañeda 2010) .....</i>	<i>10</i>
<i>Ilustración 3. Comparación de las tres herramientas. Fuente: Elaboración propia .....</i>	<i>16</i>
<i>Ilustración 4. Clasificación de los RNF Fuente: Sommerville (2010).....</i>	<i>18</i>
<i>Ilustración 5. Procesos de la Ingeniería de Requerimientos según varios autores Fuente: Elaboración propia .....</i>	<i>20</i>
<i>Ilustración 6 Vista en espiral del proceso de Ingeniería de Requerimientos. Fuente: (Sommerville 2010).....</i>	<i>21</i>
<i>Ilustración 7. Procesos IR según Sommerville 2010 Fuente: Elaboración propia .....</i>	<i>22</i>
<i>Ilustración 8. Técnicas para la IR Fuente: Elaboración propia.....</i>	<i>25</i>
<i>Ilustración 9. Agrupación de las técnicas según funciones específicas. Fuente: Elaboración propia.....</i>	<i>26</i>
<i>Ilustración 10. Clasificación de las metodologías para el desarrollo de software. Fuente: Elaboración propia .....</i>	<i>27</i>
<i>Ilustración 11. Escenarios metodología AUP-UCI según el Programa de Mejora. Fuente: Elaboración propia .....</i>	<i>29</i>
<i>Ilustración 12. Prototipo alta fidelidad/Vista principal.....</i>	<i>51</i>
<i>Ilustración 13. Prototipo alta fidelidad/Añadir Widget .....</i>	<i>51</i>
<i>Ilustración 14. Prototipo de alta fidelidad/Listar RSS.....</i>	<i>52</i>
<i>Ilustración 15. Prototipo de alta fidelidad/Importar lista de amigos .....</i>	<i>52</i>
<i>Ilustración 16. Tipos de Requerimientos del cliente Fuente: (Duarte 2005).....</i>	<i>56</i>
<i>Ilustración 17. Clasificación de los requerimientos en función de los resultados obtenidos</i>	<i>58</i>

## ÍNDICE DE TABLAS

<i>Tabla 1. Ventajas y limitaciones de los PLE. Fuente: (Almenara 2015)</i> .....	8
<i>Tabla 2. Dimensiones para la construcción de un PLE. Fuente: (Bagdanov y Wild 2010).</i> 14	
<i>Tabla 3. Comparación de herramientas según las dimensiones Fuente: Bagdanov y Wild (2010)</i> .....	15
<i>Tabla 4. Comparación entre las herramientas Fuente: (Rodríguez, Cormenzana et al. 2013)</i> .....	37
<i>Tabla 5. Requerimientos funcionales</i> .....	41
<i>Tabla 6. Requerimientos no funcionales</i> .....	43
<i>Tabla 7. Historia de Usuario Añadir Widget</i> .....	44
<i>Tabla 8 Historia de Usuario Eliminar Widget</i> .....	46
<i>Tabla 9. Listar RSS</i> .....	48
<i>Tabla 10. Añadir escritorio</i> .....	49
<i>Tabla 11. Clasificación de requerimientos</i> .....	55
<i>Tabla 12. Clasificación de los requerimientos</i> .....	57
<i>Tabla 13. Clasificación de los requerimientos en función de los resultados obtenidos</i> .....	59
<i>Tabla 14. Resumen de los resultados obtenidos al aplicar la métrica de calidad Especificación</i> .....	60
<i>Tabla 11. Comparación entre las herramientas (Rodríguez, Cormenzana et al. 2013)</i> .....	67
<i>Tabla 12. Dimensiones para la construcción de un PLE (Bagdanov and Wild 2010)</i> .....	68

## Introducción

El continuo desarrollo de las Tecnologías de la Información y la Comunicación (TIC) y su creciente incorporación a los procesos formativos en centros educacionales, hace cada vez más necesario encontrar alternativas que apoyen y enriquezcan el proceso enseñanza-aprendizaje. Esta evolución de las TIC y la emergencia de nuevos paradigmas en la web, con sus consiguientes aplicaciones en el campo de la educación, permiten poner a disposición de los estudiantes una gran diversidad de recursos educativos y herramientas informáticas. Así mismo, facilita el acceso a nuevos y variados entornos de formación virtuales y la realización de un conjunto de actividades formativas desde cualquier lugar, desde cualquier dispositivo y en cualquier momento.

El desarrollo exponencial de nuevas aplicaciones de software, brindan crecientes posibilidades a los individuos para enriquecer sus ambientes personales de formación. En el año 2001, en el marco del proyecto NIMLE (*Northern Ireland Integrated Managed Learning Environment*), se empieza a desarrollar la idea de un entorno personal de aprendizaje centrado en el alumno como evolución de los entornos virtuales de enseñanza-aprendizaje, centrados en la institución. En el 2004, el JISC (*Joint Information System Committee* de la Gran Bretaña) incluyó una sesión específica en su congreso anual dedicada a entornos telemáticos centrados en el alumno a los que llamaron *Personal Learning Environment* (PLE). Esta fue la primera vez que se empleó oficialmente dicho concepto, que posteriormente evolucionó y se ha generalizado con el acrónimo PLE.

Los PLE “*son una colección de instrumentos, materiales y recursos humanos que el estudiante emplea en un determinado momento en el contexto de un proyecto educativo*”. (Attwell 2007). Adell y Castañeda (2010) conciben el PLE como “*el conjunto de herramientas, fuentes de información, conexiones y actividades utilizadas habitualmente por el individuo, suponiendo esto un cambio en la práctica pedagógica hacia un aprendizaje abierto, social y centrado en el estudiante*”.

Väljataga y Laanpere (2010) señalan que los criterios presentados anteriormente no implican necesariamente el uso de elementos tecnológicos en los PLE. Sin embargo, este concepto necesita ser desarrollado de forma práctica, de manera que los individuos comprendan la contribución de la tecnología a la eficiencia del proceso de aprendizaje.

En la Universidad de las Ciencias Informáticas (UCI) los estudiantes emplean habitualmente diferentes herramientas informáticas, que sin estar ellos conscientes, forman parte de sus PLE, al constituir componentes del entorno donde aprenden. Entre estas herramientas se encuentran software de redes sociales (Facebook, Twitter, entre otras), herramientas de mensajería instantánea (pidgin, pandion, trillian, entre otras), sistemas de gestión del aprendizaje (Zera, Moodle), sitios para la descarga y visualización de materiales audiovisuales (Internos), entre otras. Estas herramientas se emplean de forma dispersa y no están totalmente integradas por lo que no se aprovechan sus potencialidades tecnológicas para conformar un PLE en la UCI.

En la actualidad existen software para gestionar los PLE, los cuales brindan páginas de inicio que permiten la integración de diferentes herramientas de la Web 2.0 así como la disposición de información de manera ordenada y fácil de localizar. La similitud técnica entre estas herramientas es elevada: poseen una variedad considerable de características relacionadas con la usabilidad y los aspectos estéticos; sin embargo, su empleo en la UCI no es factible porque poseen las limitaciones que se mencionan a continuación y que dificultan su empleo en la UCI:

- Poseen un sistema de recomendaciones a canales de noticias internacionales como (CNN, The New York Time) y páginas de interés que no responden a los intereses y necesidades universitarias; no recomiendan canales de noticias nacionales tales como Cubadebate, Granma, Juventud Rebelde, Trabajadores, entre otros. Estas recomendaciones no tienen en cuenta la región en el momento del registro de los usuarios.
- No admiten la configuración de correo electrónico ni la de los clientes de mensajería que se emplean en la UCI, lo cual imposibilita que estas herramientas sean parte del PLE de un estudiante de la UCI.
- No son totalmente gratis, lo cual es una limitación para estudiantes y trabajadores de la UCI que deseen utilizar todas las funcionalidades de estas herramientas por un tiempo indefinido. Ej. Netvibes solicita pasados los 14 días de uso, el depósito de un monto en USD para disfrutar de todas sus funcionalidades.

- Generan un alto consumo de Internet, lo cual no es aprobado por la comunidad universitaria, teniendo en cuenta que la navegación en la universidad está basada en un sistema de cuotas.

A partir de la situación problemática descrita anteriormente se plantea el **siguiente problema a resolver** ¿Cómo establecer una base para el diseño y la construcción de un software que soporte el entorno personal de aprendizaje de los estudiantes de la UCI?

El **objeto de estudio** de la presente investigación está enmarcado en los entornos personales de aprendizaje, dentro de este objeto se actúa en los requerimientos de un software que soporte el PLE de los estudiantes de la UCI, lo cual constituye el **campo de acción**.

El **objetivo general** de este trabajo de diploma es desarrollar el proceso de ingeniería de requerimientos de un software para soportar el entorno personal de aprendizaje de los estudiantes de la UCI.

Los **objetivos específicos** son los siguientes:

- ✓ Realizar un estudio del estado del arte de las herramientas de gestión de entornos personales de aprendizaje.
- ✓ Realizar el estudio de factibilidad, adquisición-análisis y especificación de requerimientos.
- ✓ Validar los requerimientos de software a partir de los resultados obtenidos en la fase del levantamiento de requerimientos.

Las **tareas** planteadas para darle cumplimiento a los objetivos específicos son las siguientes:

- Comparación de los principales software que soportan PLE.
- Identificación del estado actual de los PLE en la UCI.
- Identificación de la metodología de desarrollo de software a utilizar.
- Estudio de las diferentes concepciones asociadas a la ingeniería de requerimientos.
- Selección de las técnicas a utilizar en cada una de las etapas que comprenden el desarrollo de requerimientos del sistema que soporta el PLE de los estudiantes de la UCI.

- Educación de los requerimientos de software que soporta el PLE de los estudiantes de la UCI.
- Especificación de los requerimientos de software que soporta el PLE de los estudiantes de la UCI.
- Aplicación del método Kano para medir la satisfacción del cliente
- Aplicación de métricas para medir la calidad de la especificación de requerimientos.
- Desarrollo de los prototipos de alta fidelidad.

Los métodos teóricos empleados para el desarrollo de esta investigación son el Analítico-Sintético, al analizar las principales herramientas que soportan la gestión de los PLE e identificar sus principales semejanzas y diferencias, así como los conceptos y definiciones más relevantes; el Histórico-Lógico fue utilizado para estudiar la evolución de las principales definiciones sobre PLE y el Análisis documental se empleó para realizar consultas a la literatura especializada en el tema abordado en la presente investigación. Se emplearon técnicas como la Encuesta, la Entrevista, entre otras.

Este trabajo de diploma está estructurado en tres capítulos, desarrollados a partir del estudio realizado.

### ***Capítulo 1: Marco teórico referencial sobre los entornos personales de aprendizaje.***

En este capítulo se hace referencia a los elementos teóricos que soportan la investigación. Se realiza el estudio del estado del arte de las herramientas que dan soporte a PLE; estas herramientas se describen y se realizan comparaciones que tributan al diseño de la propuesta de solución. Se detalla un estudio de las diferentes concepciones asociadas a la ingeniería de requerimientos, especificando las fases o etapas de este proceso, determinadas por diferentes autores. Se especifica un estudio sobre las principales técnicas empleadas en cada una de las etapas y se justifica la selección de algunas de ellas para el desarrollo de la ingeniería de requerimiento del PLE. Se presentan el lenguaje de programación, de maquetado y modelado, así como las tecnologías que se utilizarán para desarrollar la propuesta planteada, fundamentando su selección en base al estudio realizado.

### ***Capítulo 2: Estudio de factibilidad, adquisición-análisis y especificación de requerimientos.***

En este capítulo se presenta el estudio de factibilidad realizado. Además, se expone el resultado del trabajo realizado en la etapa Adquisición y análisis de requerimientos, presentando los requerimientos funcionales y no funcionales de la solución. Finalmente se presenta la especificación de los requerimientos.

**Capítulo 3: Validación de requerimientos.**

En el Capítulo 3 se describe la propuesta de validación de requerimientos que aparecen en el documento especificado. Se presentan los prototipos de alta fidelidad del sistema como técnica para validar los requerimientos especificados. Se aplican métricas de calidad y se utiliza el método de Kano para medir la satisfacción del cliente.

---

## CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL

### Introducción

La base teórica del presente trabajo está sustentada sobre los PLE que sirven de punto de partida para la investigación. Es fundamental para tener una visión de la solución propuesta, el dominio de algunos conceptos esenciales y soluciones existentes relacionadas con la gestión de los PLE. El estudio de las herramientas y tecnologías a utilizar durante el desarrollo de la propuesta de solución, sirven de apoyo para que toda la información recopilada oriente y guíe conceptualmente el desarrollo de la investigación.

### 1.1 Conceptualización de los entornos personales de aprendizaje

Los PLE son un concepto del siglo XXI basado en tecnologías de la Web 2.0 que progresivamente ha ganado importancia como plataforma individual, eficaz para el aprendizaje del estudiante. Los PLE permiten el desarrollo de experiencias, espacios personales y sociales que motivan a los estudiantes a dirigir su propio aprendizaje.

Las definiciones sobre PLE se agrupan en dos grandes tendencias, las de carácter tecnológico y las pedagógicas. Entre las principales definiciones se pueden citar las siguientes:

*“Incluye las herramientas, personas, comunidades y servicios que desde las plataformas educativas, sobre la base de la cual los alumnos aprenden, dirigen su aprendizaje y alcanzan y logran sus objetivos cognitivos, juntos con las actividades que ellos realizan o desarrollan en el ambiente web.” (Manolis y Kalaitzidou 2018)*

*“Entorno en el que caben diferentes tipologías de aprendizaje, y que al situarse en uno de los límites de la práctica de la enseñanza reclama multiplicidad de métodos didácticos, principalmente los que se han dado en llamar “metodologías centradas en el alumno.” (Urdaneta 2015)*

*“Los PLE concretan la ecología de aprendizaje de las personas, dado que organizan y reflejan una visión amplia del aprendizaje, tanto por la demanda de aprender a lo largo de toda la vida, lo que le imprime un carácter dinámico de adaptación a los cambios como en sus posibilidades de extender el proceso de aprendizaje de las personas más allá de los límites de las instituciones formales .” (Engel y Coll 2014)*

*“Herramientas, comunidades y servicios que constituyen las plataformas educativas individuales que los aprendices utilizan para dirigir su propio aprendizaje y perseguir objetivos educativos.” (Dabbagh y Kitsantas 2012)*

*“... sistemas que ayudan a los estudiantes y a los docentes a tomar el control de gestión y de su propio aprendizaje. Lo que incluye proporcionar apoyo para que fijen sus propias metas de aprendizaje; gestionar su aprendizaje; formalizar los contenidos y procesos; y comunicarse con los demás en el proceso de aprendizaje, así como lograr los objetivos de aprendizaje.” (Almenara 2011)*

*“... un conjunto de herramientas, fuentes de información, conexiones y actividades que cada persona utiliza de forma asidua para aprender. ” (Adell y Castañeda 2010)*

Definiciones como las encontradas en (Barron 2006), (Haskins 2007), (Harmelen 2008), (Dabbagh y Kitsantas 2012) y (Ash 2013) coinciden en que los PLE son entornos de aprendizaje motivados por interés y de iniciativa propia y que ayudan a los alumnos gestionar y tener el control de su aprendizaje.

En el desarrollo de esta investigación se emplea la definición emitida por Nada Dabbagh y Anastasia Kitsantas (2012) pues la autora de este trabajo coincide en que el concepto de PLE no puede entenderse como una plataforma de software para la formación, sino más bien como un entorno constituido por diferentes herramientas y recursos que permiten a los estudiantes crear un ambiente formativo personal, a partir del cual podrán, en función de sus intereses y necesidades de aprendizaje, potenciar su proceso de formación tanto en espacios formales como informales de aprendizaje.

### **Ventajas y limitaciones de los PLE**

El contexto de los PLE se presenta como un espacio facilitador de la participación, la colaboración y la interacción; dando la posibilidad a los estudiantes de aprender de otros y con otros. A continuación se presentan algunas de las ventajas y limitaciones de estos entornos.

<i>Ventajas</i>	<i>Limitaciones</i>
<ul style="list-style-type: none"> <li>• Los alumnos se convierten en actores activos en su propio proceso de</li> </ul>	<ul style="list-style-type: none"> <li>• Al ser entornos ampliamente configurables al principio el usuario puede sentirse</li> </ul>

---

*“Las semillas de los desastres enormes del software por lo general se vislumbran en los tres primeros meses del inicio del proyecto.” Coper Jones*

<p>aprendizaje, y llegan a tener una identidad formativa más allá de los contextos tradicionales de aprendizaje.</p> <ul style="list-style-type: none"> <li>• El ritmo del aprendizaje lo marca el propio docente.</li> <li>• El docente puede gestionar sus propias herramientas y elegir las que desea usar.</li> <li>• Permiten que el docente pueda alcanzar sus objetivos de aprendizaje en un entorno que se adapta a lo que necesita para aprender.</li> <li>• Apoyan el aprendizaje informal al contar con la participación tanto de profesores como de estudiantes en las experiencias de aprendizaje.</li> <li>• Ayuda a entender cómo aprendemos las personas usando eficientemente las tecnologías</li> <li>• Son entornos abiertos a la interacción y relación con las personas independientemente de su registro oficial en los programas o cursos; es decir, potenciación con ellos de acciones formativas tanto formales, como no formales e informales</li> </ul>	<p>desorientado o no saber cuáles herramientas seleccionar.</p> <ul style="list-style-type: none"> <li>• Dedicación de tiempo y esfuerzo para elaborar el PLE y para descubrir nuevas herramientas, para analizar sus usos y si son capaces de cubrir las necesidades personales.</li> <li>• El docente puede desear un PLE lleno de herramientas, o los más actualizado posible y perder realmente los objetivos de aprendizaje.</li> </ul>
--	--

*Tabla 1. Ventajas y limitaciones de los PLE. Fuente: (Almenara 2015)*

## 1.2 Estructura básica de un entorno personal de aprendizaje

La investigación sobre los PLE ha demostrado que estos entornos ayudan a los alumnos a agrupar y compartir los resultados del aprendizaje, a participar de la producción de conocimiento colectivo, dándole un sentido propio al aprendizaje (Dabbagh y Kitsantas 2012). Sin embargo, a pesar de estas poderosas posibilidades de aprendizaje, los alumnos necesitan apoyo, orientación e intervenciones pedagógicas que les ayuden a hacer el mejor uso de las herramientas y recursos para conseguir sus objetivos y lograr éxito en la gestión de sus PLE. Para ello los siguientes autores plantean diferentes criterios sobre la estructura básica de dichos entornos.

Nada Dabbagh y Anastasia Kitsantas (2012) desarrollaron un marco pedagógico para ayudar a profesores de educación superior a potenciar las habilidades de autorregulación de los estudiantes en la creación de PLE, con el uso de las redes sociales basado en los niveles de interactividad que estas herramientas permiten. A continuación, se ilustran estos tres niveles.



*Ilustración 1. Niveles hacia el PLE Fuente: (Dabbagh y Kitsantas 2012)*

### **Gestión de Información**

En este nivel se establecen los objetivos, se planifican las tareas para alcanzarlos y se controla el tiempo y el avance hacia los mismos. Finalmente se busca y selecciona la información, para posteriormente filtrar y organizarla según su relevancia. Para la planificación y gestión de tareas se puede utilizar herramientas como: software de gestión de tareas, software de control del tiempo, software de calendario, entre otros. Para buscar y seleccionar la información se puede hacer uso de Lectores RSS, herramientas de notas, marcadores sociales, gestores de referencias bibliográficas, entre otros.

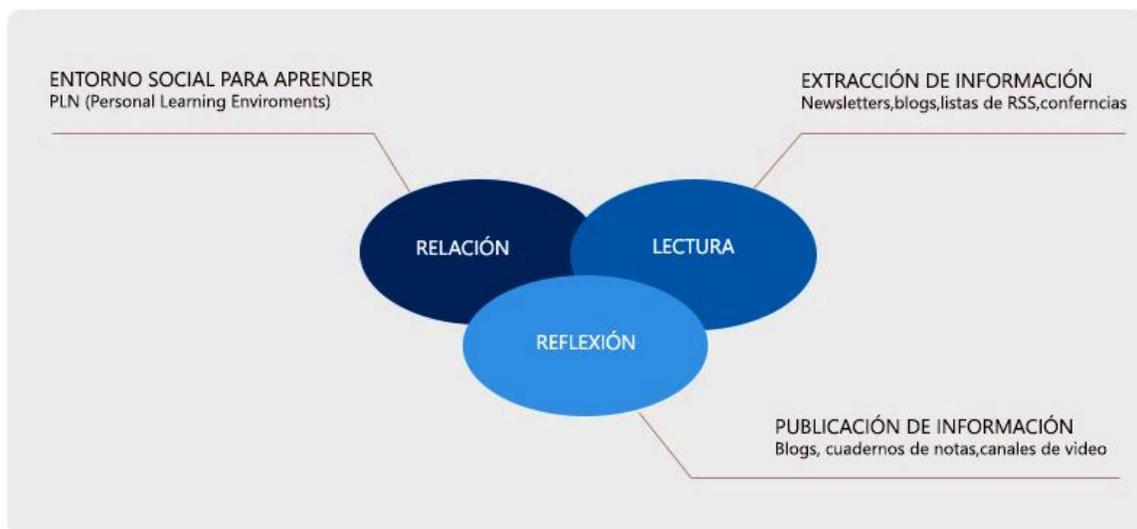
## Interacción social y colaboración

El objetivo de este nivel es conectar con comunidades online que vayan más allá del aula y realizar tareas colaborando con otros estudiantes. Para ellos se puede hacer uso de herramientas de trabajo colaborativo, herramientas de edición colaborativa de documentos, redes sociales, entre otras.

## Agregación y gestión de la información

Este nivel se enfoca en la creación y compartición de contenidos y su filtrado para otros usuarios. Se pueden emplear blog y wiki para crear y compartir la información generada.

Autores como Jordi Adell y Linda Castañeda (2010) proponen una estructura que incluye tres tipos de herramientas y estrategias: de lectura, de reflexión y de relación con otros. (*Ilustración 2*)



*Ilustración 2. Partes de un PLE. Fuente: (Adell y Castañeda 2010)*

Cada una de estas partes tienen sus herramientas y estrategias de lectura, sobre las cuales se hará referencia a continuación.

## Herramientas y estrategias de lectura

Las herramientas y estrategias de lectura son “*fuentes de información a las que se accede, que muestra la información en forma de objeto o artefacto (mediatecas)*”. (Adell y Castañeda 2010)

En el PLE de las personas no solo se integran las herramientas que se utilizan en espacios formales, sino que se incluyen todos los mecanismos a través de los cuales los individuos se informan y apropian de la información de forma habitual; tanto en contextos formales como informales. Teniendo en cuenta el planteamiento anterior, dentro de las herramientas y las estrategias de lectura, no solo puede hablarse de blogs de impacto o listas de RSS, sino también de las experiencias que, a través de la búsqueda, la iniciativa o la curiosidad del usuario le permiten acceder a la información (ej. lecturas rápidas, revisión de titulares, asistencia a conferencias, entre otras).

### **Herramientas y estrategias de reflexión**

Las herramientas y estrategias de reflexión son *“los entornos o servicios en los que se puede transformar la información (sitios donde se escribe, se comenta, se analiza, se recrea, se publica).”* (Adell y Castañeda 2010)

En los PLE también se integran las herramientas que permiten procesar y organizar la información. Todo este proceso de reconstrucción del conocimiento desencadena una serie de habilidades como la síntesis, la reflexión, la organización y la estructuración. En consecuencia, están los sitios donde se publica dicha información como los blogs personales, los sitios web de publicación de presentaciones, las páginas web, los canales de video, el muro personal de las redes sociales, entre otros.

### **Herramientas y estrategias de relación**

Las herramientas y estrategias de relación son los *“entornos donde las personas se relacionan con otras de/con las que aprende”*. (Adell y Castañeda 2010)

Todas las fuentes, mecanismos y aptitudes para aprender a las que se ha hecho referencia no solo se limitan a la información documental o la que individualmente las personas se apropian, sino que en ellas se incluyen a las personas como fuentes de información, así como las interacciones con ellas como fuentes que enriquecen el conocimiento. Es decir, no solo se hace referencia a un PLE formado por el individuo, las herramientas y servicios que consume para su aprendizaje, sino que se incluye el entorno social para aprender, que constituye su Red Personal de Aprendizaje (PLN, por sus siglas en inglés). Las oportunidades de intercambiar con las personas a través de herramientas sociales y foros en línea permiten compartir, reflexionar, discutir y reconstruir con otros el conocimiento.

## **1.3 Herramientas para la gestión de entornos personales de aprendizaje**

Existen diferentes herramientas que actualmente pueden ser utilizadas para gestionar PLE. Estas herramientas permiten centralizar y organizar tanto información de diferentes recursos web como también miniaplicaciones, widgets<sup>1</sup> o gadgets<sup>2</sup> para facilitar el acceso a funciones de uso frecuente por el usuario. A continuación se presentan las de mayor relevancia para esta investigación.

### **Symbaloo<sup>3</sup>**

Es una aplicación web, que no necesita algún tipo de instalación en el ordenador; además es gratuita, muy visual, y permite ir incorporando elementos al PLE de una manera simple. Posee un diseño flexible que posibilita ir ampliando y organizando la información. El uso de esta herramienta requiere la creación de una cuenta en su sitio web. La aplicación se compone de un conjunto de pestañas, cada una de las cuales Symbaloo las entiende como una página de inicio o escritorio independiente entre sí. Ofrece la posibilidad de compartir y hacer públicas estas páginas de inicio permitiendo a los usuarios su empleo sin necesidad de abrir una cuenta.

### **Netvibes<sup>4</sup>**

A diferencia de Symbaloo, Netvibes organiza los escritorios o páginas de inicio en diferentes pestañas, es decir, un escritorio está compuesto por un conjunto de pestañas, y se pueden tener tantos como se desee. Además, permite visualizar la información de los diferentes widgets (tanto de redes sociales, como de blogs o RSS) en la misma aplicación, sin necesidad de consultar la página de origen. Por otra parte, incorpora otro tipo de utilidades (calculadora, tiempo, agenda, notas, tareas, comparación entre hojas de cálculo, entre otras). Todo ello con una de las características que identifica la web 2.0, la

---

<sup>1</sup> Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual

<sup>2</sup> Un gadget es un dispositivo que tiene un propósito y una función específica, generalmente de pequeñas proporciones, práctico y a la vez novedoso.

<sup>3</sup> *Página oficial de la herramienta <https://www.symbaloo.com>*

<sup>4</sup> *Página oficial de la herramienta <https://www.netvibes.com>*

posibilidad de compartir widgets o incluso un escritorio, que será público y se podrá consultar sin la necesidad de registrarse.

### **iGoogle<sup>5</sup>**

Es una de las aplicaciones ofrecidas por Google para organizar información, de manera que cuando se accede, se puede ver en una sola pantalla las diferentes utilidades que el usuario emplea habitualmente. Permite una organización y personalización según preferencias, a partir de gadgets o widgets. También permite cierto grado de flexibilidad ya que se puede organizar la información en columnas (hasta cuatro) y pestañas, ampliando y clasificando las herramientas que se vayan incluyendo. Como ocurre con el resto de aplicaciones Google, es posible acceder a cualquiera de ellas mediante un único registro.

### **Análisis comparativo de las herramientas descritas**

En esta investigación se referencia la comparación establecida por Bagdanov y Wild (2010), en la cual se identifican seis dimensiones (*Tabla 2*) para la construcción de un PLE web.

<i>Dimensión</i>	<i>Definición</i>	<i>Estándares potenciales</i>
<i>Agregación</i>	Organización de varios widgets de una manera espacial	W3C Widgets 1.0 Google Gadgtes API Google GadgetTabML Netvibes UWA OpenAjax Matadata 1.0
<i>Comunicación</i>	Interoperabilidad de los datos a través de widgets y servicios. Incluye problemas con cortar y pegar, arrastrar y soltar.	Varios estándares de datos y metadatos, como RSS, SCROM, Dublin Core RDF, HTML5 Dn'D, OpenAjax Hub 2.0, Google gadget pubsub

<sup>5</sup> Página oficial de la herramienta <http://www.igooglepportal.com/>

<i>Configuración</i>	Configuración de la actualización de los widgets, estado de los datos y sincronización con otros usuarios activos cuando comparte la misma instancia del widget	COMET/Reserve Ajax, XMPP, XMPP over BOSH, Google Wave Federation Protocol
<i>Organización</i>	Interoperabilidad de la identidad del usuario, información del perfil y lista de amigos. Capacidad para definir algunos contextos grupales para compartir widgets de eventos, estado o datos.	OpenID (perfiles portátiles) Open Social API Contactos Portables Facebook
<i>Actividad</i>	Las aplicaciones en uso en el PLE se pueden controlar a través de scripts que se dedican a las actividades de aprendizaje del usuario	WS-BPEL (orientado a los negocios) IMS Learning Design (dirigido a los VLE y diseño orientado a tiempo, no en tiempo de ejecución)
<i>Tiempo de ejecución</i>	La interoperabilidad permite el intercambio de una plataforma en ejecución o sus partes con otra.	W3C Widgets 1.0: Empaquetado y configuración OPML, OpenAjax Mashup, Reference Application

Tabla 2. Dimensiones para la construcción de un PLE. Fuente: (Bagdanov y Wild 2010)

Teniendo en cuenta estas dimensiones se presenta a continuación la comparación realizada por la autora de este trabajo, sobre las herramientas iGoogle, Netvibes y Symbaloo.

<i>Dimensiones</i>	<i>Característica</i>	<i>iGoogle</i>	<i>Netvibes</i>	<i>Symbaloo</i>
<i>Agregación</i>	Pantalla compartida	X	X	X

	Widgets estándar	X	X	X
	Diseño de widgets	X	X	X
<i>Comunicación</i>	Comunicación entre los widgets	X		X
<i>Organización</i>	Lista de amigos	X	X	X
	Servidor de amigos	X	X	X
	Control de acceso	X		
<i>Configuración</i>	Permite exportar e importar	X	X	
	Importación y exportación genérica	X		
	Configuración externa			X
	Embebido <sup>6</sup>			X

Tabla 3. Comparación de herramientas según las dimensiones Fuente: Bagdanov y Wild (2010)

En la tabla anterior no se reflejaron las dimensiones Sincronización ni Recomendación, al no tener estas herramientas características que tributen a estas dimensiones. Teniendo en cuenta los resultados presentados se realizan las gráficas que se muestran en la *Ilustración 3*, las que permiten ilustrar de mejor manera el alcance de estas herramientas.

<sup>6</sup> insertar código de un lenguaje dentro de otro lenguaje



Ilustración 3. Comparación de las tres herramientas. Fuente: Elaboración propia

La comparación entre estas tres herramientas ilustra que no poseen ciertas características que resultan importantes para la concepción de un PLE, en su mayoría, asociadas a las dimensiones Comunicación, Sincronización y Recomendación.

## 1.4 Ingeniería de Requerimientos

El desarrollo de una solución informática, requiere identificar y establecer las funcionalidades que debe brindar, así como las restricciones sobre las que debe operar un software. Es esta una de las tareas más difíciles dentro del proceso de desarrollo de software. La Ingeniería de Requerimientos (IR) surge con la finalidad de comprender las necesidades exactas de los usuarios y transformarlas al lenguaje de los desarrolladores, cubriendo las actividades relacionadas con descubrir, documentar, validar y mantener un conjunto de funcionalidades para un sistema informático.

Para construir un sistema lo más importante es identificar las necesidades del usuario y lograr de esta forma un consenso final sobre las funcionalidades, las restricciones de software y hardware que tendrá el sistema deseado. De ahí la importancia de la ingeniería de requerimiento, pues si esta se realiza de forma incorrecta el producto final fallará y resulta más complejo corregir errores por un mal desarrollo no descubierto a tiempo, que si estos errores se identifican durante el proceso de ingeniería de requerimiento. Unos requerimientos bien elaborados y validados con el cliente evitan descubrir al terminar el proyecto que el sistema no era lo que se pedía.

### 1.4.1 Clasificación de los requerimientos del software

Los requerimientos para un sistema constituyen la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver un problema determinado. (Sommerville 2010)

Existen diversas formas de clasificar los requerimientos; desde un punto de vista evolutivo, Gerald Kotonya e Ian Sommerville (1995) distinguen a los requerimientos estables o duraderos de los requerimientos volátiles, con el fin de facilitar la gestión en los cambios de los mismos. Desde un punto de vista de implementación, los clasifica en obligatorios o deseables, asignándoles prioridades para determinar qué desarrollar en la siguiente etapa del proyecto.

Pete Sawyer y Gerald Kotonya (2001) presentan una clasificación de requerimientos bajo tres diferentes dimensiones; desde la perspectiva del alcance del requerimiento: si éste afecta al sistema o a un componente, desde la perspectiva de emergencia del requerimiento: si éste es derivado de otros o emergente directamente de las fuentes y desde la perspectiva de su naturaleza: requerimiento del producto o requerimiento del proceso.

Ian Sommerville (2010) clasifica los requerimientos como requerimientos del dominio, los que a su vez clasifica en requerimientos funcionales y no funcionales. Los requerimientos del dominio son requerimientos que provienen del dominio de la aplicación del sistema y que reflejan las características y restricciones de ese dominio.

Los requerimientos funcionales (RF) para un sistema refieren lo que el sistema debe hacer. Tales requerimientos dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando se escriben los requerimientos. Al expresarse como requerimientos del usuario, los requerimientos funcionales se describen por lo general de forma abstracta que entiendan los usuarios del sistema. (Sommerville 2010)

Los requerimientos no funcionales, como indica su nombre, son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios. Pueden relacionarse con propiedades emergentes del sistema, como fiabilidad, tiempo de respuesta y uso de almacenamiento. (Sommerville 2010) La *Ilustración 4* muestra la clasificación de este tipo de requerimientos.

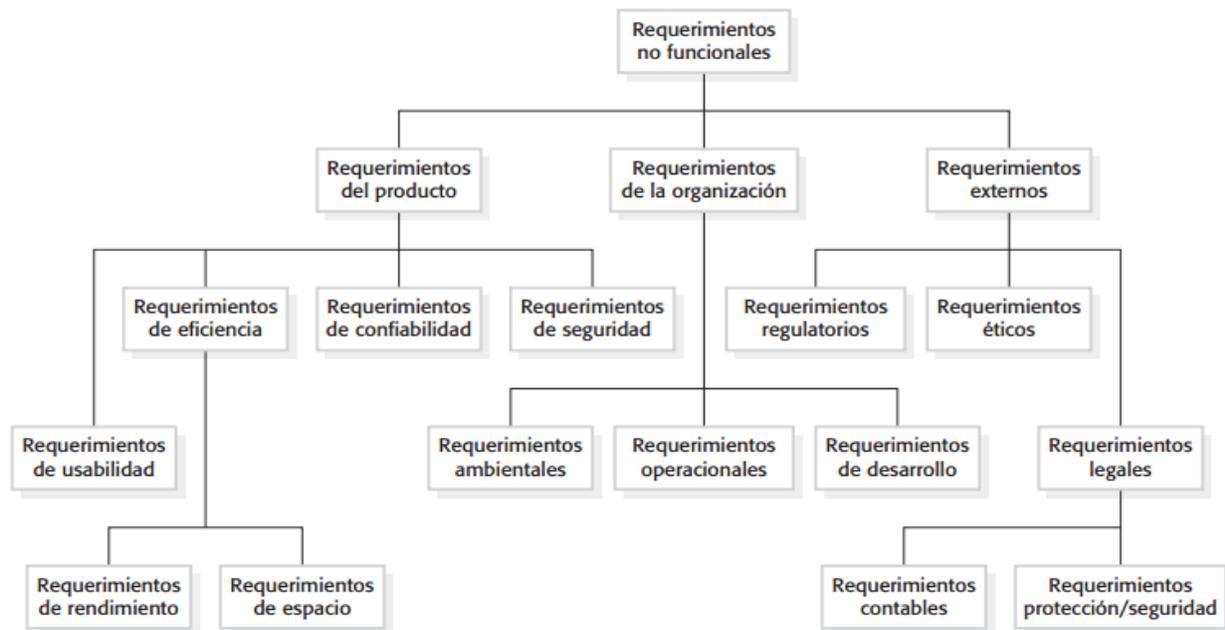


Ilustración 4. Clasificación de los RNF Fuente: Sommerville (2010)

Luego de un estudio de estas clasificaciones se decide utilizar la establecida por Sommerville (2010), pues esta clasificación es ampliamente difundida y citada en varios trabajos de IR (Escalona y Koch 2016), (Londoño y Guevara 2015) y (Toro y Jiménez 2002).

#### 1.4.2 Definición de Ingeniería de Requerimientos

Algunos investigadores, considerados entre las principales referencias dentro del tema de la IR, proponen las siguientes definiciones:

*“La ingeniería de requerimientos es el proceso de descubrir, analizar, documentar y verificar servicios y restricciones de software.”* (Sommerville 2010)

*“La IR proporciona el mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requerimientos a medida que se transforman en un sistema funcional.”* (Pressman 2010)

*“La ciencia y disciplina relacionada con el análisis y documentación de requerimientos, incluyendo el análisis de necesidades, el análisis de requerimientos y la especificación de requerimientos. También proporciona los mecanismos adecuados para facilitar las actividades de análisis, documentación y verificación de estos.”* (830-1998 2008)

*“La ingeniería de requerimientos puede considerarse como un proceso de descubrimiento y comunicación de las necesidades de clientes y usuarios y la gestión de los cambios en dichas necesidades.”* (Duarte 2005)

*“Área del conocimiento que explica cómo lograr la obtención, análisis, especificación y validación de los requerimientos del software.”* (Committee 2014)

La definición que guiará esta investigación será la dada por Sommerville (2010); señalando que la principal tarea de la IR consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo del sistema que gestionará el PLE de los estudiantes de la UCI.

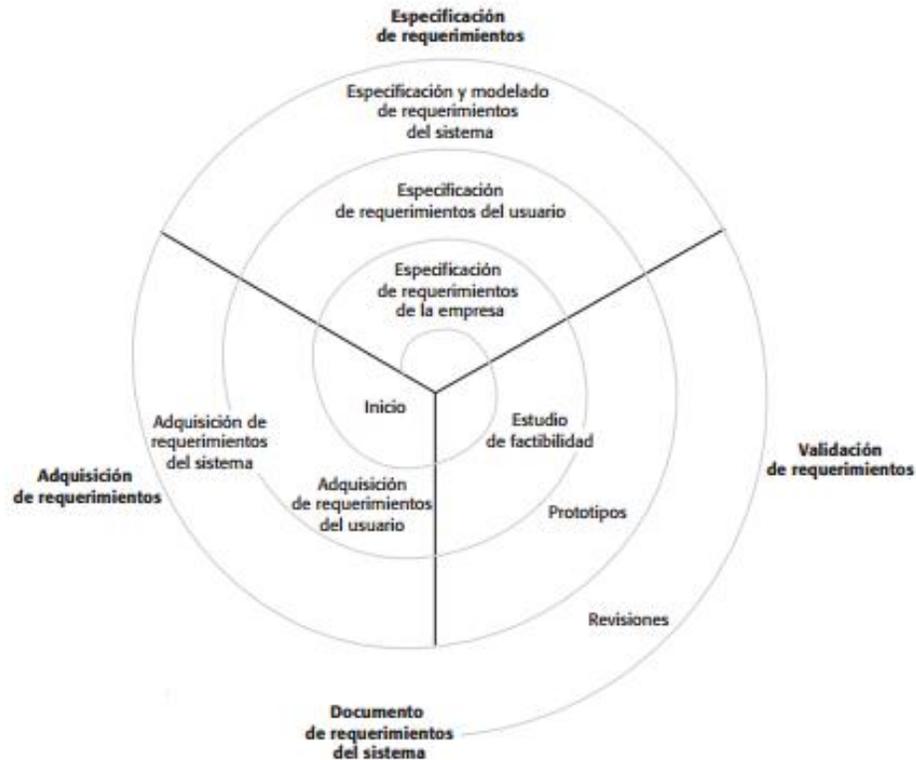
### **1.4.3 Actividades principales en el proceso de Ingeniería de Requerimientos**

Sommerville (2010), Pressman (2010), Wiegers (2013) así como Frühauf y Fuchs (2015) en el programa de estudios profesional en Ingeniería de Requerimientos de la IREB han definido un conjunto de procesos o etapas encaminadas a la obtención, el refinamiento y la validación de las funcionalidades que debe cumplir un software. Aunque los autores difieren en la cantidad y en la forma de nombrar cada etapa, al analizar cada una de las actividades que se realizan en cada una de ellas, así como los objetivos que persiguen se puede apreciar que todas las propuestas abordan los mismos aspectos, y cualquiera que se seleccione permitirá guiar el proceso de Ingeniería de Requerimientos.



*Ilustración 5. Procesos de la Ingeniería de Requerimientos según varios autores Fuente: Elaboración propia*

En la *Ilustración 5* se muestran dichas actividades como un proceso secuencial, sin embargo, en la práctica este es un proceso iterativo donde las actividades están entrelazadas. (*Ilustración 6*) Estas actividades están organizadas como un proceso iterativo alrededor de una espiral, y la salida es un documento de requerimientos del sistema.



*Ilustración 6 Vista en espiral del proceso de Ingeniería de Requerimientos. Fuente: (Sommerville 2010)*

Aunque Sommerville (2010), Pressman (2010), Frühauf y Fuchs (2015) y Wiegers (2013) difieren en cuanto a la cantidad de etapas de la IR las tres propuestas permiten establecer las bases para el diseño y la construcción de un software. La IR necesaria para el desarrollo de un sistema que gestione el PLE de los estudiantes de la UCI se desarrollará teniendo en cuenta las etapas propuestas por Sommerville para el desarrollo de los requerimientos y el aseguramiento de la calidad de los mismos (Estudio de factibilidad, Adquisición y Análisis, Especificación y Validación). En la *Ilustración 7* se muestran cada una de las etapas con sus respectivas entradas y salidas.

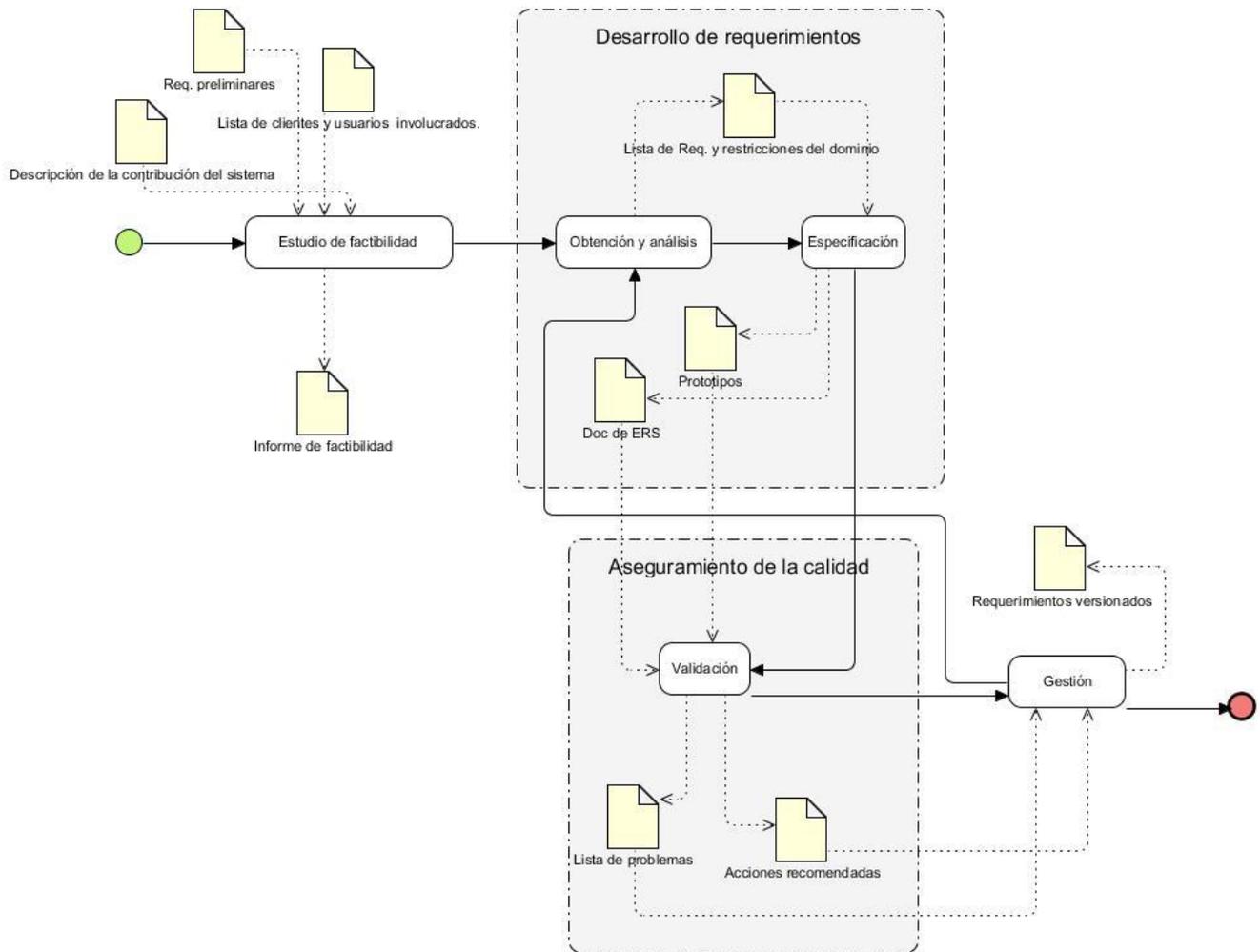


Ilustración 7. Procesos IR según Sommerville 2010 Fuente: Elaboración propia

### Estudio de Factibilidad

En esta etapa se realiza una estimación de la satisfacción de las necesidades del usuario con la explotación de las actuales tecnologías de software y hardware. El estudio considera si el sistema propuesto es útil y si éste puede desarrollarse dentro de las restricciones existentes. (Sommerville 2010) Un estudio de factibilidad debe ser rápido y relativamente corto y su resultado debe informar la decisión respecto a la continuidad de un análisis más detallado. Se debe indicar la posibilidad de implementación, ya sea por razones tecnológicas, ambientales, económicas o políticas. (Hadaad 2007)

### Adquisición y análisis de requerimientos

*“Las semillas de los desastres enormes del software por lo general se vislumbran en los tres primeros meses del inicio del proyecto.” Coper Jones*

El proceso de adquisición y análisis de requerimientos es el proceso de derivar los requerimientos del sistema mediante la observación de los sistemas existentes, discusiones con los usuarios y proveedores potenciales, así como análisis de tareas. Esto puede incluir el desarrollo de uno o más modelos de sistemas y prototipos, los que ayudan a entender el sistema que se va a especificar. (Sommerville 2010)

La aplicación de las técnicas para adquisición y análisis de requerimientos permiten obtener las necesidades que debe cumplir el sistema a desarrollar y la respectiva documentación que permita la posterior comprensión de los requerimientos. (Londoño y Guevara 2015)

### **Especificación de requerimientos**

La especificación de requerimientos consiste en la actividad de transcribir a un documento la información recopilada durante la adquisición y análisis. En el mismo se incluyen dos clases de requerimientos; los del usuario son informes abstractos de las funcionalidades del sistema para el cliente y el usuario final; y los de sistema son una descripción detallada de la funcionalidad a ofrecer. (Sommerville 2010)

En la especificación de requerimientos debe quedar descrito el sistema que se pretende desarrollar por medio de modelos y especificaciones las cuales permitan su comprensión y validación. (Londoño y Guevara 2015)

### **Validación de requerimientos**

La validación de requerimientos es el proceso de verificar que estos definan realmente el sistema que necesita el cliente. Este proceso se traslapa con el análisis, ya que se interesa por encontrar problemas con los requerimientos. Una correcta validación minimiza los costos asociados al soporte de software. (Sommerville 2010)

Durante el proceso de validación de requerimientos, tienen que realizarse diferentes tipos de comprobaciones, estas incluyen: (Sommerville 2010)

1. Comprobaciones de validez: Un usuario quizás crea que necesita un sistema para realizar ciertas funciones. Sin embargo, con mayor consideración y análisis se logra identificar las funciones adicionales o diferentes que se requieran.
2. Comprobaciones de consistencia: Los requerimientos en el documento no deben estar en conflicto; no debe haber restricciones contradictorias o descripciones diferentes de la misma función del sistema.

3. Comprobaciones de totalidad: El documento de requerimientos debe incluir requerimientos que definan todas las funciones y las restricciones pretendidas por el usuario del sistema.
4. Comprobaciones de realismo: Al usar el conocimiento de la tecnología existente, los requerimientos deben comprobarse para garantizar que en realidad pueden implementarse.
5. Verificabilidad: Los requerimientos del sistema deben escribirse siempre de manera que sean verificables. Se deben escribir un conjunto de pruebas que demuestren que el sistema entregado cumpla cada requerimiento especificado.

En esta actividad se debe confirmar que los requerimientos, una vez analizados y resueltos los posibles conflictos, corresponden realmente a las necesidades de clientes y usuarios. (Toro y Jiménez 2001) Los errores en el documento de requerimientos pueden conducir a elevar el costo en las fases de desarrollo del sistema o fases posteriores al lanzamiento del sistema. (Londoño y Guevara 2015)

#### **1.4.4 Técnicas para la Ingeniería de Requerimientos**

En el desarrollo de cada uno de los procesos de la IR antes descritos se utilizan diversas técnicas encaminadas a descubrir el dominio de aplicación, los servicios que debe proporcionar el sistema, el desempeño requerido, así como las restricciones de hardware; existen también técnicas complementarias que pueden emplearse para el descubrimiento de estos requerimientos.

A continuación se relacionan las técnicas propuestas por diferentes autores.

**ESTUDIO DE FACTIBILIDAD**

- ✓ Entrevista
- ✓ Tormenta de ideas
- ✓ Encuesta
- ✓ Observación
- ✓ Cuestionario
- ✓ Etnografía

(Sommerville 2010), (Escalona y Koch 2016),  
(Toro y Jiménez 2001)

**ADQUISICIÓN Y ANÁLISIS**

- ✓ Entrevista
- ✓ Tormenta de ideas
- ✓ Análisis de sistemas similares
- ✓ Encuesta
- ✓ Cuestionario
- ✓ JAD
- ✓ Mapa Conceptual
- ✓ Observación
- ✓ Prototipos de baja fidelidad

(Sommerville 2010), (Londoño and Guevara 2015), (Toro y Jiménez 2001)

**ESPECIFICACIÓN**

- ✓ Lenguaje natural
- ✓ Lenguaje formal
- ✓ Glosario de términos
- ✓ Prototipos de baja fidelidad
- ✓ Historias de Usuario
- ✓ Casos de Uso

(Sommerville 2010), (Toro and Jiménez 2001),  
(Londoño and Guevara 2015),

**VALIDACIÓN**

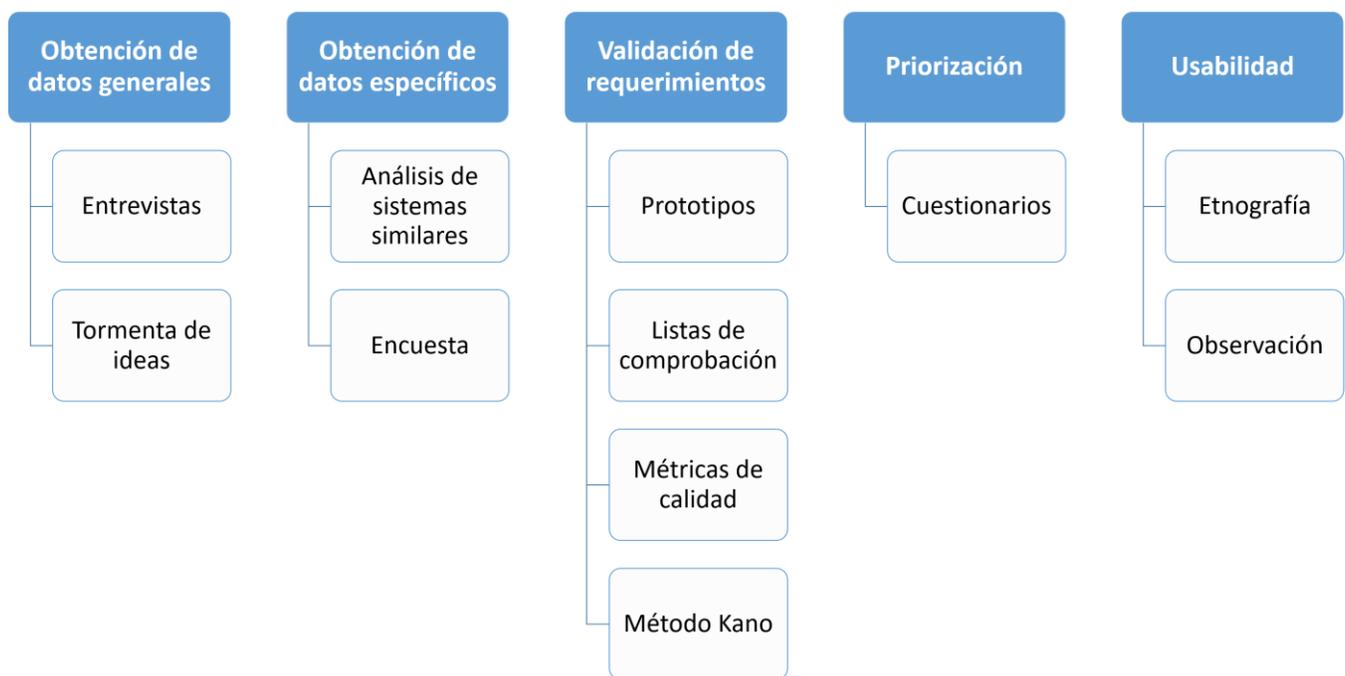
- ✓ Prototipos de alta fidelidad
- ✓ Revisiones de requerimientos
- ✓ Generación de casos de prueba
- ✓ Matrices de trazabilidad
- ✓ Opinión de expertos
- ✓ Inspecciones
- ✓ Revisiones guiadas
- ✓ Listas de comprobación
- ✓ Métricas de calidad
- ✓ Método de Kano

(Sommerville 2010), (Toro and Jiménez 2001),  
(Pressman 2010), (Duarte 2005).

Ilustración 8. Técnicas para la IR Fuente: Elaboración propia

Cada una de las técnicas seleccionadas cumple una función específica, por ello se toma como referencia la definición descrita por Londoño y Guevara (2015), con esto se propone agrupar las técnicas en las siguientes categorías: obtención de datos generales, obtención de datos específicos, validación de requerimientos, identificación de prioridades y técnicas orientadas a la usabilidad.

Teniendo en cuenta el estudio realizado, se presenta en la *Ilustración 9* las técnicas que se emplearán en cada una de las fases de la ingeniería de requerimientos asociada al desarrollo de un software de soporte al PLE de los estudiantes de la UCI, agrupadas estas según funciones específicas.



*Ilustración 9. Agrupación de las técnicas según funciones específicas. Fuente: Elaboración propia*

## 1.5 Metodologías y herramientas para el desarrollo de la ingeniería de requerimientos

Los siguientes epígrafes muestran la metodología de desarrollo de software empleada durante la realización del presente trabajo, así como las principales herramientas para el desarrollo y modelado de software.

### 1.5.1 La ingeniería de requerimientos en las metodologías de desarrollo de software

Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software, desde que surge la necesidad hasta que se cumpla el objetivo por el cual fue creado.

Algunas de las ventajas de utilizar una metodología son: (Figuroa, Solís et al. 2015)

- Facilitan las tareas de planificación.
- Optimizan el uso de recursos disponibles.
- Optimizan el conjunto y cada una de las fases del proceso de desarrollo.
- Permiten la reutilización de partes del producto.
- Contribuyen al logro de los plazos de tiempo fijados en la definición del proyecto.

Estas pueden clasificarse como se muestra en la siguiente ilustración.

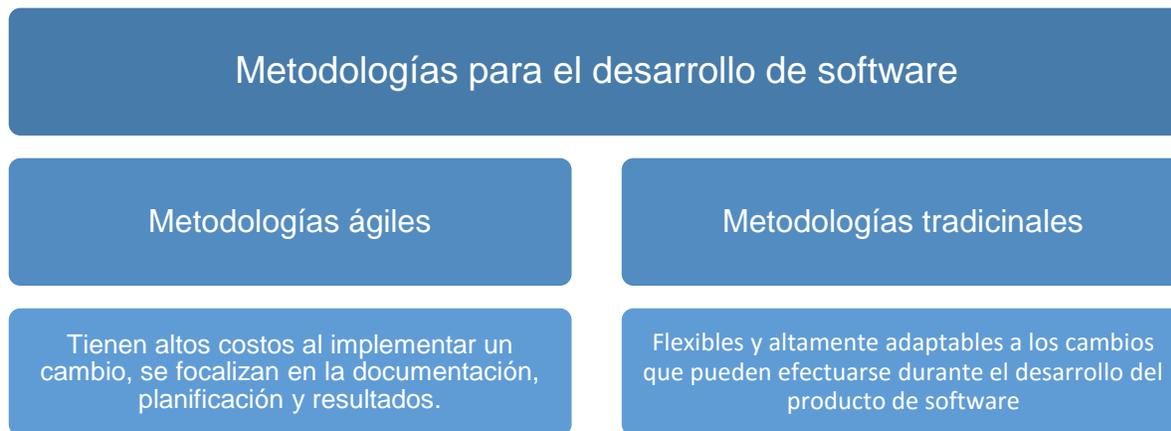


Ilustración 10. Clasificación de las metodologías para el desarrollo de software. Fuente: Elaboración propia

Entre las metodologías de desarrollo de software más empleadas se encuentran RUP, XP y AUP. A continuación, se presenta cómo se aborda en cada una de ellas la ingeniería de requerimientos.

## RUP

La gestión de requerimientos es una de las seis mejores prácticas que promueve la metodología RUP (Proceso Unificado de Desarrollo), por sus siglas en inglés Rational Unified Process. Para ello utiliza casos de uso y escenarios, los cuales capturan los RF y algunos RNF, guían el diseño, la implementación y la prueba del software.

RUP describe seis flujos de trabajo de ingeniería y tres de soporte, los que se ejecutan concurrentemente e interactúan entre sí usando los artefactos de los otros flujos. Uno de ellos es el flujo de trabajo de Requerimientos, el cual tiene por objetivo describir qué debe hacer el sistema, permitiendo que los

involucrados se pongan de acuerdo sobre dichas descripciones. Se elicitan las necesidades de los clientes y se documentan la funcionalidad requerida y restricciones. Se identifican actores y casos de uso que representan el comportamiento del sistema, creándose el modelo de casos de uso. Se describe en detalle cada caso de uso, cuya descripción muestra cómo el sistema interactúa paso a paso con los actores y qué hace el sistema. (Hadad 2007)

## **XP**

Programación extrema, en inglés Extreme Programming, es la metodología más destacada de los procesos ágiles de desarrollo de software, se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requerimientos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requerimientos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requerimientos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requerimientos. (Letelier y Penadés 2010)

Las historias de usuario son la técnica que usa XP para capturar requerimientos de software. Estas historias son confeccionadas por los usuarios, donde describen lo que el sistema debe hacer para ellos, o sea, para proveerles “valor de negocio”. Luego, las historias se descomponen en “tareas de ingeniería” que se asignan a los programadores para ser implementadas en una iteración. (Hadad 2007)

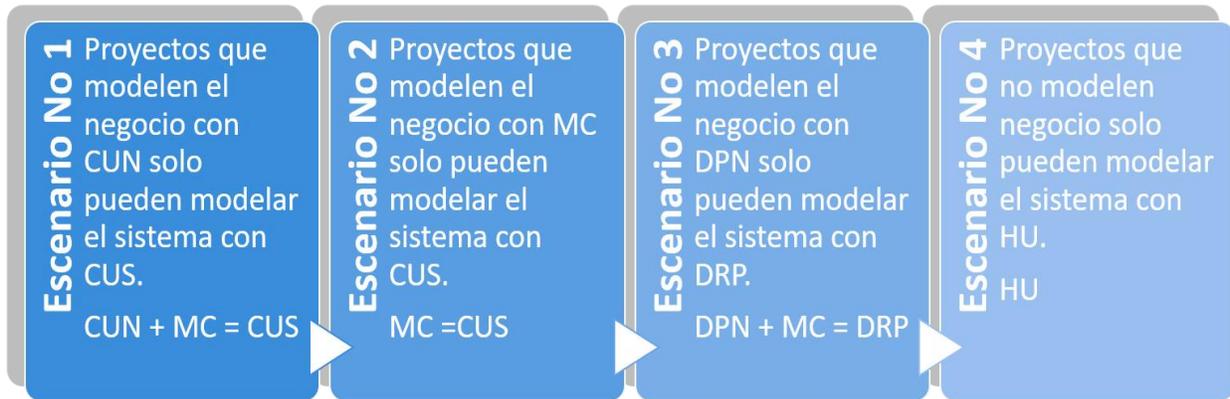
## **AUP**

Proceso ágil unificado, por sus siglas en inglés Agil Unified Process, es una versión simplificada del desarrollo de software basado en el Rational Unified Process, basado en disciplinas y entregables incrementales con el tiempo. Las disciplinas de AUP son: Modelado, Implementación, Prueba, Despliegue, Administración de la Configuración, Administración o Gerencia del Proyecto. (Pérez 2011) En la disciplina de Modelado agrupa los flujos de trabajo de Modelado de Negocio, Requerimientos y Análisis y Diseño.

## **AUP-UCI**

Para el ciclo de vida de los proyectos de la UCI se utiliza una variante de la metodología AUP llamada AUP-UCI. Esta metodología propone cuatro fases (inicio, elaboración, construcción, transición), pero modificando el objetivo de la misma, se unifican las restantes tres fases de AUP en una sola, a la que se llamó ejecución y se agrega la fase de cierre.

En la variación de la metodología AUP-UCI (Mejoras), existen tres formas de encapsular los requerimientos Casos de Uso del Sistema (CUS), Historias de usuario y Descripción de requerimientos por proceso (DRP), agrupados en cuatro escenarios, quedando como se muestra en la *Ilustración 11*.



*Ilustración 11. Escenarios metodología AUP-UCI según el Programa de Mejora. Fuente: Elaboración propia*

Teniendo en cuenta las características antes mencionadas, la necesidad de una metodología que responda con facilidad a los cambios continuos y siguiendo las políticas de desarrollo de software de la UCI, se tuvo en cuenta en el desarrollo de la ingeniería de requerimientos del sistema propuesto, lo planteado en la metodología AUP-UCI, encapsulando los requerimientos a través del escenario cuatro.

Para la selección de este escenario se tuvo en cuenta que el cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requerimientos y así poder implementarlos, probarlos y validarlos. Además, en sistemas web como el que se propone para gestionar el PLE de los estudiantes de la UCI, los requerimientos tienden a ser muy cambiantes y los documentos de especificación de requerimientos se vuelven obsoletos rápidamente. El tratamiento de las historias de usuario es muy dinámico y flexible, lo que permite que estas puedan reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas de una manera más fácil, sin invertir tanto tiempo y esfuerzo.

### **1.5.2 Lenguaje de modelado unificado**

El Lenguaje Unificado para la Construcción de Modelos (UML, por sus siglas en inglés) se define como un lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software. Es un sistema de notación (que, entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. Además es un lenguaje para construir modelos; no guía al desarrollador en la forma de realizar el análisis y diseño orientados a

objetos ni le indica cual proceso de desarrollo adoptar. (Rumbaugh, Jacobson et al. 2000). Por las razones antes expuesta se emplea este lenguaje para el modelado.

### **1.5.3 Herramienta CASE**

Las herramientas CASE<sup>7</sup> se destinan a automatizar los aspectos claves de todo el proceso de desarrollo de un sistema. Existen cuatro razones para adoptar las herramientas CASE: incrementar la productividad del analista, mejorar la comunicación entre analistas y usuarios, integrar las actividades del ciclo de vida y analizar y valorar el impacto de los cambios en el mantenimiento. (López, López et al. 2011)

Entre las herramientas CASE se encuentra:

#### **Visual Paradigm**

Visual Paradigm es una herramienta CASE. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. (Pressman 2010)

Se caracteriza por: (Pressman 2010)

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, con diferentes especificaciones.
- Licencia: gratuita y comercial.
- Soporta aplicaciones Web.
- Varios idiomas.

Por las características y funcionalidades antes expuesta se empleará esta herramienta para el modelado del sistema.

### **1.5.4 Herramienta para el diseño de los prototipos de baja fidelidad**

Para el diseño de los prototipos se utiliza Balsamiq<sup>8</sup>. Esta es una herramienta gráfica para esbozar las interfaces de usuario para los sitios web, aplicaciones móviles y de escritorio. Se centran en la fase de

---

<sup>7</sup> Ingeniería de software asistida por computadora

<sup>8</sup> <https://balsamiq.com/>

creación, proporcionando un diseño que permite enfocarse en la estructura en lugar de los colores y los iconos. Balsamiq permite escoger una gran variedad de objetos prediseñados como: barras de estado, menús, barras de progreso, entre otras.

### **1.5.5 Herramientas para el diseño de los prototipos de alta fidelidad**

#### **HTML 5**

Para el diseño de los prototipos se utiliza el lenguaje de marcas de hipertexto (HTML, por sus siglas en inglés HyperText Markup Language) porque este es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido, etc.). Además, este lenguaje permite que cualquier ordenador, independientemente del sistema operativo, puede leer o interpretar una página web. (Eguiluz 2013)

#### **CSS 3**

Se usa además las hojas de estilo en cascadas (CSS, por sus siglas en inglés Cascading Style Sheets) para definir el aspecto de cada elemento que se encuentre en el contenido como es: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, entre otros elementos del estilo del PLE. También brinda la posibilidad de poder usarse con otros lenguajes de programación como por ejemplo JavaScript para conseguir efectos dinámicos en las páginas. (Eguiluz 2013)

#### **JavaScript**

Se usa el lenguaje de programación JavaScript para crear páginas web dinámicas y con ello poder incorporar efectos como aparición y desaparición de texto, animaciones, acciones que se activan al pulsar botones u otros elementos y ventanas con mensajes de aviso al usuario. (Azustre 2016)

#### **Bootstrap**

Se hará uso de Bootstrap como framework CSS, este permite dar forma a un sitio web mediante librerías CSS que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web. Este es una excelente herramienta para crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Bootstrap es compatible con la mayoría de navegadores web del mercado, y más desde la versión tres. Además ofrece las herramientas necesarias para crear cualquier tipo de sitio web utilizando los estilos y elementos de sus librerías. (Otto y Thornton 2016)

La utilización de Bootstrap para el diseño de los prototipos de alta fidelidad simplifica el proceso de maquetado, sirviendo de guía para aplicar las buenas prácticas y los diferentes estándares. Por otro lado se puede hacer uso de muchos elementos web: desde iconos a desplegables, combinando HTML5, CSS y JavaScript . También es de código abierto, por lo que se puede usar de forma gratuita y sin restricciones.

### **Conclusiones del capítulo**

El estudio de los principales sistemas similares evidenció que los mismos no poseen las características necesarias para gestionar el PLE de los estudiantes de la UCI. El análisis de las principales definiciones de ingeniería de requerimientos y de sus etapas permitió realizar una comparación entre las propuestas de procesos de ingeniería de requerimientos establecidas por Sommerville, Pressman, Wieggers y el programa de la IREB, utilizando la propuesta de Sommerville para guiar la presente investigación. La investigación realizada a las diferentes metodologías de desarrollo de software, lenguaje de modelado, herramienta CASE y herramientas para el modelado de prototipos, permitió conocer cuáles son las que más se ajustan para el desarrollo de la ingeniería de requerimientos del PLE, en este caso las seleccionadas fueron: AUP-UCI, UML, Visual Paradigm, Balsamiq y Bootstrap como framework de desarrollo para el diseño de los prototipos de alta fidelidad.

## **CAPÍTULO 2. ADQUISICIÓN Y ESPECIFICACIÓN DE REQUERIMIENTOS**

### **Introducción**

En este capítulo se presenta el Estudio de Factibilidad, el desarrollo de la Adquisición-Análisis de los Requerimientos, así como su Especificación, las que constituyen las primeras fases planteadas por Sommerville (2010). En el desarrollo de estas etapas se detallan las técnicas empleadas para obtener la información necesaria en la identificación de los requerimientos del sistema. Con el propósito de conocer el nivel de acuerdo o desacuerdo con los objetivos, características y componentes del sistema que gestionará el PLE de los estudiantes de la UCI se aplican encuestas, haciendo uso de la escala de Likert. Se definen los requerimientos no funcionales y funcionales, encapsulando estos últimos a través de HU, según lo define el escenario cuatro de la metodología AUP-UCI.

### **2.1 Estudio de factibilidad**

A continuación, se describe el desarrollo del estudio de factibilidad y sus principales resultados, con el objetivo de valorar si el sistema que gestionará el PLE de los estudiantes de la UCI es útil para la UCI.

Previo a aplicar los instrumentos seleccionados para esta etapa se realizó un estudio de los conceptos relacionados con los PLE y los softwares que existen actualmente para gestionarlos, permitiendo de esta forma un mejor entendimiento de las necesidades del cliente y la confección de algunas interrogantes o ideas que se utilizarían como guía para lograr una mejor comunicación durante el proceso.

En la concepción del estudio de factibilidad se determinó la aplicación de una encuesta (0). Esta encuesta fue aplicada a especialistas en Ciencias Informáticas y profesores con experiencia en el desarrollo de software para el e-learning y otros temas relacionados con tecnología educativa; procedentes todos de la UCI. Se encuestaron un total de catorce expertos, de ellos dos Dr.C., tres MSc. y nueve Ingenieros en Ciencias Informáticas.

En el procesamiento de los datos aportados por los encuestados se empleó un método que consiste en identificar la frecuencia en cada categoría de la escala de Likert, definida en las preguntas 1, 2, 4 y 5 de la encuesta. El procesamiento se desarrolló con el empleo del asistente matemático SPSS. A continuación, se presentan los principales resultados de la aplicación de la encuesta.

Sobre el empleo de un software que gestiona el PLE de los estudiantes de la UCI y su contribución al logro de los objetivos generales del modelo del profesional, los encuestados manifestaron su aprobación como se puede apreciar en el *Gráfico 1*.



*Gráfico 1. Respuesta de encuestados correspondiente a la Pregunta 1*

Sobre la necesidad de integrar todas las herramientas que emplean los estudiantes de la UCI para aprender, en un software que gestiona el PLE, los encuestados respondieron según lo presentado en el *Gráfico 2*



*Gráfico 2. Respuesta de encuestados correspondiente a la Pregunta 2*

Relacionado con la afirmación “*Los estudiantes de la UCI gestionan sus PLE de forma correcta*” presentada a los encuestados, los resultados se muestran en el *Gráfico 3*, lo cual es una muestra de la necesidad de ofrecer recursos a los estudiantes para apoyarlos en la gestión de sus PLE.

Los estudiantes de la UCI gestionan sus PLE de forma correcta.

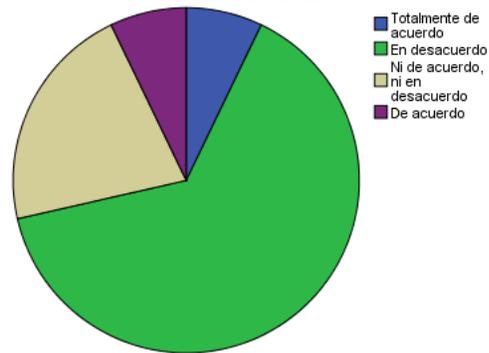


Gráfico 3. Respuesta de encuestados correspondiente a la Pregunta 4

Los encuestados coinciden en su mayoría, tal y como se presenta en el Gráfico 4, en que las herramientas que se mencionan, deben formar parte de los PLE de los estudiantes de la UCI, lo cual refuerza la necesidad de desarrollar un software que gestione sus PLE.

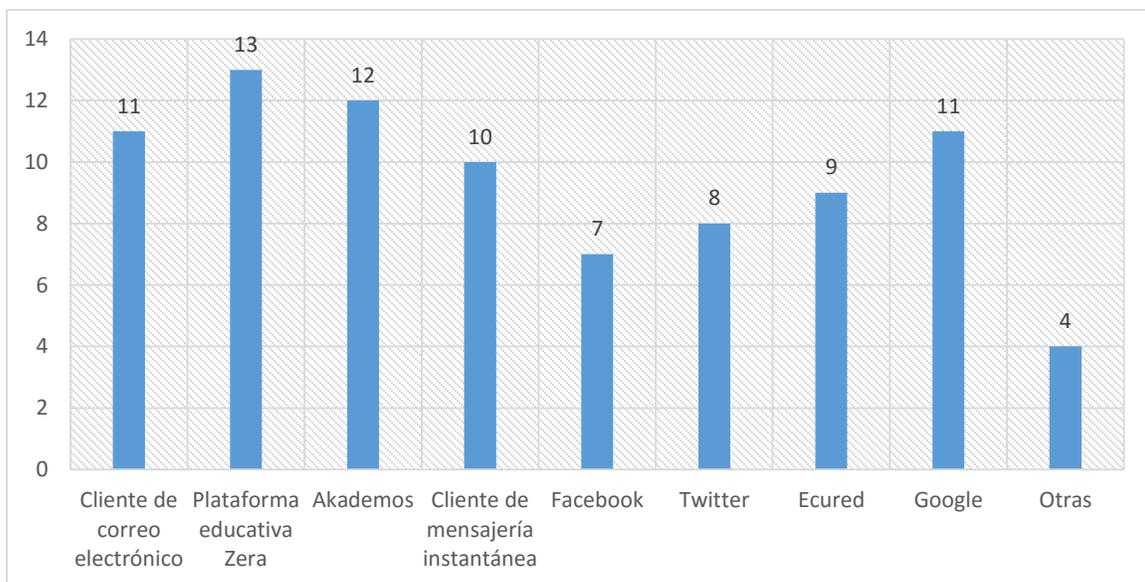


Gráfico 4 Herramientas que deben estar presentes en el PLE de los estudiantes de la UCI, según la respuesta de los encuestados en la Pregunta 3

### Análisis de sistemas similares

Se realizó el análisis de sistemas similares que presentan relación con el sistema que se va a realizar; en este caso se realizó el análisis de Symbaloo, iGoogle y Netvibes. El análisis de las diferentes funcionalidades que brindan estos sistemas permitió establecer las bases para lograr un mejor diseño del sistema a desarrollar e identificar un conjunto de funcionalidades aplicables al software que gestionará el PLE de los estudiantes de la UCI.

Las tres herramientas permiten a los usuarios organizar de forma visual todos los contenidos, tener en un mismo espacio acceso a toda la información, y todo ello compartirlo en mayor o menor medida, como queda reflejado en la siguiente tabla comparativa:

<i>Características</i>	<i>Symbaloo</i>	<i>iGoogle</i>	<i>Netvibes</i>
No necesita instalación	X	X	X
Es gratuito	X	X	X
Permite el uso compartido	X	X	X
Permite embebido en otras web	X		
Es compatible con los navegadores más usados	X	X	X
Permite integrar buscador de información	X	X	X
Incorpora por defecto widgets para algunos sitios populares	X	X	X
Permite visualizar la información dentro de cada uno de los widgets	X	X	X
Permite personalizar la interfaz	X	X	X
Permite personalizar los widgets	X		X

Permite syndicar <sup>9</sup> contenidos	X	X	X
Permite importar y exportar marcadores		X	X
Permite el uso en diferentes dispositivos	X	X	X
Permite crear diferentes pestañas para organizar los widgets	X		X

Tabla 4. Comparación entre las herramientas Fuente: (Rodríguez, Cormenzana et al. 2013)

A partir de los resultados presentados, se puede concluir que el estudio de factibilidad arrojó la utilidad de un software personalizado que gestione el PLE de los estudiantes de la UCI, lo cual valida la continuidad del proceso IR.

## 2.2 Adquisición y análisis de requerimientos

La etapa de análisis de requerimientos tomó como base de partida la información extraída de la encuesta aplicada a los especialistas y el análisis a los sistemas similares, técnicas detalladas en el epígrafe anterior. La información obtenida con el empleo de estas técnicas se transformó en los requerimientos apropiados para el diseño del sistema que dará soporte al PLE de los estudiantes de la UCI. Como técnicas de observación complementarias se utilizaron escenarios, técnica que permite a través de descripciones en la que un usuario realiza cierta actividad identificar requerimientos y la etnografía para estudiar el entorno del usuario y observar sus hábitos cotidianos. A partir de estas técnicas se infirieron requerimientos para soporte del software que gestionará el PLE de los estudiantes de la UCI.

Una vez procesada toda la información obtenida de las técnicas aplicadas se formalizaron por escrito, estableciendo los servicios que el sistema debe ser capaz de brindar y las

<sup>9</sup> Es el reenvío o reemisión de contenidos desde una fuente original (sitio web de origen) hasta otro sitio web de destino (receptor) que a su vez se convierte en emisor puesto que pone a disposición de sus usuarios los contenidos a los que en un principio sólo podían tener acceso los usuarios del sitio web de origen.

restricciones bajo las cuales debe operar. Los requerimientos identificados se clasificaron en funcionales y no funcionales; su especificación se realizó en la plantilla Especificación de Requerimientos de Software (ERS) propuesta por el Programa de Mejora.

Uno de los elementos que contiene esta plantilla es la prioridad, para ello existen criterios ó factores con el objeto de definir la priorización de los requerimientos a partir de la cuales, se establece un proceso de decisión tendiente a establecer el orden de implementación de los requerimientos de software. (Azzolini 2011)

La prioridad de los requerimientos del software que gestionará el PLE de los esudiantes de la UCI se determinó a partir del factor importancia. Este criterio de priorización se refiere al proceso en que los stakeholders manifiestan para cada requerimiento en particular un criterio de prioridad (Alta, Media, Baja) según el nivel de importancia que le confieren.

Luego de aplicar las técnicas y procesar sus resultados se idetificaron los siguientes requerimientos:

### **2.2.1 Descripción de requerimientos funcionales**

Para el correcto funcionamiento de la solución propuesta se espera que la aplicación permita:

<b>Requerimientos funcionales</b>			
<b>No</b>	<b>Requerimiento</b>	<b>Descripción</b>	<b>Prioridad</b>
<b>RF1.</b>	Añadir widget	El sistema debe permitir añadir un widget	Alta
<b>RF2.</b>	Editar widget	El sistema debe permitir cambiar el tamaño, color y posición del widget	Media
<b>RF3.</b>	Eliminar widget	El sistema debe permitir eliminar el widget	Alta
<b>RF4.</b>	Minimizar widget	El sistema debe permitir minimizar el widget	Baja
<b>RF5.</b>	Maximizar widget	El sistema debe permitir maximizar el widget	Baja

<b>RF6.</b>	Previsualizar widget	El sistema debe permitir previsualizar el widget antes de añadirlo	Media
<b>RF7.</b>	Compartir widget	El sistema debe permitir compartir el widget	Alta
<b>RF8.</b>	Autenticar usuario	El sistema debe permitir autenticarse	Alta
<b>RF9.</b>	Cerrar sesión de usuario	El sistema debe permitir cerrar la sesión	Alta
<b>RF10.</b>	Editar perfil	El sistema debe permitir editar el perfil de usuario	Media
<b>RF11.</b>	Recuperar contraseña	El sistema debe permitir recuperar la contraseña	Alta
<b>RF12.</b>	Personalizar interfaz	El sistema debe permitir personalizar la interfaz	Media
<b>RF13.</b>	Buscar	El sistema debe permitir buscar todas las coincidencias basadas en el criterio que introduzca el usuario	Alta
<b>RF14.</b>	Cambiar idioma	El sistema debe permitir seleccionar el idioma en el que desea se muestre el contenido de la aplicación	Baja
<b>RF15.</b>	Importar OPML <sup>10</sup>	El sistema debe permitir importar archivos OPML	Alta
<b>RF16.</b>	Exportar OPML	El sistema debe permitir exportar archivos OPML	Alta
<b>RF17.</b>	Añadir recordatorio	El sistema debe permitir añadir recordatorios de citas, cumpleaños, entre otras	Media
<b>RF18.</b>	Eliminar recordatorio	El sistema debe permitir eliminar los recordatorios	Media

<sup>10</sup> Outline Processor Markup Language) es un formato XML para esquemas (outlines). Se usa para listar varias fuentes RSS juntas; también se puede usar en programas gestores personales de información.

<b>RF19.</b>	Mover recordatorio	El sistema debe permitir mover el recordatorio	Media
<b>RF20.</b>	Editar recordatorio	El sistema debe permitir editar el recordatorio	Media
<b>RF21.</b>	Añadir escritorio	El sistema debe permitir añadir un escritorio	Alta
<b>RF22.</b>	Compartir escritorio	El sistema debe permitir compartir el escritorio	Alta
<b>RF23.</b>	Cambiar de vista a modo lectura	El sistema debe permitir cambiar de vista a modo lectura	Media
<b>RF24.</b>	Cambiar de vista a modo widget	El sistema debe permitir cambiar de vista a modo widget	Media
<b>RF25.</b>	Listar RSS <sup>11</sup>	El sistema debe permitir listar las RSS	Alta
<b>RF26.</b>	Refrescar listado de RSS	El sistema debe permitir refrescar el listado de RSS	Baja
<b>RF27.</b>	Marcar artículo como leído	El sistema debe permitir marcar el artículo como leído	Baja
<b>RF28.</b>	Guardar artículo	El sistema debe permitir guardar el artículo	Media
<b>RF29.</b>	Compartir artículo	El sistema debe permitir compartir el artículo	Media
<b>RF30.</b>	Añadir etiqueta	El sistema debe permitir añadir etiqueta	Media
<b>RF31.</b>	Eliminar etiqueta	El sistema debe permitir eliminar etiqueta	Media
<b>RF32.</b>	Listar etiqueta	El sistema debe permitir listar etiqueta	Baja
<b>RF33.</b>	Listar elementos leídos	El sistema debe permitir listar elementos leídos	Media
<b>RF34.</b>	Listar elementos guardados	El sistema debe permitir listar elementos guardados	Media

<sup>11</sup> Really Simple Syndication, un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS (agregador).

<b>RF35.</b>	Cambiar vista del listar a modo expandida	El sistema debe permitir cambiar vista del listar a modo expandida	Media
<b>RF36.</b>	Cambiar vista del listar a modo mosaico	El sistema debe permitir cambiar vista del listar a modo mosaico	Media
<b>RF37.</b>	Activar escritorio público	El sistema debe permitir seleccionar cual escritorios se mostrará como público	Media
<b>RF38.</b>	Listar escritorios	El sistema debe permitir listar los escritorios	Media
<b>RF39.</b>	Importar hojas de cálculo	El sistema debe permitir importar hojas de cálculo	Baja
<b>RF40.</b>	Comparar hojas de cálculo	El sistema debe permitir comparar hojas de cálculo	Baja
<b>RF41.</b>	Recomendar contenido	El sistema debe ser capaz de recomendar contenido según las preferencias del usuario	Alta
<b>RF42.</b>	Establecer un ecosistema de widget	El sistema debe adoptar normas de widgets que ayuden a establecer un ecosistema de widgets que se pueden ejecutar en diferentes plataformas	Alta
<b>RF43.</b>	Intercambiar información entre widgets	El sistema debe tener una capa de comunicación que permita la comunicación entre los widgets	Alta
<b>RF44.</b>	Importar lista de amigos	El sistema debe ser capaz de importar una lista de usuario exportados previamente por el usuario	Alta
<b>RF45.</b>	Exportar lista de amigos	El sistema debe ser capaz de exportar la lista de amigos en formato .csv	Alta

*Tabla 5. Requerimientos funcionales*

## **2.2.2 Descripción de los requerimientos no funcionales**

Dentro de los requerimientos no funcionales identificados para el curso de esta investigación están los que se muestran a continuación:

---

### Requerimientos no funcionales

---

#### Seguridad

- RNF 1. Autenticación basada en el dominio UCI.
- RNF 2. Validación de datos en el cliente, para evitar estados inconsistentes en la información y posibles ataques al sistema.
- RNF 3. Uso de certificado STS.
- RNF 4. Uso del protocolo de seguridad SSL para asegurar que los datos viajen de manera íntegra.
- RNF 5. Envío y almacenamiento encriptado de la información sensible a través del método SHA3.
- RNF 6. Usar la capa de seguridad CSP para detectar y mitigar ataques XSS.

---

#### Usabilidad

- RNF 7. Cumplir con las pautas de diseño establecidas en la Estrategia Marcaria de la Universidad. Cuando se crea/actualiza/elimina un elemento, así como el cancelar, se muestra un mensaje con el resultado de la acción.
  - RNF 8. El sistema debe ser usable por cualquier usuario sin importar la experiencia que tenga, para esto el sistema debe poseer una interfaz gráfica donde se encuentren las herramientas que el usuario necesita para organizar su PLE sin necesidad de una extensa capacitación o supervisión constante.
  - RNF 9. El sistema tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de todas las funcionalidades de la aplicación.
  - RNF 10. La interfaz debe ser adaptable a todas las resoluciones de pantalla.
  - RNF 11. El sistema será capaz de actualizar en tiempo real los datos de los widgets.
-

	RNF 12. El sistema presentará buena navegabilidad, proporcionando varias opciones para acceder a cada servicio.
<b>Soporte</b>	RNF 13. El sistema debe ser compatibles con los navegadores Firefox y Chrome.
<b>Desarrollo</b>	RNF 14. El sistema debe ser escalable <sup>12</sup> .

*Tabla 6. Requerimientos no funcionales*

### 2.3 Especificación de requerimientos

Para realizar la especificación de requerimientos del sistema que dará soporte al PLE de los estudiantes de la UCI se hace uso de las Historias de Usuario, técnica que propone el escenario 4 de la metodología AUP-UCI para encapsular los requerimientos funcionales del sistema. Con el empleo de esta técnica se describió qué se espera como salida de la implementación, y cómo se ve beneficiado el usuario final. Se expresa en lenguaje natural y sencillo, considerando que los usuarios finales tienden a desconocen el lenguaje técnico. A continuación, se muestran las historias de usuario correspondientes a los requerimientos funcionales “Añadir widget”, “Eliminar Widget”, “Listar RSS” y “Añadir escritorio”.

<b>Número:</b> 1	<b>Nombre del requerimiento:</b> Añadir widget
<b>Programador:</b> Taire Faife Rodriguez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 3 días
<b>Riesgo en Desarrollo:</b> N/A	
<b>Descripción:</b> <b>1- Objetivo:</b> Permitir añadir un widget al sistema <b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para añadir un widget hay que:	

<sup>12</sup> capacidad para adaptarse a condiciones de operación más exigentes y responder a tiempo y de forma satisfactoria a los requerimientos de los clientes.

- Estar autenticado en el sistema.

### 3- Comportamientos válidos y no válidos (flujo central y alternos):

Buscar: campo de texto que admite caracteres alfabéticos y numéricos, tiene un máximo de hasta 100 caracteres.

Aplicaciones: campo de selección.

### 4- Flujo de la acción a realizar:

- El sistema debe permitir añadir un nuevo widget a la aplicación.
- Cuando el usuario selecciona el widget que desea agregar la aplicación muestra una ventana de diálogo con la previzualización del widget, en dicha ventana se selecciona a que pestaña de la aplicación desea agregarlo. Si selecciona la opción Agregar, se crea un nuevo widget y la aplicación muestra el siguiente mensaje de información: *“Se ha incluido el elemento satisfactoriamente”*.
- Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al usuario de realizar nuevamente la acción.
- Si selecciona la opción Cancelar regresará a la vista previa.

### Prototipo de interfaz:

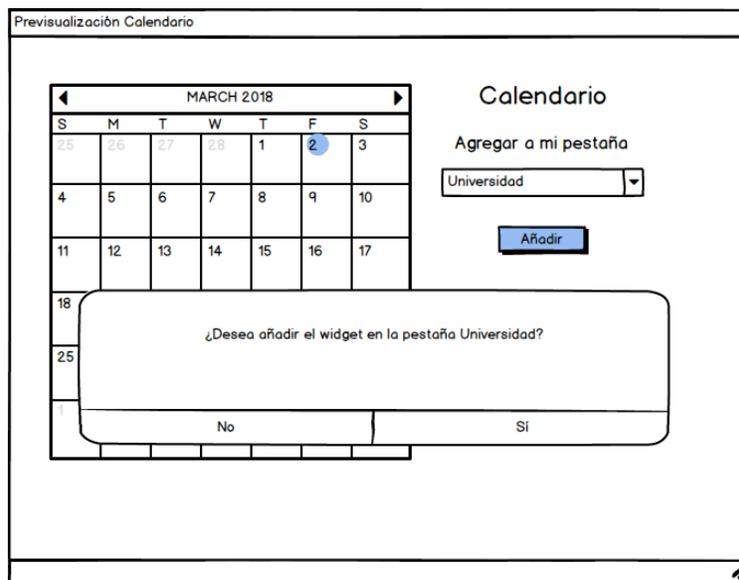


Tabla 7. Historia de Usuario Añadir Widget

<b>Número:</b> 2	<b>Nombre del requerimiento:</b> Eliminar widget	
<b>Programador:</b> Taire Faife Rodriguez		<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> N/A		<b>Tiempo Real:</b> N/A
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b></p> <p>Permitir eliminar un widget en la aplicación.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b></p> <p>Para eliminar un widget hay que:</p> <ul style="list-style-type: none"> <li>- Estar autenticado en el sistema.</li> <li>- Debe existir en el sistema al menos un widget activo.</li> </ul> <p><b>3- Flujo de la acción a realizar:</b></p> <ul style="list-style-type: none"> <li>- El sistema permite eliminar uno o varios widgets, para eliminar varios widgets el sistema muestra un listado de los mismos permitiendo marcar más de un elemento a eliminar, selecciona la opción eliminar de las acciones en lote y presiona la opción Aplicar.</li> <li>- En caso de querer eliminar un elemento se puede realizar seleccionando la opción eliminar de las opciones que muestra el propio elemento o desde la vista previa del mismo. Para ambos casos el sistema muestra un mensaje de confirmación para Aceptar o no la acción que se está realizando.</li> <li>- Si selecciona la opción Aceptar se actualiza la vista de los widgets y el sistema muestra un mensaje de información.</li> </ul>		
<p><b>Observaciones:</b> los widgets muestran el contenido de las aplicaciones por lo que sí es eliminado el usuario dejará de recibir las notificaciones de la aplicación asociada a dicho widget.</p>		

### Prototipo de interfaz:

CALENDARIO ✎ ↻ ✓ 🗑

MARCH 2018

S	M	T	W	T	F	S
25	26	27	28	1	2	3

¿Desea eliminar el widget?

No
Sí

25	26	27	28	29	30	31
1	2	3	4	5	6	7

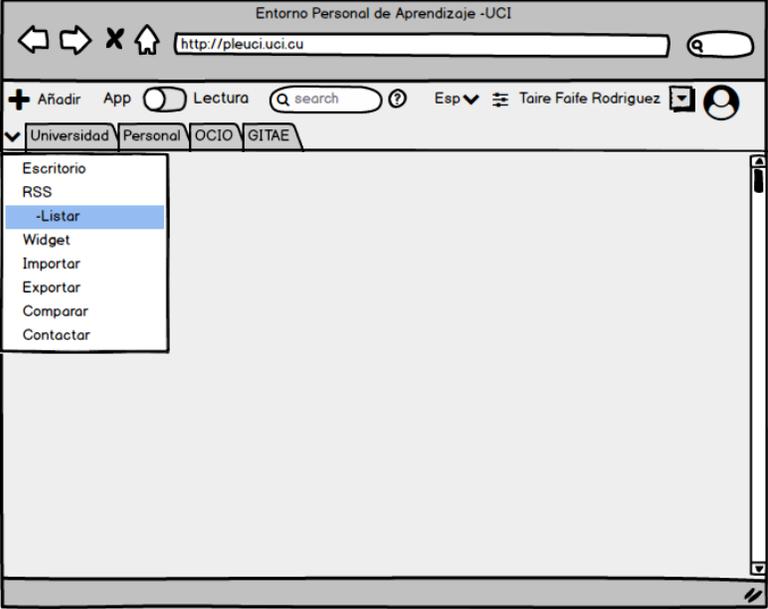
Widgets

- Seleccionar
- Calendario
- Calculadora
- Correo electrónico
- Cliente de mensajería

Eliminar

Tabla 8 Historia de Usuario Eliminar Widget

<b>Número:</b> 25	<b>Nombre del requisito:</b> Listar RSS
<b>Programador:</b> Taire Faife Rodriguez	<b>Iteración Asignada:</b> 1ra

<b>Prioridad: Alta</b>	<b>Tiempo Estimado: 1 día</b>
<b>Riesgo en Desarrollo: N/A</b>	<b>Tiempo Real: N/A</b>
<b>Descripción:</b>	
<b>1- Objetivo:</b>	
Permitir al usuario listar las RSS	
<b>2- Acciones para lograr el objetivo (precondiciones y datos):</b>	
Para listar las RSS:	
Hay que estar autenticado en el sistema	
El usuario debe haberse suscrito al menos a una página	
<b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b>	
<b>4- Flujo de la acción a realizar:</b>	
<ul style="list-style-type: none"><li>• Seleccionar la opción Listar RSS</li><li>• El sistema mostrará una lista de todas las RSS del usuario</li></ul>	
<b>Prototipo</b>	
 A screenshot of a web browser window titled "Entorno Personal de Aprendizaje -UCI". The address bar shows "http://pleuci.uci.cu". The browser interface includes a search bar, a user profile "Taire Faife Rodriguez", and a navigation menu with tabs for "Universidad", "Personal", "OCIO", and "GITAE". A dropdown menu is open under "Personal", listing options: "Escritorio", "RSS", "-Listar" (highlighted), "Widget", "Importar", "Exportar", "Comparar", and "Contactar".	

Leído	RSS	Día	Hora	Descripción	Opciones
<input type="checkbox"/>					[Download] [Share] [Delete]
<input type="checkbox"/>					[Download] [Share] [Delete]
<input type="checkbox"/>					[Download] [Share] [Delete]
<input type="checkbox"/>					[Download] [Share] [Delete]
<input type="checkbox"/>					[Download] [Share] [Delete]

Tabla 9. Listar RSS

<b>Número:</b> 21	<b>Nombre del requisito:</b> Añadir escritorio	
<b>Programador:</b> Taire Faife Rodriguez	<b>Iteración Asignada:</b> 1ra	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 1 día	
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> N/A	
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> Permitir al usuario añadir un escritorio.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para añadir un escritorio hay que estar autenticado en el sistema.</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b></p> <ul style="list-style-type: none"> <li>• Seleccionar la opción añadir escritorio.</li> <li>• El usuario podrá introducir el nombre del escritorio, o editarlo una vez añadido.</li> </ul>		

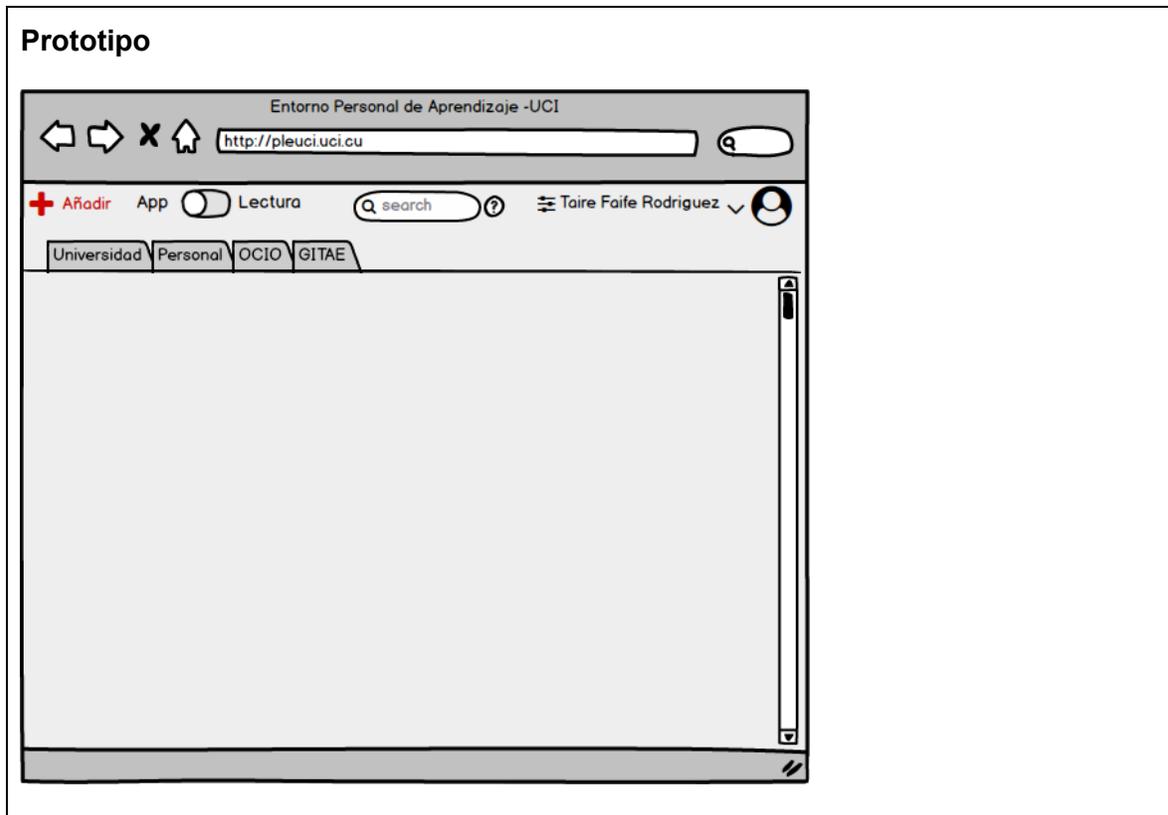


Tabla 10. Añadir escritorio

## Conclusiones del capítulo

El estudio de factibilidad desarrollado arrojó la utilidad de un software personalizado que gestione el PLE de los estudiantes de la UCI. La estrategia de obtención de requerimientos puesta en práctica, permitió identificar las funcionalidades que el sistema que gestionará el PLE de los estudiantes de la UCI debe brindar, así como las restricciones sobre las que debe operar. Las técnicas de captura de información empleadas en la etapa de adquisición y análisis dieron paso a obtener los requerimientos tanto funcionales como no funcionales, con el fin de darle solución a las necesidades del cliente. A partir de la captura de los requerimientos se especificaron los mismos en términos de Historias de Usuario. El soporte utilizado para plasmar los resultados obtenidos durante el desarrollo de las etapas detalladas en este capítulo se realizó acorde a las plantillas propuestas en el Programa de Mejora para los documentos de ERS.

## **CAPÍTULO 3. VALIDACIÓN DE REQUERIMIENTOS**

### **Introducción**

En este capítulo se describe la validación de requerimientos del software. Se presenta la aplicación de las métricas para medir la calidad de los requerimientos identificados y comprobar que estén descritos correctamente. Se presenta un prototipo de la interfaz de usuario para representar aspectos más precisos del sistema y verificar que los requerimientos cumplen con las expectativas del cliente. Además, se aplica el método Kano para medir la satisfacción del cliente.

### **3.1 Diseño del prototipo de interfaz de usuario**

Como técnica para validar los requerimientos del sistema se realiza el diseño de un prototipo de alta fidelidad, con el objetivo de proporcionar un entorno visual sencillo que permita al cliente adaptarse al trabajo con los requerimientos especificados. Esta técnica permite además detectar posibles errores en la descripción de la especificación que no habían sido identificados.

Los prototipos de alta fidelidad permiten representar aspectos más precisos del sistema, estos son útiles para detallar el proceso interactivo de una o varias tareas concretas. La propuesta se basa en obtener de la especificación de requerimientos prototipos que, sin tener la totalidad de las funcionalidades, permitan al usuario hacerse una idea de la estructura y las opciones de navegabilidad del sistema.

A continuación, se muestran algunas imágenes que ilustran algunas vistas del prototipo diseñado.

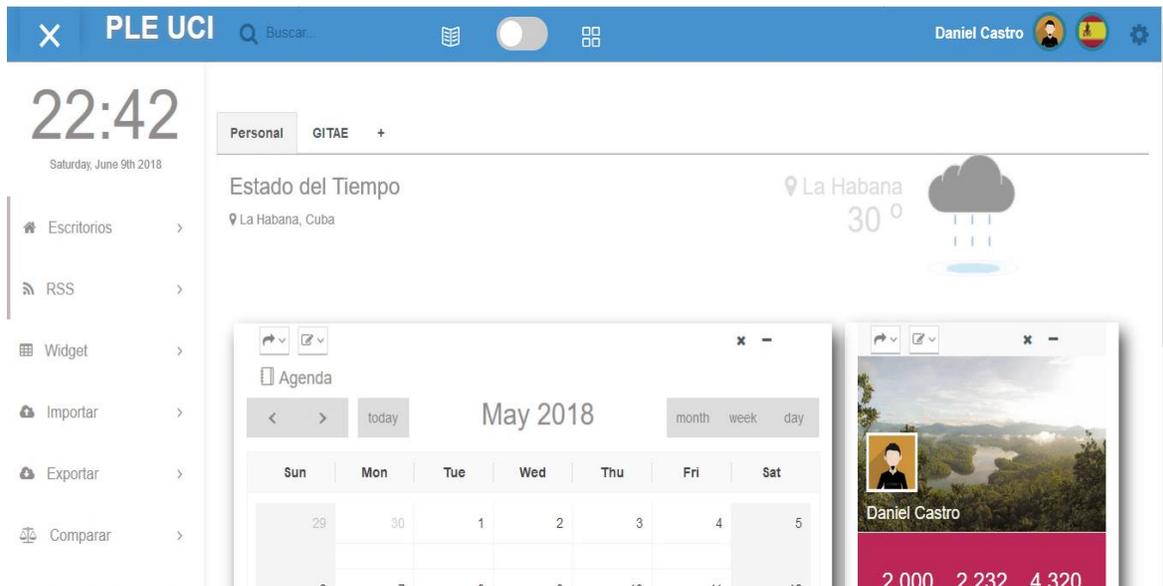


Ilustración 12. Prototipo alta fidelidad/Vista principal

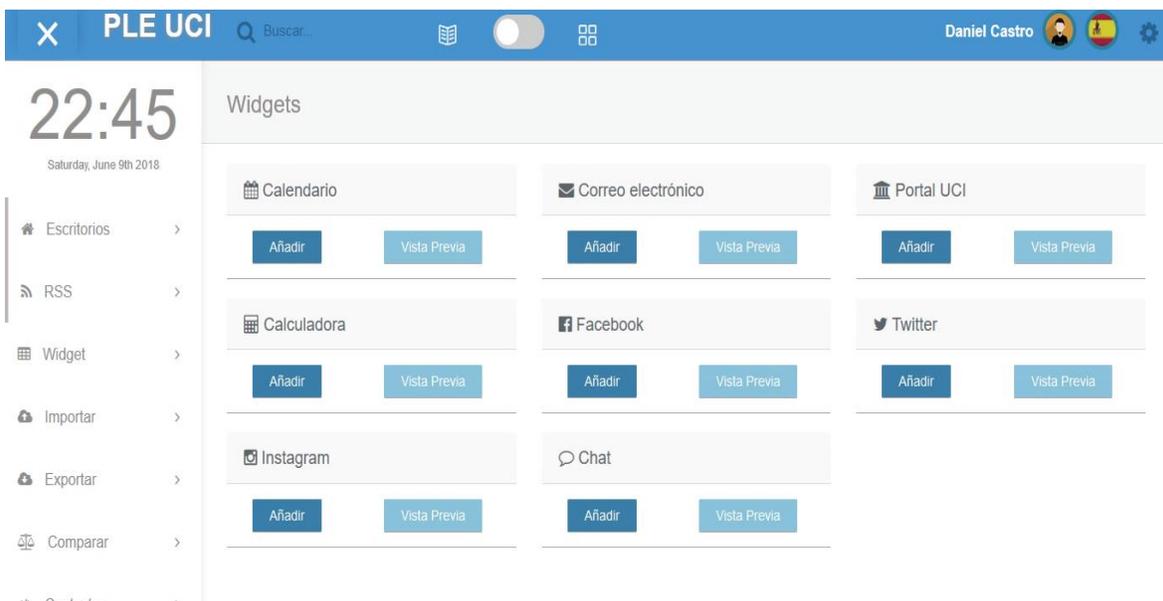


Ilustración 13. Prototipo alta fidelidad/Añadir Widget

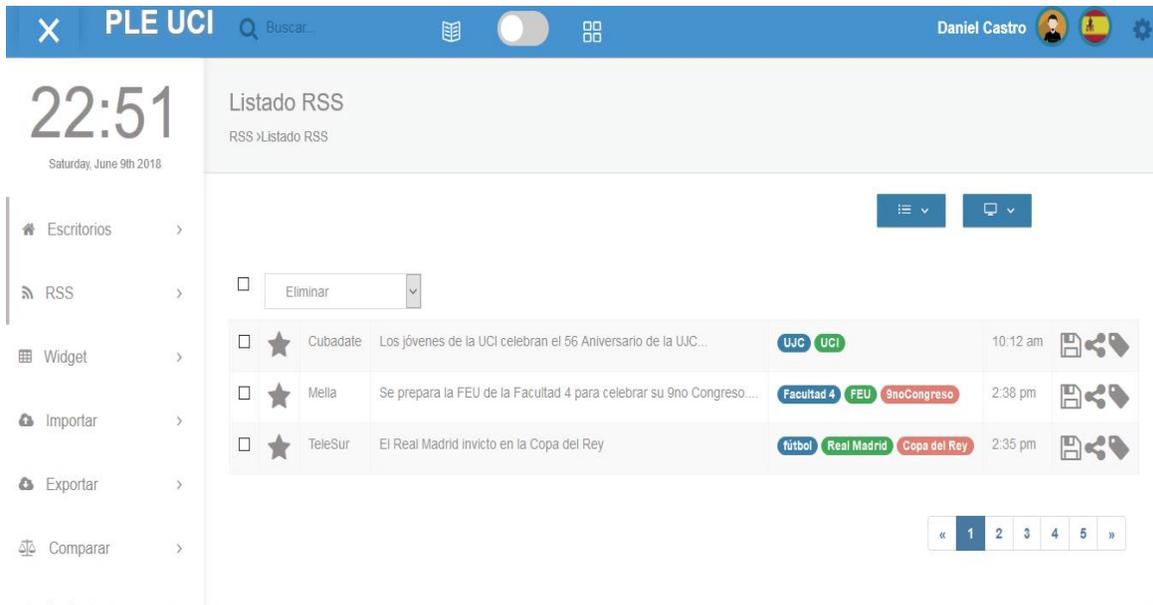


Ilustración 14. Prototipo de alta fidelidad/Listar RSS

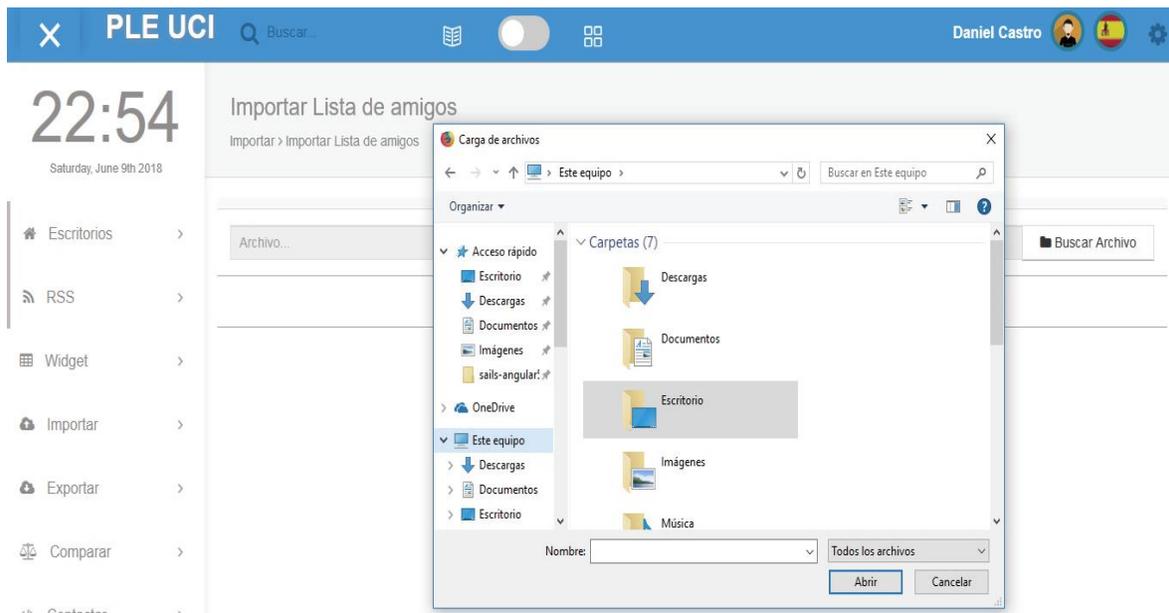


Ilustración 15. Prototipo de alta fidelidad/Importar lista de amigos

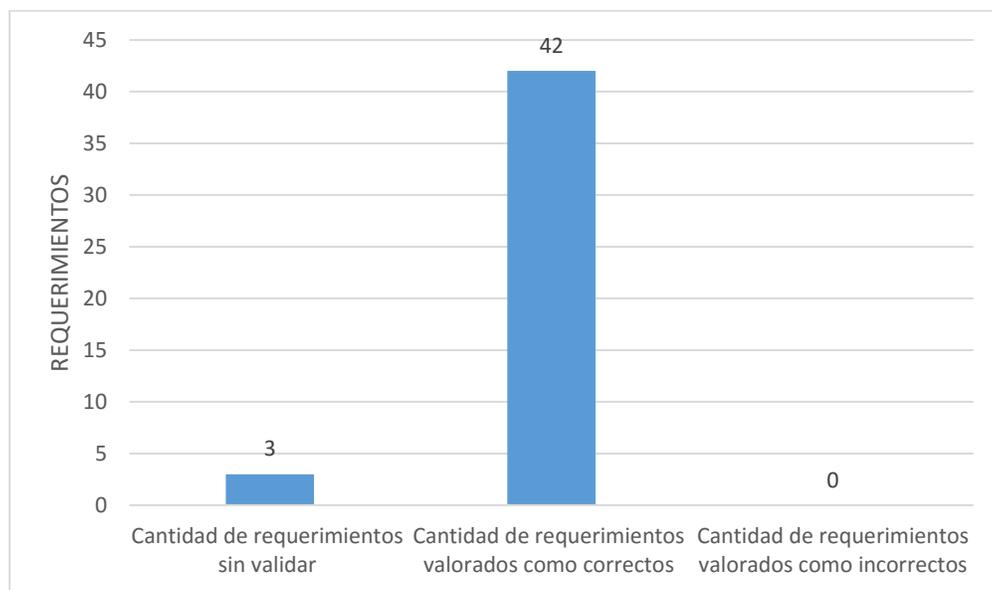
El prototipo desarrollado fue mostrado a un grupo de especialistas con el objetivo de chequear que las funcionalidades representadas estuvieran en correspondencia con los requerimientos descritos en la etapa de especificación. Para ello los especialistas debían

rellenar los datos en una encuesta donde debían indicar si el requerimiento estaba correcto o incorrecto.

Los datos arrojados por el instrumento (*Gráfico 5*) permitieron calcular el grado en el que se validaron los requerimientos ( $G_{VR}$ ) a partir de la siguiente métrica propuesta por Pressman en (Pressman 2010).

$$G_{VR} = R_{VC} / (R_{VC} + R_{NV})$$

Donde  $R_{VC}$  es la cantidad de requerimientos valorados como correctas y  $R_{NV}$  la cantidad de requerimientos sin validar.



*Gráfico 5 . Análisis del prototipo contra la descripción de requerimientos*

$$G_{VR} = 42 / (42 + 3) \quad G_{VR} = 0.93$$

Después de procesados los resultados se obtuvo que los requerimientos fueron validados con un grado de 0.93, lo que representa que el 93 % de los requerimientos descritos fueron validados por los especialistas. Los tres requerimientos que no fueron validados a través del diseño del prototipo de interfaz de usuarios del sistema serán validados a través una métrica que se describe en el apartado *Métricas cuantitativas para medir la calidad para la especificación de requerimientos*.

### **3.2 Medición del grado de satisfacción del Cliente.**

En este epígrafe se presentan los resultados obtenidos una vez aplicado el método de Kano. Para ello se muestra primeramente los requerimientos analizados, luego se muestra

el mapa de respuestas obtenidas en cada uno de los requerimientos evaluados y por último se lleva a cabo la clasificación de los requerimientos según los resultados obtenidos.

### **Selección de los Requerimientos.**

Para la selección de la muestra se tuvo en cuenta que fueran requerimientos que permitieran poder realizar acciones importantes en el sistema. En la siguiente tabla se muestra el listado de requerimientos seleccionados.

1. Añadir widget
2. Listar RSS
3. Editar perfil
4. Compartir widget
5. Eliminar widget
6. Personalizar interfaz
7. Buscar
8. Importar OPML<sup>13</sup>
9. Exportar OPML
10. Añadir recordatorio
11. Mover recordatorio
12. Añadir escritorio
13. Importar lista de amigos
14. Exportar lista de amigos

---

<sup>13</sup> Outline Processor Markup Language) es un formato XML para esquemas (outlines). Se usa para listar varias fuentes RSS juntas; también se puede usar en programas gestores personales de información.

## Cuestionario

El cuestionario (Anexo 5. ) fue realizado a una muestra de 15 estudiantes de la Facultad 4 de la UCI; 3 de segundo año, 3 de tercer año, 4 de cuarto año y 5 de quinto año. Para cada requerimiento se mostraron tres preguntas: una para medir cómo se siente si la característica está presente en el sistema, otra para medir cómo se siente si la característica no está presente en el sistema y la otra el grado de importancia que le atribuye al requerimiento especificado. Para cada pregunta el usuario podía responder entre 5 opciones.

En base a las respuestas a las dos primeras preguntas se busca su combinación en la *Tabla 11* y es así que el requerimiento del sistema se puede clasificar en una de estas 6 categorías: A: Atractivo, O: Obligatorio, U: Unidimensional, I: Indiferencia, Inv: Respuesta inversa, D: Respuesta dudosa.

		Requerimientos disfuncionales				
Requerimientos disfuncionales		1	2	3	4	5
	1	D	A	A	A	U
	2	Inv.	I	I	I	O
	3	Inv.	I	I	I	O
	4	Inv.	I	I	I	O
	5	Inv.	Inv.	Inv.	Inv.	D

*Tabla 11. Clasificación de requerimientos*

El método de Kano establece, para cada requerimiento del cliente, la relación entre satisfacción y funcionalidad y permite discriminar y clasificar los requerimientos. La *Ilustración 16* ayuda a comprender esta agrupación.

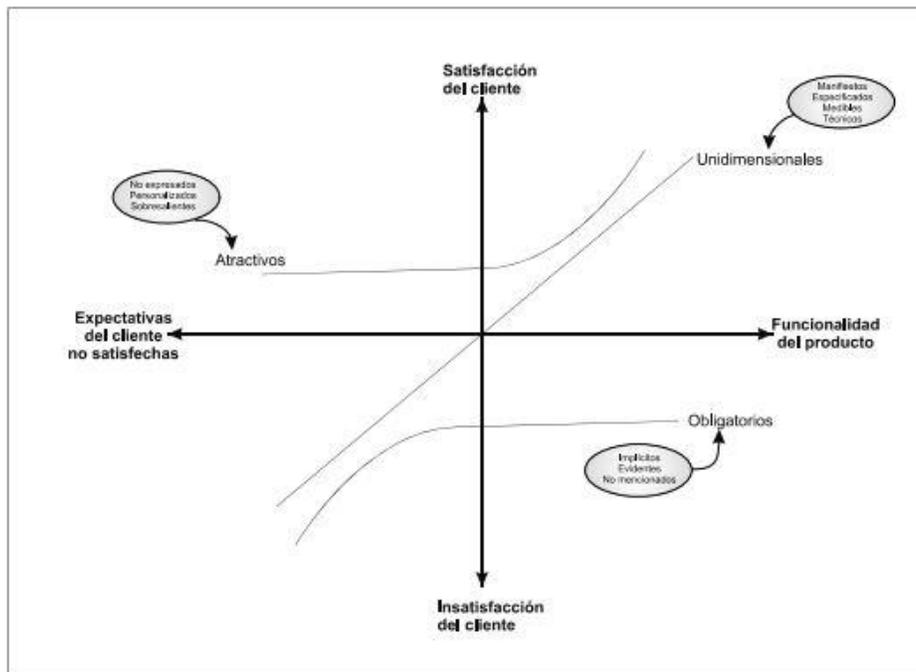


Ilustración 16. Tipos de Requerimientos del cliente Fuente: (Duarte 2005)

En el procesamiento de los datos aportados por los encuestados se empleó el método explicado en el epígrafe 2.2 y el procesamiento se desarrolló con el empleo del asistente matemático SPSS.

### Representación de las Respuestas

A continuación se muestra un análisis detallado del mapa de las respuestas, las cuales son agrupadas como corresponde según la *Tabla 11*.

Para la realización de la tabla que agrupan los datos de clasificación de cada uno de los requerimientos según las respuestas de los encuestados, se utilizó la propuesta hecha por León Duarte (2005). Esta enfatiza que conviene saber si el requerimiento es considerado importante por los clientes, es por ello que el término importancia es uno de los valores reflejados en la *Tabla 12*, que no es más que la importancia promedio que le atribuyen los encuestados a cada uno de los requerimientos (columna "Imp"). Las columnas C1 y C2 se obtienen de la multiplicación de los valores Mejor y Peor por la importancia promedio respectivamente.

Req.	A	O	U	I	Inv	D	Imp.	Mejor	Peor	C1	C2
------	---	---	---	---	-----	---	------	-------	------	----	----

1.	2	3	10	0	0	0	0.88	0,8	0,86	0,704	0,75
2.	3	9	1	1	0	0	0.76	0,28	0,72	0,21	0,54
3.	8	2	3	1	0	1	0.68	0,8	0,33	0,54	0,22
4.	8	1	4	2	0	0	0.52	0,8	0,33	0,416	0,17
5.	2	11	2	1	0	0	0.88	0,25	0,81	0,22	0,71
6.	9	2	4	0	0	0	0.75	0,86	0,4	0,645	0,3
7.	1	10	2	0	1	1	0.88	0,21	0,86	0,2024	0,75
8.	9	1	2	1	1	1	0.66	0,84	0,23	0,55	0,15
9.	9	2	1	3	0	0	0.66	0,66	0,2	0,39	0,13
10.	10	3	1	1	0	0	0.65	0,73	0,26	0,47	0,16
11.	9	4	0	2	0	0	0.65	0,6	0,26	0,39	0,16
12.	2	2	11	0	0	0	0.67	0,86	0,86	0,57	0,57
13.	2	3	9	1	0	0	0.67	0,73	0,8	0,48	0,53
14.	3	4	8	0	0	0	0.64	0,73	0,8	0,48	0,53

Tabla 12. Clasificación de los requerimientos

### Clasificación de los Requerimientos

Una vez analizados los datos recogidos en las encuestas se clasificaron los requerimientos donde resultaron ser: atractivos 6, unidimensionales 3, obligatorios 3 y ninguno indiferente.

La distribución de los requerimientos se puede ver en la *Ilustración 17*, al igual que en la *Tabla 13* donde se listan los requerimientos con su respectiva clasificación.

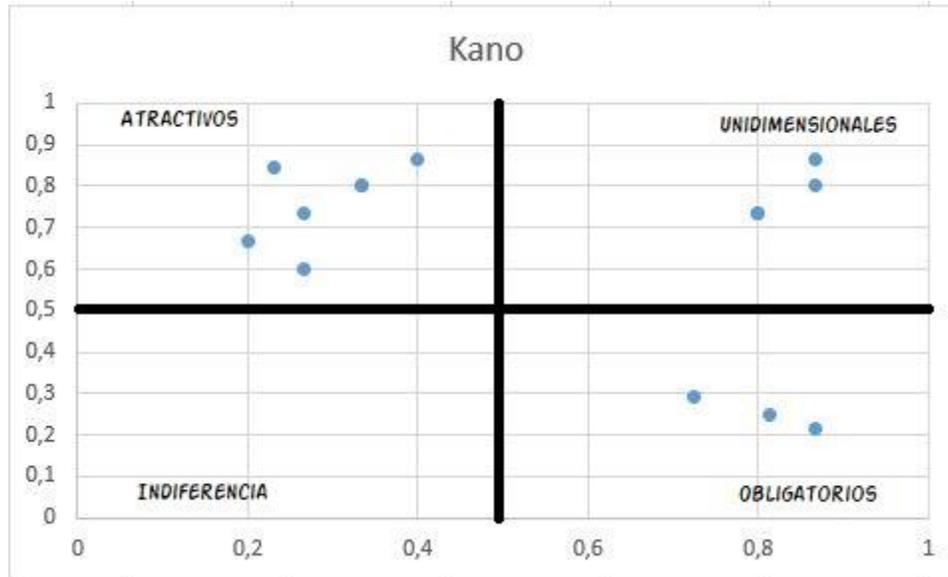


Ilustración 17. Clasificación de los requerimientos en función de los resultados obtenidos

No	Requerimiento	Clasificación
1.	Añadir widget	Unidimensional
2.	Listar RSS	Obligatorio
3.	Editar perfil	Atractivo
4.	Compartir widget	Atractivo
5.	Eliminar widget	Obligatorio
6.	Personalizar interfaz	Atractivo
7.	Buscar	Obligatorio
8.	Importar OPML <sup>14</sup>	Atractivo
9.	Exportar OPML	Atractivo

<sup>14</sup> Outline Processor Markup Language) es un formato XML para esquemas (outlines). Se usa para listar varias fuentes RSS juntas; también se puede usar en programas gestores personales de información.

10	Añadir recordatorio	Atractivo
11	Mover recordatorio	Atractivo
12	Añadir escritorio	Unidimensional
13	Importar lista de amigos	Unidimensional
14	Exportar lista de amigos	Unidimensional

*Tabla 13. Clasificación de los requerimientos en función de los resultados obtenidos*

Luego de quedar clasificados todos los requerimientos se puede apreciar que la satisfacción de los clientes es proporcional a la funcionalidad de las características del software, manifestándose esto en la concentración de la clasificación de los requerimientos en los tres tipos ideales de atributos (Obligatorios, Atractivos y Unidimensionales).

### **3.3 Métricas cuantitativas para medir la calidad para la especificación de requerimientos**

De manera ideal, los requerimientos deben ser claros, sin ambigüedades, fáciles de entender, completos y consistentes. (Sommerville 2010) A fin de comprobar que la especificación de requerimientos cumple con estas características se sometió a una revisión por especialistas. Cada revisor luego de haber inspeccionado el documento de ERS debía responder un cuestionario (Anexo 9. ). Para la confección del mismo se tuvieron en cuenta cuestiones que permitieran identificar los siguientes aspectos:

- Especificaciones de requerimientos incompletas.
- Falta de claridad y entendimiento en los requerimientos.

Para determinar la **especificidad o falta de ambigüedad** de los requerimientos, se aplica una métrica basada en la consistencia de la interpretación de los revisores para cada requerimiento (Pressman 2010):

$$Q_1 = N_{u1} / N_r$$

Donde:

- $N_{u1}$  es el número de requerimientos para los cuales todos los revisores tienen interpretaciones idénticas. Mientras más cercano a 1 esté el valor de  $Q$ , menor será la ambigüedad de la especificación.
- $N_r$  son los requerimientos que hay en una especificación. ( $N_r = N_f + N_{nf}$ )

### Resultados de las métricas de calidad para la especificación de requerimientos

Después de aplicadas las métricas para medir la especificidad, la completitud y el grado en el que se validaron los requerimientos se obtuvieron los siguientes resultados:

Métrica para medir la **especificidad** o falta de ambigüedad de los requerimientos

Atributo de calidad	Tipo de requerimiento	Interpretaciones	
		Iguals ( $N_{u1}$ )	Desiguales
Especificidad	Funcionales	43	2
	No Funcionales	13	1
Total		56	3

Tabla 14. Resumen de los resultados obtenidos al aplicar la métrica de calidad Especificación

$$Q_1 = N_{u1} / N_r \quad N_{u1} = 56 \quad N_r = 59$$

$$Q_1 = 56/59$$

$$Q_1 = 0.949$$

La expresión de completitud o falta de ambigüedad es óptima cuando su valor es 1. En este caso se obtuvo  $Q_1$  con un valor de 0.949, éste resultado indica que el grado de ambigüedad en la especificación de los requerimientos de software para gestionar el PLE de los estudiantes de la UCI fue muy bajo. Este valor de  $Q_1$  señala que los requerimientos, en su mayoría, se describieron correctamente y sin ambigüedades.

Los requerimientos ambiguos detectados, fueron sometidos a un nuevo proceso de análisis y posteriormente a su corrección, con el objetivo de lograr una mayor claridad en su interpretación. Sin embargo no se realizó una segunda iteración de la métrica, pues se consideró haber alcanzado un valor de especificidad suficientemente alto.

### Conclusiones del capítulo

En este capítulo se llevó a cabo la última etapa propuesta para el desarrollo de requerimientos, la validación de los requerimientos definidos en las etapas iniciales. Se aplicó la técnica de prototipos de alta fidelidad, empleada para realizar el prototipo de interfaz de usuario del sistema, permitiendo detectar errores y deficiencias existentes. La aplicación de las métricas para la calidad de la especificación de los requerimientos de software demostró que el grado de ambigüedad de los mismos fue muy bajo. Mientras que la aplicación de las métricas para la calidad de la funcionalidad de las HU reflejó que se construyó una especificación con calidad. Por su parte la aplicación del método de Kano también arrojó resultados satisfactorios, asegurándose de esta manera la satisfacción del cliente. A partir de estos resultados se puede plantear que los artefactos obtenidos tienen la calidad necesaria, lo que contribuirá al posterior diseño del software que gestionará el PLE de los estudiantes de la UCI.

## CONCLUSIONES

El trabajo realizado permite llegar a las siguientes conclusiones:

- ✓ Con los elementos teóricos y prácticos más actuales de las ciencias informáticas en lo referido a tecnología educativa, se establecieron las bases para el diseño y la construcción de un sistema que soporte el PLE de los estudiantes de la UCI, a través del desarrollo de una ingeniería de requerimientos, validada con métodos científicos.
- ✓ La ingeniería de requerimientos presentada constituye una base para la ejecución de las posteriores fases del desarrollo de software, dado que se encuentra totalmente desarrollada y documentada.
- ✓ El conjunto de métodos científicos utilizados para la validación de los requerimientos (prototipos de alta fidelidad, método Kano y métricas de calidad) permitió comprobar que:
  - Los requerimientos propuestos responden a las necesidades de los estudiantes de la UCI.
  - Existe una alta satisfacción de los usuarios actuales y potenciales con respecto a la necesidad, utilidad y actualidad del sistema propuesto.

**RECOMENDACIONES**

1. Realizar un seguimiento de los requerimientos de software durante las posteriores fases de desarrollo, aplicando la Administración de Requerimientos que propone la Ingeniería de Requerimientos.
2. Continuar con la identificación de nuevas aplicaciones que emplean los estudiantes y que pueden ser incorporadas al PLE.
3. Realizar un estudio para identificar los servicios web que el PLE necesita consumir para la actualización de los widgets en tiempo real.
4. Incorporar funcionalidades asociadas a la recomendación de contenidos basados en las preferencias del usuario.

**REFERENCIAS BIBLIOGRÁFICAS**

1. 830-1998, E. S. (2008). "Especificación de Requerimientos según el estándar de IEEE 830. ."
2. Adell, J. and L. Castañeda (2010). "Los Entornos Personales de Aprendizaje (PLEs): una nueva manera de entender el aprendizaje."
3. Almenara, J. C. (2015). "El diseño de entornos personales de aprendizaje y la formación de profesores en TIC."
4. Almenara, J. C., V. M. Díaz and A. Infante (2011). "CREACIÓN DE UN ENTORNO PERSONAL PARA EL APRENDIZAJE: DESARROLLO DE UNA EXPERIENCIA." *Revista Electrónica de Tecnología Educativa* **38**.
5. Ash, J. (2013). "Personal Learning Environments' Focus on the Individual." *Education Week*.
6. Attwell, G. (2007). "Personal Learning Environments - the future of eLearning?"
7. Azzolini, C. M. (2011). "Un Enfoque de Priorización de Requerimientos, a partir de la Segmentación de las Preferencias de los Stakeholders."
8. Bagdanov, E. and F. Wild (2010). "Mapping Web Personal Learning Environments."
9. Barron, B. (2006). "Interest and Self-Sustained Learning as Catalysts of Development: A Learning Ecology Perspective." *Human Development* **49**.
10. Committee, S. E. C. (2014). "SWEBOOK, Software Engineering Body of Knowledge."
11. Dabbagh, N. and A. Kitsantas (2012). "Personal Learning Environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning." *Internet and Higher Education* **15**.
12. Dabbagh, N. and A. Kitsantas (2012). "Personal Learning Environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning." *Internet and Higher Education*.
13. Duarte, J. A. L. (2005). "METODOLOGÍA PARA LA DETECCIÓN DE REQUERIMIENTOS SUBJETIVOS EN EL DISEÑO DE PRODUCTO."
14. Engel, A. and C. Coll (2014). "Los entornos personales de aprendizaje en contextos de educación formal." *Revista Cultura y Educación* **24**(6): 617-630.
15. Escalona, M. J. and N. Koch (2016). "Ingeniería de Requerimientos en Aplicaciones para la Web."
16. Figueroa, R. G., C. J. Solís and A. A. Cabrera (2015). "METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES."
17. Frühauf, K. and E. Fuchs (2015). "Programa de estudios. Profesional Certificado en Ingeniería de Requerimientos de IREB."
18. Hadad, G. D. S. (2007). "Uso de Escenarios en la Derivación de Software."
19. Harmelen, F. v. (2008). "A Semantic Web Primer." *The MIT Press*.
20. Haskins, C. (2007). "A Systems Engineering Framework for Eco-Industrial Park Formation." *Systems Engineering* **10**.

21. Kotonya, G. and I. Sommerville (1995). "Requirements Engineering With Viewpoints."
22. Letelier, P. and M. d. C. Penadés (2010). "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP) " Laboratorio de Sistemas de Información. Departamento de Sistemas Informáticos y Computación.
23. Facultad de Informática. Universidad Politécnica de Valencia. .
24. Londoño, C. A. D. L. C. and G. A. C. Guevara (2015). "METODOLOGÍA PARA LA ADQUISICIÓN Y GESTIÓN DE REQUERIMIENTOS EN EL DESARROLLO DE SOFTWARE PARA PEQUEÑAS Y MEDIANAS EMPRESAS (PYMES) DEL DEPARTAMENTO DE RISARALDA."
25. López, P. G., A. A. G. López and J. A. G. Lázaro (2011). "Herramientas CASE. ¿Cómo incorporarlas con éxito en nuestra organización?"
26. Manolis, C. and E. Kalaitzidou (2018). Personal Learning Environments and Social Networks in the Traditional School System: An Applied Case Study in the Greek Educational System.
27. Mejoras, P. d. "Metodología de desarrollo para la Actividad productiva de la UCI." from <http://mejoras.prod.uci.cu/>.
28. Otto, M. and J. Thornton (2016). "Bootstrap 3, el manual oficial."
29. Pérez, L. T. (2011). "METODOLOGÍA DE DESARROLLO DE SOFTWARE PARA EL CITI." Engineering for a Smart Planet, Innovation, Information Technology and Computational Tools for Sustainable Development.
30. Pressman, R. S. (2010). Ingeniería del Software. Un enfoque práctico.
31. Rodríguez, S. U., M. A. Cormenzana, M. Conde and C. O. Pons (2013). "LAS PÁGINAS DE INICIO COMO HERRAMIENTAS DE AYUDA PARA ORGANIZAR EL PLE. UN ANÁLISIS COMPARATIVO." EDUTEC. Revista Electrónica de Teconología Educativa **43**.
32. Rumbaugh, J., I. Jacobson and G. Boch (2000). El lenguaje Unificado de Modelado.
33. Sawyer, P. and G. Kotonya (2001). "SWEBOK."
34. Sommerville, I. (2010). Ingeniería de software.
35. Toro, A. D. and B. B. Jiménez (2001). "Metodología para el Análisis
36. de Requerimientos de Sistemas Software."
37. Toro, A. D. and B. B. Jiménez (2002). "Metodología para la Elicitación de Requerimientos de Sistemas Software".
38. Urdaneta, C. E. R., M. S. Fernández, S. J. Izquierdo and L. R. G. Pérez (2015). "Entornos personales de aprendizaje." Revista Iberoamericana de Producción Académica y Gestión Educativa.
39. Våljataga, T. and M. Laanpere (2010). "Learner control and personal learning environment: a challenge for instructional design." Interactive Learning Environments Volumen **18**.

## ANEXOS

### Anexo 1. Comparación entre herramientas

<i>Características</i>	<i>Symbaloo</i>	<i>iGoogle</i>	<i>Netvibes</i>
No necesita instalación	X	X	X
Es gratuito	X	X	X
Permite el uso compartido	X	X	X
Permite embebido en otras web	X		
Es compatible con los navegadores más usados	X	X	X
Permite integrar buscador de información	X	X	X
Incorpora por defecto widgets para algunos sitios populares	X	X	X
Permite visualizar la información dentro de cada uno de los widgets	X	X	X
Permite personalizar la interfaz	X	X	X
Permite personalizar los widgets	X		X

## REFERENCIAS BIBLIOGRÁFICAS

Permite syndicar <sup>15</sup> contenidos	X	X	X
Permite importar y exportar marcadores		X	X
Permite el uso en diferentes dispositivos	X	X	X
Permite crear diferentes pestañas para organizar los widgets	X		X

Tabla 15. Comparación entre las herramientas (Rodríguez, Cormenzana et al. 2013)

## Anexo 2. Dimensiones para la construcción de un PLE web.

<i>Dimensión</i>	<i>Definición</i>	<i>Estándares potenciales</i>
<i>Agregación</i>	Organización de varios widgets de una manera espacial	W3C Widgets 1.0 Google Gadgtes API Google GadgetTabML Netvibes UWA OpenAjax Matadata 1.0
<i>Comunicación</i>	Interoperabilidad de los datos a través de widgets y servicios. Incluye problemas con cortar y pegar, arrastrar y soltar.	Varios estándares de datos y metadatos, como RSS, SCROM, Dublin Core RDF, HTML5 Dn'D, OpenAjax Hub 2.0, Google gadget pubsub

<sup>15</sup> Es el reenvío o reemisión de contenidos desde una fuente original (sitio web de origen) hasta otro sitio web de destino (receptor) que a su vez se convierte en emisor puesto que pone a disposición de sus usuarios los contenidos a los que en un principio sólo podían tener acceso los usuarios del sitio web de origen.

## REFERENCIAS BIBLIOGRÁFICAS

<i>Configuración</i>	Configuración de la actualización de los widgets, estado de los datos y sincronización con otros usuarios activos cuando comparte la misma instancia del widget	COMET/Reserve Ajax, XMPP, XMPP over BOSH, Google Wave Federation Protocol
<i>Organización</i>	Interoperabilidad de la identidad del usuario, información del perfil y lista de amigos. Capacidad para definir algunos contextos grupales para compartir widgets de eventos, estado o datos.	OpenID(perfiles portátiles) Open Social API Contactos Portables Facebook
<i>Actividad</i>	Las aplicaciones en uso en el PLE se pueden controlar a través de scripts que se dedican a las actividades de aprendizaje del usuario	WS-BPEL(orientado a los negocios) IMS Learning Design(dirigido a los VLE y diseño orientado a tiempo, no en tiempo de ejecución)
<i>Tiempo de ejecución</i>	La interoperabilidad permite el intercambio de una plataforma en ejecución o sus partes con otra.	W3C Widgets 1.0: Empaquetado y configuración OPML, OpenAjax Mashup, Reference Application

Tabla 16. Dimensiones para la construcción de un PLE (Bagdanov and Wild 2010)

### Anexo 3. Resultados arrojados por el software SPSS

**Estadísticos**

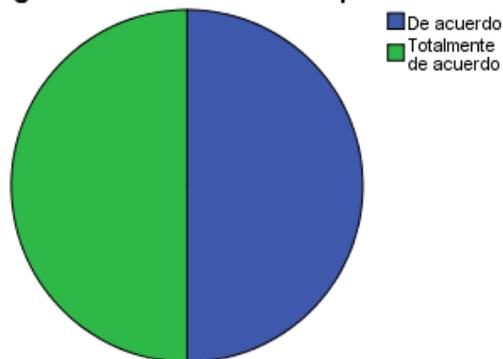
El empleo de un software que gestiona el PLE de los estudiantes de la UCI, contribuye al logro de los objetivos generales del modelo del profesional.

Válidos	14
Perdidos	0

El empleo de un software que gestiona el PLE de los estudiantes de la UCI, contribuye al logro de los objetivos generales del modelo del profesional.

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos De acuerdo	7	50,0	50,0	50,0
Totalmente de acuerdo	7	50,0	50,0	100,0
Total	14	100,0	100,0	

**El empleo de un software que gestiona el PLE de los estudiantes de la UCI, contribuye al logro de los objetivos generales del modelo del profesional.**



**Estadísticos**

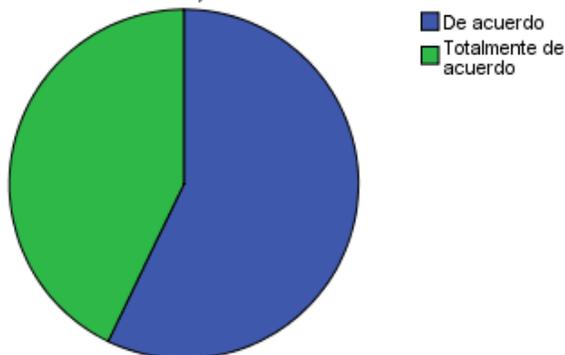
La integración de todas las herramientas que emplean los estudiantes de la UCI para aprender, en un software que gestiona el PLE, es necesaria

N	Válidos	14
	Perdidos	0

La integración de todas las herramientas que emplean los estudiantes de la UCI para aprender, en un software que gestiona el PLE, es necesaria

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	De acuerdo	8	57,1	57,1	57,1
	Totalmente de acuerdo	6	42,9	42,9	100,0
	Total	14	100,0	100,0	

**La integración de todas las herramientas que emplean los estudiantes de la UCI para aprender, en un software que gestiona el PLE, es necesaria**



**Estadísticos**

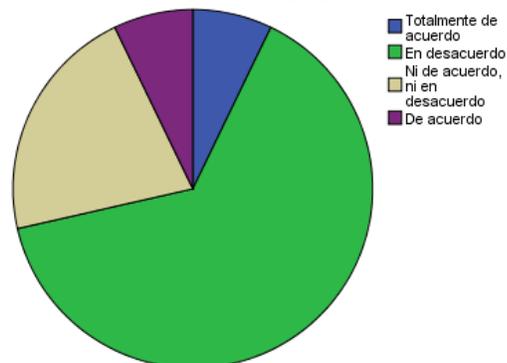
Los estudiantes de la UCI gestionan sus PLE de forma correcta.

N	Válidos	14
	Perdidos	0

Los estudiantes de la UCI gestionan sus PLE de forma correcta.

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Totalmente de acuerdo	1	7,1	7,1	7,1
	En desacuerdo	9	64,3	64,3	71,4
	Ni de acuerdo, ni en desacuerdo	3	21,4	21,4	92,9
	De acuerdo	1	7,1	7,1	100,0
	Total	14	100,0	100,0	

Los estudiantes de la UCI gestionan sus PLE de forma correcta.



**Estadísticos**

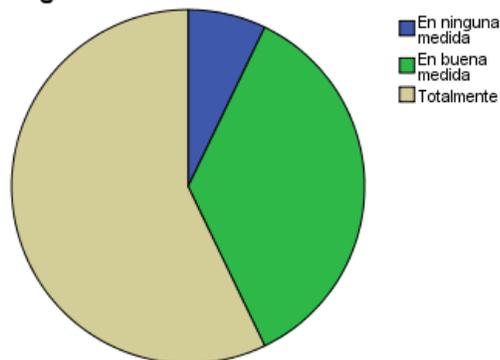
¿En qué medida Ud. considera que las siguientes dificultades pueden afectar la gestión de su PLE a los estudiantes?

N	Válidos	14
	Perdidos	0

¿En qué medida Ud. considera que las siguientes dificultades pueden afectar la gestión de su PLE a los estudiantes?

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	En ninguna medida	1	7,1	7,1	7,1
	En buena medida	5	35,7	35,7	42,9
	Totalmente	8	57,1	57,1	100,0
	Total	14	100,0	100,0	

¿En qué medida Ud. considera que las siguientes dificultades pueden afectar la gestión de su PLE a los estudiantes?

**Anexo 4. Encuesta para el estudio de factibilidad**

## REFERENCIAS BIBLIOGRÁFICAS

Estimado (a), en el mundo existen soluciones de software denominados Entornos Personales de Aprendizaje (PLE, por sus siglas en inglés), cuya finalidad es integrar en un mismo espacio todas aquellas soluciones que emplean los individuos a diario para aprender. En la UCI, las tecnologías y herramientas informáticas que emplean a diario los estudiantes para aprender son variadas, se encuentran dispersas y en muchas ocasiones no se integran. Además, existe desconocimiento por parte de algunos alumnos sobre la existencia de todas las soluciones disponibles que tienen como objetivo favorecer su aprendizaje. En el marco del desarrollo del proyecto de investigación *Entorno de Aprendizaje Ubicuo* se plantea la necesidad de diseñar e implementar un software para gestionar el PLE de los estudiantes de la UCI, de manera tal que se facilite el empleo de los medios y recursos, que la Universidad pone a su alcance para su formación.

Con el objetivo de desarrollar el estudio de factibilidad, relacionado con la implementación de un software para gestionar el PLE de los estudiantes de la UCI, solicitamos su colaboración en el desarrollo del siguiente instrumento.

1. El empleo de un software que gestiona el PLE de los estudiantes de la UCI, contribuye al logro de los objetivos generales del modelo del profesional.

Totalmente de acuerdo	
De acuerdo	
Ni de acuerdo, ni en desacuerdo	
En desacuerdo	
Totalmente en desacuerdo	

2. La integración de todas las herramientas que emplean los estudiantes de la UCI para aprender, en un software que gestiona el PLE, es necesaria.

Totalmente de acuerdo	
De acuerdo	

## REFERENCIAS BIBLIOGRÁFICAS

Ni de acuerdo, ni en desacuerdo	
En desacuerdo	
Totalmente en desacuerdo	

3. Seleccione cuál (es) herramientas Ud. considera que deben estar presentes en el PLE de los estudiantes de la UCI.

- Cliente de correo electrónico
- Plataforma Educativa Zera
- Akademos
- Cliente de mensajería instantánea
- Facebook
- Twitter
- Ecured
- Google
- Otras \_\_\_\_\_

4. Los estudiantes de la UCI gestionan sus PLE de forma correcta.

Totalmente de acuerdo	
De acuerdo	
Ni de acuerdo, ni en desacuerdo	
En desacuerdo	
Totalmente en desacuerdo	

3. ¿En qué medida Ud. considera que las siguientes dificultades pueden afectar la gestión de su PLE a los estudiantes?

- Desconocimiento de tecnologías
- Inversión de tiempo considerable
- Gran cantidad de información y recursos disponibles.
- Amplia variedad de herramientas de comunicación que ofrece la Internet para reelaborar el conocimiento
- No identifican las oportunidades para analizar las tareas desde perspectivas diversas, usando una amplia variedad de recursos.
- Poca discusión y reflexión sobre el término PLE

En ninguna medida		En poca medida		En alguna medida		En buena medida		Totalmente	
-------------------	--	----------------	--	------------------	--	-----------------	--	------------	--

#### Anexo 5. Historias de Usuario

<b>Número:</b> 14	<b>Nombre del requisito:</b> Importar OPML
<b>Programador:</b> Taire Faife Rodriguez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 3 día
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> N/A
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> Permitir al usuario importar OPML</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para importar OPML hay que estar autenticado en el sistema</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b></p> <p>- El usuario debe seleccionar la opción de importar OPML</p> <p>-El usuario debe especificar el directorio donde se encuentra el archivo y para finalmente</p>	

importarlo.

**Observación:** El archivo que está en el directorio debe estar correctamente exportado, si el archivo está dañado o no coincide con la extensión de los archivos OPML no podrá ser exportado

### Prototipo

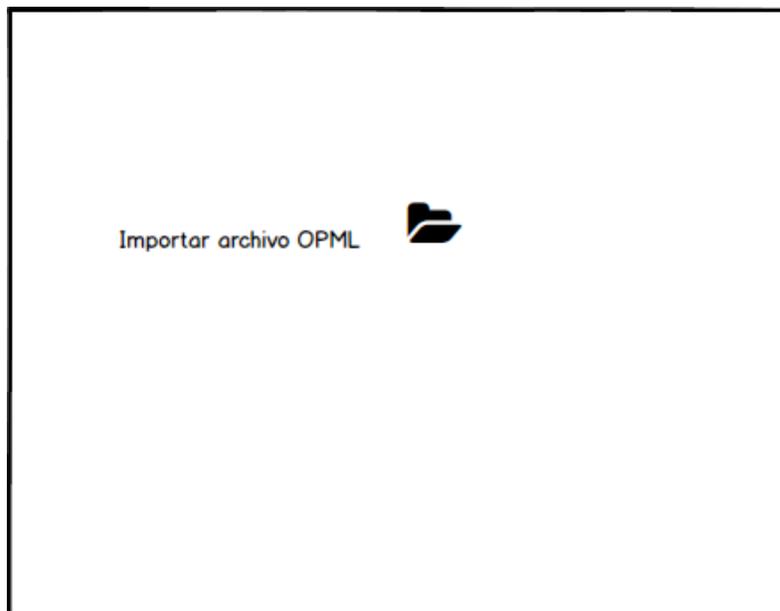
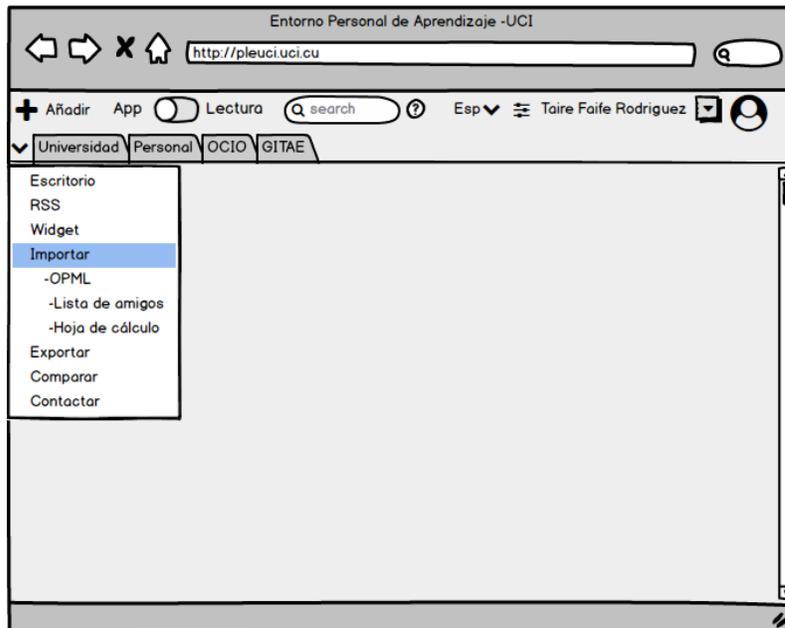


Tabla 17.HU Importar OPML

Número: 7

Nombre del requisito: Previsualizar widget

<b>Programador:</b> Taire Faife Rodriguez	<b>Iteración Asignada:</b> 1era
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> N/A
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b></p> <p>Permitir un widget a través de las redes sociales.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b></p> <p>Para compartir un widget hay que:</p> <ul style="list-style-type: none"> <li>- Estar autenticado en el sistema.</li> <li>- El widget debe haber sido añadido a uno de los escritorios disponibles</li> </ul> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b></p> <p>Seleccionar del widget que desea compartir la opción que aparece en la parte superior derecha y seleccionar la red social a través de la cual desea que sea compartido.</p>	
<b>Observaciones:</b>	
<p><b>Prototipo de interfaz:</b></p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="224 1297 711 1715"> </div> <div data-bbox="760 1297 1271 1715"> </div> </div>	

Tabla 18. HU Previsualizar widget

Número: 10

Nombre del requisito: Editar Perfil

<b>Programador:</b> Taire Faife Rodriguez	<b>Iteración Asignada:</b> 1ra
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 2 días
<b>Riesgo en Desarrollo:</b> N/A	<b>Tiempo Real:</b> N/A
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b></p> <p>Permitir al usuario editar su perfil en el sistema.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b></p> <p>Para editar el perfil hay que:</p> <ul style="list-style-type: none"> <li>- Estar autenticado en el sistema</li> </ul> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p>Los campos nombre, apellidos, usuario, sexo, facultad y año académico son obligatorios.</p> <ul style="list-style-type: none"> <li>• Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 20 caracteres.</li> <li>• Apellidos: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 20 caracteres.</li> <li>• Usuario: Nombre: campo de texto que admite caracteres alfabéticos y numéricos y tiene un máximo de hasta 20 caracteres.</li> <li>• Twitter: Campo de texto que admite caracteres alfabéticos y alfanuméricos tiene un máximo de hasta 20 caracteres.</li> <li>• Facebook: Campo de texto que admite caracteres alfabéticos y alfanuméricos tiene un máximo de hasta 20 caracteres.</li> </ul> <p><b>4- Flujo de la acción a realizar:</b></p> <ul style="list-style-type: none"> <li>- El usuario debe rellenar los datos y presionar el botón editar para actualizar la información</li> <li>- Cuando el usuario deja en blanco alguno de los campos obligatorios el sistema muestra un mensaje indicándole que debe rellenar el campo.</li> </ul>	

**Prototipo**

*Editar perfil*

Nombre

Apellidos

Usuario

Twitter

Facebook

Preferencias  Diseño  Programación Web

Sexo  femenino  Masculino

Año académico

Facultad

Tabla 19. HU Editar perfil

### Anexo 6. Cuestionario para medir la funcionalidad de los requerimientos del sistema

Estimado (a), con el objetivo de desarrollar la fase de validación, relacionado con la implementación de un software para gestionar el PLE de los estudiantes de la UCI, solicitamos su colaboración en el desarrollo del siguiente instrumento.

Requerimiento "Añadir widget"				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				

## REFERENCIAS BIBLIOGRÁFICAS

__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
Requerimiento " <i>Listar RSS</i> "				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
Requerimiento " <i>Editar perfil</i> "				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
Requerimiento " <i>Compartir widget</i> "				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				

## REFERENCIAS BIBLIOGRÁFICAS

__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
Requerimiento " <i>Eliminar widget</i> "				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
Requerimiento " <i>Personalizar interfaz</i> "				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada

Requerimiento “ <i>Buscar</i> ”				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
Requerimiento “ <i>Activar escritorio público</i> ”				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
Requerimiento “ <i>Importar OPML</i> ”				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada

¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
Requerimiento “ <i>Exportar OPML</i> ”				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
Requerimiento “ <i>Añadir recordatorio</i> ”				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada

Requerimiento “ <i>Mover recordatorio</i> ”				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
Requerimiento “ <i>Importar lista de amigos</i> ”				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada
Requerimiento “ <i>Exportar lista de amigos</i> ”				
¿Cómo se siente si esta funcionalidad está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada

¿Cómo se siente si esta funcionalidad no está presente en el sistema que gestionará el PLE de los estudiantes de la UCI?				
__Me agrada	__Es de esperarse	__Neutral	__Lo acepto	__Me desagrada

**Anexo 7. Cuestionario para validar el prototipo de interfaz de usuario**

Requerimientos	Correcto	Incorrecto	Observaciones
RF 1. Añadir widget			
RF 2. Editar widget			
RF 3. Eliminar widget			
RF 4. Minimizar widget			
RF 5. Maximizar widget			
RF 6. Previsualizar widget			
RF 7. Compartir widget			
RF 8. Autenticar usuario			
RF 9. Cerrar sesión de usuario			
RF 10. Editar perfil			
RF 11. Recuperar contraseña			
RF 12. Personalizar interfaz			
RF 13. Buscar			
RF 14. Cambiar idioma			

## REFERENCIAS BIBLIOGRÁFICAS

RF 15. Importar OPML <sup>16</sup>			
RF 16. Exportar OPML			
RF 17. Añadir recordatorio			
RF 18. Eliminar recordatorio			
RF 19. Mover recordatorio			
RF 20. Editar recordatorio			
RF 21. Añadir escritorio			
RF 22. Compartir escritorio			
RF 23. Cambiar de vista a modo lectura			
RF 24. Cambiar de vista a modo widget			
RF 25. Listar RSS <sup>17</sup>			
RF 26. Refrescar listado de RSS			
RF 27. Marcar artículo como leído			
RF 28. Guardar artículo			
RF 29. Compartir artículo			

<sup>16</sup> Outline Processor Markup Language) es un formato XML para esquemas (outlines). Se usa para listar varias fuentes RSS juntas; también se puede usar en programas gestores personales de información.

<sup>17</sup> Really Simple Syndication, un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS (agregador).

## REFERENCIAS BIBLIOGRÁFICAS

RF 30. Añadir etiqueta			
RF 31. Eliminar etiqueta			
RF 32. Listar etiqueta			
RF 33. Listar elementos leídos			
RF 34. Listar elementos guardados			
RF 35. Cambiar vista del listar a modo expandida			
RF 36. Cambiar vista del listar a modo mosaico			
RF 37. Activar escritorio público			
RF 38. Listar escritorios			
RF 39. Importar hojas de cálculo			
RF 40. Comparar hojas de cálculo			
RF 41. Interoperabilidad con sistema recomendador de contenidos			
RF 42. Ecosistema de widget			
RF 43. Comunicación entre widgets			
RF 44. Importar lista de amigos			
RF 45. Exportar lista de amigos			



## REFERENCIAS BIBLIOGRÁFICAS

RF 9. Cerrar sesión de usuario	1	2	3	4	5	6	7	8	9
RF 10. Editar perfil	1	2	3	4	5	6	7	8	9
RF 11. Recuperar contraseña	1	2	3	4	5	6	7	8	9
RF 12. Personalizar interfaz	1	2	3	4	5	6	7	8	9
RF 13. Buscar	1	2	3	4	5	6	7	8	9
RF 14. Cambiar idioma	1	2	3	4	5	6	7	8	9
RF 15. Importar OPML <sup>18</sup>	1	2	3	4	5	6	7	8	9
RF 16. Exportar OPML	1	2	3	4	5	6	7	8	9
RF 17. Añadir recordatorio	1	2	3	4	5	6	7	8	9
RF 18. Eliminar recordatorio	1	2	3	4	5	6	7	8	9
RF 19. Mover recordatorio	1	2	3	4	5	6	7	8	9
RF 20. Editar recordatorio	1	2	3	4	5	6	7	8	9

---

<sup>18</sup> Outline Processor Markup Language) es un formato XML para esquemas (outlines). Se usa para listar varias fuentes RSS juntas; también se puede usar en programas gestores personales de información.

## REFERENCIAS BIBLIOGRÁFICAS

RF 21. Añadir escritorio	1	2	3	4	5	6	7	8	9
RF 22. Compartir escritorio	1	2	3	4	5	6	7	8	9
RF 23. Cambiar de vista a modo lectura	1	2	3	4	5	6	7	8	9
RF 24. Cambiar de vista a modo widget	1	2	3	4	5	6	7	8	9
RF 25. Listar RSS <sup>19</sup>	1	2	3	4	5	6	7	8	9
RF 26. Refrescar listado de RSS	1	2	3	4	5	6	7	8	9
RF 27. Marcar artículo como leído	1	2	3	4	5	6	7	8	9
RF 28. Guardar artículo	1	2	3	4	5	6	7	8	9
RF 29. Compartir artículo	1	2	3	4	5	6	7	8	9
RF 30. Añadir etiqueta	1	2	3	4	5	6	7	8	9

---

<sup>19</sup> Really Simple Syndication, un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS (agregador).

## REFERENCIAS BIBLIOGRÁFICAS

RF 31. Eliminar etiqueta	1	2	3	4	5	6	7	8	9
RF 32. Listar etiqueta	1	2	3	4	5	6	7	8	9
RF 33. Listar elementos leídos	1	2	3	4	5	6	7	8	9
RF 34. Listar elementos guardados	1	2	3	4	5	6	7	8	9
RF 35. Cambiar vista del listar a modo expandida	1	2	3	4	5	6	7	8	9
RF 36. Cambiar vista del listar a modo mosaico	1	2	3	4	5	6	7	8	9
RF 37. Activar escritorio público	1	2	3	4	5	6	7	8	9
RF 38. Listar escritorios	1	2	3	4	5	6	7	8	9
RF 39. Importar hojas de cálculo	1	2	3	4	5	6	7	8	9
RF 40. Comparar hojas de cálculo	1	2	3	4	5	6	7	8	9

## REFERENCIAS BIBLIOGRÁFICAS

RF 41. Interoperabilidad con sistema recomendador de contenidos	1	2	3	4	5	6	7	8	9
RF 42. Ecosistema de widget	1	2	3	4	5	6	7	8	9
RF 43. Comunicación entre widgets	1	2	3	4	5	6	7	8	9
RF 44. Importar lista de amigos	1	2	3	4	5	6	7	8	9
RF 45. Exportar lista de amigos	1	2	3	4	5	6	7	8	9

**Anexo 9. Cuestionario para medir la calidad de la especificación**

No	Preguntas	Sí	No
1.	¿Los requerimientos están escritos en un lenguaje no técnico y comprensible para el usuario/cliente?		
2.	¿Hay algún requerimiento que pueda tener más de una interpretación?		
3.	¿Existen requerimientos contradictorios?		
4.	¿Puede ser eliminado algún requerimiento?		
5.	¿Existen requerimientos redundantes?		
6.	¿Hay algún requerimiento no satisfactorio?		

7.	¿Para cada requerimiento hay un proceso que puede ser ejecutado para verificarlo?		
8.	¿Hay alguna redundancia en los requerimientos?		
9.	¿Cada requerimiento es relevante al problema y a su solución?		
10.	¿Todas las funciones del software están especificadas?		
11.	¿Todos los requerimientos de seguridad para cada función están definidos?		