

## SGR: Sistema para la Gestión de Requerimientos

Rípodas Alejandra, Aguirre Verónica, Delía Lisandro, Marrero Luciano  
{aripodas, vaguirre, ldelia, lmarrero}@lidi.info.unlp.edu.ar

Instituto de Investigación en Informática LIDI  
Facultad de Informática UNLP

**Abstract.** La ingeniería de requerimientos es una etapa clave en el ciclo de vida del software. Actualmente existen en el mercado diversas herramientas que dan soporte a dicha etapa. Al analizar y comparar un conjunto de herramientas, se identificaron las características que consideramos debe tener un producto de software para la gestión de requerimientos. Del análisis comparativo se concluye que no existe una herramienta, que cumpla con la totalidad de las características consideradas. Es por esto que se decidió desarrollar un producto de software SGR (Sistema para la Gestión de Requerimientos) que brinde soporte a la fase de ingeniería de requerimientos y que reúna todas las características consideradas. A través de la creación de SGR se pretende aportar una herramienta colaborativa que fomente la participación del cliente para lograr obtener un producto final exitoso y que genere el SRS basado en el estándar IEEE 830-1998.

**Keywords:** Requerimiento, Ingeniería de Requerimientos, Herramienta CASE

### 1. Introducción

Los requerimientos reflejan las necesidades de los clientes. Son un elemento clave en el desarrollo de un proyecto de software, ya que marcan el punto de partida para actividades como planeación, estimación de esfuerzo, definición de recursos, costos, etc. Inician cuando comienza un proyecto y pueden variar a lo largo de este.

La extracción y el análisis de los requerimientos sirve para dos propósitos diferentes, pero relacionados. Por una parte, la extracción nos permite escribir un documento de definición de requerimientos; escrita en términos que el cliente puede entender. Representa una comprensión entre el cliente y el desarrollador de lo que se necesita o desea, y por lo general es escrito en forma conjunta entre el cliente y el desarrollador. Por otra parte, la especificación de requerimientos reitera la definición en los términos técnicos apropiados para el desarrollo del diseño de un sistema [1].

La especificación de requerimientos deber ser clara, completa y estar libre de ambigüedades. Es por esto que, se considera que la mejor forma de realizarla es mediante el uso del método estructurado. Este método propone la definición y el uso de plantillas para asegurar que los requerimientos estén expresados de forma uniforme. De esta manera se minimizan los errores relacionados con la interpretación durante las distintas etapas de un proyecto. Por otro lado, la especificación

estructurada de requerimientos guía a analistas inexpertos en el desarrollo de la especificación de requerimientos.

En [1] a través de una investigación empírica sobre errores de software, se observa que el 48% de los defectos encontrados en proyectos de software de mediana escala eran “atribuidos a especificaciones funcionales o requerimientos incorrectos o mal interpretados”. Asimismo, en [1] se afirma que “encontrar y corregir los defectos en la fase de desarrollo del software es más costoso que hacerlo en la fase de requerimientos”.

Con el paso de los años el uso de herramientas para dar soporte a la gestión de requerimientos se convirtió en un factor importante en el desarrollo del software. Estas herramientas suelen ser herramientas CASE (Computer-Aided Software Engineering), y tienen como objetivo hacer que el desarrollo de un proyecto sea un proceso eficientemente organizado. Existe una gran cantidad de estas herramientas, las cuales permiten controlar proyectos complejos y brindan funcionalidades para especificar, organizar, almacenar y gestionar requerimientos. De esta manera se intenta reducir errores que a largo plazo puedan perjudicar el correcto desarrollo de un proyecto [2].

Con el objetivo de detectar un conjunto mínimo y deseable de características a cumplir por una herramienta CASE, se analizaron y compararon una serie de herramientas para la gestión de requerimientos. Del análisis comparativo se concluye que no existe una herramienta entre las analizadas, que cumpla con la totalidad de los aspectos esperados.

Es por esto que se decidió desarrollar un producto de software que reúna las características evaluadas brindando soporte a una de las actividades más importantes en el desarrollo de un proyecto, como es la fase de Ingeniería de Requerimientos.

En la sección 2 se muestra un análisis de un conjunto de herramientas de gestión de requerimientos y se establecen características que consideramos debe contener una herramienta CASE para la gestión de requerimientos. En la sección 3 se presentan SGR (Sistema para la Gestión de Requerimientos) y sus funcionalidades más importantes. Por último se presentan conclusiones y trabajos futuros.

## **2. Análisis de herramientas para la gestión de requerimientos**

Los autores de [3] determinaron las características que debería tener una herramienta para la gestión de requerimientos en base al documento ISO/IEC TR 24766: 2009. Este establece un total de 157 capacidades, las cuales fueron agrupadas en 6 categorías principales: obtención de requerimientos, análisis de requerimientos, especificación de requerimientos, verificación y validación de requerimientos, administración de requerimientos y otras capacidades.

Tomando como referencia las categorías definidas anteriormente se evaluaron tres herramientas del ámbito académico (REM [4], Heler [5] y Let's Req! [6]) y siete del ámbito comercial (Visure Requirements [7], DOORSNG [8], Rommana [9], TraceCloud [10], Psoda [11], TrackStudio [12] Y Redmine [13]). Como resultado del análisis realizado se identificaron las características mínimas y deseables que debería

tener una herramienta CASE para la gestión de requerimientos. Estas características son:

- **Diferencia entre requerimientos:** brindar la posibilidad de definir requerimientos funcionales y no funcionales de forma separada.
- **Objetivos de proyecto:** crear objetivos del proyecto y asignarlos a los requerimientos para establecer sus fines.
- **Roles y permisos:** permitir el acceso al sistema mediante autenticación, restringiendo las funcionalidades que se pueden realizar.
- **Versiones de requerimientos:** asignar números de versiones a los requerimientos para indicar los cambios que transitó.
- **Estado:** asignar el estado en que se encuentra tanto un requerimiento como un proyecto.
- **Stakeholder:** definir personas que influyan directa o indirectamente en la gestión de requerimientos.
- **Prioridad:** asignar prioridad para indicar qué requerimientos son más importantes.
- **Estabilidad:** definir y asignar estabilidad a requerimientos funcionales y no funcionales para indicar el nivel de modificación.
- **Historial de cambios de requerimientos:** llevar un registro de los cambios de cada requerimiento para realizar su seguimiento.
- **Documentación externa:** permitir el almacenamiento de documentación externa en diferentes formatos (pdf, doc, odt, jpg, etc.), de esta manera se complementa la información tanto de proyectos como de requerimientos funcionales y no funcionales.
- **Generación automática del IEEE Std. 830-1998:** generar automáticamente documentos basados en el estándar de la IEEE 830-1998 en formato PDF para contar con los datos sin necesidad de acceder al sistema.
- **Trazabilidad:** permitir identificar el origen y las relaciones entre los requerimientos con el fin de mostrar la matriz de trazabilidad correspondiente.
- **Trabajo colaborativo:** permitir que los diferentes usuarios de un proyecto puedan trabajar de forma colaborativa para agilizar la gestión de requerimientos.
- **Comunicación interna:** brindar un canal de mensajería interna para realizar la comunicación a través de la herramienta.
- **Notificaciones por email:** realizar el envío automático de notificaciones por email para mantener actualizados a los miembros del proyecto.
- **Claridad y facilidad de uso:** brindar una interfaz intuitiva para agilizar su utilización.
- **Interfaz web adaptable:** brindar una interfaz web adaptable para satisfacer la necesidad de utilizar la herramienta desde diferentes dispositivos como Tablets, Smartphones, etc.
- **Multiplataforma:** permitir utilizar la herramienta desde distintas plataformas independientemente del sistema operativo del usuario.

Luego, se realizó un cuadro comparativo (Ver Tabla 1) en el que se puede visualizar el comportamiento de cada herramienta con las características

anteriormente mencionadas. En Tabla 1 se puede observar que ninguna de las herramientas estudiadas satisface la totalidad éstas características.

A partir de esta comparación, se decidió realizar un sistema de software denominado SGR que reúna todas las características que consideramos necesarias y deseables que debe tener un sistema para la etapa de gestión de requerimientos.

**Tabla 1.** Cuadro comparativo de herramientas seleccionadas.

| Herramientas                                 | H1 | H2 | H3 | H4 | H5 | H6 | H7 | H8 | H9 | H10 |
|--|----|----|----|----|----|----|----|----|----|-----|
| <b>Características</b>                       |    |    |    |    |    |    |    |    |    |     |
| Diferencia entre requerimientos              | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓   |
| Objetivos de proyecto                        | ✓  | ✓  | ✓  | ✓  | ✓  | -  | -  | ✓  | -  | -   |
| Roles y permisos                             | -  | ✓  | -  | ✓  | -  | ✓  | ✓  | ✓  | ✓  | ✓   |
| Versiones de requerimientos                  | ✓  | ✓  | ✓  | -  | ✓  | ✓  | -  | -  | ✓  | -   |
| Estado                                       | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓   |
| Stakeholders                                 | ✓  | -  | ✓  | ✓  | ✓  | -  | -  | -  | -  | -   |
| Prioridad                                    | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓   |
| Estabilidad                                  | ✓  | -  | ✓  | -  | -  | ✓  | ✓  | ✓  | -  | -   |
| Historial de cambios de requerimientos       | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓   |
| Documentación externa                        | -  | ✓  | -  | -  | -  | ✓  | ✓  | ✓  | ✓  | ✓   |
| Generación automática del IEEE Std. 830-1998 | -  | -  | ✓  | -  | -  | -  | -  | -  | -  | -   |
| Trazabilidad                                 | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | -  | ✓   |
| Trabajo colaborativo                         | -  | ✓  | -  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓   |
| Comunicación interna                         | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓   |
| Notificaciones por email                     | -  | ✓  | -  | ✓  | -  | ✓  | ✓  | ✓  | ✓  | ✓   |
| Claridad y facilidad de uso                  | ✓  | ✓  | ✓  | -  | ✓  | -  | -  | -  | -  | ✓   |
| Interfaz web adaptable                       | -  | -  | -  | -  | -  | -  | -  | -  | -  | ✓   |
| Multiplataforma                              | -  | -  | -  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓  | ✓   |

H1: REM, H2: Visure Requirements, H3: Heler, H4: DOORSNG, H5: Let's Req!, H6: Rommana, H7: TraceCloud, H8: Psoda, H9: TrackStudio, H10: Redmine

✓= Cumple con la funcionalidad, - = No cumple

### 3. SGR (Sistema para la Gestión de Requerimientos)

En esta sección se presenta SGR, una herramienta CASE para la etapa de elicitación de requerimientos. Además se expone un caso de estudio donde se la utilizó y por último, se muestra una comparación entre SGR y las herramientas analizadas en la sección 2.

### 3.1 Funcionalidades destacadas

#### 3.1.1 Administración de Proyectos

SGR permite la creación de proyectos compuestos por: requerimientos funcionales, no funcionales, objetivos, usuarios asignados, archivos adjuntos, atributos, stakeholders y versiones. Además, permite mantener historial de cambios e indicar el estado del proyecto en un momento determinado (Figura 1).

Es posible realizar filtros y establecer órdenes para mejorar la visualización de los datos.

The screenshot shows the SGR (Sistema de Gestión de Requerimientos) interface for a project named 'Albergue Universitario UNLP'. The interface includes a navigation bar with 'SGR', 'Tablero', 'Configuración', 'Datos generales', and 'Proyectos'. The project status is 'Activo'. A progress bar indicates that 21% of requirements are in the 'Terminado' state. Below this, there are tabs for 'Datos Generales', 'Requerimientos Funcionales', 'Requerimientos No Funcionales', 'Objetivos', 'Usuarios Asignados', 'Archivos Adjuntos', 'Atributos', and 'Stakeholders'. A table displays project details:

| Fecha de inicio estimada | Fecha de fin estimada | Fecha de inicio real | Fecha de fin real | Versión                         |
|--------------------------|-----------------------|----------------------|-------------------|---------------------------------|
| 16/04/2018               | 16/10/2018            |                      |                   | 1.0 12/06/2018 [desarrollo] + - |

Below the table, there are sections for 'Alcance', 'Propósito', 'Resumen', and 'Descripción'. A description box contains the text: 'Actualmente, el albergue no utiliza ningún sistema de software para realizar la gestión de comidas del comedor. El sistema que se desarrollará permitirá gestionar los alumnos que se presentan diariamente en el comedor del albergue, y permite al llevar un control cuantitativo de las inscripciones y retiros de comida que se realizan.' At the bottom right, there are buttons for 'Eliminar Proyecto' and 'Volver'. The footer of the page reads 'Facultad de Informática - UNLP'.

Figura 1. Interfaz de un proyecto.

#### 3.1.2 Administración de Requerimientos

Los requerimientos deben ser creados con el mayor detalle posible, con el fin de generar documentos completos y consistentes. Éstos están compuestos por estado, prioridad y estabilidad. Además se permite asignarles stakeholders, los cuales no pueden interactuar directamente con el sistema, pero son asociados como fuente de información de los requerimientos. También permite realizar el seguimiento de los requerimientos a través de su historial de cambios, administrar versiones, asignar objetivos y adjuntar archivos. Estas características enriquecen su especificación.

### 3.1.3 Trabajo Colaborativo

Es importante mantener informados a los usuarios acerca de los cambios en la información para favorecer la comunicación y obtener mejores resultados en el desarrollo de un proyecto de software.

Para ello se realizan dos tipos de notificaciones: una en la pantalla inicial del sistema denominada *Tablero de usuario* y otra a través del envío de emails.

En el *Tablero del usuario* se muestra: un listado con los proyectos próximos a concluir, los proyectos retrasados y un listado con los proyectos que fueron actualizados recientemente.

Las notificaciones por email se generan por dos motivos: cambio de estado de un requerimiento por parte de un cliente o por actualización en un proyecto.

Además, SGR permite que los usuarios asignados a un proyecto comenten tanto el proyecto como los requerimientos pertenecientes a este, logrando de esta manera, una comunicación fluida entre ellos.

### 3.1.4 Generación automática del IEEE Std. 830-1998

En algunas ocasiones es conveniente que los usuarios puedan disponer de los datos sin necesidad de recurrir al sistema. Es por esto que SGR provee la funcionalidad de generar y exportar automáticamente documentos de especificación de requerimientos SRS basado en el estándar IEEE Std. 830-1998. Según IEEE, un buen Documento de Requerimientos, pese a no ser obligatorio que siga estrictamente la organización y el formato dados en el estándar 830, deberá incluir, de una forma o de otra, toda la información presentada en dicho estándar. Este tiene como objetivo ayudar a los clientes a describir claramente lo que se desea obtener mediante un determinado software, ayudar a los desarrolladores a entender qué necesita exactamente el cliente y servir de base para desarrollos de estándares de SRS particulares para cada organización [14].

### 3.1.5 Roles y Usuarios

La gestión de requerimientos involucra, generalmente, a varias personas que se relacionan con el proyecto de diferentes formas. Debido a esto, el sistema diferencia entre diversos roles de usuarios.

Para brindar privacidad de los datos almacenados, los usuarios que quieran operar con SGR deben autenticarse. Cada usuario autenticado puede realizar diferentes tareas dependiendo de los permisos que tenga asignado dentro del proyecto.

## 3.2 Caso de estudio

Con el objetivo de evaluar el comportamiento y desempeño de SGR, actualmente se lo está utilizando en un proyecto llevado a cabo en el Instituto de Investigación en Informática - LIDI (III-LIDI). El proyecto surge como respuesta a la solicitud del Albergue Universitario de la UNLP. Esta institución se encarga de dar hospedaje y

alimentación (almuerzo y cena) de forma gratuita a más de 130 jóvenes estudiantes de carreras de grado de la Universidad. La problemática presentada se basó puntualmente en la necesidad de llevar a cabo la gestión de comidas brindadas por el comedor del albergue. Actualmente no utilizan ningún sistema informático para esto, por lo que se propuso desarrollar un sistema de software para la gestión del comedor.

Para llevar a cabo la etapa de elicitación de requerimientos se realizaron entrevistas y con la información obtenida, se procedió a la carga de los datos del proyecto en el sistema.

A partir de este caso de estudio, se concluyó que a través de SGR es posible llevar adelante la etapa de elicitación de requerimientos de una manera ordenada y guiada, como también exportar el documento SRS basado en IEEE Std. 830-1998, mantener la información centralizada y realizar el trabajo de forma colaborativa. Además fue posible fomentar la participación de los clientes permitiendo negociar los requerimientos, logrando de esta manera alinear las necesidades del cliente con el software resultante. Cabe destacar que las funcionalidades de exportar el documento SRS basado en IEEE Std. 830-1998 y el trabajo colaborativo no se encuentran satisfechas de forma simultánea en las herramientas analizadas.

### 3.3 Tecnología utilizada

Para el desarrollo de SGR se utilizó el framework Symfony [15], el cual se basa en PHP [16], diseñado para optimizar el desarrollo de aplicaciones web. Al ser un sistema web permite que el sistema sea multiplataforma. La decisión de utilizar el framework Symfony se debió a que cuenta con una amplia y completa documentación y además cuenta con una gran comunidad de colaboradores que lo apoyan.

Como sistema de gestión de base de datos se utilizó MYSQL [17] ya que se considera una de las bases de datos de uso libre más utilizada en entornos web. Para realizar la interacción con la base de datos se utilizó Doctrine [18], una librería que permite realizar consultas para modelos orientados a objetos.

Con respecto a la interfaz, se utilizó Twig [19] y Bootstrap [20]. Twig es un motor de creación de plantillas que facilita la implementación de las vistas haciendo el trabajo más simple y ordenado. Bootstrap es un conjunto de herramientas de uso libre pensado para sitios web adaptables.

La figura 2 muestra la arquitectura utilizada para SGR.

Una funcionalidad importante de SGR es el envío automático de notificaciones a los integrantes de un proyecto para mantenerlos actualizados sobre nuevos cambios, para esto se utilizó Swiftmailer [21] debido a que brinda una solución simple y flexible.

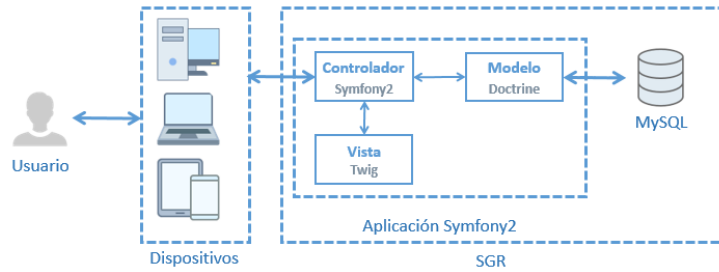


Figura 2. Modelo cliente-servidor de SGR.

### 3.4 Interfaz de usuario

SGR cuenta con una interfaz de usuario amigable, ya que, tratándose de un sistema para la gestión de requerimientos, se espera que el usuario pueda utilizarla de manera intuitiva y sin grandes dificultades. Al ser una herramienta web se amplía su utilización para cualquier usuario que posea un dispositivo con acceso a un navegador web y conexión a internet. El sistema está desarrollado utilizando diseño web adaptable, para cumplir con los objetivos de una interfaz amigable y de fácil utilización en cualquier dispositivo electrónico como Smartphones, Tablets, Notebook, etc.

Las figuras 3 y 4 muestran la interfaz móvil del detalle de un proyecto y un requerimiento funcional respectivamente.

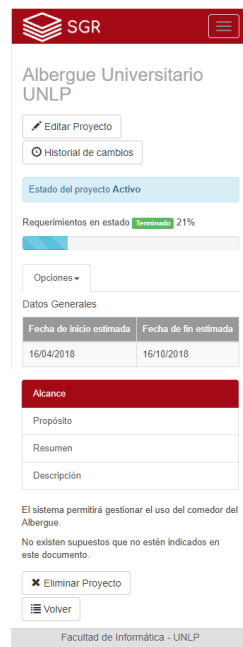


Figura 3. Interfaz de Proyecto.

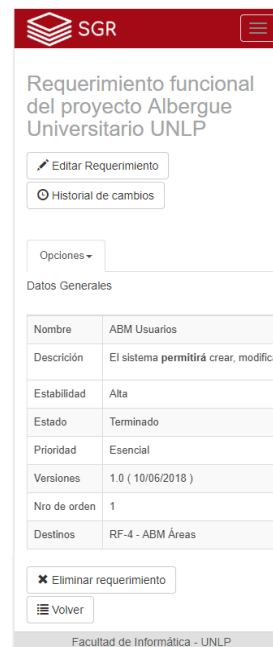


Figura 4. Interfaz de Requerimiento.



### 3.5 Objetivos alcanzados

SGR fue incluido en la comparación de las herramientas evaluadas en la sección 2 (ver Tabla 1). En consecuencia, se desprende la Tabla 2 la cual muestra la cantidad de características mencionadas en este artículo, que satisface cada herramienta. Se puede observar que SGR cumple con la totalidad de éstas ya que se puso énfasis en las debilidades de las otras herramientas como lo son la generación de documentos basados en el estándar IEEE 830-1998 y el diseño web adaptable, entre otros.

**Tabla 2.** Resultados obtenidos.

| Herramientas | Características Satisfechas/Total |
|--------------|-----------------------------------|
| REM          | 11-18                             |
| VISURE       | 13-18                             |
| HELER        | 12-18                             |
| DOORSNG      | 12-18                             |
| LETS REQ!    | 12-18                             |
| ROMMANA      | 13-18                             |
| TRACECLOUD   | 12-18                             |
| PSODA        | 13-18                             |
| TRACKSTUDIO  | 11-18                             |
| REDMINE      | 13-18                             |
| SGR          | 18-18                             |

## 4. Conclusiones

Siendo la elicitación de requerimientos una etapa clave en el ciclo de vida del software, encontrar y corregir los defectos en la fase de desarrollo es más costoso que hacerlo en la fase de requerimientos. Por lo tanto, la especificación de requerimientos debe ser clara, completa y estar libre de ambigüedades.

Con el avance de la tecnología informática, la utilización de las herramientas CASE destinadas a la Ingeniería de Software ha tomado gran importancia. Uno de los principales objetivos de estas herramientas es facilitar y mejorar las actividades involucradas en la gestión de requerimientos.

Se realizó un análisis comparativo de un conjunto de herramientas del ámbito académico y comercial. A partir de este análisis se determinó que no existe una herramienta que cumpla con la totalidad de las características mínimas y deseables que debería contener una herramienta CASE para la etapa de gestión de requerimientos. Es por esto que se decidió desarrollar SGR, un sistema que permita: gestionar los requerimientos de manera detallada, trabajar colaborativamente, facilitar la comunicación entre los usuarios del sistema, almacenar toda la información de manera centralizada, brindar seguridad y privacidad de los datos a partir de definición de roles y permisos, permitir realizar trazabilidad de los requerimientos, permitir su utilización desde cualquier plataforma, brindar interfaces amigables y web adaptables.

Se presentaron las funcionalidades más importantes de SGR y a través del proyecto desarrollado para el Albergue Universitario UNLP se pudo observar que a partir de su utilización se llevó adelante la etapa de elicitación de requerimientos de manera ordenada y guiada; se generó el documentos SRS basado en IEEE Std 890-1998 que

están siendo utilizados actualmente por los desarrolladores para la implementación del sistema con resultados y feedback positivos.

## 5. Trabajos futuros

Se propone agregar nuevas funcionalidades a SGR: diccionario de frases del dominio, notificaciones a través del navegador, calendario con fechas importantes para los proyectos y generación de diversos reportes con el fin de mejorar la experiencia de los usuarios.

Además ampliar las pruebas de campo en asignaturas relacionadas con la Ingeniería de Software de la Facultad de Informática (UNLP) para realizar mejoras o agregar funcionalidades a partir de la retroalimentación de éstas.

## 6. Bibliografía

1. Pfleeger, Shari Lawrence. Ingeniería de software: teoría y práctica (2002).
2. Bárbara A. McDonald Landazuri. Definición de Perfiles en Herramientas de Gestión de Requisitos (2005).
3. IEEE Software Editado por Christof Ebert. Requirements Engineering Tools. Páginas 86-91. Volumen 28 Julio/Agosto (2011).
4. Durán Amador. Un Entorno Metodológico de ingeniería de requisitos para sistemas de información. Tesis doctoral. Universidad de Sevilla (2010).
5. Mauro Callegas Cuervo, Luz Yadira Castillo Estupoñán, Ruby Mónica Fernández Álvarez. HELER: una herramienta para la ingeniería de requisitos automatizada. Págs. 184-200. Colombia: Entramado (2010).
6. Pablo Torrecilla González, María N. Moreno García. Proyecto TCUE Universidades de Castilla y León. (2011).
7. Visure Requirements [www.visuresolutions.es](http://www.visuresolutions.es) (4-6-2018)
8. Doorsng [www.doorsng.com](http://www.doorsng.com) (4-6-2018)
9. Rommana <https://rommanasoftware.com/> (7-6-2018)
10. Tracecloud <https://www.tracecloud.com/> (14-6-2018)
11. Psoda <https://www.psoda.com> (14-6-2018)
12. Trackstudio <http://www.trackstudio.com/> (14-6-2018)
13. Redmine <https://www.redmine.org/> (14-6-2018)
14. Especificación de Requisitos según el estándar de IEEE 830, IEEE Std. 830- 1998. (1998).
15. Symfony <https://symfony.com> (27-6-2018)
16. PHP <http://php.net/> (27-6-2018)
17. MySql <https://www.mysql.com/> (27-6-2018)
18. Doctrine <https://symfony.com/doc/2.8/doctrine.html> (27-6-2018)
19. Twig <https://twig.symfony.com/> (27-6-2018)
20. Bootstrap <http://getbootstrap.com/> (27-6-2018)
21. Swiftmailer <http://swiftmailer.org/> (27-6-2018)