



# Universidad Nacional de La Plata

## Facultad de Informática

Tesina de Licenciatura en Sistemas

### SGR: Sistema para la Gestión de Requerimientos

**Autores: Aguirre Verónica Anahí - Rípodas Alejandra**

Director: Esp. Delía Lisandro

Codirector: Lic. Marrero Luciano

# Agradecimientos

## *Verónica*

A mis padres, por acompañarme en este largo camino, incentivándome en todo momento para terminar la carrera.

A mis hermanas, que sin ellas nada sería igual.

A Sebastián, por acompañarme siempre, por su paciencia y por brindarme serenidad cuando fue necesario.

A mis amigos, por estar a mi lado y siempre darme ánimo a lo largo de toda la carrera.

A Ale, por acompañarme y hacer este proceso final más liviano y divertido.

## *Alejandra*

A mi familia que me dio la oportunidad de estudiar y me brindó su apoyo incondicional en todo momento.

A “el gordi” y a mis amigos que me bancaron todos estos años y siempre estuvieron presentes.

A mi compañera Vero por hacer inolvidable esta etapa, por ponerle siempre mucha garra y buena onda.

## *Verónica y Alejandra*

A nuestros directores Lisandro y Luciano, por habernos incentivado y guiado para que sea posible llevar a cabo este trabajo, pero principalmente, por habernos tenido toda la paciencia del mundo.

Al Instituto III-LIDI, por darnos la posibilidad de iniciar nuestra experiencia laboral y de crecer tanto en el ámbito profesional como personal.

# ÍNDICE

INTRODUCCIÓN.....	10
OBJETIVOS.....	10
OBJETIVOS GENERALES .....	10
OBJETIVOS ESPECÍFICOS.....	10
MOTIVACIÓN.....	10
1. PREFACIO .....	12
1.1 INTRODUCCIÓN.....	12
1.2 REQUERIMIENTOS.....	12
1.2.1 QUÉ ES UN REQUERIMIENTO.....	12
1.2.2 TIPOS DE REQUERIMIENTOS .....	13
1.2.3 CARACTERÍSTICAS DE LOS REQUERIMIENTOS .....	14
1.3 INGENIERÍA DE REQUERIMIENTOS .....	15
1.3.1 ANÁLISIS DEL PROBLEMA.....	15
1.3.2 ELABORACIÓN Y NEGOCIACIÓN.....	16
1.3.3 ESPECIFICACIÓN .....	16
1.3.4 VALIDACIÓN .....	17
1.3.5 ADMINISTRACIÓN DE REQUERIMIENTOS .....	17
1.4 PUNTOS DE VISTA .....	18
1.5 IMPORTANCIA DE LA EXTRACCIÓN CORRECTA DE LOS REQUERIMIENTOS.....	20
1.6 IMPACTO DE LOS ERRORES EN LA ETAPA DE ELICITACIÓN DE REQUERIMIENTOS.....	20
1.7 ESPECIFICACIÓN DE REQUERIMIENTOS.....	21
1.7.1 CLASES DE DOCUMENTOS DE REQUERIMIENTOS .....	21
1.7.2 ESPECIFICACIÓN EN LENGUAJE NATURAL .....	22
1.7.3 ESPECIFICACIÓN ESTRUCTURADA.....	23
1.7.4 EL ESTÁNDAR IEEE-830 PARA DOCUMENTOS DE REQUERIMIENTOS.....	24
1.8 TRAZABILIDAD.....	25
1.9 PRIORIZACIÓN .....	26
1.10 HERRAMIENTAS CASE .....	26
1.11 RESUMEN DEL CAPÍTULO.....	27
2. ANÁLISIS DE HERRAMIENTAS PARA LA GESTIÓN DE REQUERIMIENTOS.....	29
2.1 INTRODUCCIÓN.....	29
2.2 RELEVAMIENTO DE FUNCIONALIDADES DESEABLES .....	29
2.3 HERRAMIENTAS EXISTENTES .....	30

2.3.1 VISURE REQUIREMENTS.....	31
2.3.2 DOORSNG.....	31
2.3.3 ROMMANA.....	32
2.3.4 TRACECLOUD.....	32
2.3.5 PSODA .....	33
2.3.6 TRACKSTUDIO .....	34
2.3.7 REDMINE .....	34
2.3.8 REM .....	35
2.3.9 HELER .....	35
2.3.10 LET’S REQ! .....	36
2.4 ANÁLISIS COMPARATIVO .....	36
2.5 RESUMEN DEL CAPÍTULO .....	38
3. SGR: SISTEMA PARA LA GESTIÓN DE REQUERIMIENTOS.....	40
3.1 INTRODUCCIÓN.....	40
3.2 ALCANCE .....	40
3.3 PROPÓSITO.....	40
3.4 CARACTERÍSTICAS DE SGR.....	41
3.4.1 USUARIOS, ROLES Y PERMISOS.....	41
3.4.2 STAKEHOLDERS .....	41
3.4.3 ESTADOS.....	41
3.4.4 PRIORIDAD .....	42
3.4.5 ESTABILIDAD .....	43
3.4.6 NOTIFICACIONES DE CAMBIOS .....	43
3.4.7 INFORMACIÓN ADJUNTA.....	43
3.4.8 GENERACIÓN AUTOMÁTICA DEL DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS BASADO EN IEEE STD. 830-1998 .....	44
3.4.9 TRAZABILIDAD.....	44
3.5 CARACTERÍSTICAS NO FUNCIONALES DE SGR.....	44
3.5.1 INTERFAZ DE USUARIO.....	44
3.5.2 PORTABILIDAD .....	44
3.5.3 WEB ADAPTABLE.....	44
3.5.4 DISPONIBILIDAD .....	45
3.5.5 MANTENIBILIDAD.....	45
3.5.6 CONCURRENCIA .....	45

3.6 CARACTERÍSTICAS FUNCIONALES DE SGR.....	45
3.6.1 USUARIOS.....	45
3.6.2 ROLES .....	45
3.6.3 GRUPOS.....	45
3.6.4 ESTADOS.....	46
3.6.5 ESTABILIDADES.....	46
3.6.6 INSTITUCIONES.....	46
3.6.7 PROYECTOS .....	46
3.6.8 USUARIOS CON GRUPOS ASIGNADOS A PROYECTO .....	47
3.6.9 STAKEHOLDERS DE UN PROYECTO.....	47
3.6.10 HISTORIAL DE CAMBIOS DE UN PROYECTO .....	47
3.6.11 OBJETIVOS DE UN PROYECTO .....	47
3.6.12 ATRIBUTOS DE UN PROYECTO .....	48
3.6.13 GESTIÓN DE REQUERIMIENTOS .....	48
3.6.14 HISTORIAL DE CAMBIOS DE UN REQUERIMIENTO.....	48
3.6.15 STAKEHOLDERS DE UN REQUERIMIENTO .....	49
3.6.16 OBJETIVOS DE UN REQUERIMIENTO.....	49
3.6.17 APROBAR/RECHAZAR UN REQUERIMIENTO .....	49
3.6.18 ARCHIVOS ADJUNTOS .....	49
3.6.19 VERSIONES .....	49
3.6.20 CONVERSACIONES.....	50
3.6.21 COMENTARIOS.....	50
3.6.22 TABLERO DE USUARIOS.....	50
3.6.23 ENVÍO DE NOTIFICACIONES .....	51
3.6.24 REPORTE BASADO EN SRS ESTÁNDAR IEEE Std. 830-1998 .....	51
3.7 RESUMEN DEL CAPÍTULO.....	51
4. DISEÑO .....	53
4.1 INTRODUCCIÓN.....	53
4.2 MODELO CONCEPTUAL.....	53
4.3 DIAGRAMA DE CLASES .....	55
4.4 MOCKUPS.....	57
4.4.1 LOGIN .....	57
4.4.2 DISEÑO GENERAL .....	57
4.4.3 PANTALLA DE INICIO .....	58

4.4.4 LISTADO DE ELEMENTOS.....	59
4.4.5 ALTA DE ELEMENTO .....	59
4.4.6 EDICIÓN DE ELEMENTO .....	60
4.4.7 DETALLE DE ELEMENTO .....	61
4.4.8 DETALLE DE PROYECTOS .....	61
4.4.9 DETALLE DE REQUERIMIENTOS.....	62
4.5 RESUMEN DEL CAPÍTULO .....	63
5. IMPLEMENTACIÓN .....	64
5.1 INTRODUCCIÓN.....	64
5.2 TECNOLOGÍA UTILIZADA .....	64
5.2.1 PHP.....	64
5.2.2 FRAMEWORK SYMFONY .....	64
5.2.3 DOCTRINE.....	65
5.2.4 TWIG .....	65
5.2.5 BASE DE DATOS MYSQL .....	65
5.2.6 BOOTSTRAP.....	66
5.2.7 JQUERY .....	66
5.2.8 KNPPAGINATORBUNDLE.....	66
5.2.9 KNPSNAPPYBUNDLE.....	67
5.2.10 SWIFTMAILER.....	67
5.2.11 ENTITYAUDIT.....	67
5.2.12 GLIFFY.....	67
5.2.13 GITLAB.....	68
5.3 DISEÑO ARQUITECTÓNICO .....	68
5.4 RESUMEN DEL CAPÍTULO .....	69
6. CASO EXPERIMENTAL: ALBERGUE UNIVERSITARIO DE LA UNLP .....	70
6.1 INTRODUCCIÓN.....	70
6.2 GESTIÓN DE USUARIOS Y GRUPOS .....	70
6.3 GESTIÓN DE PROYECTOS.....	71
6.3.1 DETALLE DEL PROYECTO .....	72
6.3.2 OBJETIVOS.....	74
6.3.3 USUARIOS ASIGNADOS .....	74
6.3.4 ARCHIVOS ADJUNTOS .....	75
6.3.5 ATRIBUTOS.....	76

6.3.6 STAKEHOLDERS .....	76
6.3.7 VERSIONES .....	77
6.3.8 NOTIFICACIONES DE CAMBIOS Y COMUNICACIÓN ENTRE USUARIOS .....	78
6.3.9 MATRIZ DE TRAZABILIDAD .....	80
6.4 GESTIÓN DE REQUERIMIENTOS FUNCIONALES .....	81
6.4.1 DETALLE DE REQUERIMIENTO .....	84
6.4.2 ASIGNACIÓN DE STAKEHOLDERS .....	85
6.4.3 ASIGNACIÓN DE OBJETIVOS.....	86
6.4.4 VALIDACIÓN DE REQUERIMIENTOS .....	86
6.5 GENERACIÓN AUTOMÁTICA DEL SRS.....	87
6.6 RESUMEN DEL CAPÍTULO .....	88
7. CONCLUSIONES .....	89
8. TRABAJOS FUTUROS .....	91
BIBLIOGRAFÍA.....	92
ANEXO I .....	94
SRS ALBERGUE UNIVERSITARIO UNLP generado por SGR .....	94

## ÍNDICE DE FIGURAS

Figura 1.1. Imagen ilustrativa sobre los distintos puntos de vista.....	19
Figura 1.2. Costo relativo a la corrección de una falla en cada fase del ciclo de vida del software tradicional .....	21
Figura 1.3. Usuarios de un documento de requerimientos.....	22
Figura 1.4. Estructura de un documento de requerimientos .....	25
Figura 3.1. Flujo de estados esperado para un requerimiento.....	42
Figura 3.2. Flujo de estados esperado para un proyecto.....	42
Figura 4.1. Modelo conceptual de SGR.....	54
Figura 4.2. Diagrama de clases de SGR .....	56
Figura 4.3. Inicio de sesión del sistema (PC) .....	57
Figura 4.4. Inicio de sesión del sistema (Dispositivo móvil).....	57
Figura 4.5. Diseño de interfaz de SGR (PC) .....	58
Figura 4.6. Diseño de interfaz de SGR (Dispositivo móvil) .....	58
Figura 4.7. Tablero de usuario (PC).....	58
Figura 4.8. Tablero de usuario (Dispositivo móvil) .....	58
Figura 4.9. Listado de elementos (PC) .....	59
Figura 4.10. Listado de elementos (Dispositivo móvil) .....	59
Figura 4.11. Alta de un elemento (PC) .....	60
Figura 4.12. Alta de un elemento (Dispositivo móvil).....	60
Figura 4.13. Edición de un elemento (PC).....	60
Figura 4.14. Edición de un elemento (Dispositivo móvil) .....	60
Figura 4.15. Detalle de un elemento (PC) .....	61
Figura 4.16. Detalle de un elemento (Dispositivo móvil).....	61
Figura 4.17. Detalle de un proyecto (PC) .....	62
Figura 4.18. Detalle de un proyecto (Dispositivo móvil).....	62
Figura 4.19. Detalle de un requerimiento (PC) .....	62
Figura 4.20. Detalle de un requerimiento (Dispositivo móvil).....	62
Figura 5.1. Formato elegido para el paginador.....	66
Figura 5.2. Encabezado de tabla ordenado por Abreviatura Ascendente .....	66
Figura 5.3. Flujo HTTP .....	68
Figura 5.4. Arquitectura de SGR.....	69
Figura 6.1. Listado de usuarios .....	71
Figura 6.2. Listado de grupos .....	71
Figura 6.3. Listado de proyectos .....	71
Figura 6.4. Creación del proyecto Albergue Universitario UNLP .....	72

Figura 6.5. Detalle del proyecto Albergue Universitario UNLP.....	73
Figura 6.6. Detalle del proyecto Albergue Universitario UNLP (Dispositivo Móvil).....	73
Figura 6.7. Listado de objetivos .....	74
Figura 6.8. Listado de usuarios asignados.....	75
Figura 6.9. Listado de archivos adjuntos.....	75
Figura 6.10. Formulario para el alta de atributos .....	76
Figura 6.11. Listado de stakeholders .....	77
Figura 6.12. Versiones de un proyecto .....	77
Figura 6.13. Tablero de usuario .....	78
Figura 6.14. Tablero de usuario (Dispositivo Móvil) .....	78
Figura 6.15. Email enviado a un usuario participante de un proyecto actualizado.....	79
Figura 6.16. Email enviado a todos los usuarios asignados a un proyecto.....	79
Figura 6.17. Nueva Conversación de un Proyecto .....	80
Figura 6.18. Conversación y comentarios de un proyecto .....	80
Figura 6.19. Matriz de trazabilidad .....	81
Figura 6.20. Formulario de creación de un requerimiento funcional.....	82
Figura 6.21. Listado de requerimientos .....	83
Figura 6.22. Listado de requerimientos (Dispositivo Móvil).....	83
Figura 6.23. Detalle de un requerimiento funcional.....	84
Figura 6.24. Detalle de un requerimiento funcional (Dispositivo Móvil).....	85
Figura 6.25. Listado de stakeholders asignados a un requerimiento .....	86
Figura 6.26. Listado de objetivos asignados a un requerimiento .....	86
Figura 6.27. Listado de requerimientos validados y rechazados .....	87

# INTRODUCCIÓN

## OBJETIVOS

### OBJETIVOS GENERALES

Desarrollar una aplicación web denominada SGR (Sistema para la Gestión de Requerimientos) para la administración de requerimientos de manera colaborativa, que permita brindar acceso desde diferentes dispositivos, realizar trazabilidad sobre los requerimientos y almacenar información de manera centralizada.

Fomentar la participación de clientes para que el producto final se ajuste a lo que realmente necesitan es un factor clave para el éxito de un proyecto. Por esto, SGR permitirá validar los requerimientos y realizarles comentarios para llegar al producto final esperado.

A partir de esto, se busca disminuir los defectos en los requerimientos y/o encontrar errores en etapas tempranas.

### OBJETIVOS ESPECÍFICOS

- Evaluar herramientas de gestión de requerimientos actuales, tanto comerciales como académicas.
- Construir una aplicación web que brinde soporte a la Ingeniería de Requerimientos, reuniendo las características más destacadas de las herramientas evaluadas.
- Guiar a analistas con escasa experiencia en el proceso de desarrollo de especificaciones de requerimientos.
- Facilitar la transferencia de conocimiento entre los diferentes miembros de un proyecto.
- Permitir el uso de la aplicación desde cualquier dispositivo con acceso a internet, y utilizar diseño web adaptable para su correcta visualización.

## MOTIVACIÓN

Los requerimientos son un elemento clave en el desarrollo de un proyecto de software ya que marcan el punto de partida para actividades tales como planeación, estimación de esfuerzo, definición de recursos, costos, entre otras. Inician cuando comienza un proyecto y pueden variar a lo largo de este.

La gestión de requerimientos debe realizar el seguimiento y control de los cambios de estos durante el ciclo de vida de un proyecto.

En [1] mediante 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, 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 pasar 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 son herramientas CASE (Computer-Aided Software Engineering) que tienen por objetivo hacer que el desarrollo de un proyecto sea un proceso eficientemente organizado. Existen gran cantidad de estas herramientas, con interfaces y funcionalidades muy variadas. Las mismas permiten controlar proyectos complejos; determinar nivel de complejidad; facilitar el trabajo de especificación, organización, almacenamiento y gestión de requerimientos. Además, añaden beneficios significativos al reducir errores que puedan perjudicar el correcto desarrollo de un proyecto a largo plazo [2]. Actualmente, existen en el mercado varias herramientas comerciales tales como Visure Requirements, DOORSNG, Rommana, TraceCloud, Psoda, TrackStudio, Redmine, entre otras; que permiten llevar a cabo todo el proceso de gestión de requerimientos. Como principales características se puede apreciar que están bien documentadas, brindan soporte a sus usuarios y tienen la posibilidad de ampliar su funcionalidad mediante el pago de diferentes licencias.

Por otra parte, se encuentran las herramientas académicas como REM, Heler o Let's req! Beta, entre otras; las cuales no poseen gran documentación de acceso público y han sido realizadas dentro de alguna metodología propia de la organización desarrolladora, lo que dificulta entenderlas.

Se decidió desarrollar SGR, un software que reúne las características más destacadas de las herramientas evaluadas y que brinda soporte a una de las actividades más importantes en el desarrollo de un proyecto de software, como es la fase de Ingeniería de Requerimientos. SGR será de uso libre, no comercial, multiplataforma y se podrá acceder desde cualquier dispositivo, ya que será realizado utilizando técnicas de diseño web adaptable.

# 1. PREFACIO

## 1.1 INTRODUCCIÓN

Los autores definen a la Ingeniería de Software desde distintos puntos de vista:

“La ingeniería de Software es la aplicación de principios científicos a la transformación ordenada de un problema en una solución elaborada de software, y el mantenimiento subsecuente de ese software hasta el final de su vida útil. La Ingeniería de Software es más que sólo programación; el proceso de ingeniería de software generalmente comienza mucho antes que una línea de código sea escrita y continúa mucho después de que la versión inicial del programa ha sido completada.” [4].

“La Ingeniería de Software es una disciplina de ingeniería que se interesa por todos los aspectos de la producción de software, desde las primeras etapas de la especificación del sistema hasta el mantenimiento del sistema después de que se pone en operación. En esta definición se presentan dos frases clave:

- **Disciplina de ingeniería:** Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde es adecuado. Siempre tratan de encontrar soluciones a problemas, incluso cuando no hay teorías ni métodos aplicables. Los ingenieros también reconocen que deben trabajar ante restricciones organizacionales y financieras, de modo que buscan soluciones dentro de tales limitaciones.
- **Todos los aspectos de la producción del software:** La Ingeniería de Software no sólo se interesa por los procesos técnicos del desarrollo de software, sino también incluye actividades como la administración del proyecto de software y el desarrollo de herramientas, así como métodos y teorías para apoyar la producción de software.” [5]

“La Ingeniería de Software es: 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software. 2) El estudio de enfoques según el punto 1” [6].

En este capítulo se presentan conceptos sobre Ingeniería de Software, fundamentalmente sobre ingeniería de requerimientos, que permitirán comprender el contexto sobre el cual se desarrolla la presente tesina.

## 1.2 REQUERIMIENTOS

### 1.2.1 QUÉ ES UN REQUERIMIENTO

Los requerimientos especifican qué es lo que el sistema debe hacer (sus funciones) y sus propiedades esenciales y deseables. La captura de los requerimientos tiene como objetivo principal la comprensión de lo que los clientes y los usuarios esperan que haga el sistema. Un requerimiento expresa el propósito del sistema sin considerar cómo se va a implementar. La captura y el análisis de los requerimientos del sistema es una de las fases más importantes para que el proyecto tenga éxito. El costo de reparar un error se incrementa en un factor de diez de una fase de desarrollo a la

siguiente, por lo tanto, la preparación de una especificación adecuada de requerimientos reduce los costos y el riesgo general asociado con el desarrollo [7].

Los requerimientos no especifican cómo debe implementarse un sistema. Las descripciones específicas de la implementación no se consideran como requerimientos, a menos que así lo establezca el cliente. Es decir, un requerimiento apunta al propósito del sistema sin considerar cómo se va a implementar. Se dice que los requerimientos identifican el **qué** del sistema, en tanto el diseño establece el **cómo** del sistema [1].

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información [5].

## 1.2.2 TIPOS DE REQUERIMIENTOS

Los requerimientos describen el comportamiento de un sistema. A medida que este actúa sobre los datos o las instrucciones, los objetivos y las entidades se mueven de un estado a otro, por ejemplo, de *ocupado a libre*, de *lleno a vacío*, etc. En un estado dado el sistema satisface un conjunto de condiciones. Los requerimientos expresan los estados del sistema, de sus objetivos y, además, describen las transacciones de un estado a otro. En particular, describen las actividades del sistema como una entrada y el estado de cada entidad en el sistema antes y después de que ocurra dicha actividad [1].

Tanto para Sommerville [5] como para Pfleeger [1] los requerimientos pueden dividirse en dos grupos significativos:

- **Requerimientos funcionales:** Los requerimientos funcionales 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. Los requerimientos funcionales se describen por lo general de forma simplificada para que puedan ser comprendidos por los usuarios del sistema. Sin embargo, requerimientos funcionales más específicos del sistema detallan las funciones del sistema, sus entradas y salidas, sus excepciones, etcétera [5].

Los requerimientos funcionales del sistema varían desde requerimientos generales (que cubren lo que tiene que hacer el sistema), hasta requerimientos muy específicos (que reflejan maneras locales de trabajar o los sistemas existentes de una organización) [5].

En principio, la especificación de los requerimientos funcionales de un sistema debe ser completa y consistente. La primera característica tiene que ver con que se definan todos los servicios requeridos por el usuario. Mientras que la segunda hace referencia a evitar definiciones contradictorias [5].

Un requerimiento funcional describe una interacción entre el sistema y su ambiente. En otras palabras, describen cómo debe comportarse el sistema ante un determinado estímulo. Los requerimientos funcionales son independientes de la implementación de la solución para el problema del cliente. Se describe lo que el sistema hará sin considerar donde se implementará, el lenguaje de programación a utilizar, las estructuras internas de los datos a elegir, etcétera [1].

- **Requerimientos no funcionales:** 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. De forma alternativa, pueden definir restricciones sobre la implementación del sistema, como las capacidades de los dispositivos I/O o las representaciones de datos usadas en las interfaces con otros sistemas.

Los requerimientos no funcionales, como el rendimiento, la seguridad o la disponibilidad, especifican o restringen por lo general características del sistema como un todo. Los requerimientos no funcionales a menudo son más significativos que los requerimientos funcionales individuales. Es común que los usuarios del sistema encuentren formas para trabajar en torno a una función del sistema que realmente no cubra sus necesidades. No obstante, el fracaso para cubrir los requerimientos no funcionales haría que todo el sistema fuera inútil.

Los requerimientos no funcionales surgen a través de necesidades del usuario, debido a restricciones presupuestales, políticas de la organización, necesidad de interoperabilidad con otro software o sistemas de hardware, o factores externos como regulaciones de seguridad o legislación sobre privacidad [5].

Según Pfleeger los requerimientos no funcionales describen una restricción sobre el sistema que limita las elecciones en la construcción de una solución al problema. Por ejemplo, se puede solicitar que el sistema debe implementarse para un determinado tipo de computadora o el tiempo de respuesta del sistema ante un determinado evento. Estas restricciones por lo general limitan la elección del lenguaje, plataformas, técnicas y herramienta de implementación. Sin embargo, la elección se realiza en la etapa de diseño, después de haber especificado los requerimientos [1].

### 1.2.3 CARACTERÍSTICAS DE LOS REQUERIMIENTOS

Según Pfleeger, para asegurar que tanto los clientes como los desarrolladores comprendan y utilicen correctamente los requerimientos, es importante que estos sean de alta calidad. Para esto, los requerimientos deben poseer las siguientes características:

- **Deben ser correctos:** tanto el cliente como el desarrollador deben revisarlos para asegurar que no tienen errores.
- **Deben ser consistentes:** dos requerimientos son inconsistentes cuando es imposible satisfacerlos simultáneamente.
- **Deben ser completos:** el conjunto de requerimientos es completo si todos los estados posibles, cambios de estado, entradas, productos y restricciones están descritos en alguno de los requerimientos.
- **Deben ser realistas:** todos los requerimientos deben ser revisados para asegurar que sean posibles de llevar a cabo.
- **Deben describir lo que es necesario para el cliente:** a veces un requerimiento restringe innecesariamente a los desarrolladores, o incluye funciones que no están directamente relacionadas con el problema que se está tratando. Los requerimientos deben ser revisados para conservar sólo aquellos que inciden directamente en la resolución del problema del cliente.

- Deben ser verificables: se deben poder preparar pruebas que demuestren que se han cumplido con los requerimientos.
- Deben ser rastreables: se debe poder rastrear cada función del sistema hasta el conjunto de requerimientos que la establece. De esta forma, se puede examinar minuciosamente cada posible requerimiento, mejorándolo y cambiándolo a medida que se avanza.

Además de las características expuestas anteriormente, la IEEE [8] agrega las siguientes:

- Todo requerimiento deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.
- Los requerimientos no deben ser ambiguos, deben tener una sola interpretación. Para eliminar la ambigüedad relacionada a los requerimientos expresados, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que habitualmente poseen más de una interpretación, se definirán con precisión en el glosario.
- Los requerimientos deben ser clasificados. Normalmente, no todos los requerimientos son igual de importantes. Los requerimientos pueden clasificarse por importancia (esencial, condicional u opcional) o por estabilidad (cambios que se espera que afecten al requerimiento). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requerimientos no esenciales.

## 1.3 INGENIERÍA DE REQUERIMIENTOS

Según Pressman [6] la ingeniería de requerimientos 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.

Las actividades relacionadas con la Ingeniería de Software pueden generalizarse en: Análisis del Problema, Elaboración y Negociación, Especificación, Validación y Evolución. A continuación, se describirán cada una de estas actividades.

### 1.3.1 ANÁLISIS DEL PROBLEMA

La mayor parte de los proyectos comienzan cuando se identifica una necesidad del negocio o se descubre un nuevo mercado o servicio potencial. En la concepción del proyecto se establece el entendimiento básico del problema, las personas que quieren una solución, la naturaleza de la solución que se desea, así como la eficacia de la comunicación y la colaboración entre los participantes y el equipo de software [6].

Los problemas que se encuentran cuando ocurre el análisis son:

- Problemas de alcance: La frontera de los sistemas está mal identificada o los clientes o usuarios finales especifican detalles técnicos innecesarios que confunden los objetivos del sistema.
- Problemas de entendimiento: Los clientes o usuarios no están seguros de lo que necesitan, comprenden mal las capacidades y limitaciones de su ambiente de computación, no entienden todo el dominio del problema, tienen problemas para comunicar sus necesidades al ingeniero de sistemas, omiten información que consideran obvia, especifican

requerimientos que están en conflicto con las necesidades de otros clientes o usuarios, o solicitan requerimientos ambiguos o que no pueden someterse a prueba.

- Problemas de volatilidad: Los requerimientos cambian con el tiempo [6].

### 1.3.2 ELABORACIÓN Y NEGOCIACIÓN

Pressman [6] considera que la información obtenida del cliente durante el análisis se expande y refina durante la elaboración de este. Esta tarea se centra en desarrollar un modelo refinado de los requerimientos que identifique distintos aspectos de la construcción del software, su comportamiento e información.

El autor también expresa que en ocasiones los clientes y usuarios piden más de lo que puede lograrse dado lo limitado de los recursos del negocio. Además, es relativamente común que distintos clientes o usuarios propongan requerimientos conflictivos. Estos desacuerdos deben ser resueltos por medio de un proceso de negociación.

Por otra parte, Pfleeger [1] se refiere a esta actividad como la extracción de los requerimientos: “Es una parte especialmente crítica del proceso. Podemos utilizar una gran variedad de técnicas para determinar qué es lo que los usuarios y los clientes necesitan realmente. Una de las maneras de realizar un análisis del problema consiste en identificar las personas, los procesos y los recursos involucrados y después documentar las relaciones existentes entre ellos. Se les pregunta a los clientes quién está involucrado y se intenta determinar el límite del sistema. Se averigua qué elementos de datos pasan de un rol a otro, y cuáles son los procesos que transforman los datos de una forma o estado a otro. Durante la extracción de requerimientos se interroga a los usuarios y clientes sobre los mismos aspectos, de muchas maneras, hasta tener la seguridad de que se comprende lo que quieren y necesitan”.

### 1.3.3 ESPECIFICACIÓN

Una especificación puede ser un documento escrito, un conjunto de modelos gráficos, un modelo matemático formal, un conjunto de escenarios de uso, un prototipo o cualquier combinación de éstos [6].

Los requerimientos del usuario para un sistema deben describir los requerimientos funcionales y no funcionales, de forma que sean comprensibles para los usuarios del sistema que no cuentan con un conocimiento técnico detallado.

Los requerimientos del sistema son versiones extendidas de los requerimientos del usuario que los ingenieros de software usan como punto de partida para el diseño del sistema. Añaden detalles y explican cómo el sistema debe brindar los requerimientos del usuario.

La especificación de requerimientos es el proceso de escribir, en un documento, los requerimientos del usuario y del sistema. De manera ideal, los requerimientos del usuario y del sistema deben ser claros, sin ambigüedades, fáciles de entender, completos y consistentes. Esto en la práctica es difícil de lograr, ya que los participantes interpretan los requerimientos de formas diferentes y es frecuente encontrar requerimientos en conflicto y con inconsistencias inherentes [5].

Más adelante en este capítulo se explicarán detalladamente los distintos aspectos de la especificación de requerimientos, haciendo énfasis en la especificación mediante el uso de documentos escritos.

### 1.3.4 VALIDACIÓN

La validación de requerimientos es el proceso de verificar que éstos definen realmente el sistema que en verdad necesita el cliente. Se superpone con la etapa del análisis, ya que se interesa por encontrar problemas con los requerimientos. La validación de requerimientos es importante porque los errores en un documento de requerimientos pueden conducir a grandes costos por tener que rehacer partes, cuando dichos problemas se descubren durante el desarrollo del sistema o después de que éste se encuentra en servicio. En general, el costo por corregir un problema de requerimientos al hacer un cambio en el sistema es mucho mayor que reparar los errores de diseño o codificación. La razón es que un cambio a los requerimientos significa generalmente que también debe cambiar el diseño y la implementación del sistema. Más aún, el sistema debe ponerse a prueba nuevamente [5].

Pfleeger propone los siguientes pasos para realizar el proceso de la validación de requerimientos:

1. Revisar las metas declaradas y los objetivos del sistema.
2. Comparar los requerimientos con las metas y objetivos para verificar que todos los requerimientos son necesarios.
3. Describir el ambiente en el cual debe operar el sistema. Examinar las interfaces entre el sistema y los otros sistemas, y verificar que sean correctas y completas. Revisar nuevamente el flujo de información y la estructura del sistema para asegurar que los requerimientos reflejan exactamente el significado y la intención del cliente. Las funciones del sistema deben ser consistentes con el alcance y la intención del cliente. Además, las funciones y restricciones deben ser realistas y estar dentro de las capacidades de desarrollo del equipo. Todos los requerimientos se deben corroborar para localizar omisiones, falta de completitud e inconsistencias.
4. Evaluar y documentar todo riesgo implicado con el desarrollo o el funcionamiento real del sistema. Discutir y comparar alternativas entre el cliente y los desarrolladores para acordar los enfoques a utilizar. Siempre que se identifique un problema, el equipo de revisión lo debe documentar, determinar su causa y establecer las acciones a realizar para subsanarlos antes de comenzar el proceso de diseño.
5. Discutir sobre las pruebas del sistema, ¿Cómo continuará la verificación y la validación de los requerimientos a medida que avanza el proyecto? Si el sistema tendrá un desarrollo escalonado, ¿cómo se hará la comprobación de los requerimientos durante las fases intermedias?

Siempre que se identifica un problema en esta etapa, el equipo de revisión documenta, determina su causa y establece la acción para subsanarlo antes de comenzar el proceso de diseño [1].

### 1.3.5 ADMINISTRACIÓN DE REQUERIMIENTOS

La administración de requerimientos es el proceso de comprender y controlar los cambios en los requerimientos del sistema. Es necesario seguir la pista de requerimientos individuales y mantener

los vínculos entre los requerimientos dependientes, de manera que pueda valorarse el efecto del cambio en los requerimientos. También es preciso establecer un proceso formal para hacer cambios a las propuestas y vincular éstos con los requerimientos del sistema. El proceso formal de la administración de requerimientos debe comenzar tan pronto como esté disponible un borrador del documento de requerimientos. Sin embargo, hay que empezar a planear cómo administrar el cambio en los requerimientos durante el proceso de adquisición de estos [5].

La administración del cambio es esencial porque es necesario determinar si los beneficios de implementar nuevos requerimientos están justificados por los costos de la implementación. La ventaja de usar un proceso formal para la administración del cambio es que todas las propuestas de cambio se tratan de manera consistente y los cambios al documento de requerimientos se realizan en una forma controlada. Existen tres etapas principales de un proceso de administración del cambio:

- **Análisis del problema y especificación del cambio:** el proceso comienza con la identificación de un problema en los requerimientos o, en ocasiones, con una propuesta de cambio específica. Durante esta etapa, el problema o la propuesta de cambio se analizan para comprobar que es válida. Este análisis retroalimenta al solicitante del cambio, quien responderá con una propuesta de cambio de requerimientos más específica, o decidirá retirar la petición.
- **Análisis del cambio y estimación del costo:** el efecto del cambio propuesto se valora usando información de seguimiento y conocimiento general de los requerimientos del sistema. El costo por realizar el cambio se estima en términos de modificaciones al documento de requerimientos y, si es adecuado, al diseño y la implementación del sistema. Una vez completado este análisis, se toma una decisión acerca de si se procede o no con el cambio de requerimientos.
- **Implementación del cambio:** se modifica el documento de requerimientos y, donde sea necesario, el diseño y la implementación del sistema. Hay que organizar el documento de requerimientos de forma que sea posible realizar cambios sin reescritura o reorganización extensa. Conforme a los programas, la variabilidad en los documentos se logra al minimizar las referencias externas y al hacer las secciones del documento tan reducidas e independientes como sea posible. De esta manera, secciones individuales pueden modificarse y sustituirse sin afectar otras partes del documento [5].

## 1.4 PUNTOS DE VISTA

Debido a que existen distintos tipos de participantes, los requerimientos del sistema se exploran desde diversos puntos de vista. Cada uno de los integrantes, también denominados stakeholders, aportará información al proceso de ingeniería en requerimientos [6].

Para Pfleeger los stakeholders en el proceso de recolección de requerimientos son:

- Los supervisores del contrato, quienes sugieren hitos de control y cronogramas que restringen el desarrollo del sistema.
- Los clientes y usuarios, quienes deben comprender los requerimientos de modo que puedan estar seguros de que el sistema satisface sus necesidades.

- Los gerentes de negocios, que pueden comprender las probables consecuencias de construir y utilizar el sistema.
- Los diseñadores, quienes utilizan los requerimientos como base para el desarrollo de una solución aceptable que se implementará como un sistema basado en software.
- Los verificadores, que desarrollarán datos de prueba y secciones de prueba para asegurar que el sistema de software satisface cada uno de los requerimientos.

A medida que se recaba información procedente de múltiples puntos de vista, los requerimientos que surjan tal vez sean inconsistentes o estén en conflicto uno con otro (Figura 1.1 [9]). Debe clasificarse toda la información de los stakeholders (incluso los requerimientos inconsistentes y conflictivos) de forma que permita a quienes toman las decisiones, escoger para el sistema un conjunto de requerimientos que tenga coherencia interna [6].

El análisis del dominio del software es la identificación, análisis y especificación de los requerimientos comunes, a partir de un dominio de aplicación específica, normalmente para usarlo varias veces en múltiples proyectos dentro del dominio de la aplicación. La meta del análisis del dominio es clara: encontrar o crear aquellas clases o patrones de análisis que sean aplicables en lo general, de modo que puedan volverse a usar. Un punto de vista complementario del análisis del dominio involucra el modelado de éste, de manera que los ingenieros del software y otros stakeholders aprendan más al respecto. El análisis del dominio es una actividad de la Ingeniería de Software que no está conectada con ningún proyecto de software [6].



Figura 1.1. Imagen ilustrativa sobre los distintos puntos de vista

## 1.5 IMPORTANCIA DE LA EXTRACCIÓN CORRECTA DE LOS REQUERIMIENTOS

En su libro sobre la Ingeniería de Software, Pfleeger [1] cita a varios autores que mencionan la importancia de la correcta extracción de los requerimientos; algunas de las cuales serán referenciadas a continuación.

- Boehm y Paccio en uno de sus estudios manifiestan que la mayoría de los errores se producen durante el diseño, y que usualmente existen tres defectos de diseño por cada dos de codificación. Puntualizan que el alto número de defectos atribuidos a la etapa de diseño podrían derivar de errores de requerimientos.
- Basili y Perricone en una investigación empírica sobre errores de software, informan que el 48% de los defectos observados en proyectos de software de mediana escala eran “atribuidos a especificaciones funcionales o requerimientos incorrectos o mal interpretados”.
- Beizer expresa que los problemas resultan de la forma en la que están especificados o implementados los requerimientos, así como también los requerimientos incorrectos, requerimientos ilógicos o no razonables, ambiguos, incompletos o sobre especificados, requerimientos no compatibles o imposibles de probar, requerimientos mal procesados o cambiados.
- Los requerimientos expresados en una especificación (o a menudo no expresados porque no existe una especificación), son la mayor fuente de costos de *bugs*<sup>1</sup>. El rango de errores va desde un pequeño porcentaje hasta más del 50% dependiendo de la aplicación y del ambiente. Estos defectos pueden provocar grandes daños en el sistema, ya que lo invaden tempranamente y, generalmente, son los últimos en ser detectados. No es raro que un requerimiento defectuoso pase todas las pruebas de desarrollo y sea descubierto después de que se ha instalado el sistema.
- Perry y Stieg concluyen que el 79,6% de los defectos de las interfaces y el 20,4% de los defectos de implementación se deben a requerimientos omitidos o incorrectos.

## 1.6 IMPACTO DE LOS ERRORES EN LA ETAPA DE ELICITACIÓN DE REQUERIMIENTOS

La mayoría de los fracasos de los proyectos de software están asociados directamente con los requerimientos y el diseño preliminar. Encontrar y corregir los defectos en el software es más costoso que hacerlo en la fase de requerimientos.

---

<sup>1</sup> Bugs: es un error de software, error o simplemente fallo es un problema en un programa de computador o sistema de software que desencadena un resultado indeseado (Figura 1.2 [10]).



Figura 1.2. Costo relativo a la corrección de una falla en cada fase del ciclo de vida del software tradicional

Los errores en la etapa de elicitación de requerimientos tienen un importante impacto según [11]:

- El software resultante puede no satisfacer a los usuarios.
- Las interpretaciones no precisas de los requerimientos pueden causar desacuerdos entre clientes y desarrolladores.
- Puede consumirse tiempo y dinero construyendo el sistema incorrecto.

Estos errores pueden deberse a:

- Comunicación deficiente entre analistas y clientes o entre analistas y desarrolladores.
- Baja participación de los interesados.
- Gran cantidad de requerimientos, lo que dificulta manejarlos.
- Cambios no controlados en los requerimientos.
- Documentación escasa o errónea.

El porcentaje de fracasos en los proyectos de software generados por los problemas en la elicitación de requerimientos puede disminuirse a partir del uso de herramientas CASE. Éstas contribuyen a estructurar la definición y carga de los requerimientos, agilizar la comunicación entre clientes y analistas, mantener un historial de cambios de los requerimientos, así como también generar y centralizar la información, lo que se traduce en documentación más completa y correcta.

## 1.7 ESPECIFICACIÓN DE REQUERIMIENTOS

### 1.7.1 CLASES DE DOCUMENTOS DE REQUERIMIENTOS

Debido a que el foco de la atención está puesto sobre el cliente, 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**; escrito en términos que el cliente pueda entender, la definición de requerimientos es un listado completo de todo lo que el cliente espera que haga el sistema propuesto. 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 ellos.

Por otra parte, la **especificación de requerimientos** reitera la definición en los términos técnicos apropiados para el desarrollo del diseño de un sistema. A veces un único documento puede servir para ambos propósitos, pero a menudo, se necesitan ambos documentos y se debe ser cuidadoso para no perder información ni se introduzcan cambios cuando se reinterpreta la definición como una especificación. Existe una correspondencia directa entre cada requerimiento del documento de definición y aquellos documentados en la especificación [1].

El documento de requerimientos tiene un conjunto variado de usuarios, desde el administrador ejecutivo de la organización que paga por el sistema, hasta los ingenieros responsables del desarrollo del software. La figura 1.3 muestra a los posibles usuarios del documento y cómo ellos lo utilizan. La diversidad de posibles usuarios significa que el documento de requerimientos debe ser un compromiso entre la comunicación de los requerimientos a los clientes, la definición de los requerimientos con detalle preciso para desarrolladores y examinadores, y la inclusión de información sobre la posible evolución del sistema [5].

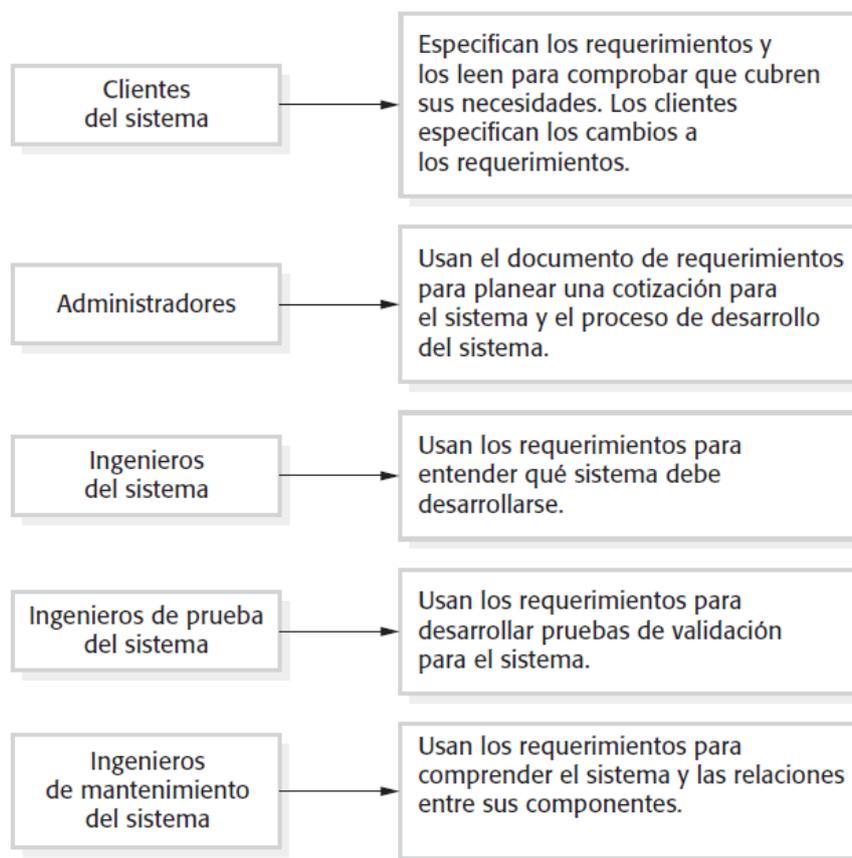


Figura 1.3. Usuarios de un documento de requerimientos

### 1.7.2 ESPECIFICACIÓN EN LENGUAJE NATURAL

Desde los comienzos de la Ingeniería de Software, el lenguaje natural se usa para escribir los requerimientos de software. Es expresivo, intuitivo y universal. También es potencialmente vago,

ambiguo y su significado depende de los antecedentes del lector. Como resultado, hay varias formas alternativas de escribir los requerimientos. Sin embargo, ninguna se ha adoptado de manera amplia, por lo que el lenguaje natural seguirá siendo la forma más usada para especificar los requerimientos del sistema y del software.

Para minimizar la interpretación errónea al escribir los requerimientos en lenguaje natural, se recomienda seguir algunos lineamientos sencillos:

1. Se debe elaborar un formato estándar y asegurarse de que todas las definiciones de requerimientos se adhieran a dicho formato. Al estandarizar el formato es menos probable cometer omisiones y más sencillo comprobar los requerimientos.
2. Se debe utilizar el lenguaje de manera clara para distinguir entre requerimientos obligatorios y deseables. Los primeros son requerimientos que el sistema debe soportar y, por lo general, se escriben en futuro “debe ser”. En tanto que los requerimientos deseables no son necesarios y se escriben en tiempo pospretérito o como condicional “debería ser”.
3. Se recomienda utilizar texto resaltado (negrita, cursiva o color) para seleccionar partes clave del requerimiento.
4. No se debe deducir que los lectores entienden el lenguaje técnico de la Ingeniería de Software. Es fácil que se malinterpreten palabras como “arquitectura” y “módulo”. Por lo tanto, se debe evitar el uso de jerga, abreviaturas y acrónimos.
5. Siempre que sea posible, es deseable asociar una razón con cada requerimiento de usuario. La razón debe explicar por qué se incluyó el requerimiento. Es particularmente útil cuando los requerimientos cambian, pues ayuda a decidir cuáles cambios serían indeseables [5].

Por muchos años, los requerimientos fueron expresados en el lenguaje natural del cliente, utilizando enunciados y frases convencionales. Sin embargo, ha quedado en claro que existen varios problemas con el uso exclusivo del lenguaje natural. En primer lugar, si los requerimientos deben resultar de alguna utilidad, todos los integrantes del proyecto deben interpretar su significado de la misma manera. Si el cliente piensa sobre un objetivo o característica de una manera y los desarrolladores lo hacen de otra, los requerimientos llevan a confusión; es improbable que ambos tengan la misma convención acerca de las palabras utilizadas. En segundo lugar, los requerimientos no siempre se separan fácilmente según los elementos del sistema con los que interactúan. A veces es difícil (sino imposible) rastrear desde una característica del sistema hasta el requerimiento que la define o afecta [1].

### 1.7.3 ESPECIFICACIÓN ESTRUCTURADA

El lenguaje natural estructurado es una manera de escribir requerimientos del sistema, donde se limita la libertad del escritor de requerimientos y todos se anotan en una forma estándar. Aunque este enfoque conserva la mayoría de la expresividad y comprensibilidad del lenguaje natural, asegura que haya cierta uniformidad sobre la especificación. Las anotaciones en el lenguaje estructurado emplean plantillas para especificar los requerimientos del sistema. Para usar un enfoque estructurado que especifique los requerimientos del sistema, hay que definir una o más plantillas estándar para requerimientos y representar dichas plantillas como formas estructuradas.

Al usar especificaciones estructuradas se eliminan algunos de los problemas de la especificación del lenguaje natural. La variabilidad en la especificación se reduce y los requerimientos se organizan de

forma más efectiva. Sin embargo, en ocasiones todavía es difícil escribir requerimientos sin ambigüedades, en particular cuando deben especificarse cálculos complejos [5].

#### 1.7.4 EL ESTÁNDAR IEEE-830 PARA DOCUMENTOS DE REQUERIMIENTOS

El documento IEEE-830[8] presenta un formato de especificación de requerimientos de software (Software Requirements Specification, SRS). En él se indica que un buen documento de requerimientos debe contemplar toda la información presentada y, aunque propone una organización de la información, no exige que se siga estrictamente el formato.

Los principales objetivos que se identifican en IEEE-830 para la especificación de requerimientos son:

1. Ayudar a los clientes a describir claramente lo que se desea obtener mediante un determinado software: El cliente debe participar activamente en la especificación de requerimientos, ya que éste tiene una visión mucho más detallada de los procesos que se llevan a cabo.
2. Ayudar a los desarrolladores a entender qué necesita exactamente el cliente: En muchas ocasiones el cliente no sabe exactamente qué es lo que necesita. El SRS permite al cliente definir todos los requerimientos que desea y al mismo tiempo los desarrolladores tienen una base fija en la que trabajar. Si no se realiza una buena especificación de requerimientos, los costes de desarrollo pueden incrementarse considerablemente, ya que se deben hacer cambios durante la creación de la aplicación.
3. Servir de base para desarrollos de estándares de SRS particulares para cada organización.

Una buena especificación de requerimientos de software ofrece una serie de ventajas entre las que se destacan el contrato entre cliente y desarrolladores, la reducción del esfuerzo en el desarrollo, una buena base para la estimación de costos y planificación, un punto de referencia para procesos de verificación y validación, y una base para la identificación de posibles mejoras en los procesos analizados [12].

La figura 1.4 indica una posible organización para un documento de requerimientos basada en un estándar del IEEE para documentos de requerimientos. Este estándar es genérico y se adapta a usos específicos. La información que se incluya en un documento de requerimientos depende del tipo de software que se va a desarrollar y del enfoque que se utilice para el desarrollo [5].

Capítulo	Descripción
Prefacio	Debe definir el número esperado de lectores del documento, así como describir su historia de versiones, incluidas las causas para la creación de una nueva versión y un resumen de los cambios realizados en cada versión.
Introducción	Describe la necesidad para el sistema. Debe detallar brevemente las funciones del sistema y explicar cómo funcionará con otros sistemas. También tiene que indicar cómo se ajusta el sistema en los objetivos empresariales o estratégicos globales de la organización que comisiona el software.
Glosario	Define los términos técnicos usados en el documento. No debe hacer conjeturas sobre la experiencia o la habilidad del lector.
Definición de requerimientos del usuario	Aquí se representan los servicios que ofrecen al usuario. También, en esta sección se describen los requerimientos no funcionales del sistema. Esta descripción puede usar lenguaje natural, diagramas u otras observaciones que sean comprensibles para los clientes. Deben especificarse los estándares de producto y proceso que tienen que seguirse.
Arquitectura del sistema	Este capítulo presenta un panorama de alto nivel de la arquitectura anticipada del sistema, que muestra la distribución de funciones a través de los módulos del sistema. Hay que destacar los componentes arquitectónicos que sean de reutilización.
Especificación de requerimientos del sistema	Debe representar los requerimientos funcionales y no funcionales con más detalle. Si es preciso, también pueden detallarse más los requerimientos no funcionales. Pueden definirse las interfaces a otros sistemas.
Modelos del sistema	Pueden incluir modelos gráficos del sistema que muestren las relaciones entre componentes del sistema, el sistema y su entorno. Ejemplos de posibles modelos son los modelos de objeto, modelos de flujo de datos o modelos de datos semánticos.
Evolución del sistema	Describe los supuestos fundamentales sobre los que se basa el sistema, y cualquier cambio anticipado debido a evolución de hardware, cambio en las necesidades del usuario, etc. Esta sección es útil para los diseñadores del sistema, pues los ayuda a evitar decisiones de diseño que restringirían probablemente futuros cambios al sistema.
Apéndices	Brindan información específica y detallada que se relaciona con la aplicación a desarrollar; por ejemplo, descripciones de hardware y bases de datos. Los requerimientos de hardware definen las configuraciones, mínima y óptima, del sistema. Los requerimientos de base de datos delimitan la organización lógica de los datos usados por el sistema y las relaciones entre datos.
Índice	Pueden incluirse en el documento varios índices. Así como un índice alfabético normal, uno de diagramas, un índice de funciones, etcétera.

Figura 1.4. Estructura de un documento de requerimientos

## 1.8 TRAZABILIDAD

Una especificación de un requerimiento de software es trazable si:

- (i) el origen de cada requerimiento está claro y
- (ii) se facilita la referencia de cada requerimiento en el desarrollo futuro o en la documentación.

La trazabilidad es la capacidad de describir y de seguir la vida de un requerimiento, tanto en dirección hacia adelante como hacia atrás, es decir, desde sus orígenes, a través de su desarrollo y

especificación, a su despliegue y uso subsecuente, y a través de períodos de refinamiento y de la iteración en curso en cualquiera de estas fases.

En [13] se describe la trazabilidad desde el origen del requerimiento hasta la implementación de este en el sistema y viceversa, es decir, hacia adelante y hacia atrás. Esto implica que un requerimiento debe de ser rastreable desde que se define y durante todo el desarrollo del software, lo cual garantiza una adecuada administración del cambio con el fin de evaluar el impacto en el resto del sistema.

En el caso que se esté en la etapa de desarrollo de los requerimientos, se podrá evaluar cómo afectaría un cambio de un requerimiento en otro. Por otro lado, estando en etapa de implementación y, en caso de que haya un cambio en los requerimientos, la trazabilidad permitirá hacer una evaluación en el diseño y en la implementación. Si el cambio se da cuando el sistema está implementado, la trazabilidad permitirá hacer una evaluación de cómo serán afectados los requerimientos involucrados [13].

## 1.9 PRIORIZACIÓN

La tarea de asignar prioridades requiere de la participación de clientes y usuarios con cierto nivel de decisión y puede realizarse de diversas maneras, tales como reuniones, cuestionarios y otras. Se deben tener en cuenta diferentes criterios como: la dependencia entre requerimientos, las limitaciones de recursos, las necesidades del negocio, las imposiciones del mercado y los costos de implementación, la volatilidad de los requerimientos, el riesgo para el negocio (por ausencia o por presencia de los servicios asociados al requerimiento) y factibilidad de implementación en el negocio; así como también la importancia relativa que tiene un requerimiento, y organizar aquellos requerimientos que deben implementarse inicialmente frente a aquellos que pueden postergarse [14][15].

La asignación de prioridades se basa en estrategias simples o complejas, que van desde una asignación que califica al requerimiento como importante o no, como obligatorio o postergable (suspendido) hasta un ranking de importancia de los requerimientos.

## 1.10 HERRAMIENTAS CASE

Elizabeth Vargas [16] en su trabajo final, cita a varios autores que definen a las herramientas CASE de diferentes formas:

- Son un complemento del conjunto de herramientas del ingeniero del software que le proporciona, la posibilidad de automatizar actividades manuales y de mejorar su visión general de la ingeniería. Las herramientas CASE ayudan a asegurar que la calidad sea un aspecto diseñado antes de llegar a construir el producto.
- CASE (Computer-Aided Systems Engineering) cuyo significado en español es Ingeniería del Software Asistida por Computadora, es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias del desarrollo de sistemas y, al igual que las herramientas CAD (Diseño Asistido por Computadora) o CAM (Manufactura Asistida por Computadora), su objetivo es acelerar el proceso para el que han sido diseñadas. En el caso

de CASE, para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

- La tecnología CASE corresponde a la ingeniería del software asistido por computadora. Es un conjunto de herramientas de software para automatizar las tareas del desarrollo del software. La idea es proporcionar un conjunto integrado de herramientas, que enlazan y automatizan todas las fases del ciclo de vida de software y su administración. Es una combinación de herramientas, software y metodología enfocada sobre el problema de producción de software.
- La tecnología CASE es la automatización del desarrollo de Software. La idea básica es proporcionar un conjunto de herramientas bien integradas, que ahorren trabajo enlazando y automatizando todas las fases del ciclo de vida del software.

Este tipo de herramientas se enfocan hacia el soporte de diferentes fases del ciclo de vida del Software o al desarrollo de diferentes tipos de sistemas. La disparidad de herramientas que se engloban dentro del concepto CASE hace necesaria una clasificación que nos permita realizar comparaciones entre las herramientas existentes. Por lo tanto, pueden ser clasificadas en tres categorías:

- **Juego de herramientas (Toolkit):** conjunto de herramientas integradas que automatizan un tipo de tarea del ciclo de vida del software. Constituyen el tipo más simple de herramientas.
- **Banco de trabajo (Workbench):** entorno de propósito general que soporta la totalidad de las tareas del ciclo de vida del desarrollo del software. Este tipo de herramientas se integran de forma que la salida de una fase del ciclo de vida pase directa y automáticamente a la siguiente.
- **Compañero de metodología (Methodology Companion):** es un tipo de herramienta CASE que estructura el proceso de desarrollo de software de acuerdo con los pasos y las reglas de una determinada metodología [17].

## 1.11 RESUMEN DEL CAPÍTULO

Mediante la utilización de métodos y herramientas la Ingeniería de Software ayuda a los analistas en los distintos aspectos de la resolución de problemas para la producción de software.

Los requerimientos reflejan las necesidades de los clientes, es decir, lo que ellos esperan que haga el sistema.

La especificación de requerimientos deber ser clara, completa y estar libre de ambigüedades. Por esto, 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 muchos casos, los fracasos en proyectos de software se deben a errores en los requerimientos. El costo de encontrar y corregir errores aumenta a medida que avanza el proyecto.

Otro factor relevante para tener en cuenta es la priorización de los requerimientos, la cual permite asignar a cada requerimiento un orden de importancia. Esta información ayuda a los analistas a determinar en qué orden deben ser implementados los distintos requerimientos.

La planificación de requerimientos es igual de importante que su especificación. Debido a la variabilidad de los requerimientos, es necesario llevar un estricto control de los cambios realizados. Así mismo, es deseable realizar un seguimiento desde la creación de cada requerimiento hasta su implementación y prueba. De esta forma, si se detecta un error de especificación, se podrá verificar rápidamente su origen.

Por último, otro aspecto importante durante el ciclo de vida de un proyecto es la comunicación entre los distintos stakeholders intervinientes. Si esta falla, aumenta la posibilidad de que se interpreten mal los distintos requerimientos y se produzcan errores. Por esto la comunicación entre ellos debe ser simple y fluida. Actualmente, gracias a la evolución de la tecnología y la aparición de las herramientas CASE es posible utilizar dispositivos informáticos como Tablets, Smartphones y PCs para resolver este tipo de problemas.

# 2. ANÁLISIS DE HERRAMIENTAS PARA LA GESTIÓN DE REQUERIMIENTOS

## 2.1 INTRODUCCIÓN

Los requerimientos 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 gestión de requerimientos debe realizar el seguimiento y control de cambios durante el ciclo de vida de un proyecto.

El presente capítulo muestra un listado de las funcionalidades deseables en una herramienta CASE para la gestión de requerimientos. Además, se estudia un conjunto de herramientas y se presenta una comparación entre éstas.

## 2.2 RELEVAMIENTO DE FUNCIONALIDADES DESEABLES

Con el pasar 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; añaden beneficios significativos al reducir errores que, a largo plazo, puedan perjudicar el correcto desarrollo de un proyecto.

Estas herramientas son las llamadas CASE que facilitan el trabajo de especificación, organización, almacenamiento y gestión de requerimientos.

En [3] determinaron las características que debería tener una herramienta para la gestión de requerimientos en base al documento ISO/IEC TR 24766: 2009. Este es un informe técnico que se centra en la evaluación de herramientas de Ingeniería de Software asistida por computadora que agrupa las capacidades establecidas en 6 categorías principales:

- Elicitación de requerimientos.
- Análisis de documentos.
- Especificación de requerimientos.
- Verificación y Validación de requerimientos.
- Manejo/ Administración de requerimientos.
- Otras capacidades (trazabilidad, análisis de trazabilidad, etc.).

Tomando como referencia las categorías definidas anteriormente, se evaluaron un conjunto de herramientas del ámbito comercial y académico. Como resultado de dicha evaluación 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 por los que ha transitado el mismo.
- **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 si están siendo modificados frecuentemente o si ya no se les realizarán cambios.
- **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 de archivos (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 posibilitar 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.

## 2.3 HERRAMIENTAS EXISTENTES

Actualmente existe en el mercado una variedad de herramientas comerciales que permiten llevar a cabo el proceso completo de gestión de requerimientos. Muchas de ellas están bien documentadas, brindan soporte a sus usuarios, y proveen la posibilidad de ampliar su funcionalidad mediante el pago de diferentes licencias.

Por otra parte, se encuentran las herramientas académicas las cuales no poseen documentación de acceso público y están realizadas dentro de alguna metodología propia de la organización desarrolladora, lo que dificulta su aprendizaje.

Asimismo, en muchos proyectos aún se sigue utilizando aplicaciones ofimáticas como Word y Excel debido al amplio conocimiento de su funcionamiento por parte de los usuarios y a su bajo costo con relación a las herramientas de gestión de requerimientos. Sin embargo, la productividad y

rendimiento que se puede conseguir con herramientas de gestión de requerimientos, difícilmente se logren con aplicaciones ofimáticas.

A partir de las características definidas en la sección anterior, se evaluaron siete herramientas del ámbito comercial (Visure Requirements [18], DOORSNG [19], Rommana [20], TraceCloud [21], Psoda [22], TrackStudio [23] y Redmine [24]) y tres del ámbito académico (REM [25], Heler [26] y Let's req! [27]).

### 2.3.1 VISURE REQUIREMENTS

Visure Requirements [18] es una herramienta equipada con diferentes módulos integrados entre sí para llevar a cabo la gestión de requerimientos. Permite definir requerimientos funcionales, no funcionales, establecer estado de avance, versiones, prioridad, nivel de completitud, llevar historial de cambios y definir relaciones entre ellos. El área de trabajo se presenta como interfaces de Office y las descripciones de estos pueden referenciar a documentos externos como pueden ser tablas, gráficos y hojas de cálculo de Microsoft Excel.

En cuanto a la usabilidad, presenta varias interfaces para las distintas actividades de la gestión de requerimientos. El manual de usuario es muy completo, y la empresa brinda a sus clientes diferentes vías de comunicación (email, teléfono, etc.) para darles soporte. Al ser una aplicación de escritorio no tiene la posibilidad de ser utilizada en diferentes dispositivos.

Con respecto al trabajo colaborativo, es posible llevarlo a cabo con los integrantes de una misma intranet, ya que el sistema se instala en servidores propios de la empresa cliente. Así dos analistas podrían acceder al mismo requerimiento simultáneamente y trabajar sobre él sin ninguna dificultad. El sistema realiza envío de emails ante cambios en los datos almacenados.

También brinda la posibilidad de definir usuarios y roles.

Visure Requirements, a pesar de ser una de las herramientas más completas del mercado presenta desventajas como: su licencia no es libre, la versión de prueba dura 30 días y luego hay que realizar un pago para adquirirla.

### 2.3.2 DOORSNG

Es un software que entre sus funcionalidades se encuentra la gestión de requerimientos. DOORSNG [19] no cuenta con un elemento "requerimiento" específico, sino que lo hace a partir de un objeto llamado "artefacto" el cual se utiliza para definir tanto requerimientos funcionales como no funcionales. Consta de un editor de texto enriquecido a partir del cual se crean los diferentes elementos que un analista necesite; además, es posible asignarles estados, prioridad y ver el historial de cambios.

Lo primero que se observa al ingresar en la herramienta es su compleja interfaz compuesta por múltiples elementos. Sus funcionalidades se encuentran divididas en diferentes módulos, con un panel para cada tipo de proyecto.

La herramienta cuenta con la posibilidad de asignar roles y permisos por usuario para un proyecto determinado.

Si bien es una aplicación web, no utiliza un modelo web adaptativo correctamente. En dispositivos como Tablet o Smartphones, el contenido se reduce por lo que no es posible navegar sin hacer un acercamiento, o desplazamiento de las ventanas.

A pesar de contar con amplia documentación, la gestión de requerimientos resulta difícil ya que cuenta con una gran cantidad de funcionalidades por lo que se requiere extensa experiencia con el sistema para manejarlas.

### 2.3.3 ROMMANA

Es un software para la gestión de requerimientos. Presenta diferentes versiones que permiten acceder a variadas funcionalidades para completar la gestión total de un proyecto de software. Es posible cargar requerimientos funcionales y no funcionales (los cuales a su vez se dividen en categorías tales como requerimiento de Corrección, Calidad, Portabilidad, Reusabilidad, etc.); asignar estados, prioridad, versiones y realizar comentarios. Además, permite crear atributos personalizados para asociarlos a los requerimientos y asociar requerimientos entre ellos para realizar trazabilidad.

Rommana [20] cuenta con una interfaz amigable, pero tiene la particularidad de generar ventanas emergentes cada vez que se quiere utilizar alguna funcionalidad. En el caso de que un usuario no tenga permisos para acceder a un proyecto, se genera una nueva ventana emergente, en la que se visualiza lo mismo que en la anterior. Esto dificulta su utilización y resulta confuso.

La herramienta cuenta con la posibilidad de agregar usuarios al grupo de trabajo de un proyecto. Ellos sólo pueden ser agregados por el líder de proyecto, el cual asigna permisos de lectura, creación y edición o borrado. Un usuario, dueño de un proyecto, puede crear ilimitados proyectos. Estos estarán visibles para todos los usuarios, pero solamente aquellos que formen parte del grupo de trabajo podrán acceder.

Los requerimientos almacenados permiten ver un historial de los cambios. Estos cambios son notificados a los integrantes del proyecto a través de emails en tiempo real.

También es posible anexar documentación y modelos de datos, pero sólo contratando el paquete que contiene dicha funcionalidad. La herramienta es muy completa, pero presenta la desventaja de que ofrece un período de prueba de 14 días. Luego hay que acceder a alguna de sus versiones pagas que se diferencian por las distintas funcionalidades según el paquete que se adquiera.

### 2.3.4 TRACECLOUD

Es un sistema web para gestionar proyectos que además permite realizar actividades relacionadas con la metodología ágil Scrum como pueden ser: definir sprint, daylis, notas de scrum y tablero de tareas.

Es posible definir diferentes tipos de requerimientos (de negocio, funcionales, test case, test result, defectos) estructurados en carpetas y subcarpetas anidadas. De cada uno se guarda un nombre y una descripción. Para esta última se muestra un editor de texto muy completo que permite dar formato a la información almacenada; es posible definir el dueño, la prioridad, indicar el porcentaje de completitud, adjuntar archivos y crear la matriz de trazabilidad.

La interfaz de TraceCloud [21] no está optimizada para diferentes dispositivos ya que los elementos se superponen y no es posible ingresar a opciones como el administrador de proyectos o al área de trabajo de Scrum en dispositivos pequeños. Por otro lado, los listados se encuentran paginados de 20 ítems por páginas lo que favorece su visualización.

El dueño de un proyecto puede agregar usuarios al mismo y asignarles permisos para permitirle realizar acciones como: agregar, editar, actualizar, borrar y aprobar. Estos permisos pueden ser asignados por carpeta dentro de un proyecto.

Es posible ver el historial de cambios de los requerimientos y enviarlos por email individualmente. También permite configurar el envío de emails por días y por proyectos; y realizar comentarios en los requerimientos.

La herramienta permite generar documentos en formato Word, Excel y PDF con la información almacenada en una carpeta. En el caso de querer la información de todas las carpetas, se pueden generar diferentes informes en formato Word y combinarlos.

Presenta mucha documentación integrada en el sistema brindando tutoriales a través de videos de YouTube, presentaciones y documentos de texto para orientar al usuario en el trabajo con la herramienta.

La versión de evaluación tiene una vigencia de 60 días, pasado este período ofrece dos posibilidades: una basada en usuarios (pagando un precio fijo por usuario según el rol que desempeñe) y otra por proyecto (es posible tener un número ilimitado de usuarios pero una cantidad limitada de requerimientos, dependiendo del monto que se pague). En la versión de evaluación, no es posible crear nuevos usuarios.

### 2.3.5 PSODA

Psoda [22] es un sistema compuesto por diferentes módulos dependiendo de la actividad que realice la empresa que lo adquiera. En esta oportunidad, se analizó el módulo de gestión de requerimientos. Éste permite almacenar requerimientos funcionales y no funcionales; definirles estados, prioridad y peso. Es posible almacenarlos en carpetas y subcarpetas para organizar la información.

Otros elementos que permite crear la aplicación son excepciones, evaluaciones, solicitudes de cambios, casos de prueba y defectos; así como también comentarios para cada elemento mencionado anteriormente. También es posible importar requerimientos desde archivos con formato CSV.

La herramienta permite generar diferentes reportes como listado de dependencia de requerimientos, progreso del proyecto, especificación de requerimientos agrupados por carpetas, etc., los cuales pueden ser exportados en formato doc, text, HTML, XML y PDF.

Psoda permite agregar ilimitados usuarios a un proyecto, asignar roles y grupos. Al ser una aplicación web, los usuarios que cuenten con datos de acceso pueden ingresar desde cualquier dispositivo conectado a internet.

Con respecto a la interfaz no adapta su contenido a dispositivos pequeños y además genera ventanas emergentes para las diferentes funcionalidades, ocasionando así dificultad en la navegabilidad.

Esta herramienta proporciona envío de reportes por email, los cuales pueden configurarse por día, mes, día de la semana y hora en que se quiere recibir el reporte.

La documentación es muy completa, brinda secciones de ayuda integradas en la herramienta y cuenta con videos tutoriales.

La licencia tiene una versión de evaluación por 30 días. Pasado ese período hay que adquirir una licencia paga.

### 2.3.6 TRACKSTUDIO

Para utilizar el sistema es necesario instalar un paquete de herramientas compuestos por un servidor web, un manejador de base de datos y el código de la herramienta. Una vez realizada la instalación, es necesario configurar la base de datos a través de TrackStudio Enterprise Server Manager.

El paquete de instalación está disponible para sistemas operativos GNU/Linux y Windows. Es posible instalarlo solamente en un servidor físico.

TrackStudio [23] permite crear requerimientos, asignarle estados, prioridad, presupuesto destinado, casos de prueba (previamente creados), adjuntar archivos y realizar comentarios. Los requerimientos están diferenciados con colores según el estado en que se encuentren (nuevo, aprobado, rechazado o pausado). Es posible agruparlos en carpetas y subcarpetas para organizar la información. Esto permite diferenciar requerimientos funcionales y no funcionales dependiendo de la carpeta en que se encuentre. Permite exportar la información almacenada en formato HTML, PDF, Excel, Word, CSV y XML.

Con relación a la interfaz no aplica la filosofía de diseño web adaptable dado que el tamaño del texto se reduce para entrar en la pantalla del dispositivo, por lo que hay que acercar la imagen para poder navegar.

Para cada proyecto es posible definir usuarios asignados, grupos y roles. Cada usuario puede tener distintos roles en diferentes proyectos.

La herramienta ofrece 5 tipos de licencias, dependiendo de la cantidad de usuarios que la organización adquirente necesite. Entre las licencias se ofrece la posibilidad de comprar el código fuente del sistema, lo que permitiría integrar otras aplicaciones para mejorar la funcionalidad de este.

Respecto a la documentación anexa, es posible adjuntar archivos tanto a proyectos como a requerimientos.

### 2.3.7 REDMINE

Redmine [24] es una herramienta web para la gestión de proyectos. Para utilizarlo es necesario realizar una instalación local o en algún servidor privado. Se debe contar con conocimientos previos ya que es necesario realizar configuraciones específicas para su correcto funcionamiento.

Los requerimientos funcionales y no funcionales se pueden diferenciar a través de la creación de tipos de tareas; es posible agregarle estados, prioridad, fecha de inicio y de fin, tiempos estimados en horas y usuario asignado. Además, permite adjuntar archivos a proyectos y peticiones para

complementar la documentación. La información almacenada puede ser exportada en formato Word o PDF.

La interfaz resulta muy intuitiva ya que las diferentes funcionalidades se muestran ordenadas y resulta fácil encontrarlas. Presenta diseño web adaptable, lo que permite acceder a la herramienta desde cualquier dispositivo.

Para favorecer el trabajo colaborativo cuenta con varios canales de comunicación como un portal de noticias, email de notificaciones por cambios, citas y foros de mensajes tanto en proyectos como en peticiones.

Es posible definir usuarios y roles diferentes para cada proyecto.

Permite generar reportes de la información almacenada, tanto en archivos de tipo PDF como CSV. Estos reportes tienen formatos de tablas predefinidas, presentados esencialmente como un listado.

### 2.3.8 REM

Es una herramienta CASE para la ingeniería de requerimientos creada como un prototipo de una tesis doctoral. Entre sus funcionalidades se encuentran la administración de requerimientos funcionales y no funcionales, definir objetivos y relacionarlos con éstos para poder determinar su trazabilidad. REM [25] permite agregar nivel de urgencia, estado de desarrollo y estabilidad a cada requerimiento; agregar autores, fuentes de información, versiones, comentarios, historial de cambios, entre otras funcionalidades con el fin de registrar un detalle más específico.

Con respecto a la navegabilidad de la herramienta, presenta una interfaz amigable y fácil de aprender; la documentación disponible es escasa, no tiene soporte y la última actualización oficial es de noviembre de 2004.

REM es una aplicación de escritorio, por lo que no es posible usarla en diferentes dispositivos como Tablet o Smartphones y sólo puede ser usada sobre sistemas operativos Windows; no permite trabajar colaborativamente ya que no es posible agregar otros usuarios a un proyecto (no maneja usuarios ni roles). Si varios usuarios quisieran acceder a una especificación de requerimientos deberían hacerlo a través de un archivo PDF previamente generado o un archivo propio (con extensión .rem) exportado desde la herramienta, el cual puede ser utilizado sólo con REM. Por esto, los analistas no tienen la posibilidad de estar al tanto de los cambios realizados por otras personas, lo que dificulta el trabajo en equipo y la comunicación.

Esta aplicación no brinda la posibilidad de adjuntar archivos generados de manera externa al sistema, por lo que no se podría complementar la documentación.

### 2.3.9 HELER

Es un software creado para dar soporte a la especificación de requerimientos en la fase de entendimiento del problema basado en el Proceso Unificado<sup>2</sup> (UP). Permite definir requerimientos

---

<sup>2</sup> El Proceso Unificado de Desarrollo de Software o simplemente Proceso Unificado es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura, y por ser iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational o simplemente RUP.

funcionales mediante Casos de Uso<sup>3</sup> (CU), requerimientos no funcionales, objetivos y stakeholders; es posible agregar comentarios a cada requerimiento y seguir su ciclo de vida.

En cuanto a la usabilidad, presenta una interfaz sencilla, la cual se ejecuta en ambientes Windows, por lo que no es posible acceder desde dispositivos con otros sistemas operativos.

Por ser una herramienta monousuario, no permite trabajar colaborativamente. Los analistas no podrían tener conocimiento de las modificaciones realizadas sobre los requerimientos, ni consultarlos. Por otro lado, permite imprimir la especificación de requerimientos basado en IEEE 830-1998 en formato PDF.

Heler [26] no permite adjuntar documentación generada externamente al sistema, pero es posible indicar en un cuadro de texto cuál es la fuente de información de un proyecto (como puede ser entrevistas, cuestionarios, etc.), asignarle un nombre y una descripción de dicha fuente.

Este software no maneja el concepto de roles ni permisos ya que no brinda la posibilidad de compartir un proyecto entre los usuarios.

### 2.3.10 LET'S REQ!

Es una herramienta web para la gestión de requerimientos. Permite definir requerimientos funcionales, no funcionales y de información (para almacenar información relativa al requerimiento); definir objetivos y relacionarlos con éstos.

Let's req! [27] presenta una interfaz amigable y fácil de aprender. Puede utilizarse en cualquier dispositivo que disponga de un navegador web. A pesar de esto, su diseño no se adapta completamente a ellos; el tamaño del texto se reduce para entrar en la pantalla del dispositivo, por lo que hay que acercar la imagen para poder navegar.

Es posible invitar a un número limitado de usuario, previamente registrados, para participar en un proyecto. No es posible asignar roles por lo que no se podría tener diferentes perfiles de usuarios, de esta forma todos los integrantes tendrían la posibilidad de realizar las mismas acciones. En su versión gratuita, no permite administrar más de un proyecto por usuario.

Cada funcionalidad posee una lista de revisiones que permite volver al estado en que se encontraba dicha funcionalidad en un momento determinado. Estos cambios son informados en el panel general, pero no presenta detalle de estos. No realiza notificaciones a los usuarios, por lo que éstos deberían entrar al sistema para estar informados de las acciones o cambios realizados por otros usuarios.

No permite cargar archivos previamente generados de manera externa a la herramienta.

## 2.4 ANÁLISIS COMPARATIVO

Del estudio de las herramientas mencionadas en la sección anterior se ha observado que ellas permiten diferenciar requerimientos funcionales y no funcionales, llevar un historial de cambios,

---

<sup>3</sup> Un Caso de Uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso.

insertar conversaciones o comentarios en los requerimientos para mantener informados a los analistas en caso de que exista alguna modificación o detalle que se deba tener en cuenta.

Lo que respecta a la claridad en las funcionalidades y facilidad de uso, las que se destacan son REM, Heler y Redmine ya que nos resultaron muy intuitivas y fáciles de aprender. Visure Requirements no es muy simple, pero brinda mucha documentación en formato PDF, videos tutoriales y soporte por medio de emails lo que facilita que el usuario aprenda a utilizar la herramienta. En los casos de PSODA y DOORSNG, la interfaz nos resultó compleja, difícil de aprender y de utilizar al momento de realizar la gestión de requerimientos.

Entre las diez herramientas analizadas, solamente Redmine cuenta con diseño web adaptable. Las restantes herramientas no cumplen con esta característica, ya sea por ser aplicaciones de escritorio como es el caso de REM, Visure Requirements y Heler o porque implementan la funcionalidad parcialmente, como Let's req!, Rommana, TraceCloud, Psoda y TrackStudio que adaptan algunos de sus contenidos (formularios, cuerpo principal, etc.), pero no otros (como el menú), o como DOORSNG que no adapta ningún contenido, lo que dificulta su uso en dispositivos móviles.

De las herramientas analizadas, DOORSNG, Let's req!, Rommana, TraceCloud, Psoda, TrackStudio y Redmine son multiplataforma ya que utilizan un navegador web como interfaz. El resto de las herramientas sólo funcionan bajo sistemas operativos Windows. Visure Requirements ofrece la posibilidad de acceder desde sistemas operativos Mac y Linux a través de aplicaciones de escritorio remoto.

En cuanto al trabajo colaborativo, REM y Heler no tiene esta funcionalidad. El resto lo cubren correctamente.

Con respecto al envío de notificaciones, las herramientas REM, Heler y Let's req! no lo implementan. Esto dificulta la comunicación entre los integrantes de un proyecto. El resto, realiza envío de emails para mantener al tanto de las modificaciones o actualizaciones a los diferentes usuarios del sistema.

Con respecto a adicionar documentación externa a la herramienta, Visure, Rommana, TraceCloud, Psoda, TrackStudio y Redmine permiten hacerlo, mientras que REM, Heler, DOORSNG y Let's req! no.

La asignación de roles y permisos está implementada en herramientas como Visure Requirements, DOORSNG Rommana, TraceCloud y Psoda en su versión paga. En el caso de TrackStudio y Redmine, es posible asignar roles y grupos ya que se accede al código fuente y se instala en servidores locales. Let's req!, REM, y Heler no hace diferencia entre los diferentes usuarios del proyecto.

Todas las herramientas permiten generar varios reportes a partir de la información almacenada, pero solo Heler permite exportar un reporte basado en IEEE Std. 830-1998.

Por último, una funcionalidad muy importante de una herramienta CASE para la gestión de requerimientos, es realizar trazabilidad entre estos. Menos TrackStudio todas realizan trazabilidad, ya sea con una matriz o con un listado.

Se realizó una comparación entre las herramientas evaluadas, cuyo resultado se muestra en la tabla 1.

Tabla 1: Comparación entre herramientas para la gestión de requerimientos.

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

✓= Cumple con la funcionalidad, - = No cumple

## 2.5 RESUMEN DEL CAPÍTULO

En [3] se determinaron las capacidades que debe tener una herramienta CASE para la gestión de requerimientos. En dicho trabajo se establece un total de 157 capacidades, las cuales fueron agrupadas en 6 categorías principales: Elicitación de requerimientos, Análisis de documentos, Especificación de documentos, Verificación y Validación de requerimientos, Manejo/ Administración de requerimientos y Otras capacidades.

Teniendo en cuenta estas categorías se analizaron las funcionalidades de siete herramientas comerciales y tres del ámbito académico. Luego se establecieron las capacidades mínimas y deseables que debería tener una herramienta de gestión de requerimientos.

A continuación, se realizó una comparación entre dichas herramientas y se observó que cumplen satisfactoriamente con la mayoría de las características necesarias, pero presentan inconvenientes como licencias pagas (en el caso de las comerciales) o poca documentación (en el caso de las académicas). Se pudo observar que ninguna de las herramientas estudiadas satisface la totalidad estas características.

A partir de esto se decidió realizar un sistema denominado SGR, Sistema para la Gestión de Requerimiento que reúna todas las características deseables que debe tener un sistema para la etapa de elicitación de requerimientos.

# 3. SGR: SISTEMA PARA LA GESTIÓN DE REQUERIMIENTOS

## 3.1 INTRODUCCIÓN

Como se detalló en el capítulo anterior, las herramientas analizadas no cumplieron con el total de las funcionalidades que se consideran más importantes, y que son necesarias para llevar a cabo una gestión de requerimientos exitosa.

A partir de esto, surgió la necesidad de implementar un sistema de software que reúna todas las características necesarias para llevar a cabo la gestión de requerimientos de forma correcta y completa. Para definir los requerimientos del sistema, se escogieron las funcionalidades que las herramientas ya resuelven satisfactoriamente y se hizo hincapié en las funcionalidades omitidas o en las que se encontraron falencias.

En este capítulo se presenta el alcance, propósito y funcionalidades de nuestra propuesta, denominada SGR (Sistema para la Gestión de Requerimientos).

## 3.2 ALCANCE

El principal objetivo de SGR es brindar asistencia en la etapa de elicitación y gestión de requerimientos, minimizando errores de interpretación y facilitando el mantenimiento de éstos. El software brindará un ambiente colaborativo donde líderes de proyectos, analistas funcionales y usuarios finales (entre otros stakeholders) podrán participar en la elicitación y comunicarse entre ellos.

Cabe destacar que la etapa de elicitación de requerimientos es importante en el ciclo de vida del proyecto ya que permite comprobar si se alcanzaron los objetivos del proyecto y las necesidades del cliente.

## 3.3 PROPÓSITO

Manejar gran cantidad de requerimientos puede ser una tarea compleja. Por esto, el propósito del sistema es dar soporte al proyecto durante la etapa de gestión de requerimientos con el fin de permitir el análisis de objetivos, avances y desviaciones en cuanto a tiempos del proyecto.

Debido a que los requerimientos son el hilo conductor de un proyecto de software, es esencial que se lleve a cabo todo el proceso de manera ágil y completa. Con el uso del sistema será posible reducir los errores a través de la participación de los clientes, mejorar la comunicación interna y el trabajo colaborativo.

## 3.4 CARACTERÍSTICAS DE SGR

### 3.4.1 USUARIOS, ROLES Y PERMISOS

Almacenar y mantener la información perteneciente a la actividad de gestión de requerimientos involucra, generalmente, a varias personas que se relacionan con el proyecto de diferentes formas. Por esto, el sistema propuesto diferencia entre diversos grupos de usuarios.

Para brindar privacidad de los datos almacenados, los usuarios que quieran operar con SGR, se deberán autenticar. Un usuario puede ser asignado a uno o más proyectos con diferentes grupos. Los grupos estarán compuestos por permisos que determinan las funcionalidades que un usuario, tendrá permitido realizar. Cada usuario autenticado podrá realizar las tareas definidas para el grupo al que esté asignado en un proyecto determinado.

Los grupos definidos serán:

- **Super administrador:** tendrá la posibilidad de acceder a todas las funcionalidades del sistema. Podrá crear, editar y eliminar usuarios; asignarles grupos y roles, y tendrá acceso a todos los proyectos almacenados en el sistema.
- **Administrador de proyecto:** podrá crear proyectos y realizar todas las funciones relacionadas con estos.
- **Administrador de requerimientos:** podrá consultar los datos de un proyecto y realizar todas las funcionalidades relacionadas con los requerimientos funcionales y no funcionales.
- **Invitado:** sólo podrá consultar los proyectos y exportar documentos.
- **Cliente:** podrá consultar proyectos, aprobar o rechazar requerimientos que estén en estado *Validación* y realizar comentario sobre estos.

Todos los permisos de un grupo pueden ser modificados por un usuario perteneciente al grupo administrador. De esta manera, se flexibilizan las acciones permitidas por cada grupo.

### 3.4.2 STAKEHOLDERS

Un stakeholder es una persona que influye directa o indirectamente en la gestión de requerimientos. Al cargar un proyecto o un requerimiento, se podrá establecer al stakeholder como la fuente de la información.

### 3.4.3 ESTADOS

A partir de las herramientas analizadas, se definieron dos listados con los estados por los que pueden pasar tanto los proyectos como los requerimientos a lo largo de la etapa de elicitación de requerimientos.

Para los requerimientos funcionales y no funcionales:

- **Desarrollo:** en el que caso de que el requerimiento esté siendo elaborado por los analistas.
- **Validación:** el requerimiento está completo, sin confirmación del cliente.
- **Terminado:** el requerimiento está verificado por el cliente.
- **Cancelado:** para el caso en que no se haya comprendido un requerimiento, o simplemente, no es lo que el cliente requiere.

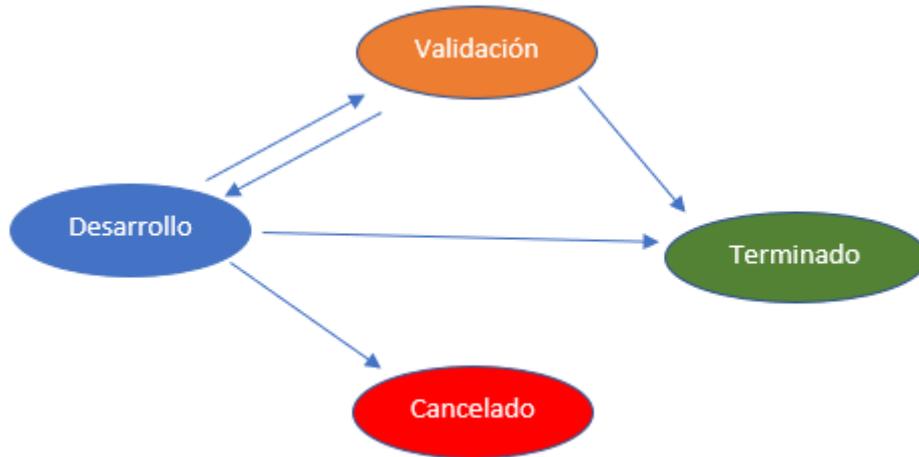


Figura 3.1. Flujo de estados esperado para un requerimiento

Cada requerimiento contará con un solo estado en un momento dado.

En el caso del proyecto, los estados se definirán como:

- **Activo:** para aquellos que se encuentran vigentes.
- **Suspendido:** no se puede continuar por un tiempo limitado a raíz de algún conflicto en los requerimientos u otros motivos ajenos a estos.
- **Finalizado:** se concluyó exitosamente el proyecto.
- **Cancelado:** el proyecto no se continuará y quedará inconcluso.

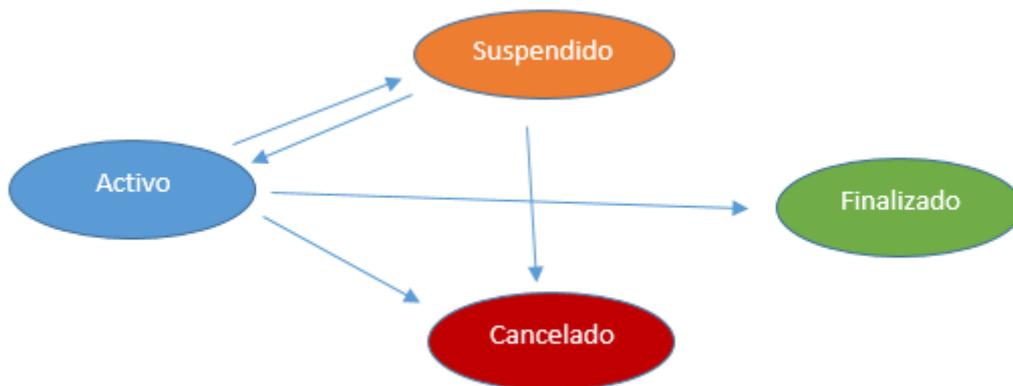


Figura 3.2. Flujo de estados esperado para un proyecto

Todos los estados anteriormente definidos se mostrarán diferenciándolos con colores asociados para determinar su importancia, lo que permitirá al usuario reconocer cada uno simplemente.

#### 3.4.4 PRIORIDAD

Además de las funcionalidades mencionadas anteriormente, el sistema permitirá asignar prioridad a los requerimientos funcionales y no funcionales.

El estándar IEEE 830-1998 recomienda la "Asignación Numérica": este es un método de fácil uso, el cual se basa en clasificar los requerimientos en tres categorías: "Esenciales", "Condicionales",

"Opcionales". De esta manera, se indaga a los stakeholders respecto de la "importancia percibida" sobre los mismos.

### 3.4.5 ESTABILIDAD

El sistema permitirá asignar Estabilidad, lo que indicará si el requerimiento está siendo modificado frecuentemente o si ya no sufre cambios.

La estabilidad se definirá como:

- Alta
- Media
- Baja

### 3.4.6 NOTIFICACIONES DE CAMBIOS

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.

Las notificaciones del Tablero del usuario muestran dos listados: uno con los proyectos que se encuentran atrasados (ya superaron su fecha de fin estimada) y otro con los próximos a su fecha de fin estimada (aquellos que su fecha de fin estimada se encuentra entre los 7 días siguientes a la fecha actual).

Además, cuenta con una sección donde se muestran los proyectos que fueron actualizados luego de la última sesión del usuario y durante la misma. Con esta funcionalidad, el usuario puede acceder en todo momento a ver estas actualizaciones. Para cada proyecto, requerimiento funcional, no funcional, objetivo, usuario asignado, adjunto, atributo y stakeholder se puede ver un enlace para acceder el ítem directamente.

Las notificaciones por email se generarán ante dos eventos: cambio de estado de un requerimiento por parte de un cliente o por la actualización en un proyecto.

Un usuario perteneciente a un proyecto recibirá un email cuando un usuario perteneciente al grupo *Cliente* apruebe o rechace un requerimiento en estado *Validación*. Esto permitirá a todos los miembros de un proyecto, estar al tanto en tiempo real, de los cambios que sufre un requerimiento.

Un usuario integrante de un proyecto recibirá un email periódicamente si alguno de los proyectos sufre alguna actualización luego de su último inicio de sesión. Los datos incluidos en estos serán los proyectos, requerimiento funcional, no funcional, objetivo, usuario asignado, adjunto, atributo y stakeholder.

### 3.4.7 INFORMACIÓN ADJUNTA

El sistema brinda la posibilidad de adjuntar archivos en diversos formatos que complementan la documentación del proyecto. Es posible agregar documentos de texto, como minutas de reunión,

entrevistas, cuestionarios, imágenes de modelo conceptual, mockups<sup>4</sup> de las interfaces, etc. También es posible anexar documentos a cada requerimiento.

Esta funcionalidad enriquece la documentación del proyecto, ya que además de la información cargada en el sistema, también se puede relacionar toda la documentación generada externamente manteniendo la centralización.

### 3.4.8 GENERACIÓN AUTOMÁTICA DEL DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS BASADO EN IEEE STD. 830-1998

Además de almacenar la información en el sistema, es importante que los usuarios puedan disponer de ella sin necesidad de recurrir a la aplicación. Por esto se provee la funcionalidad de generar y exportar documentos de especificación de requerimientos SRS basado en el IEEE Std. 830-1998. Este tipo de documento permite presentar una descripción completa del comportamiento del sistema que se va a realizar, en un lenguaje informal, de forma que sea fácil de entender por todos los involucrados.

### 3.4.9 TRAZABILIDAD

El sistema permitirá asignar a cada requerimiento los requerimientos funcionales y no funcionales dependientes. A partir de estas asignaciones se generará automáticamente la matriz de trazabilidad, lo que permitirá determinar de manera gráfica qué requerimientos se verán afectados al momento de realizar un cambio.

## 3.5 CARACTERÍSTICAS NO FUNCIONALES DE SGR

### 3.5.1 INTERFAZ DE USUARIO

El sistema deberá tener una interfaz amigable e intuitiva que permita a los usuarios interactuar con ella invirtiendo su tiempo en la definición del proyecto y sus requerimientos y no en comprender su usabilidad.

### 3.5.2 PORTABILIDAD

SGR será desarrollado en un lenguaje web, por lo tanto, será accesible desde cualquier dispositivo con acceso a un navegador web e internet sin realizar configuración ni instalación previa.

### 3.5.3 WEB ADAPTABLE

Deberá ser posible acceder al sistema desde diferentes dispositivos. El contenido visualizado se debe adaptar al tamaño de cada dispositivo para brindarle al usuario una mejor experiencia de uso.

---

<sup>4</sup> Mockup: Un mockup, o maqueta, es un modelo a escala o tamaño real de un diseño o un dispositivo; utilizado para la demostración, evaluación del diseño, promoción, y para otros fines.

### 3.5.4 DISPONIBILIDAD

El sistema deberá estar disponible de forma continua para ser accedida desde distintos dispositivos.

### 3.5.5 MANTENIBILIDAD

El código del sistema deberá estar estructurado de forma consistente y predecible para que su funcionamiento pueda ser mejorado en el futuro.

### 3.5.6 CONCURRENCIA

El sistema deberá soportar múltiples usuarios conectados al mismo tiempo sin reducir su rendimiento ni producir inconsistencia en los datos.

## 3.6 CARACTERÍSTICAS FUNCIONALES DE SGR

### 3.6.1 USUARIOS

El sistema permitirá crear, editar y eliminar usuarios (solo en los casos en que no exista información relacionada) con los siguientes datos:

- Nombre
- Apellido
- Nombre de usuario
- Contraseña
- Email
- Activo (se utiliza para determinar si el usuario está activo o no, pudiendo deshabilitarlos sin borrarlo)
- Salt (para cifrar la contraseña)
- Borrado
- Grupo (o grupos) al que pertenece

### 3.6.2 ROLES

El sistema permitirá la visualización de roles de usuario con los siguientes datos:

- Nombre
- Descripción
- Rol (o roles) que componen el grupo

### 3.6.3 GRUPOS

El sistema permitirá crear, editar y eliminar grupos de usuarios con los siguientes datos:

- Nombre
- Descripción

### 3.6.4 ESTADOS

El sistema permitirá la visualización de los estados de proyectos y requerimientos con los siguientes datos:

- Nombre
- Descripción
- Abreviatura

Los únicos estados permitidos serán: Desarrollo, Validación, Terminado, Cancelado, Activo, Suspendido, Finalizado y Cancelado.

### 3.6.5 ESTABILIDADES

El sistema permitirá crear, editar y eliminar estabilidades (sólo en los casos en que no esté relacionado con un requerimiento) con los siguientes datos:

- Nombre
- Descripción
- Abreviatura

### 3.6.6 INSTITUCIONES

El sistema permitirá crear, editar y eliminar instituciones (sólo si no existen stakeholders asociados a dicha institución) con los siguientes datos:

- Nombre
- Descripción
- Dirección
- Teléfono
- Email

### 3.6.7 PROYECTOS

El sistema permitirá crear, editar y eliminar proyectos con los siguientes datos:

- Nombre
- Resumen
- Descripción
- Alcance
- Propósito
- Fecha de inicio estimada
- Fecha de fin estimada
- Fecha de inicio real
- Fecha de fin real
- Estado
- Listado de versiones
- Listado de conversaciones
- Listado de requerimientos funcionales

- Listado de requerimientos no funcionales
- Listado de objetivos
- Listado de usuarios asignados
- Listado de archivos adjuntos
- Listado de atributos
- Listado de stakeholders
- Listado de requerimientos funcionales eliminados
- Listado de requerimientos no funcionales eliminados

En el caso de eliminar un proyecto, el sistema mostrará todos los elementos que se eliminarán y requerirá que se ingrese el motivo por el cual se realiza esta acción. Una vez aceptado, se borrarán todos los datos relacionados sin la posibilidad de revertirlo.

El listado de proyectos podrá ser filtrado por nombre y estado.

### 3.6.8 USUARIOS CON GRUPOS ASIGNADOS A PROYECTO

El sistema permitirá crear y eliminar usuarios con un grupo a un proyecto con los siguientes datos:

- Usuario
- Grupo

Solo se podrán asignar usuarios y grupos previamente creados en el sistema.

Esto permitirá al usuario realizar las acciones que le correspondan según el grupo al que pertenece en ese proyecto.

El listado de usuarios asignados podrá ser filtrado por usuario y grupo.

### 3.6.9 STAKEHOLDERS DE UN PROYECTO

El sistema permitirá crear, editar y eliminar stakeholders de un proyecto con los siguientes datos:

- Nombre
- Apellido
- Puesto que ocupa
- Institución a la que pertenece
- Proyecto (será asignado automáticamente dependiendo del proyecto en el que esté)

El listado de stakeholders podrá ser filtrado por nombre y apellido.

### 3.6.10 HISTORIAL DE CAMBIOS DE UN PROYECTO

El sistema permitirá consultar el historial de cambios de un proyecto. Se mostrarán qué campos fueron modificados y permitirá realizar comparaciones entre dos versiones.

### 3.6.11 OBJETIVOS DE UN PROYECTO

El sistema permitirá crear, editar y eliminar los objetivos de un proyecto con los siguientes datos:

- Nombre
- Descripción

- Proyecto (será asignado automáticamente dependiendo del proyecto en el que esté)

El listado de objetivos podrá ser filtrado por nombre y descripción.

### 3.6.12 ATRIBUTOS DE UN PROYECTO

El sistema permitirá crear, editar y eliminar atributos de un proyecto con los siguientes datos:

- Nivel
- Subnivel
- Nombre
- Descripción
- Proyecto (será asignado automáticamente dependiendo del proyecto en el que esté).

El sistema no permitirá crear dos atributos con los mismos datos para un mismo proyecto.

El listado de atributos podrá ser filtrado por nombre, nivel y subnivel.

### 3.6.13 GESTIÓN DE REQUERIMIENTOS

El sistema permitirá crear, editar y eliminar requerimientos funcionales y no funcionales con los siguientes datos:

- Nombre
- Descripción
- Usuario (será asignado automáticamente el usuario que realice la acción)
- Proyecto (será asignado automáticamente dependiendo del proyecto en el que esté)
- Estado
- Estabilidad
- Prioridad
- Listado de archivos adjuntos
- Listado de conversaciones
- Listado de versiones
- Listado de stakeholders
- Listado de objetivos
- Listado de dependencias funcionales
- Listado de dependencias no funcionales

En el caso de eliminar un requerimiento, el sistema requerirá que se ingrese un motivo y se borren todos los datos relacionados sin la posibilidad de revertirlo.

El listado de requerimientos podrá ser filtrado por nombre, estado y prioridad.

### 3.6.14 HISTORIAL DE CAMBIOS DE UN REQUERIMIENTO

El sistema permitirá consultar el historial de cambios de requerimientos funcionales y no funcionales. Se mostrarán qué campos fueron modificados y permitirá realizar comparaciones entre dos versiones.

### 3.6.15 STAKEHOLDERS DE UN REQUERIMIENTO

El sistema permitirá crear y eliminar asignaciones de stakeholders a requerimientos funcionales y no funcionales.

Solo se podrán asignar los stakeholders previamente creados en un proyecto.

El listado de stakeholders podrá ser filtrado por apellido y nombre.

### 3.6.16 OBJETIVOS DE UN REQUERIMIENTO

El sistema permitirá crear y eliminar asignaciones de objetivos a requerimientos funcionales y no funcionales.

Solo se podrán asignar los objetivos previamente creados en un proyecto.

El listado de objetivos podrá ser filtrado por nombre.

### 3.6.17 APROBAR/RECHAZAR UN REQUERIMIENTO

El sistema permitirá que un usuario, con permisos para realizar esta acción, acepte o rechace un requerimiento en estado *Validación*. Una vez que sea marcado como aceptado o rechazado, el estado del requerimiento deberá cambiar a *Terminado o Desarrollo* respectivamente.

### 3.6.18 ARCHIVOS ADJUNTOS

El sistema permitirá crear, editar y eliminar archivos a proyectos y requerimientos con los siguientes datos:

- Archivo
- Nombre
- Descripción
- Proyecto (será asignado automáticamente dependiendo del proyecto en el que esté).
- Requerimiento Funcional (será asignado automáticamente dependiendo del requerimiento en el que esté).
- Requerimiento No Funcional (será asignado automáticamente dependiendo del requerimiento en el que esté).

Un adjunto podrá pertenecer solo a un proyecto, requerimiento funcional o requerimiento no funcional al mismo tiempo.

El listado de archivos adjuntos podrá ser filtrado por nombre y descripción.

### 3.6.19 VERSIONES

El sistema permitirá agregar y modificar (solo la última) la versión de proyectos y requerimientos con los siguientes datos:

- Número mayor
- Número menor
- Fecha

- Proyecto (será asignado automáticamente dependiendo del proyecto en el que esté).
- Requerimiento Funcional (será asignado automáticamente dependiendo del requerimiento en el que esté).
- Requerimiento No Funcional (será asignado automáticamente dependiendo del requerimiento en el que esté).

La fecha de una versión debe ser mayor a la última cargada.

Los números mayor y menor de una versión no podrán ser menor al último cargado.

Una versión podrá pertenecer solo a un proyecto, requerimiento funcional o requerimiento no funcional al mismo tiempo.

El sistema permitirá consultar el historial de las versiones que se crearon para un proyecto o requerimiento.

### 3.6.20 CONVERSACIONES

El sistema permitirá crear conversaciones para un proyecto o requerimiento con los siguientes datos:

- Título
- Usuario que lo realizó
- Listado de comentarios
- Fecha y hora de creación
- Proyecto (será asignado automáticamente dependiendo del proyecto en el que esté).
- Requerimiento Funcional (será asignado automáticamente dependiendo del requerimiento en el que esté).
- Requerimiento No Funcional (será asignado automáticamente dependiendo del requerimiento en el que esté).

Una conversación podrá pertenecer solo a un proyecto, requerimiento funcional o requerimiento no funcional al mismo tiempo.

### 3.6.21 COMENTARIOS

El sistema permitirá crear comentarios para una conversación con los siguientes datos:

- Mensaje
- Conversación (será asignada automáticamente de la conversación en el que esté)
- Fecha y hora de creación.

### 3.6.22 TABLERO DE USUARIOS

El sistema contará con una pantalla en la se mostrará la siguiente información:

- Proyectos atrasados: son aquellos que pasaron su fecha de fin estimada y no poseen fecha de fin real. De cada proyecto se mostrará nombre, fecha de fin estimada y un enlace a los datos del proyecto.

- **Proyectos próximos a concluir:** son aquellos cuya fecha estimada de finalización se cumplirá en los próximos siete días. De cada proyecto se mostrará nombre, fecha de fin estimada y un enlace a los datos del proyecto.
- **Últimas actualizaciones:** son aquellos proyectos que fueron modificados luego del último inicio de sesión del usuario y durante la sesión actual. De cada proyecto se visualizará su nombre y un listado con las actualizaciones ocurridas en él. Cada actualización contendrá un enlace para acceder a los datos.

### 3.6.23 ENVÍO DE NOTIFICACIONES

El sistema realizará envío automático de notificaciones por email por dos tipos de eventos.

Uno será ante cambios en los requerimientos funcionales, requerimientos no funcionales, objetivos, usuarios asignados, adjuntos, atributos o stakeholder a aquellos usuarios que pertenezcan a proyectos que hayan sido actualizados luego de la última sesión del usuario. Estas notificaciones serán enviadas una vez al día.

El otro será ante el cambio de estado de un requerimiento funcional o no funcional al ser aceptado o rechazado. Estas notificaciones serán enviadas luego de que se produzca el cambio.

### 3.6.24 REPORTE BASADO EN SRS ESTÁNDAR IEEE Std. 830-1998

El sistema permitirá generar un reporte basado en SRS estándar IEEE Std. 830-1998 con los siguientes datos:

- Nombre del proyecto
- Versión actual del proyecto
- Fecha
- Propósito
- Alcance
- Descripción del proyecto
- Atributos
- Requerimientos funcionales
  - Nombre
  - Descripción
- Requerimientos no funcionales
  - Nombre
  - Descripción

Dicho reporte contendrá una portada con el nombre del proyecto, la versión y la fecha, y un índice con hipervínculos dentro del mismo documento.

## 3.7 RESUMEN DEL CAPÍTULO

Al momento de desarrollar un sistema, es importante especificar lo que debe hacer y las cualidades que debe exhibir como performance, portabilidad, disponibilidad y seguridad.

SGR permitirá gestionar proyectos, los cuales estarán compuestos por requerimientos funcionales, no funcionales, objetivos, usuarios asignados, archivos adjuntos y stakeholders.

Los requerimientos funcionales serán almacenados con el mayor detalle posible, con el fin de generar documentos completos y consistentes.

Para poder acceder al sistema se deberá contar con un nombre de usuario y contraseña generados por un usuario administrador. SGR no permitirá que una persona se registre por sí misma. Cada usuario tendrá grupos y permisos que lo habilitarán a realizar acciones específicas. El sistema permitirá asociar, como fuente de información, a los distintos stakeholder de un proyecto y/o de un requerimiento.

Tanto los requerimientos como los proyectos tendrán asignado un estado en un momento determinado. Además, los requerimientos tendrán asignados prioridad y estabilidad. Estas características enriquecen la especificación de cada requerimiento.

Se podrán realizar filtros y establecer órdenes para mejorar la visualización de los datos.

Para contribuir al trabajo colaborativo, SGR realizará, de forma automática, el envío de notificaciones a través del correo electrónico y del Tablero del usuario logrando así mantener actualizados a todos los participantes de un proyecto.

También será posible realizar el seguimiento de los cambios en los requerimientos a través de un historial de cambios.

El sistema permitirá adjuntar información anexa tanto a nivel de proyecto como de requerimiento.

Por último, será posible generar de manera automática y exportar en formato PDF la especificación de requerimientos para un proyecto.

SGR contará 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 fácil y sin grandes dificultades. Además, el sistema deberá ser multiplataforma, lo que permitirá que pueda utilizarse independientemente del sistema operativo que se elija. De esta manera, se amplía su utilización para cualquier usuario que posea un dispositivo con acceso a un navegador web y conexión a internet. Juntamente con esto, se determinó que el sistema deberá tener un diseño web adaptable, para cumplir con los objetivos de una interfaz amigable y de fácil utilización en cualquier dispositivo electrónico como smartphone, Tablet, notebook, etc.

# 4. DISEÑO

## 4.1 INTRODUCCIÓN

El diseño es un proceso comprendido por múltiples etapas en el que se detalla la representación de los datos y la estructura del sistema. Es el proceso creativo del problema en una solución. Una vez que se analizan y especifican los requerimientos, el diseño es la primera actividad técnica por realizar.

El diseño de software agrupa el conjunto de principios, conceptos y prácticas que llevan al desarrollo de un sistema o producto de alta calidad. Es la única manera de traducir con exactitud un a producto o sistema terminado los requerimientos. Sin diseño se corre el riesgo de obtener un sistema inestable, que falle cuando se hagan cambios pequeños o que sea difícil de someter a pruebas [6].

Este capítulo presenta el diseño conceptual y el modelo de clase realizado para el desarrollo del sistema. Asimismo se muestra el diseño de las interfaces de SGR en versiones para PC y para dispositivos móviles.

## 4.2 MODELO CONCEPTUAL

Antes de comenzar el desarrollo de SGR, se llevó a cabo el modelo conceptual (Figura 4.1) que contiene las entidades, los atributos y las relaciones necesarias para representar toda la información que el sistema requiere.

Los modelos conceptuales son medios para representar la información de un problema con un alto nivel de abstracción. Están compuesto por entidades, relaciones y atributos.

- Las entidades representan elementos u objetos del mundo real con identidad.
- Las relaciones representan agregaciones entre dos o más entidades. Describen las dependencias o asociaciones entre dichas entidades.
- Los atributos representan una propiedad básica de una entidad o relación. [28]

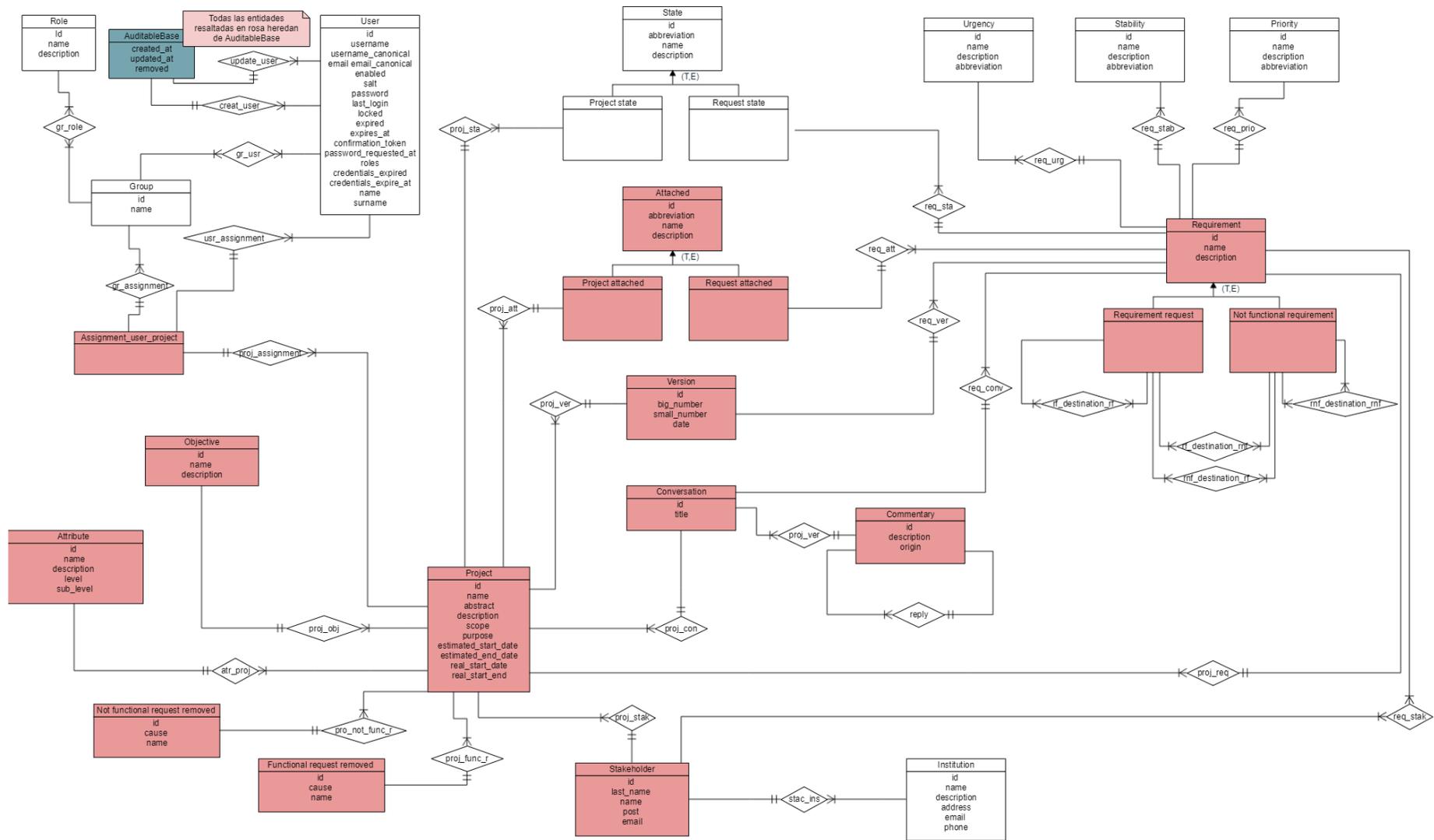


Figura 4.1. Modelo conceptual de SGR

## 4.3 DIAGRAMA DE CLASES

Luego de realizar el modelo conceptual, elaboramos un diagrama de clases (Figura 4.2) que muestra las clases del sistema y las relaciones que existen entre estas. Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Una relación es una conexión semántica entre objetos, en otras palabras, provee un camino de comunicación entre ellos [29].

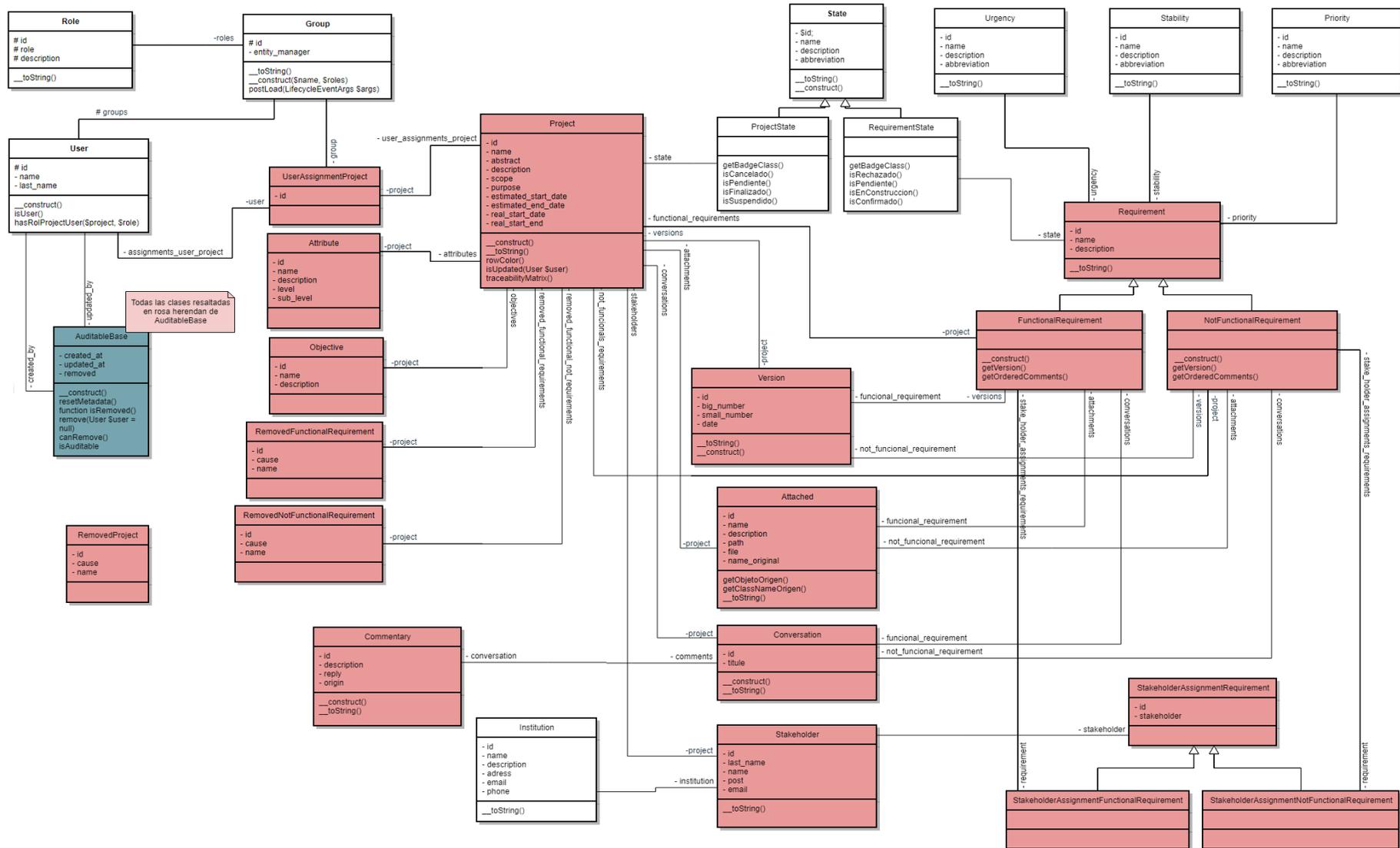


Figura 4.2. Diagrama de clases de SGR

## 4.4 MOCKUPS

Una interfaz difícil de utilizar provoca que los usuarios cometan errores o incluso que se rehúsen a utilizar el sistema. La interfaz debe contribuir a que el usuario consiga un rápido acceso al contenido de sistemas complejos, sin pérdida de la comprensión mientras se desplaza a través de la información. Mediante el uso de mockups se diseñó la forma en la que se va a navegar por el sistema y cómo se disponen los elementos que componen las diferentes páginas. Cada una respeta la misma interfaz para que el usuario pueda navegar el sistema de forma intuitiva.

### 4.4.1 LOGIN

En el diseño de la interfaz para el inicio de sesión se presenta el logo del sistema y el correspondiente formulario que permite la autenticación (Figuras 4.3 y 4.4).

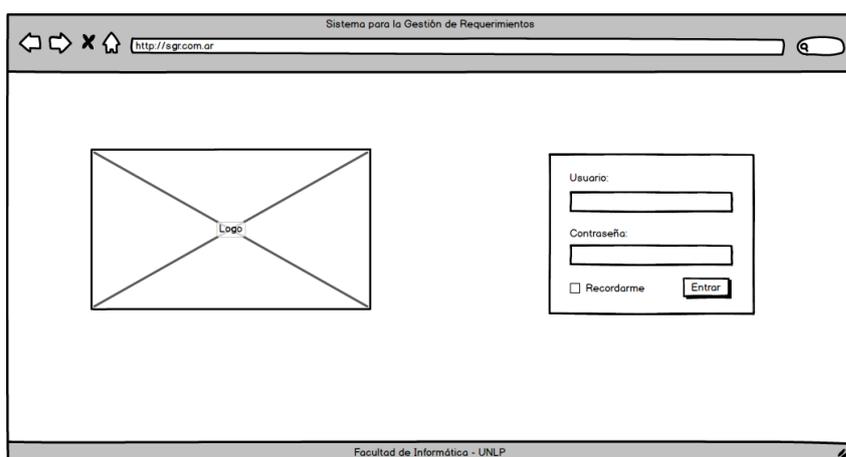


Figura 4.3. Inicio de sesión del sistema (PC)

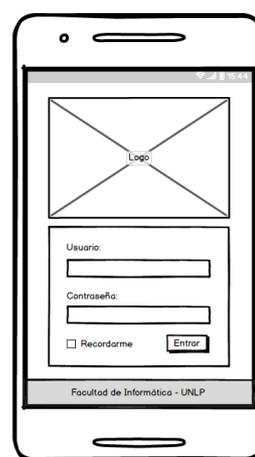


Figura 4.4. Inicio de sesión del sistema (Dispositivo móvil)

### 4.4.2 DISEÑO GENERAL

SGR cuenta con un diseño general para todas las interfaces compuesto por: un área superior o encabezado, el contenido y un área inferior o pie de página (Figuras 4.5 y 4.6).

El área superior contiene: el logo del sistema, un menú de navegación que le permite al usuario acceder a las distintas funcionalidades del sistema y un botón desplegable con las opciones *cerrar sesión* y *cambiar la contraseña*.

El área del contenido varía depende del elemento que se esté visualizando.

El área inferior o pie de página contiene un texto centrado.



Figura 4.5. Diseño de interfaz de SGR (PC)

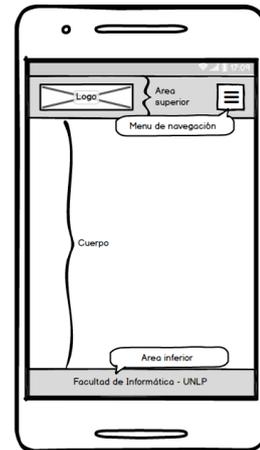


Figura 4.6. Diseño de interfaz de SGR (Dispositivo móvil)

#### 4.4.3 PANTALLA DE INICIO

La interfaz del Tablero de usuario (Figuras 4.7 y 4.8) muestra las actualizaciones de los proyectos desde la última sesión.

En el contenido de la pantalla se muestran los siguientes datos:

- **Listado de proyectos atrasados:** muestra un listado de proyectos cuya fecha estimada de finalización es anterior a la fecha actual.
- **Listado de proyectos próximos a concluir:** muestra un listado de los proyectos cuya fecha estimada de finalización se cumple en los próximos siete días.
- **Listado de las últimas actualizaciones:** muestra un listado con las últimas actualizaciones de los proyectos en los que es miembro el usuario.

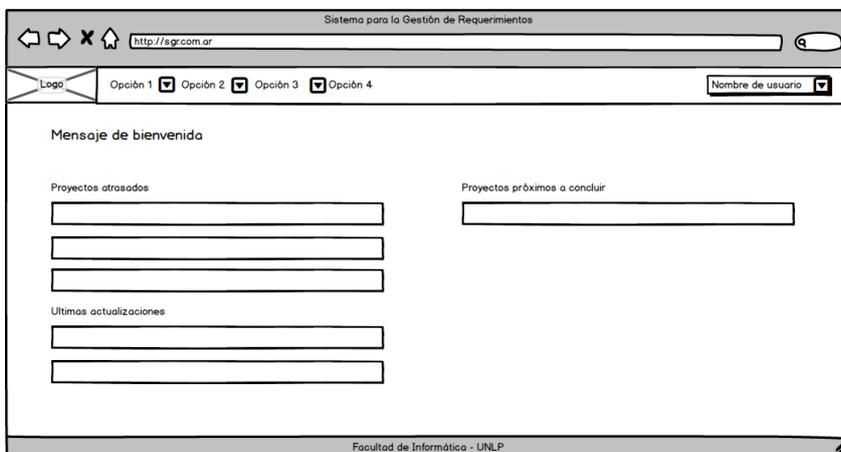


Figura 4.7. Tablero de usuario (PC)

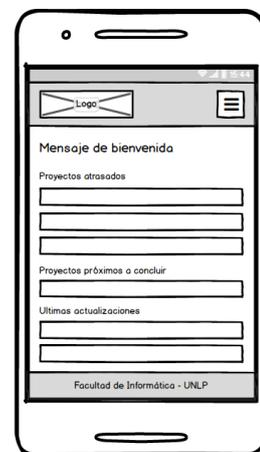


Figura 4.8. Tablero de usuario (Dispositivo móvil)

#### 4.4.4 LISTADO DE ELEMENTOS

Un listado de elementos (Figuras 4.9 y 4.10) muestra un área que permite al usuario filtrar la información por distintos criterios. Las opciones de filtrado dependen de la información de cada elemento. Por ejemplo, los proyectos se pueden filtrar por nombre y estado, mientras que los archivos adjuntos se pueden filtrar por nombre, descripción y nombre de archivo.

Los elementos se muestran en formato tabular con la información relevante para cada caso y una columna de acciones con los íconos correspondientes a las funcionalidades disponibles para cada elemento.

Al final de la tabla, se muestran los siguientes datos:

- Del lado izquierdo, muestra la cantidad de elementos que contiene el listado
- Del lado derecho, se muestra un paginador que permite navegar los resultados. El primer botón permite ir a la página 1. El segundo, permite ir a la siguiente de la actual. Luego habrá de 1 a 5 botones (dependiendo de la cantidad de elementos). El ante último botón permite ir a la página siguiente de la actual. Y, por último, un botón para ir a la página final.

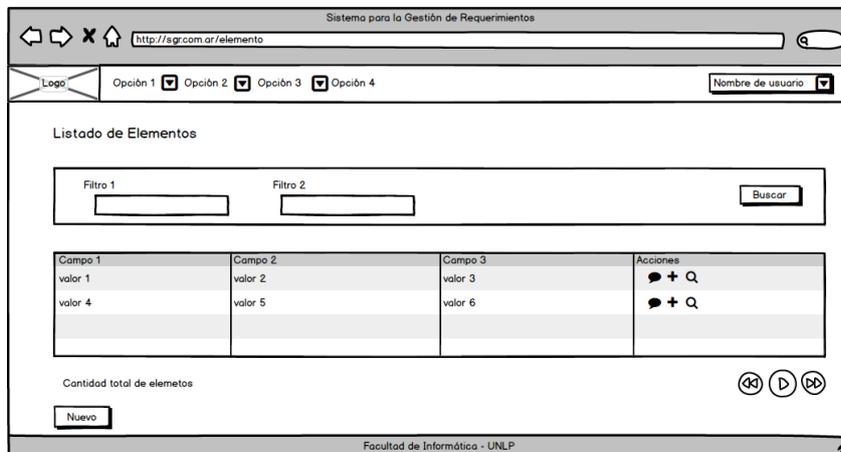


Figura 4.9. Listado de elementos (PC)

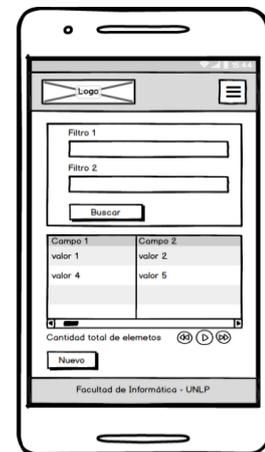


Figura 4.10. Listado de elementos (Dispositivo móvil)

#### 4.4.5 ALTA DE ELEMENTO

Para realizar el alta de un elemento (Figuras 4.11 y 4.12), se visualizan todos los campos de información requeridos para realizar la carga del elemento y dos botones *Guardar* y *Volver*. La opción *Guardar*, confirmará los datos cargados, mientras que la opción *Volver*, cancelará la operación y retornará al listado correspondiente al tipo de elemento que se quiso crear.

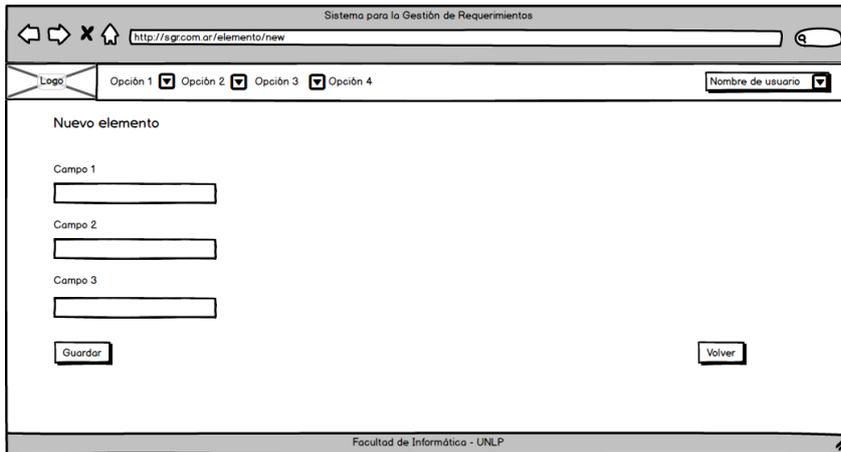


Figura 4.11. Alta de un elemento (PC)

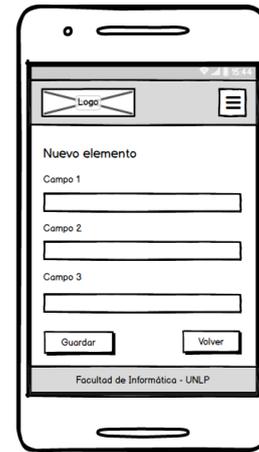


Figura 4.12. Alta de un elemento (Dispositivo móvil)

#### 4.4.6 EDICIÓN DE ELEMENTO

Para realizar la edición de un elemento, se muestra una pantalla similar a la del alta con los datos precargados (Figuras 4.13 y 4.14) en donde el usuario podrá modificar la información almacenada. Debajo del formulario se encuentran dos botones *Guardar* y *Volver*. La opción *Guardar*, confirmará los datos cargados, mientras que la opción *Volver*, cancelará la operación y retornará al listado correspondiente al tipo de elemento que se quiso crear.

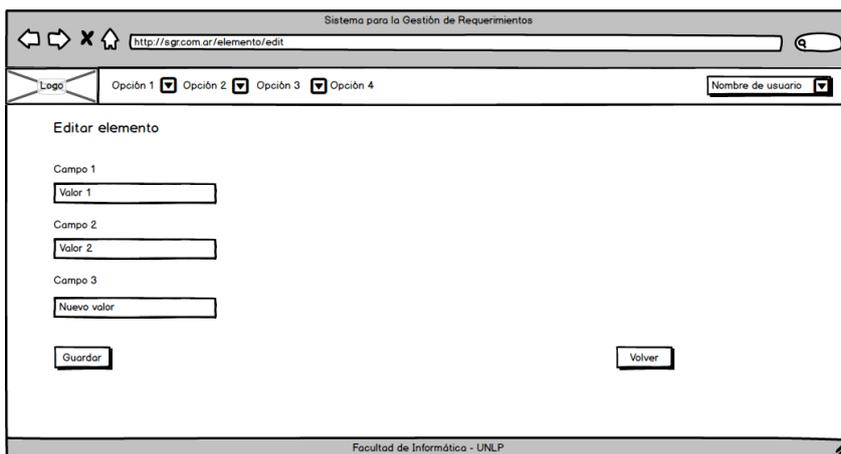


Figura 4.13. Edición de un elemento (PC)

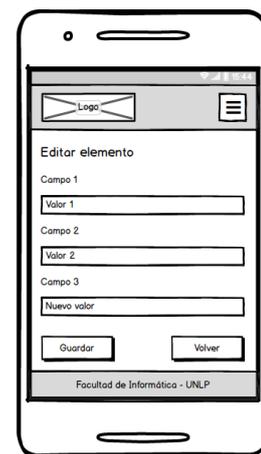


Figura 4.14. Edición de un elemento (Dispositivo móvil)

#### 4.4.7 DETALLE DE ELEMENTO

La pantalla que permite visualizar el detalle de un elemento se diseña de forma tabular (Figuras 4.15 y 4.16). Debajo de la tabla se muestran tres botones: *Editar*, *Eliminar* y *Volver*. El primero permite ir a la pantalla de edición del elemento, el segundo muestra una ventana de confirmación para la eliminación del elemento que se está visualizando y el tercero permite volver al listado de elementos.

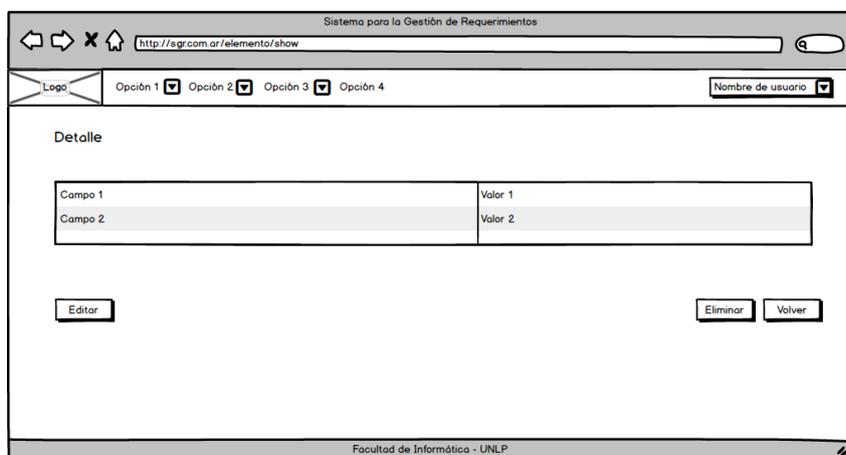


Figura 4.15. Detalle de un elemento (PC)

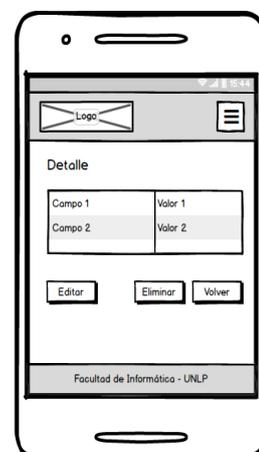


Figura 4.16. Detalle de un elemento (Dispositivo móvil)

#### 4.4.8 DETALLE DE PROYECTOS

En la pantalla de detalle de un proyecto encontramos los siguientes elementos:

- **Botón de edición de proyecto y botón de historial:** el primero permite ir a la pantalla de edición de un proyecto, mientras que el segundo muestra el historial de cambios realizados para ese proyecto.
- **Mensaje:** muestra el estado actual del proyecto.
- **Barra de porcentaje de avance de requerimientos:** muestra el porcentaje de requerimientos funcionales y no funcionales que se encuentran en estado finalizado.
- **Panel de elementos vinculados a un proyecto:** visualiza dividido en pestañas, los elementos relacionados con un proyecto. Estos pueden ser requerimientos funcionales, requerimientos no funcionales, objetivos, archivos adjuntos, usuarios asignados, stakeholder, etc.
- **Botón Eliminar Proyecto y botón Volver:** Eliminar muestra una ventana de confirmación para eliminar el proyecto actual y Volver retorna al listado de proyectos.

Las figuras 4.17 y 4.18 muestran el detalle de un proyecto.

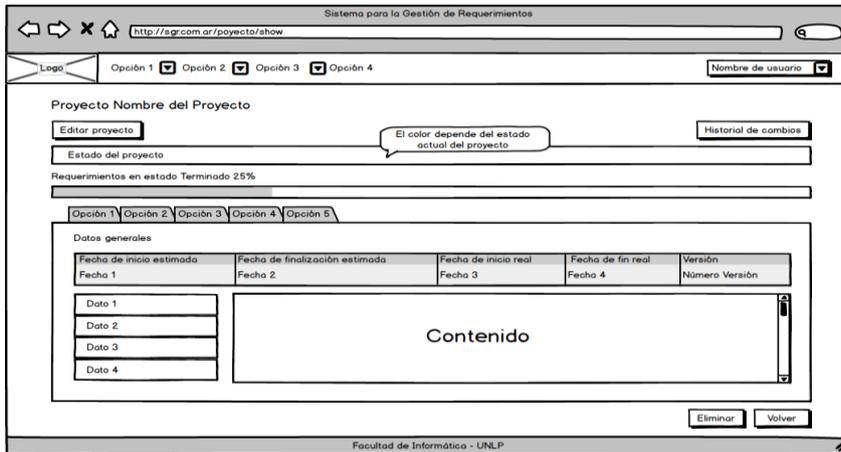


Figura 4.17. Detalle de un proyecto (PC)

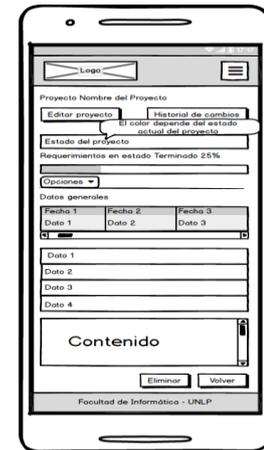


Figura 4.18. Detalle de un proyecto (Dispositivo móvil)

#### 4.4.9 DETALLE DE REQUERIMIENTOS

En la pantalla de detalle de un requerimiento funcional o no funcional se encuentran los siguientes elementos:

- **Botón de edición de requerimiento y botón de historial:** el primero permite ir a la pantalla de edición de un requerimiento, mientras que el segundo muestra el historial de cambios realizados para el requerimiento actual.
- **Panel:** contiene todos los elementos relacionados a un requerimiento.
- **Botón Eliminar requerimiento y botón Volver:** el botón “Eliminar” muestra una ventana de confirmación para eliminar el requerimiento actual y el botón “Volver” retorna al detalle del proyecto.

Las figuras 4.19 y 4.20 muestran el detalle de un requerimiento.

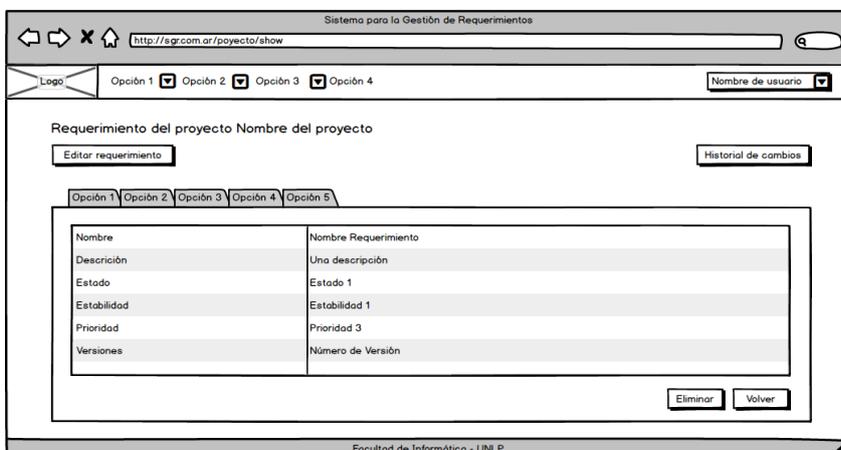


Figura 4.19. Detalle de un requerimiento (PC)



Figura 4.20. Detalle de un requerimiento (Dispositivo móvil)

## 4.5 RESUMEN DEL CAPÍTULO

En primer lugar, se presentó el modelo conceptual, el cual es una representación gráfica del dominio del problema, a través de entidades y relaciones.

Luego, se realizó el diagrama de clases para establecer las clases y sus relaciones necesarias para el desarrollo de SGR.

Por último, se definieron los mockups del sistema. Estos definen las posiciones en donde se ubican los elementos que componen cada pantalla. En base a estos elementos de diseño, se implementaron las diferentes interfaces del sistema.

# 5. IMPLEMENTACIÓN

## 5.1 INTRODUCCIÓN

La arquitectura de software es un conjunto de métricas, técnicas y patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo al equipo responsable del desarrollo de un proyecto de software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Un aspecto importante en los grandes sistemas de software es su estructura base, es decir, qué elementos la componen, cómo se organizan y se relacionan entre ellos [30].

A continuación, se detallarán los componentes que se utilizaron para el desarrollo de SGR y su arquitectura.

## 5.2 TECNOLOGÍA UTILIZADA

### 5.2.1 PHP

PHP [31] (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser embebido en HTML. Tiene la ventaja de ser extremadamente simple para usuarios principiantes, pero sumamente potentes para usuarios avanzados. También es posible usar la versión de línea de comando PHP, llamada PHP-CLI (Command Line Interface). Entre sus características se destaca que permite utilizar excepciones, técnicas de programación orientada a objetos y posee la capacidad de ampliar su funcionalidad a través de módulos. También se destaca por su popularidad, posee conexión con la mayoría de las bases de datos (destacando su conectividad con MySQL y PostgreSQL), gran cantidad de documentación en su sitio web oficial (en la cual se ejemplifica y detalla el funcionamiento de todas las funciones) y es libre, por lo que es una alternativa de fácil acceso.

Por su flexibilidad ha tenido gran repercusión como lenguaje base para aplicaciones web. Grandes sistemas fueron realizados utilizando PHP como Facebook y WordPress.

### 5.2.2 FRAMEWORK SYMFONY

Un framework es un conjunto de componentes reutilizables que puede ser utilizado para producir aplicaciones concretas y específicas. Los objetivos principales son: agilizar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas.

Symfony [32] es un completo framework diseñado para optimizar el desarrollo de aplicaciones web. Separa la lógica del negocio (modelo), la lógica del servidor (controlador) y la presentación de la aplicación web (vista). Proporciona herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web. Además, automatiza las tareas más comunes permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

A través de la herramienta Composer<sup>5</sup>[33], creada por miembros de la comunidad Symfony es posible instalar y gestionar dependencias de las aplicaciones PHP, lo que facilita el trabajo del desarrollador.

La decisión de utilizar Symfony para desarrollar SGR se debió a que es un framework maduro, con amplia y completa documentación, y además cuenta con una gran comunidad de colaboradores que lo apoyan. La versión elegida fue la 2.8 ya que, al momento de comenzar el desarrollo, era la versión estable y con mayor soporte a largo plazo (hasta noviembre de 2019). Gracias a los Bundles<sup>6</sup> desarrollados por esta comunidad, es posible agregar nuevas funcionalidades al sistema de manera ágil y flexible. Es compatible con la mayor parte de gestores de bases de datos y se puede ejecutar tanto en plataformas Windows como Unix, Linux, etc.

### 5.2.3 DOCTRINE

Doctrine [34] es una librería centrada en proporcionar persistencia y funcionalidades relacionadas. Una de las características más importantes de Doctrine es la posibilidad de escribir consultas de bases de datos en Doctrine Query Language (DQL), un lenguaje de consultas para modelos orientados a objetos.

Se utilizó esta librería ya que se integra de manera completa en Symfony y es una herramienta poderosa para interactuar con bases de datos.

### 5.2.4 TWIG

Twig [35] es un motor de creación de plantillas. Brinda una solución al tratamiento de las cuestiones visuales alrededor de una aplicación desarrollada en PHP. Facilita la implementación de las vistas, haciendo el trabajo más simple y ordenado. Permite separar la lógica del diseño, lo que minimiza la complejidad. Permite flexibilidad en relación con el HTML, CSS y JavaScript, sin necesidad de saber o aprender PHP.

### 5.2.5 BASE DE DATOS MYSQL

MySQL [36] es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos libre más popular del mundo, sobre todo para entornos de desarrollo web. Sus características más importantes son la seguridad (a través de la utilización de usuario y contraseña), la portabilidad (es posible utilizarlo en Linux, Mac OS X, Solaris, Windows, etc.) y su fácil utilización, entre otras.

---

<sup>5</sup> Composer: manejador de dependencias basado en PHP utilizado para descargar, instalar y actualizar librerías PHP.

<sup>6</sup> Bundle: es similar a un complemento en otro software. La diferencia clave es que todo es un bundle en Symfony, incluido el código del framework.

## 5.2.6 BOOTSTRAP

Bootstrap [37] es un framework o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web frontend. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales. El framework está desarrollado para facilitar el desarrollo de sitios web adaptables. Una característica importante es que brinda compatibilidad con todos los navegadores web modernos en diferentes plataformas.

El framework fue elegido por su fácil aprendizaje y su gran comunidad de desarrollo y soporte; cuenta con ejemplos claros, simples y reutilizables.

## 5.2.7 JQUERY

jQuery [38] es una librería multiplataforma de JavaScript [39]. Es software libre y de código abierto. Posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta librería se logran grandes resultados en menos tiempo y espacio.

SGR almacenará los requerimientos funcionales y no funcionales que se releven para un proyecto, así como también objetivos, archivos adjuntos, stakeholders, entre otras cosas. Teniendo en cuenta esto, es probable que se generen extensos listados de información; con el objetivo de mejorar su utilización y visualización es necesario incorporar la función de paginación que se encarga de mostrar la información en fragmentos, denominados páginas. Para esto se utilizó una herramienta conocida como paginador.

## 5.2.8 KNPPAGINATORBUNDLE

KNPPAGINATORBUNDLE [40] es un bundle de KNPLABS. Es posible integrarlo a nuestro proyecto de manera sencilla. Puede ser renderizado <sup>7</sup>y extendido con plantillas Twig. Cuenta con la posibilidad de redefinir todas las plantillas, cantidad de ítems por páginas y cantidad de páginas que se muestran a la vez (Figura 5.1).



Figura 5.1. Formato elegido para el paginador

Este bundle también cuenta con la posibilidad de realizar ordenaciones de la información listada a través de links en los encabezados de las columnas (Figura 5.2).

Nombre	Abreviatura ▲	Descripción	Acciones
--------	---------------	-------------	----------

Figura 5.2. Encabezado de tabla ordenado por Abreviatura Ascendente

<sup>7</sup> Renderizar: una renderización es una imagen final creada o generada por ecuaciones o cálculos matemáticos que realiza una computadora en un programa determinado.

### 5.2.9 KNPSNAPPYBUNDLE

KnpsnappyBundle [41] permite generar archivos PDF o de imagen directamente desde documento HTML. Mediante un archivo de configuración es posible agregar elementos como portada, número de página, una tabla de contenidos, etc. Su integración con Symfony es simple, lo que facilita su utilización.

### 5.2.10 SWIFTMAILER

Una funcionalidad importante de SGR es el envío de notificaciones a los integrantes de un proyecto para mantenerlos actualizados sobre nuevos cambios, para esto se utilizó SWIFTMAILER [42].

SWIFTMAILER es un componente para el envío de emails en aplicaciones PHP. Se eligió esta componente debido a que brinda una solución simple y flexible, orientada a objetos con múltiples características, entre ellas: configuración del medio de envío (SMTP, mail, sendmail o gmail), host, puerto, usuario y contraseña de cuenta de envío, encriptación. Permite adjuntar archivos, insertar imágenes y realizar prácticamente las mismas funciones que se realizan con un cliente de correo. Permite utilizar Twig para maquetar emails. Esto facilita su utilización ya que es el mismo motor de plantillas utilizado por Symfony.

A través de un bundle llamado SwiftmailerBundle es posible integrar SWIFTMAILER al sistema de manera sencilla.

### 5.2.11 ENTITYAUDIT

ENTITYAUDITBUNDLE [43] Es una extensión para Doctrine que permite realizar versiones completas de entidades y sus asociaciones.

Esta extensión crea una tabla idéntica a la original para cada entidad auditable, las cuales tienen el sufijo "\_audit". Además de todas las columnas, agrega dos campos más para indicar el número de revisión a partir de una tabla de "revisiones" y un campo revtype que indica si se trata de una actualización, inserción o borrado

La tabla "revisions" almacena que usuario realizó la revisión y cuando.

Con estos datos, es posible determinar en qué momento se produjeron los cambios en el sistema, lo que permite llevar un control más estricto de las acciones realizadas sobre los requerimientos de un proyecto.

### 5.2.12 GLIFFY

Gliffy [44] es una aplicación online para crear y compartir diagramas. Funciona a través de un navegador web sin la necesidad de instalar el software. Permite almacenar, realizar copias de documentos y crear una gran variedad de diagramas. Permite el trabajo colaborativo entre varios usuarios, los que pueden interactuar sobre un determinado diagrama.

### 5.2.13 GITLAB

Debido a que el sistema fue desarrollado de manera colaborativa, fue necesario utilizar un sistema de control de versiones. El control de versiones es un sistema que registra cambios en un archivo o conjunto de archivos a lo largo del tiempo para que se pueda recuperar versiones específicas más tarde.

Dentro de las herramientas para el control de versiones se cuenta con una gran variedad, pero se optó por GitLab [45] ya que proporciona repositorios privados con una interfaz cómoda y amigable.

GitLab es un sistema de control de versiones distribuidas de código abierto y gratuito diseñado para manejar proyectos grandes y pequeños. A través de una cuenta de usuario es posible crear los proyectos deseados, crear tareas, asignar a usuarios a esas tareas, entre otras cosas.

## 5.3 DISEÑO ARQUITECTÓNICO

SGR un sistema web, por lo que utiliza HTTP. HTTP (HyperText Transfer Protocol) define como los clientes web solicitan páginas a los servidores web y como estos servidores transfieren las páginas web a los clientes. HTTP se implementa en dos programas: un programa cliente y uno servidor. Tanto el programa cliente como el servidor se ejecutan en sistemas terminales diferentes y se comunican entre sí intercambiando mensajes HTTP [46].

La figura 5.3 muestra el flujo de información desde el acceso a una página web hasta que se muestra en el navegador.

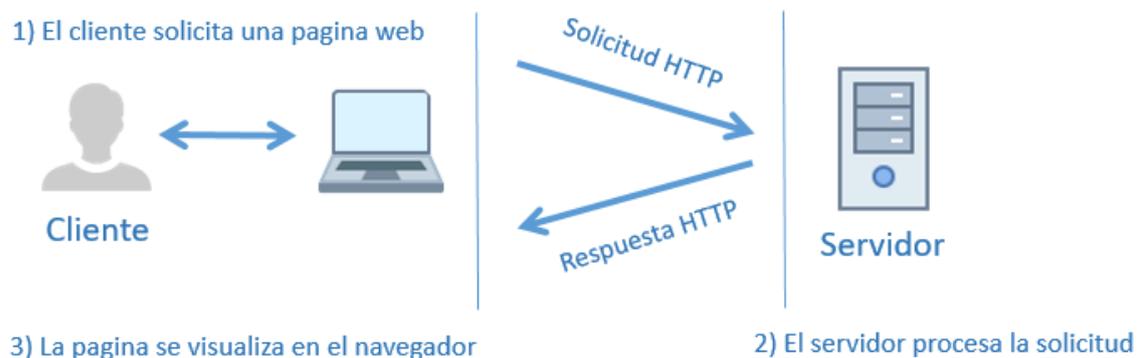


Figura 5.3. Flujo HTTP

La arquitectura del software representa un diseño de alto nivel del sistema. Define de manera abstracta los componentes que llevan a cabo alguna tarea, sus interfaces y la comunicación entre ellos. La figura 5.4 muestra la arquitectura definida para el desarrollo de SGR.

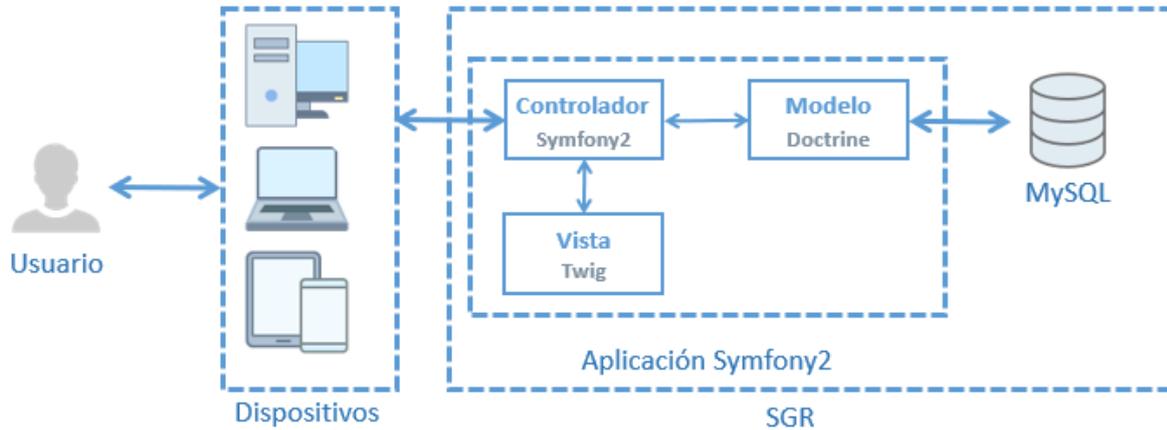


Figura 5.4. Arquitectura de SGR

## 5.4 RESUMEN DEL CAPÍTULO

En este capítulo se describió la arquitectura y las herramientas utilizadas en el desarrollo de SGR.

Se eligió utilizar PHP, un lenguaje de programación web simple para usuarios principiantes, y sumamente potente para usuarios expertos, posee gran cantidad de documentación y es libre.

Para simplificar el trabajo, se decidió utilizar el framework Symfony en su versión 2.8, el cual se basa en PHP, es multiplataforma y es compatible con la mayor parte de los gestores de bases de datos. Este framework está diseñado para agilizar el desarrollo de aplicaciones web, posee buena y amplia documentación, y cuenta con una gran comunidad que lo apoya.

Con respecto a la gestión de base de datos, se utilizó Doctrine ya que tiene como objetivo brindar herramientas que permitan realizar consultas e interacciones de manera fácil y flexible.

En cuanto al manejo de las vistas, se decidió utilizar el motor de creación de plantillas Twig ya que permite trabajar de manera simple y ordenada y es de fácil integración con el framework elegido. Symfony fue complementado con Bootstrap y jQuery para realizar la interfaz de usuario de manera que sea web adaptable y amigable.

Además, se utilizaron una serie de herramientas que facilitaron la implementación del sistema. Por un lado, se utilizó un conjunto de bundles como KnpPaginator (realiza ordenaciones sobre tablas y pagina resultados), KnpSnappyBundle (genera archivos en formato PDF junto con su tabla de contenido), EntityAudit (almacena historiales de cambios de aquellas entidades configuradas para esto). Por otro lado, se usó un componente basado en PHP llamado Swiftmailer, para el envío de notificaciones. Ésta es una funcionalidad muy importante para favorecer el trabajo colaborativo y la comunicación entre analistas, desarrolladores y clientes.

Como herramienta de soporte para el desarrollo del sistema se utilizó GitLab como servicio web de control de versiones.

Por último, se utilizó la aplicación web Gliffy para la elaboración del diagrama de clases y el modelo conceptual.

# 6. CASO EXPERIMENTAL: ALBERGUE UNIVERSITARIO DE LA UNLP

## 6.1 INTRODUCCIÓN

En los últimos meses, el Instituto de Investigación en Informática LIDI (III-LIDI) utiliza SGR para llevar a cabo la gestión de requerimientos de un proyecto de software para el 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 Nacional de La Plata. La problemática presentada se origina 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 se estableció que el sistema deberá cumplir los siguientes objetivos:

- Automatizar el proceso de inscripción a cada comida
- Registrar el proceso de retiro de comida de cada alumno
- Generar reportes con la información almacenada
- Determinar restricciones del sistema

En el presente capítulo se mostrará el proceso que debe llevar a cabo un analista para realizar la gestión de requerimientos mediante el uso de SGR, utilizando como caso de estudio el proyecto del Albergue Universitario UNLP.

## 6.2 GESTIÓN DE USUARIOS Y GRUPOS

Todos los participantes de un proyecto deben tener un usuario para poder ingresar al sistema.

Debido a que no todos los usuarios realizan las mismas funciones en los proyectos, SGR permite la creación de grupos que sirven para limitar las funcionalidades a las que puede acceder cada uno. Los usuarios pueden pertenecer a un grupo por proyecto.

A través de la opción *Configuración* del menú, SGR permite la administración de usuarios (Figura 6.1) y grupos (Figura 6.2).

Para el proyecto del Albergue Universitario de la UNLP se crearon los siguientes usuarios:

- un usuario con perfil de líder de proyecto
- un usuario con perfil de analista
- y otro con perfil de cliente

Usuario	Nombre	Apellido	Email	Activo	Acciones
super_admin	Super	Admin	admin@admin	SI	🔍 ✎
lider_proyecto	Lider de proyecto	Lider	lider@lider	SI	🔍 ✎
cliente	Cliente	Cliente	invitado@invitado	SI	🔍 ✎
analista	Analista	analista	analista@analista	SI	🔍 ✎

Resultados: 4

+ Nuevo

Figura 6.1. Listado de usuarios

Nombre	Acciones
Administrador	🔍 ✎
Desarrollador	🔍 ✎
Analista	🔍 ✎
Cleinte	🔍 ✎
Administrador de proyecto	🔍 ✎

Resultados: 5

+ Nuevo

Figura 6.2. Listado de grupos

## 6.3 GESTIÓN DE PROYECTOS

SGR permite que los usuarios puedan ver los proyectos en los que participan (Figura 6.3). Dependiendo del grupo al que pertenece un usuario, el sistema le permitirá realizar distintas acciones. Estas acciones pueden ser: ver y editar los datos del proyecto; administrar versiones, requerimientos funcionales, no funcionales, objetivos, usuarios asignados, archivos adjuntos y stakeholders.

Nombre	Fecha de inicio estimada	Fecha de fin estimada	Fecha de inicio real	Fecha de fin real	Estado	Acciones
Albergue Universitario UNLP	16/04/2018	16/10/2018			Activo	🔍 ✎ 🗨️ 📄

Resultados: 1

+ Nuevo

Finalizado Atrasado Fecha de fin estimada en menos de 7 días

Figura 6.3. Listado de proyectos

Luego del análisis del problema y de la información obtenida, se procedió a realizar la carga de los datos del proyecto. Para esto, el usuario líder de proyecto perteneciente al grupo Administrador de proyecto creó el proyecto llamado Albergue Universitario UNLP. En este se cargó la descripción del

proyecto, el propósito del sistema, el alcance del documento, su fecha de inicio real y fecha de fin estimada (Figura 6.4).

Datos Proyecto

Nombre  
Albergue Universitario UNLP

Descripción

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 permitirle al llevar un control cuantitativo de las inscripciones y retiros de comida que se realizan.

Resumen

Este documento establece la Especificación de Requerimientos del Sistema de Software a desarrollar para el comedor del albergue de la UNLP. Este documento sirve como marco contractual; se describen los requerimientos funcionales del Sistema de Software para la Gestión del Comedor.

Propósito

Este documento establece la Especificación de Requerimientos del Sistema de Software a desarrollar para el comedor del albergue de la UNLP. Este documento sirve como marco contractual; se describen los requerimientos funcionales del Sistema de Software para la Gestión del Comedor.

Alcance

El sistema permitirá gestionar el uso del comedor del Albergue.  
No existen supuestos que no estén indicados en este documento.

Fecha de inicio estimada: 16/04/2018  
Fecha de fin estimada: 16/10/2018

Fecha de inicio real:   
Fecha de fin real:   
Estado: Activo

Guardar Volver

Facultad de Informática - UNLP

Figura 6.4. Creación del proyecto Albergue Universitario UNLP

### 6.3.1 DETALLE DEL PROYECTO

Una vez creado el proyecto, el sistema muestra cómo se organiza la información relacionada a través de nav tabs (Figuras 6.5 y 6.6).

En esta pantalla es posible: editar los datos del proyecto, acceder al historial de cambios y eliminarlo. Así como también crear una nueva versión, editar sólo la última, y ver el historial de versiones.

Además, se muestra una barra con el porcentaje de requerimientos funcionales y no funcionales terminados. Esto permite que los analistas puedan conocer rápidamente el estado de avance del proyecto.

SGR Tablero Proyectos lider\_proyecto

## Albergue Universitario UNLP

Editar Proyecto Historial de cambios

Estado del proyecto Activo

Requerimientos en estado Terminado 0%

Fecha de inicio estimada	Fecha de fin estimada	Fecha de inicio real	Fecha de fin real	Versión
16/04/2018	16/10/2018			0.1 15/04/2018 [lider_proyecto] +

Alcance  
Propósito  
Resumen  
Descripción

El sistema permitirá gestionar el uso del comedor del Albergue. No existen supuestos que no estén indicados en este documento.

Eliminar Proyecto Volver

Facultad de Informática - UNLP

Figura 6.5. Detalle del proyecto Albergue Universitario UNLP

SGR

## Albergue Universitario UNLP

Editar Proyecto Historial de cambios

Estado del proyecto Activo

Requerimientos en estado Terminado 21%

Opciones

- Datos Generales
- Requerimientos Funcionales
- Requerimientos No Funcionales
- Objetivos
- Usuarios Asignados
- Archivos Adjuntos
- Atributos
- Participantes
- resumen
- Descripción

El sistema permitirá gestionar el uso del comedor del Albergue. No existen supuestos que no estén indicados en este documento.

Eliminar Proyecto

Volver

Facultad de Informática - UNLP

Figura 6.6. Detalle del proyecto Albergue Universitario UNLP (Dispositivo Móvil)

### 6.3.2 OBJETIVOS

En el nav tab *Objetivos* (Figura 6.7) es posible visualizar el listado de los objetivos relacionados a un proyecto con la posibilidad de ordenarlos y filtrarlos por Nombre y Descripción; así como también realizar las acciones de crear, editar y eliminar.

Para el caso de estudio, se creó la lista de objetivos previamente descritos en la introducción, que dieron origen al sistema. De esta manera será posible establecer qué objetivos del proyecto debe satisfacer un requerimiento.

The screenshot displays the 'Objetivos' (Objectives) tab within the 'Albergue Universitario UNLP' project management system. The header includes the SGR logo, navigation tabs for 'Tablero' and 'Proyectos', and a user profile dropdown for 'lider\_proyecto'. The main content area shows the project name 'Albergue Universitario UNLP' and a status bar indicating 'Estado del proyecto Activo'. Below this, a progress indicator shows 'Requerimientos en estado Terminado 0%'. A horizontal menu allows switching between 'Datos Generales', 'Requerimientos Funcionales', 'Requerimientos No Funcionales', 'Objetivos' (selected), 'Usuarios Asignados', 'Archivos Adjuntos', 'Atributos', and 'Stakeholders'. A search interface includes input fields for 'Nombre' and 'Descripción', and a 'Buscar' button. The results are shown in a table with columns for 'Nombre' and 'Acciones'. The table lists four objectives, each with search and edit icons. At the bottom, there are buttons for '+ Nuevo', 'Eliminar Proyecto', and 'Volver', along with a footer for 'Facultad de Informática - UNLP'.

Nombre	Acciones
Automatizar el proceso de inscripción a cada comida	🔍 ✎
Registrar el proceso de retiro de cada alumno	🔍 ✎
Generar reportes con la información almacenada	🔍 ✎
Determinar restricciones del sistema	🔍 ✎

Figura 6.7. Listado de objetivos

### 6.3.3 USUARIOS ASIGNADOS

El nav tab *Usuarios Asignados* (Figura 6.8) muestra el listado de los usuarios asignados a un proyecto. Además, permite crear y eliminar asignaciones. Dicho listado puede ser ordenado y filtrado por nombre de Usuario y Grupo. Al realizar la asignación se deberá elegir el grupo al que pertenece, lo que indicará el rol que cumple este dentro del proyecto.

Los usuarios definidos para el desarrollo del sistema del Albergue Universitario UNLP realizaron las siguientes acciones:

- **el líder del proyecto** quién se encargó de crear el proyecto, asignar los usuarios, cargar los objetivos y cargar los stakeholders.
- **el analista** el cual llevó a cabo la carga de los requerimientos.
- **el cliente** quién se encargó de llevar a cabo el proceso de validación de los requerimientos.

A partir de estas asignaciones, se limitaron las acciones que podían realizar los usuarios dentro del proyecto.

Albergue Universitario UNLP

Estado del proyecto Activo

Requerimientos en estado Terminado 21%

Datos Generales | Requerimientos Funcionales | Requerimientos No Funcionales | Objetivos | **Usuarios Asignados** | Archivos Adjuntos | Atributos | Stakeholders

Usuario:  Grupo:  Buscar

Usuario	Grupo	Notificar	
cliente	Invitado	Si	🔍 ✎
lider_proyecto	Administrador	No	🔍 ✎
analista	Analista	Si	🔍 ✎

Resultados: 3

+ Nuevo

✖ Eliminar Proyecto | 🏠 Volver

Facultad de Informática - UNLP

Figura 6.8. Listado de usuarios asignados

### 6.3.4 ARCHIVOS ADJUNTOS

En Archivos Adjuntos (Figura 6.9) es posible ver el listado de archivos adjuntos de un proyecto, agregar nuevos, editarlos, eliminarlos y recuperarlos; también se podrá ordenar y filtrar dicho listado. Esta opción también está disponible para requerimientos funcionales y no funcionales.

Para el caso de estudio se adjuntó al proyecto la minuta de reunión y documentación relacionada con el diseño, de manera tal que se tenga centralizada toda la información.

Albergue Universitario UNLP

Estado del proyecto Activo

Requerimientos en estado Terminado 0%

Datos Generales | Requerimientos Funcionales | Requerimientos No Funcionales | Objetivos | Usuarios Asignados | **Archivos Adjuntos** | Atributos | Stakeholders

Nombre:  Descripción:  Nombre del archivo:  Buscar

Nombre	Descripción	Archivo	Acciones
Minuta primera reunión	Realizada en el LIDI. Duración 1 hora	Minuta 1.doc	🔍 ✎
Modelo conceptual	Primera versión del Modelo conceptual del sistema	Modelo Conceptual V1.jpeg	🔍 ✎

Resultados: 2

+ Nuevo

Recuperar Adjuntos

✖ Eliminar Proyecto | 🏠 Volver

Facultad de Informática - UNLP

Figura 6.9. Listado de archivos adjuntos

### 6.3.5 ATRIBUTOS

A través del nav tab *Atributos* (Figura 6.10) es posible cargar información complementaria, la cual será incorporada al documento de especificación de requerimientos.

Si bien SGR comprende los elementos incluidos en el estándar de la IEEE Std. 830-1998, permite que los usuarios agreguen los ítems que consideren necesarios (ver Anexo I).

The image shows a web interface for adding a new attribute. At the top, there is a red navigation bar with the SGR logo and tabs for 'Tablero' and 'Proyectos'. A user profile 'lider\_proyecto' is shown in the top right corner. The main content area is titled 'Nuevo atributo' and contains four input fields: 'Nombre', 'Descripción', 'Nivel', and 'Sub nivel'. Below the 'Nombre' field is a 'Guardar' button, and below the 'Sub nivel' field is a 'Volver' button. The footer of the page reads 'Facultad de Informática - UNLP'.

Figura 6.10. Formulario para el alta de atributos

### 6.3.6 STAKEHOLDERS

El nav tab *Stakeholders* (Figura 6.11) permite agregar datos de aquellas personas que influyen directa o indirectamente en un proyecto, pero no tienen un usuario en el sistema para poder acceder; permitirá ver un listado, el cuales puede ser ordenado y filtrado por Apellido y Nombre.

Para el proyecto, se agregó un nuevo stakeholder para determinar la fuente de información de la especificación de los requerimientos, lo que contribuye a complementar los datos de la especificación de requerimientos.

Albergue Universitario UNLP

Estado del proyecto Activo

Requerimientos en estado Terminado 0%

Datos Generales | Requerimientos Funcionales | Requerimientos No Funcionales | Objetivos | Usuarios Asignados | Archivos Adjuntos | Atributos | Stakeholders

Apellido:  Nombre:  Buscar

Apellido	Nombre	Institución	Puesto	Acciones
Mella	Franco	Albergue	-	

Resultados: 1

+ Nuevo

Eliminar Proyecto | Volver

Facultad de Informática - UNLP

Figura 6.11. Listado de stakeholders

### 6.3.7 VERSIONES

A través del detalle de un proyecto (Figuras 6.5 y 6.6) es posible agregar una nueva versión, editar la última y ver el historial (Figura 6.12). Estas funcionalidades también están disponibles para requerimientos funcionales y no funcionales.

El número de versión está compuesto por dos valores. El primero se utiliza para indicar que hubo cambios fundamentales es un proyecto y que estos fueron revisados y aprobados. El segundo valor indica que hubo cambios menos significativos (también deben estar aprobados). Cada proyecto es creado con el número 1.0 por defecto.

Número	Fecha	Usuario	Acciones
1.0	07/02/2018	lider_proyecto	
1.1	16/05/2018	lider_proyecto	

Cerrar

Figura 6.12. Versiones de un proyecto

### 6.3.8 NOTIFICACIONES DE CAMBIOS Y COMUNICACIÓN ENTRE USUARIOS

Debido a que en un proyecto generalmente trabajan varias personas, SGR brinda distintos medios de notificaciones por los cuales mantiene actualizados a todos los usuarios integrantes de un proyecto. Por un lado, cuenta con un tablero y por el otro, a través del envío automático de emails.

El Tablero (Figuras 6.13 y 6.14) le indica al usuario en sesión los proyectos atrasados (aquellos que pasaron su fecha de fin estimada y no tiene fecha de fin real), los proyectos próximos a concluir (aquellos cuya fecha estimada de finalización se cumple en los próximos siete días) y las últimas actualizaciones (se muestran todos proyectos que fueron modificados luego del último inicio de sesión del usuario y durante la sesión actual).



Figura 6.13. Tablero de usuario

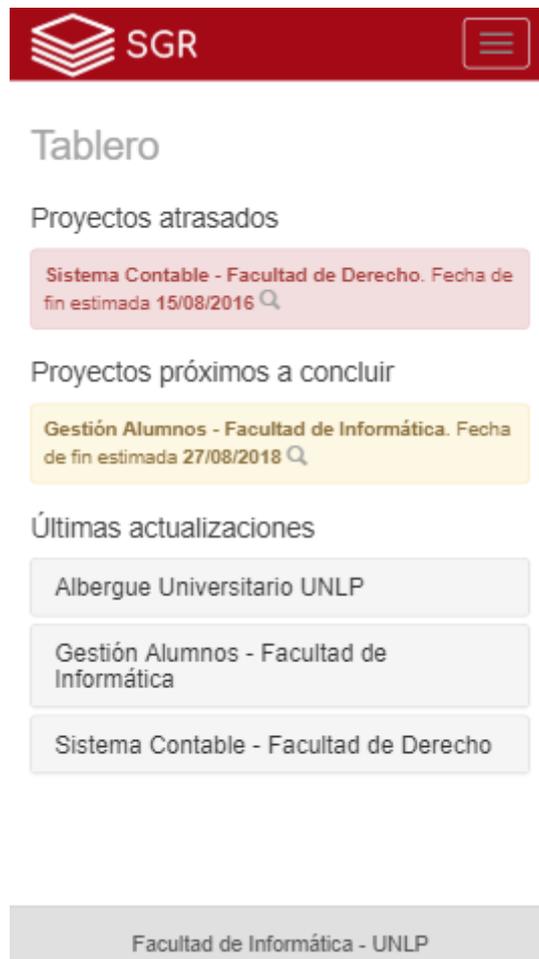


Figura 6.14. Tablero de usuario (Dispositivo Móvil)

El envío automático de emails se realiza por dos eventos:

1. Cuando un proyecto haya sido actualizado después de la última sesión del usuario se lo notificará de los cambios ocurridos. Estas notificaciones se realizan una vez al día (Figura 6.15).

## Notificaciones SGR

**Desde su último inicio de sesión se produjeron los siguientes cambios**

Proyecto: Albergue Universitario UNLP [Ver](#)

- Datos del proyecto
- Requerimientos Funcionales
  - ABM Usuarios
- Objetivos
  - Automatizar el proceso de inscripción a cada comida
- Usuarios asignados
  - Usuario cliente - Grupo cliente
- Adjuntos
  - Modelo Conceptual

Saludos, Equipo SGR



Figura 6.15. Email enviado a un usuario participante de un proyecto actualizado

2. Cuando un usuario con permisos para aprobar o rechazar un requerimiento realiza alguna de las dos acciones, SGR envía un email (como el de la Figura 6.16) para notificar a los usuarios integrantes de un proyecto acerca del cambio de estado de un requerimiento. Los emails solo serán enviados a los usuarios que se hayan indicado al realizar la asignación de estos en el proyecto.

## Notificaciones SGR

**Requerimiento Aprobado por desarrollo**

Requerimiento: ABM de Usuarios [Ver](#)

Saludos, Equipo SGR



Figura 6.16. Email enviado a todos los usuarios asignados a un proyecto.

Asimismo, SGR permite crear conversaciones (Figura 6.17) y comentarios (Figura 6.18) lo que brinda comunicarse entre usuarios y fomentar la participación de estos en los proyectos. Tanto las conversaciones como los comentarios pueden realizarse a nivel de proyecto o de requerimiento.



Figura 6.17. Nueva Conversación de un Proyecto

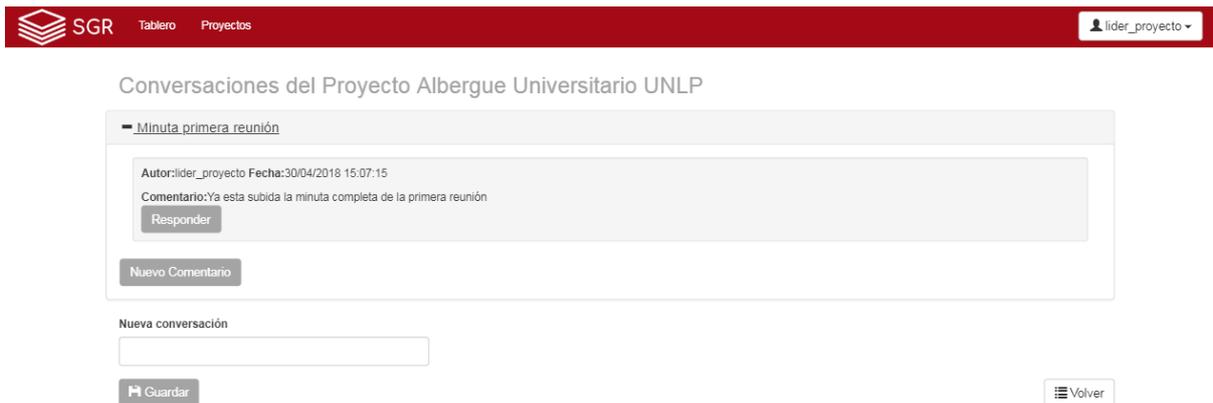


Figura 6.18. Conversación y comentarios de un proyecto

### 6.3.9 MATRIZ DE TRAZABILIDAD

Desde el listado de proyectos (Figura 6.3) se puede acceder a la matriz de trazabilidad (Figura 6.19) en donde se puede ver fácilmente las dependencias que fueron asignadas a los requerimientos funcionales y no funcionales. En el caso de estudio realizado, no fue necesario definir requerimientos no funcionales, por lo que no se encuentran graficados en dicha matriz.

## Matriz de trazabilidad Albergue Universitario UNLP

	RF-1	RF-2	RF-3	RF-4	RF-5	RF-6	RF-7	RF-8	RF-9	RF-10	RF-11	RF-12	RF-13	RF-14
RF-1			✓											
RF-2				✓										
RF-3														
RF-4					✓									
RF-5														
RF-6														
RF-7						✓								
RF-8														
RF-9		✓					✓							
RF-10							✓							
RF-11							✓							
RF-12							✓							
RF-13									✓	✓	✓	✓		
RF-14									✓					

[Volver](#)

Facultad de Informática - UNLP

Figura 6.19. Matriz de trazabilidad

## 6.4 GESTIÓN DE REQUERIMIENTOS FUNCIONALES

Sobre un proyecto ya creado, es posible ver el listado de requerimientos funcionales (Figura 6.19). Desde este listado se puede realizar un conjunto de acciones relacionadas con un requerimiento. Estas acciones son: crear, ver y editar un requerimiento; asignar una nueva versión, editar solo la última y ver el historial de versiones. Es posible realizar búsquedas y ordenaciones por nombre, estado y prioridad.

Una vez creado el proyecto y definida la información complementaria (usuarios asignados, objetivos, stakeholders y adjuntos), se procedió a cargar de los requerimientos funcionales a través del formulario de la figura 6.20.

**SGR** Tablero Proyectos analista

### Nuevo requerimiento funcional

**Nombre**

**Estabilidad**

**Descripción**  
 Archivo Editar Ver Formato  
 Formatos **B** *I* [List of icons]  
 El sistema **permitirá** crear, modificar y eliminar usuarios con los siguientes datos: nombre (requerido), apellido (requerido), DNI (requerido y único), teléfono, email y área (selecciona).  
 p

**Estado**

**Prioridad**

**Dependencias funcionales**

**Dependencias no funcionales**

**Nro orden**

Facultad de Informática - UNLP

Figura 6.20. Formulario de creación de un requerimiento funcional

El listado de requerimientos funcionales cargado es:

- ABM Áreas
- ABM Usuarios
- ABM Facultades
- ABM Carreras
- ABM Alumnos
- Parámetros de Configuración
- ABM Menús
- ABM Feriados
- Inscripción a comida
- Ticket de retiro con inscripción - Personal
- Ticket de retiro con inscripción - Otro
- Ticket de retiro sin inscripción
- Reporte General
- Reporte de inscripción finalizada

Si bien SGR no impone un orden para la carga de la información, lo descrito en los ítems anteriores determina el orden que, a nuestro parecer, debería seguirse para sacar el mejor provecho de las funcionalidades del sistema.

Las figuras 6.21 y 6.22 muestran los requerimientos cargados en el sistema. El detalle completo estos se presentan a través del documento SRS generado (ver Anexo I).

SGR Tablero Proyectos analista

## Albergue Universitario UNLP

Editar Proyecto Historial de cambios

Estado del proyecto Activo

Requerimientos en estado Terminado 0%

Datos Generales **Requerimientos Funcionales** Requerimientos No Funcionales Objetivos Usuarios Asignados Archivos Adjuntos Atributos Stakeholders

Nombre Estado Prioridad

Nº	Nombre	Versión	Estado	Prioridad	Creado Por	Fecha	Acciones
1	ABM Usuarios	1.0 16/04/2018 [analista]	Validación	Esencial	analista	16/04/18	🔍 + 🗑️ 🗨️
2	ABM de Alumnos	1.0 16/04/2018 [analista]	Validación	Esencial	analista	16/04/18	🔍 + 🗑️ 🗨️
3	ABM Áreas	1.0 16/04/2018 [analista]	Validación	Esencial	analista	16/04/18	🔍 + 🗑️ 🗨️
4	ABM Carreras	1.0 16/04/2018 [analista]	Validación	Esencial	analista	16/04/18	🔍 + 🗑️ 🗨️
5	ABM Facultades	1.0 17/04/2018 [analista]	Validación	Esencial	analista	17/04/18	🔍 + 🗑️ 🗨️
6	AMB Feriados	1.0 17/04/2018 [analista]	Validación	Esencial	analista	17/04/18	🔍 + 🗑️ 🗨️
7	ABM Menu	1.0 17/04/2018 [analista]	Validación	Esencial	analista	17/04/18	🔍 + 🗑️ 🗨️
8	Parámetros de Configuración	1.0 17/04/2018 [analista]	Validación	Esencial	analista	17/04/18	🔍 + 🗑️ 🗨️
9	ABM Inscripción a Comida	1.0 17/04/2018 [analista]	Validación	Esencial	analista	17/04/18	🔍 + 🗑️ 🗨️
10	Ticket de retiro con inscripción - Personal	1.0 20/04/2018 [analista]	Validación	Esencial	analista	20/04/18	🔍 + 🗑️ 🗨️

Resultados: 14

Facultad de Informática - UNLP

Figura 6.21. Listado de requerimientos

SGR

## Albergue Universitario UNLP

Estado del proyecto Activo

Requerimientos en estado Terminado 21%

Opciones

Requerimientos Funcionales

Nombre

Estado

Prioridad

Nº	Nombre	Versión
1	ABM Usuarios	1.0 10/0
2	ABM de Alumnos	1.0 10/0
3	ABM Áreas	1.0 10/0
4	ABM Carreras	1.0 10/0
5	ABM Facultades	1.0 10/0
6	AMB Feriados	1.0 10/0
7	ABM Menu	1.0 10/0
8	Parámetros de Configuración	1.0 10/0
9	ABM Inscripción a Comida	1.0 10/0
10	Ticket de retiro con inscripción - Personal	1.0 10/0

Resultados: 14

Facultad de Informática - UNLP

Figura 6.22. Listado de requerimientos (Dispositivo Móvil)

## 6.4.1 DETALLE DE REQUERIMIENTO

Una vez creado los requerimientos (Figura 6.20) es posible acceder al detalle de un requerimiento funcional (Figuras 6.23 y 6.24), editar los datos, acceder a su historial de cambios y eliminarlos.

La información relacionada con un requerimiento, al igual que el detalle de un proyecto, se muestra organizada en nav tabs.

Requerimiento funcional del proyecto Albergue Universitario UNLP

Editar Requerimiento Historial de cambios

Datos Generales Archivos Adjuntos Stakeholders Objetivos

Nombre	Reporte General
Descripción	El sistema permitirá elegir un período de tiempo indicando fecha desde y fecha hasta, el tipo de comida (se selecciona entre almuerzo, cena o ambas). El reporte mostrará • Inscriptos o cantidad de inscriptos en el menú vegetariano o cantidad de inscriptos en el menú común • Retiros o cantidad de retiros de inscriptos de menú vegetariano o cantidad de retiros de inscriptos de menú común o cantidad de retiros de NO inscriptos de menú vegetariano o cantidad de retiros de NO inscriptos de menú común o cantidad total de retiros de menú vegetariano o cantidad total de retiros de menú común
Estabilidad	Alta
Estado	Validación
Prioridad	Esencial
Versiones	1.0 (21/04/2018)
Nro de orden	13
Dependencias	RF-11 - ABM Inscripción a Comida RF-12 - Ticket de retiro con inscripción - Personal RF-13 - Ticket de retiro con inscripción - Otro RF-14 - Ticket de retiro sin inscripción

Volver

Facultad de Informática - UNLP

Figura 6.23. Detalle de un requerimiento funcional

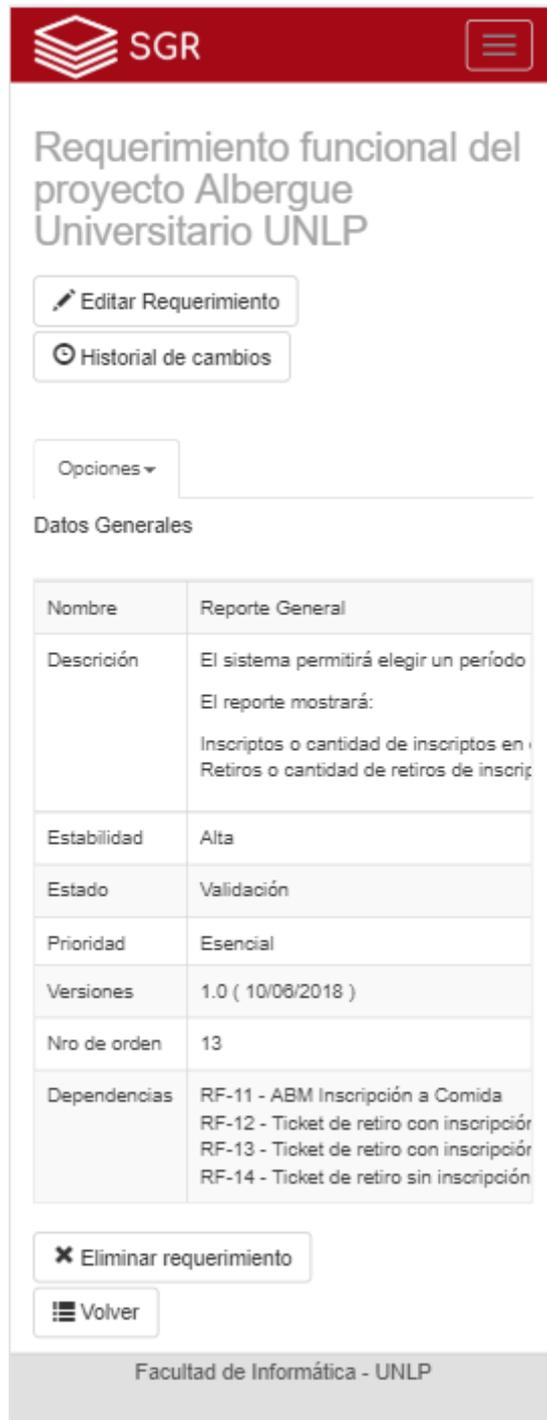


Figura 6.24. Detalle de un requerimiento funcional (Dispositivo Móvil)

#### 6.4.2 ASIGNACIÓN DE STAKEHOLDERS

El nav tab *Stakeholder* (Figura 6.25) muestra el listado de aquellos stakeholders asignados a un proyecto. Solo se podrán asignar stakeholders previamente creados en el proyecto. Desde esta opción solo se podrá agregar y eliminar la asignación (no el stakeholder).

Requerimiento funcional del proyecto Albergue Universitario UNLP

Editar Requerimiento Historial de cambios

Datos Generales Archivos Adjuntos Stakeholders Objetivos

Apellido Nombre Buscar

Solo se podrán asignar stakeholders que pertenezcan al proyecto.

Apellido y Nombre	Institución	Puesto	Acciones
Mella Franco	Albergue	-	

Resultados: 1

+ Nuevo Volver

Facultad de Informática - UNLP

Figura 6.25. Listado de stakeholders asignados a un requerimiento

### 6.4.3 ASIGNACIÓN DE OBJETIVOS

El nav tab *Objetivos* (Figura 6.26) permite visualizar el listado de objetivos asignados y crear nuevas asignaciones. Desde esta opción, solo se podrán agregar y eliminar asignaciones (no objetivos).

La asignación de objetivos es una función muy importante ya que establecen el fundamento del desarrollo del software a implementar.

Requerimiento funcional del proyecto Albergue Universitario UNLP

Se ha creado correctamente el registro.

Editar Requerimiento Historial de cambios

Datos Generales Archivos Adjuntos Stakeholders Objetivos

Nombre Buscar

Solo se podrán asignar objetivos que pertenezcan al proyecto.

Nombre	Acciones
Generar reportes con la información almacenada	

Resultados: 1

+ Nuevo Volver

Facultad de Informática - UNLP

Figura 6.26. Listado de objetivos asignados a un requerimiento

### 6.4.4 VALIDACIÓN DE REQUERIMIENTOS

En el caso de contar con el permiso correspondiente, se mostrarán las acciones para aceptar o rechazar un requerimiento que se encuentre en estado *Validación* (Figuras 6.27). Por cada

requerimiento que fue revisado y aprobado o rechazado por el usuario *cliente*, se enviaron automáticamente emails a aquellos integrantes del proyecto que se indicó que debían ser notificados (ver ítem 6.3.3).

Los requerimientos aprobados, se pasan a estado *Terminado* lo que indica que cumplen con lo solicitado por el cliente.

Los requerimientos rechazados, se volvieron a estado *Desarrollo* ya que no satisfacen lo que el cliente solicitó.

SGR Tablero Proyectos analista

Albergue Universitario UNLP

Editar Proyecto Historial de cambios

Estado del proyecto Activo

Requerimientos en estado **Terminado** 21%

Datos Generales **Requerimientos Funcionales** Requerimientos No Funcionales Objetivos Usuarios Asignados Archivos Adjuntos Atributos Stakeholders

Nombre Estado Prioridad Buscar

Nº	Nombre	Versión	Estado	Prioridad	Creado Por	Fecha	Acciones
1	ABM Usuarios	1.0 16/04/2018 [analista]	Validación	Esencial	analista	16/04/18	🔍 ✎ + 🗑️
2	ABM de Alumnos	1.0 16/04/2018 [analista]	Terminado	Esencial	analista	16/04/18	🔍 ✎ + 🗑️
3	ABM Áreas	1.0 16/04/2018 [analista]	Desarrollo	Esencial	analista	16/04/18	🔍 ✎ + 🗑️
4	ABM Carreras	1.0 16/04/2018 [analista]	Desarrollo	Esencial	analista	16/04/18	🔍 ✎ + 🗑️
5	ABM Facultades	1.0 17/04/2018 [analista]	Validación	Esencial	analista	17/04/18	🔍 ✎ + 🗑️
6	AMB Feriados	1.0 17/04/2018 [analista]	Terminado	Esencial	analista	17/04/18	🔍 ✎ + 🗑️
7	ABM Menu	1.0 17/04/2018 [analista]	Desarrollo	Esencial	analista	17/04/18	🔍 ✎ + 🗑️
8	Parámetros de Configuración	1.0 17/04/2018 [analista]	Validación	Esencial	analista	17/04/18	🔍 ✎ + 🗑️
9	ABM Inscripción a Comida	1.0 17/04/2018 [analista]	Terminado	Esencial	analista	17/04/18	🔍 ✎ + 🗑️
10	Ticket de retiro con inscripción - Personal	1.0 20/04/2018 [analista]	Validación	Esencial	analista	20/04/18	🔍 ✎ + 🗑️

Resultados: 14

+ Nuevo

Volver

Facultad de Informática - UNLP

Figura 6.27. Listado de requerimientos validados y rechazados

## 6.5 GENERACIÓN AUTOMÁTICA DEL SRS

A través de la acción  del listado de proyectos (Figura 6.3) es posible generar documentos IEEE 830-1998 en formato PDF de la especificación de requerimientos con los datos almacenados en SGR como (descripción, alcance, propósito, listados de requerimientos funcionales y no funcionales, atributos definidos, etc.). Asimismo, éste archivo contará con una portada con el nombre del proyecto, la versión y la fecha; y un índice con hipervínculos dentro del mismo documento.

El anexo I muestra el SRS para el Albergue Universitario UNLP generado con SGR.

## 6.6 RESUMEN DEL CAPÍTULO

En este capítulo se tomó como caso de estudio el sistema para la gestión del comedor del Albergue Universitario UNLP, utilizando SGR para realizar una de las etapas más importantes en el desarrollo de un proyecto de software como es la gestión de requerimientos.

Si bien SGR no impone un orden para la carga de la información, lo descrito en este capítulo determina el orden que, a nuestro parecer, debería seguirse para sacar el mejor provecho de las funcionalidades del sistema.

Para la elicitación de requerimientos, se realizó una entrevista a un representante del Albergue. A partir de la información obtenida, se procedió a la creación y carga de los datos del proyecto como: la descripción del sistema que se iba a desarrollar, el propósito y alcance de la especificación de requerimientos, entre otros; además, se asignaron los datos relacionados como: los objetivos, los usuarios que participaron en la etapa de gestión de requerimientos y el stakeholder que se encargó de brindar toda la información. También se adjuntaron los archivos generados externamente para mantener centralizada la información.

Asimismo, se crearon todos los requerimientos funcionales a partir de las necesidades del cliente.

Por otra parte, se mostraron las funcionalidades de versionado, historiales y conversaciones las cuales son comunes tanto para los proyectos como para los requerimientos funcionales y no funcionales.

Además se mostró la participación del cliente a través de las acciones de *Aceptar* y *Rechazar* los requerimientos. Esto permitió realizar las negociaciones de los requerimientos, logrando de esta manera alinear las necesidades del cliente con el software resultante.

La comunicación entre los usuarios se realizó a través del Tablero del usuario y envíos de emails automáticos. Los emails son enviados a los usuarios asignados a un proyecto, cuando se produzcan actualizaciones sobre éstos y cuando los requerimientos en estado *Validación* sean aprobados o rechazados por el cliente.

Por último, se mostró la generación del reporte de especificación de requerimientos (SRS) basados en el estándar de la IEEE 830-1998 para el sistema de gestión del comedor del Albergue Universitario.

## 7. CONCLUSIONES

La obtención de requerimientos tiene, como principal objetivo, traducir o captar lo que el cliente y los usuarios esperan o desean de un sistema de software. Esta tarea marca el punto inicial para planear, estimar tiempos y costos, definir recursos, elaborar cronogramas, etc. Preparar una especificación de requerimientos adecuada permite definir objetivos y alcances para verificar si se alcanzaron las metas definidas logrando reducir los costos y riesgos asociados con el desarrollo.

En el marco de la presente tesina se estudiaron diferentes maneras de llevar a cabo una especificación de requerimientos y se concluyó que este proceso debe ser claro, completo y estar libre de ambigüedades. Por esto, la mejor forma de realizar la especificación es a través de un lenguaje natural estructurado que asegura que haya cierta uniformidad sobre la especificación a través de la definición de una o más plantillas estándar para requerimientos. Además, la especificación estructurada de requerimientos guía a analistas inexpertos en el desarrollo de la especificación de requerimientos.

Una buena especificación sirve como un contrato entre clientes y desarrolladores; y ayuda a la reducción de errores atribuidos a defectos en la especificación de ambos tipos de requerimientos.

Una comunicación simple y fluida entre los distintos stakeholders es otro factor importante y fundamental para el éxito de un proyecto de software. Una mala comunicación aumenta la posibilidad de obtener requerimientos incorrectos y llegar a producir resultados no esperados por el cliente.

En base al análisis realizado se evidenció el impacto de llevar a cabo una mala gestión de requerimientos en un proyecto de software, y se determinó la necesidad e importancia de contar con una herramienta CASE para dar soporte a la etapa de 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. Por esto 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

Posteriormente, se desarrolló un modelo conceptual y de clases para representar la información del sistema, y por último se establecieron los diseños de interfaz a través del uso de mockups.

Para la implementación de SGR se establecieron previamente la arquitectura y las herramientas a utilizar para el desarrollo de manera que el sistema se realice de forma eficiente y completa para satisfacer las necesidades de los clientes.

SGR fue incluido en el análisis comparativo de las herramientas evaluadas en el capítulo 3. En consecuencia, se depende la Tabla 2, la cual muestra la cantidad de características que satisface cada herramienta, de entre todas las que se consideran necesarias para un software de este estilo (sección 2.2). Se puede observar que SGR cumple con la totalidad de estas características, ya que se

puso énfasis en las debilidades de las otras herramientas, como 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

Finalmente, con el objetivo de evaluar su desempeño, se utilizó SGR en un proyecto de software 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 generaron documentos SRS que están siendo utilizados actualmente por los desarrolladores para la implementación del sistema con resultados y feedback positivos.

## 8. TRABAJOS FUTUROS

Si bien SGR cumple con las características necesarias para llevar a cabo la etapa de gestión de requerimientos, es posible mejorar su funcionamiento incorporando nuevas funcionalidades para potenciar su uso. A partir de la experiencia obtenida sobre el caso de estudio descrito en la sección anterior se desprenden los siguientes trabajos a futuro como nuevos desafíos para ser llevados a cabo.

- **Ampliar las funcionalidades de SGR:** Con el propósito de mejorar la experiencia del usuario, algunas de las funcionalidades que se podrían agregar al SGR son: definir los objetivos por los que un requerimiento fue creado (lo cual permitirá ampliar la matriz de trazabilidad); incorporar la opción de generar documentos en formato PDF de requerimientos funcionales y no funcionales de un proyecto permitiendo aplicar filtros y criterios de ordenamiento; visualizar notificaciones desde el navegador ante determinado eventos; mostrar un calendario con las fechas importantes tanto de los requerimientos como de los proyectos; permitir la gestión de historias de usuarios para mejorar la comunicación con los clientes posibilitando responder rápidamente a los requerimientos cambiantes; y generar para cada proyecto un diccionario con las frases y palabras más utilizadas en el dominio o ambiente del negocio.
- **SGR como aplicación móvil:** Crear una aplicación para dispositivos móviles que, a través de la información provista por SGR, les permita a los usuarios acceder a los proyectos en los que participa y recibir notificaciones en tiempo real de las actualizaciones en los mismos.
- **SGR en entornos académicos:** Poner a SGR a disposición de asignaturas relacionadas con la Ingeniería de Software de la Facultad de Informática (UNLP) para obtener retroalimentación de los alumnos a partir de la cual realizar mejoras o agregar funcionalidades al sistema.

# BIBLIOGRAFÍA

- [1] Pfleeger, Shari L. *Ingeniería de software: teoría y práctica*. Prentice Hall. 2002.
- [2] Bárbara A. Mcdonald Landazuri. *Definición de Perfiles en Herramientas de Gestión de Requisitos*. Revista de las Sedes Regionales. 2006.
- [3] IEEE Software. *Requirements Engineering Tools*. Páginas 86-91. Volumen 28. Julio/Agosto de 2011. <http://ieeexplore.ieee.org/document/5929527/>
- [4] Davis Alan M. *Software requirements: objects, functions, and states*. Prentice Hall. 1993.
- [5] Somerville, Ian. *Ingeniería de Software 9ma Edición*. Pearson. 2011
- [6] Pressman, Roger. *Ingeniería del software un enfoque práctico 7ma Edición*. The McGraw-Hill, 2010.
- [7] María del Carmen Gómez Fuentes. *Notas del curso: análisis de requerimientos*, Universidad Autónoma Metropolitana. 2011
- [8] IEEE. *Especificación de Requisitos según el estándar de IEEE 830, IEEE Std. 830-1998*. 1998.
- [9] *Ingeniería de Software 1. Teoría 1*. Facultad Informática UNLP. 2017.
- [10] ALARCÓN, Andrea. Sandoval, Erika. *Herramientas CASE para ingeniería de Requisitos*. Cultura Científica JDC. 2008.
- [11] Enciclopedia Cubana creada por el Ministerio de Informática y Comunicaciones. [www.ecured.cu](http://www.ecured.cu)
- [12] Raúl Monferrer Agut. *Especificación de Requisitos Software según el estándar de IEEE 830*. Universidad Jaume I. 2001
- [13] Master María Marta Sandoval Carvajal. *La trazabilidad en el proceso de requerimientos de software*. Universidad Nacional Escuela de Informática. Costa Rica. 2008.
- [14] Graciela Hadad, Jorge Doorn, Marcela Ridaio, Gladys Kaplan. *Facilitando la Asignación de Prioridades a los Requisitos*. Tecnológicas UNLaM. Argentina. 2009.
- [15] Graciela D.S. Hadad, Gerardo A. Riera, Jorge H. Doorn. *Priorizar Requisitos: un Estudio sobre sus Propósitos*. Universidad Nacional de La Matanza. 2018.
- [16] Blanca Elizabeth Vargas Martínez. *Panorama general de las "herramientas case"*. Universidad autónoma del estado de hidalgo. 2007.
- [17] González López P, González López A, Gallud Lázaro J. *Herramientas CASE ¿Cómo incorporarlas con éxito en nuestra organización?* Revista de la Facultad de Educación de Albacete. 1995.
- [18] Visure Requirements [www.visuresolutions.es](http://www.visuresolutions.es)
- [19] DOORSNG [www.doorsng.com](http://www.doorsng.com)
- [20] Rommana <https://rommanasoftware.com/>
- [21] TraceCloud <https://www.tracecloud.com/>
- [22] Psoda <https://www.psoda.com>
- [23] TrackStudio <http://www.trackstudio.com/>
- [24] Redmine <https://www.redmine.org/>

- [25] Durán Amador. *Un Entorno Metodológico de ingeniería de requisitos para sistemas de información*. Tesis doctoral. Universidad de Sevilla. Mayo 2010.
- [26] Mauro Callegas Cuervo, Luz Yadira Castillo Estupoñán, Ruby Mónica Fernández Álvarez. *HELER: una herramienta para la ingeniería de requisitos automatizada*. Colombia: Entramado 2010. Págs. 184-200.
- [27] Pablo Torrecilla González, María N. Moreno García. *Let's Req*. Proyecto TCUE Universidades de Castilla y León. <http://letsreq.com/>
- [28] Bertone y Thomas. *Introducción a las Bases de Datos*. Pearson. 2012.
- [29] Orientación a Objetos 1. *Teoría Introducción al lenguaje de modelado*. Facultad Informática UNLP. 2017
- [30] David Garlan, Mary Shaw. *Software Architecture: Practice, Potential, and Pitfalls*. 1993
- [31] PHP [www.php.net/](http://www.php.net/)
- [32] Symfony <https://symfony.com/>
- [33] Composer <https://getcomposer.org/>
- [34] Doctrine <https://symfony.com/doc/2.8/doctrine.html>
- [35] Twig <https://twig.symfony.com>
- [36] MySQL <https://www.mysql.com/>
- [37] Bootstrap <http://getbootstrap.com/>
- [38] JQuery <http://jquery.com/>
- [39] JavaScript <https://www.javascript.com>
- [40] KnpPaginatorBundle <https://github.com/KnpLabs/KnpPaginatorBundle>
- [41] SnappyBundle <https://ourcodeworld.com/articles/read/250/how-to-create-a-pdf-from-html-using-knpsnappybundle-wkhtmltopdf-in-symfony-3>
- [42] Swiftmailer <http://swiftmailer.org/>
- [43] EntityAuditBundle <https://github.com/simplethings/EntityAuditBundle/blob/1.0/README.md>
- [44] Gliffy <https://www.gliffy.com/>
- [45] GitLab <https://gitlab.com/>
- [46] Kurose, James F. *Redes de computadoras: un enfoque descendente*. Pearson, 2010.

Todas las páginas fueron accedidas entre los meses de febrero y septiembre de 2018.

# ANEXO I

## SRS ALBERGUE UNIVERSITARIO UNLP generado por SGR