Universidad Nacional de La Plata
Facultad de Informática
Postgrado

Tesis de Maestría en Ingeniería de Software

**Una propuesta para facilitar la trazabilidad
de los requerimientos en un desarrollo ágil**

Alumno: Lic. Fernando Javier Guerra
Director: Dr. Gustavo Rossi
Director: Dr. Matías Urbieta

La Plata, Argentina, Junio 2021

Dedicatoria

“Si no hay esfuerzo no hay progreso...”

Frederick Douglass, 3 Agosto 1857

A mi Tata y a mi Mama

Agradecimientos

A mi esposa Lili, nuestros hijos Agustín y Andrés, presentes siempre sin saberlo.

A Matías Urbieto, quien me acompañó y guió como un amigo de siempre.

Índice

1	INTRODUCCIÓN.....	1
1.1	MOTIVACIÓN.....	1
1.2	OBJETIVOS.....	3
1.2.1	<i>Objetivo general.....</i>	<i>3</i>
1.2.2	<i>Objetivos específicos.....</i>	<i>3</i>
2	TRAZABILIDAD.....	4
2.1	DEFINICIÓN.....	4
2.2	TÉCNICAS DE TRAZABILIDAD.....	6
2.2.1	<i>Matriz de trazabilidad de requerimientos.....</i>	<i>6</i>
2.2.2	<i>Referencia cruzada.....</i>	<i>6</i>
2.2.3	<i>Modelado específico de dominio.....</i>	<i>7</i>
2.2.4	<i>Etiquetado.....</i>	<i>8</i>
2.2.5	<i>Integración de documentos.....</i>	<i>8</i>
2.2.6	<i>Recuperación de información (IR).....</i>	<i>9</i>
2.3	MODELO DE TRAZABILIDAD.....	10
2.3.1	<i>Metamodelo de Ramesh y Jarke.....</i>	<i>10</i>
2.3.2	<i>Metamodelo de Carniel y Pegoraro.....</i>	<i>11</i>
2.3.3	<i>Metamodelo de Letelier.....</i>	<i>14</i>
2.4	TRAZABILIDAD & INGENIERÍA DE SOFTWARE.....	16
2.4.1	<i>Ingeniería de requerimientos y gestión.....</i>	<i>17</i>
2.4.2	<i>Gestión de proyectos.....</i>	<i>18</i>
2.4.3	<i>Demostración de cumplimiento.....</i>	<i>19</i>
2.4.4	<i>Diseño e Implementación.....</i>	<i>19</i>
2.4.5	<i>Prueba.....</i>	<i>19</i>
2.4.6	<i>Mantenimiento y evolución.....</i>	<i>20</i>
3	DESARROLLO ÁGIL.....	21
3.1	DEFINICIÓN.....	21
3.2	VALORES.....	21
3.3	PRINCIPIOS.....	22
3.4	DESARROLLO ÁGIL Y REQUERIMIENTOS.....	23
4	REQUERIMIENTOS & HISTORIAS DE USUARIO.....	26
4.1	REQUERIMIENTOS.....	26
4.1.1	<i>Clasificación de los requerimientos.....</i>	<i>26</i>
4.1.2	<i>Calidad de los requerimientos.....</i>	<i>27</i>
4.2	HISTORIAS DE USUARIO.....	28
4.2.1	<i>Calidad de una Historia de Usuario.....</i>	<i>29</i>
4.2.2	<i>Criterios de aceptación.....</i>	<i>32</i>
4.3	REQUERIMIENTOS VS HISTORIAS DE USUARIO.....	33
5	LÉXICO EXTENDIDO DEL LENGUAJE.....	34
5.1	DEFINICIÓN.....	34
5.2	CARACTERÍSTICAS.....	34
5.3	PROCESO DE CONSTRUCCIÓN.....	35

6	CONSTRUCCIÓN DE LEL DESDE LAS HISTORIAS DE USUARIO.....	39
6.1	INTRODUCCIÓN.....	39
6.2	PASOS.....	39
6.3	REGLAS.....	40
7	PROPUESTA DE TRAZABILIDAD.....	42
7.1	TÉCNICA DE TRAZABILIDAD DESARROLLADA.....	42
7.2	PROCESO DE REGISTRAR LA TRAZABILIDAD.....	43
7.2.1	<i>Visión general</i>	43
7.2.2	<i>Redactar ítems</i>	44
7.2.3	<i>Extracción de símbolos</i>	44
7.2.4	<i>Vincular Ítems con símbolos</i>	45
7.2.5	<i>Vincular fragmentos de código</i>	45
7.2.5.1	Forma Manual.....	46
7.2.5.2	Forma Semiautomática.....	47
7.2.6	<i>Buscar vínculos en el código</i>	47
7.3	MODELO DE TRAZABILIDAD DESARROLLADO.....	48
8	HERRAMIENTA DE TRAZABILIDAD.....	51
8.1	DESCRIPCIÓN GENERAL.....	51
8.2	ARQUITECTURA.....	51
8.3	ALMACENAMIENTO.....	52
8.4	FUNCIONALIDAD DE LA HERRAMIENTA.....	53
8.4.1	<i>Establecer trazas</i>	53
8.4.2	<i>Consultar trazas</i>	58
9	VALIDACIÓN DE LA PROPUESTA DE TRAZABILIDAD.....	61
9.1	PROYECTO PETCLINIC.....	61
9.2	PROYECTO SYLIUS.....	63
9.2.1	<i>Enfoque de validación</i>	64
9.2.2	<i>Análisis de tickets</i>	66
9.2.3	<i>Limitaciones</i>	72
9.2.4	<i>Beneficios</i>	73
9.2.5	<i>Resultado</i>	73
10	DISCUSIÓN SOBRE LA PROPUESTA DE TRAZABILIDAD.....	74
10.1	ACTIVIDADES REVISADAS.....	74
10.1.1	<i>Ingeniería y gestión de requerimientos</i>	74
10.1.2	<i>Gestión de proyectos</i>	76
10.1.3	<i>Demostración de cumplimiento</i>	77
10.1.4	<i>Diseño e Implementación</i>	78
10.1.5	<i>Prueba</i>	79
10.1.6	<i>Mantenimiento y evolución</i>	80
10.2	RESULTADO.....	83
10.3	BENEFICIOS.....	83
10.4	DESVENTAJAS.....	84
11	CONCLUSIÓN.....	85
11.1	CONSIDERACIONES FINALES.....	86
11.2	TRABAJO FUTUROS.....	86
12	TRABAJO RELACIONADOS.....	87

13	APÉNDICES.....	I
14	BIBLIOGRAFÍA.....	XV

Lista de Figuras

Figura 1: Tipos de trazabilidad - (Winkler & von Pilgrim, 2010).....	5
Figura 2: Referencia cruzada. (Cleland-Huang, 2012).....	7
Figura 3: Técnica de Etiquetado. (Moore, 1999).....	8
Figura 4: Plantilla de requerimientos y casos pruebas (Kolla & Banka, 2014).....	9
Figura 5: Metamodelo de Trazabilidad - (Ramesh & Jarke, s. f.).....	10
Figura 6: Metamodelo de trazabilidad en métodos ágiles - (Carniel & Pegoraro, 2018)	12
Figura 7: Metamodelo para trazabilidad de requerimientos - (Letelier, 2002).....	14
Figura 8: Uso de técnicas de captura de requerimientos 2003, 2008 y 2013.....	24
Figura 9: Ejemplo de termino del LEL.....	35
Figura 10: Proceso de construcción del LEL - (Antonelli et al., 2017).....	39
Figura 11: Vínculo entre ítems, símbolos y fragmentos de código.....	42
Figura 12: Proceso de trazabilidad propuesto.....	44
Figura 13: Relación entre ítems y símbolos.....	45
Figura 14: Relación entre ítem, símbolo y fragmento de código.....	46
Figura 15: Psuedocódigo de la generación de anotaciones.....	47
Figura 16: Proceso de definición del modelo de trazabilidad - (Taromirad & Paige, 2012).....	49
Figura 17: Modelo de trazabilidad desarrollado.....	50
Figura 18: Paquetes de la herramienta.....	51
Figura 19: Patrón Strategy para distintos lenguajes.....	52
Figura 20: DER de la herramienta.....	53
Figura 21: Opciones del menú procesamiento.....	54
Figura 22: Resultado de Find Objects.....	55
Figura 23: Resultado de buscar Owner en el código.....	56
Figura 24: Resultado de realizar la búsqueda de Snippets.....	56
Figura 25: Información del ítem US2.....	57
Figura 26: Resultado de buscar owner desde US2.....	57
Figura 27: Detalle de insertString.....	58
Figura 28: Opción de menu Get Items.....	58
Figura 29: Lista de ítems filtrados según criterio ingresado.....	59
Figura 30: Detalle del ítem US484.....	60
Figura 31: Cantidad de archivos por tipo. Petclinic.....	61
Figura 32: Lista de items de Petclinic.....	62
Figura 33: Cantidad de archivos por tipo. Sylus.....	63
Figura 34: Lista parcial de Items de Sylus.....	64
Figura 35: FlatRateConfigurationType.php.....	67
Figura 36: PerUnitRateConfigurationType.php.....	67
Figura 37: Commit asociado al ticket 10781.....	68
Figura 38: Cambios en OrderContext.php.....	70
Figura 39: Cambios en CreatePage.php UpdatePage.php.....	72
Figura 40: Actividades que pueden ser soportadas por la trazabilidad de requerimientos.....	83

Lista de Tablas

Tabla 1: Actividades respaldadas por la trazabilidad.....	16
Tabla 2: Artefactos en la Ingeniería de Requerimientos Agiles.....	25
Tabla 3: Posibilidad de vincular Símbolos con Clases/Métodos.....	46
Tabla 4: Criterios de búsqueda.....	55
Tabla 5: Tickets revisados.....	65
Tabla 6: Ticket 10790.....	66
Tabla 7: Ticket 10781.....	68
Tabla 8: Ticket 10679.....	68
Tabla 9: Ticket 10695.....	69
Tabla 10: Ticket 10658.....	69
Tabla 11: Ticket 10603.....	70
Tabla 12: Ticket 10806.....	71
Tabla 13: Ticket 10793.....	71
Tabla 14: Soporte de la propuesta a las actividades de Ingeniería y Gestión de Requerimientos.....	75
Tabla 15: Soporte de la propuesta a las actividades de Gestión de Proyectos.....	77
Tabla 16: Soporte de la propuesta a las actividades de Demostración de cumplimiento.....	78
Tabla 17: Soporte de la propuesta a las actividades de Diseño e Implementación.....	79
Tabla 18: Soporte de la propuesta a las actividades de Prueba.....	80
Tabla 19: Soporte de la propuesta a las actividades de Mantenimiento y evolución.	82

Apéndices

Apéndice I: Lista de ítems del proyecto Sylius.....	I
---	---

Lista de Abreviaciones y Símbolos

BDD	Desarrollo guiado por el comportamiento
CMMI	Modelo de capacidad de madurez integrado
DSML	Lenguaje de modelado específico del dominio
ER	Especificación de requerimientos
ERS	Especificación de requerimientos de software
IEEE	Instituto de ingenieros eléctricos y electrónicos
IDE	Entorno de desarrollo integrado
ISO	Organización internacional de estandarización
IR	Ingeniería de requerimientos
JITT	Trazabilidad justo a tiempo
LEL	Léxico extendido del Lenguaje
LSI	Indexación semántica latente
MDE	Ingeniería dirigida por modelos
MVC	Modelo vista controlador
PM	Modelo probabilístico
PO	Dueño del producto
ROMPL	Lenguaje de programación y modelado orientado a los requerimientos
RT	Trazabilidad de requerimientos
SLC	Ciclo de Vida del Software
TIM	Modelo de trazabilidad de información
UdeD	Universo de discurso
VSM	Modelo de espacio vectorial
XP	Programación Extrema

1 Introducción

1.1 Motivación

Como en muchas oportunidades la motivación principal de este trabajo radica en la necesidad del tesista de realizar bien el trabajo diario, como así también la menor cantidad de trabajo innecesario. Diariamente como programador, el tesista enfrenta la tarea de realizar mantenimiento a sistemas que no han sido documentados exhaustivamente o bien si en algún momento lo fueron a medida que se realizan cambios la documentación existente queda desactualizada, en otras oportunidades la documentación ni siquiera existe, el problema central es que al momento de realizar cambios en el software debido a un requerimiento nuevo o a la modificación de uno existente, se torna difícil cuando no imposible medir el impacto del cambio.

Ante esta situación, la solución que aquí se pretende desarrollar, se basa en dos premisas, la primera que el cambio en los requerimientos está siempre presente (*Principios del Manifiesto Ágil*, 2001), hace a la esencia del software (Brooks, 1986), por lo que es necesario buscar una forma adecuada para manejar los requerimientos como así también los cambios en los mismos.

Para la tarea de registrar requerimientos se utilizarán las historias de usuario, y se complementarán las mismas construyendo un LEL, como se menciona en (Antonelli et al., 2017). Esta transformación realiza una descomposición de los elementos que conforman una historia de usuario y posteriormente mediante la aplicación de una serie de reglas, construye un LEL. Este proceso permitirá consolidar la información proveniente de las historia de usuario (funcionalidad del sistema, roles, entidades, contextos de uso) y facilitar el acceso a dicha información, por otro lado este LEL también servirá como índice de requerimientos.

La segunda premisa que orienta este trabajo es que “el código es la descripción más precisa y confiable de lo que hace un sistema y cómo fue diseñado” (Turk et al., 2005), por esto se buscará una forma de vincular los requerimientos con las porciones de código (clases, métodos, atributos, etc) que estén relacionadas con el mismo, además se pretende registrar junto al código mismo la información necesaria que permita gestionar la trazabilidad de los requerimientos.

Revisando en la clasificación de (Gotel & Finkelstein, 1994), se puede dividir la trazabilidad en pre-trazabilidad, que se refiere a aquellos aspectos de la vida de un requisi-

to antes de su inclusión en la Especificación de Requerimientos (ER) y pos-trazabilidad, que se refiere a aquellos aspectos de la vida de un requisito que resultan de su inclusión en la ER. Utilizando las historias de usuario como herramienta para la ER, con el LEL derivado de las mismas se puede abarcar la pre-trazabilidad de los requerimientos, restaría elaborar una forma de relacionar las historia de usuario con el código, para abarcar la pos-trazabilidad de los requerimientos. Finalmente se pretende utilizar anotaciones como medio para registrar el vínculo entre las historias de usuario, los elementos del LEL y el código fuente.

Junto con el pretendido beneficio de medir el impacto del cambio en los requerimientos, se espera obtener beneficios adicionales, en áreas tales como: gestión de proyecto, visibilidad del proyecto, verificación & validación y mantenimiento. Se suma a esto el hecho que la trazabilidad es un componente importante en estándares tales como CMMI e ISO 9001:2000 (Kannenberg & Saiedian, 2009).

Además de los beneficios que ofrece la trazabilidad, “su implementación enfrenta muchos desafíos, especialmente en las áreas de costo, gestión del cambio, problemas organizacionales y soporte deficiente de herramientas” (Kannenberg & Saiedian, 2009). El costo se hace presente en términos de tiempo y esfuerzo: “La captura de estas interdependencias (trazas) proporciona beneficios explícitos para la identificación de conflictos de requisitos, análisis de impacto de cambios, planificación de lanzamientos, etc., pero la alta complejidad y el esfuerzo necesario de rastreo hace que el seguimiento de los requisitos sea demasiado costoso para su uso en la práctica.” (Heindl & Biffel, 2005).

El problema que se presenta al gestionar el cambio es que cuando los cambios ocurren, es necesario actualizar los datos de trazabilidad para reflejar dichos cambios. Para poder realizar esta tarea, es necesario que quienes actualizan la información de trazabilidad sean disciplinados. “Enfocarse en los beneficios a largo plazo de la trazabilidad en lugar de los costos a corto plazo, puede ayudar a una organización a mantener una actitud positiva frente a los costos de mantenimiento de los datos de trazabilidad en medio del cambio” (Kannenberg & Saiedian, 2009).

Algunos de los problemas organizacionales asociados a la trazabilidad pueden ser los siguientes, muchas organizaciones ven la trazabilidad como un mandato o como una herramienta para el cumplimiento de un estándar. “Muchas organizaciones no capacitan a sus empleados con respecto a la importancia de trazabilidad” (Gotel & Finkels-

tein, 1994) o bien no cuentan con políticas definidas a relacionadas con la trazabilidad.

Con la propuesta a desarrollar se busca solucionar algunos de los desafíos que están presentes en la trazabilidad, principalmente aquellos relacionados con el costo y la gestión del cambio.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar un proceso que vincule las Historia de Usuario y el Léxico Extendido del Lenguaje (LEL) derivado de ellas, con el código fuente para facilitar la trazabilidad de los requerimientos.

1.2.2 Objetivos específicos

- Proponer un proceso que vincule las Historia de Usuario y el LEL resultante con el código fuente.
- Implementar el proceso propuesto.
- Analizar los resultados del proceso implementado.
- Determinar las ventajas y desventajas del proceso propuesto.

El resultado de esta tesis, será contar con un proceso que facilite la trazabilidad de los requerimientos, esto será posible a partir de contar con el requerimiento expresado mediante una historia de usuario, desde allí se derivará un LEL cuyos elementos se corresponderán posteriormente con elementos del paradigma orientado a objetos (clases, atributos o métodos).

En este trabajo también se desarrollará una librería, la cual hará efectiva la vinculación entre los elementos del LEL y el código fuente. Esta librería será probada mediante la implementación del proceso propuesto en un caso práctico.

2 Trazabilidad

En el presente capítulo se presentan distintos conceptos de trazabilidad, las principales técnicas de trazabilidad y los modelos definidos en la actualidad.

2.1 Definición

Inicialmente se menciona a la trazabilidad como (Greenspan & McGowan, 1978) una propiedad de una técnica de descripción del sistema que permite realizar cambios en una de las tres descripciones del sistema (requisitos, especificaciones, implementación) para rastrear las porciones correspondientes de las otras descripciones. La correspondencia debe mantenerse durante la vida útil del sistema. Este concepto ha sido refinado y según el estándar (*IEEE Std. 610-1990 - IEEE Standard Computer Dictionary*, 1991), la trazabilidad es:

“(1) El grado en que una relación puede ser establecida entre dos o más productos del proceso de desarrollo, especialmente en aquellos que presentan relaciones predecesor-sucesor o maestro-subordinado entre sí; por ejemplo, el grado en que un requerimiento y el diseño de un componente de software dado coinciden.

(2) El grado en que cada elemento de un producto de desarrollo de software establece su razón de existir; por ejemplo, el grado en que cada elemento de un diagrama de burbuja referencia el requerimiento que éste satisface”

Una definición ampliamente aceptada, es la de (Gotel & Finkelstein, 1994), “La trazabilidad de los requerimientos se refiere a la capacidad de describir y seguir la vida de un requerimiento, tanto hacia delante como hacia atrás (es decir, desde su origen, a través de su desarrollo y especificación, hasta su posterior despliegue y uso, y durante todos los períodos de refinamiento e iteración en curso en cualquiera de estas fases) ”.

Junto a este concepto, se introduce la idea de dos tipos básicos de trazabilidad, pre-trazabilidad, que se refiere a aquellos aspectos de la vida de un requisito antes de su inclusión en la Especificación de Requerimientos (ER) y pos-trazabilidad, que se refiere a aquellos aspectos de la vida de un requisito que resultan de su inclusión en la ER.

Ramesh menciona los conceptos de trazabilidad vertical y horizontal (Ramesh & Edwards, 1992); “la trazabilidad vertical se refiere a la asociación de diferentes tipos de objetos del ciclo de vida de software, normalmente creados en diferentes procesos de Ciclo de Vida del Software (SLC). Un ejemplo de una relación de trazabilidad vertical sería entre la declaración de requisitos y la declaración de diseño.”

La trazabilidad horizontal se refiere a la "asociación de objetos SLC del mismo tipo (típicamente creados en el mismo proceso SLC). Un ejemplo de este tipo de rastreabilidad incluye las relaciones padre/hijo entre los diagramas de flujo de datos descompuestos y la relación "derivada de" entre los enunciados de requisitos". "Cuando te mueves horizontalmente es cuando estás analizando qué proceso está dentro de qué proceso". La trazabilidad horizontal equivale a “el subproceso transfiere datos a otro subproceso como los niveles primitivos tienen que hablar entre sí, etc. ". La trazabilidad horizontal también se refiere a las relaciones entre diferentes vistas del mismo nivel de representación.

Figura 1: Tipos de trazabilidad - (Winkler & von Pilgrim, 2010)

El presente trabajo pretende abarcar las trazas que corresponden con la trazabilidad vertical que tiene a lugar entre las historias de usuario y el código fuente, como así también la trazabilidad horizontal que ocurre entre distintas historias de usuario, vinculadas mediante una relación historias épicas e historia de usuario, considerando solamente las trazas correspondiente a la post-trazabilidad.

2.2 Técnicas de trazabilidad

Existen diversas técnicas utilizadas para gestionar las trazas de los requerimientos en todo tipos de proyectos y desde hace mucho tiempo (R. Watkins, y M. Neal, 1994) algunas de ellas han sido adaptadas o creadas para proyectos ágiles (Kodali, 2015).

A continuación se describen algunas de las técnicas más frecuentemente utilizadas, finalmente se indica cual es la empleada en este trabajo.

2.2.1 Matriz de trazabilidad de requerimientos

La matriz de trazabilidad de requerimientos es utilizada para relacionar requerimientos con distintos artefactos software. Es una grilla de dos dimensiones que tiene requerimientos en una de las dimensiones, mientras que en la restante, tiene requerimientos o bien otros tipos de elementos, (casos de prueba, usuarios, etc). Mas precisamente si la celda c_{ij} está marcada, significa que el requerimiento de la columna i está vinculado con el artefacto de la fila j .

2.2.2 Referencia cruzada

En esta técnica, cada artefacto se enumera utilizando lenguaje natural y se proporciona una lista de enlaces a otros artefactos relacionados.

Un ejemplo del uso de ésta técnica, es el trabajo realizado por (Cleland-Huang, 2012), en el mismo se puede ver como se relacionan las historia de usuario con los test de aceptación. Sin embargo, el código se trata como un único artefacto de alto nivel, y no hay información sobre qué fragmento de código está relacionado con qué casos de prueba. Otro ejemplo de esta técnica es el trabajo realizado por (Ratanotayanon et al., 2009) quienes diseñaron una herramienta, Zelda, para trabajar con un proceso de desarrollo ágil que captura y mantiene enlaces entre la información de alto nivel y el código.

go fuente. Esta herramienta es un complemento de Eclipse que ayuda a los desarrolladores a crear enlaces desde historias de usuarios y tareas a código fuente, casos de prueba y varios archivos basados en texto.

Figura 2: Referencia cruzada. (Cleland-Huang, 2012)

2.2.3 Modelado específico de dominio

Algunos investigadores intentaron investigar la combinación de Model Driven Engineering (MDE) y la ingeniería de requisitos. En esta dirección, (Badreddin et al., 2014) propuso un Lenguaje de Programación y Modelado Orientado al Requisito (ROMPL). En esta técnica, los autores proporcionan una sintaxis textual para los conceptos de requisitos e integrarlos dentro del código orientado a modelos. ROMPL admite la generación automática y el mantenimiento de los enlaces entre requisitos, modelos y código.

Un esfuerzo similar es el realizado por (Taromirad & Paige, 2012), que propuso un enfoque de trazabilidad de requerimientos en MDE mediante un modelado específico de dominio. El autor sugirió usar el lenguaje de modelado específico del dominio (DSML) para construir un esquema de trazabilidad para un dominio o proyecto específico, de forma incremental. La idea aquí es describir un meta-modelo RT específico de dominio y representar las estructuras, comportamientos y características del domi-

nio objetivo con respecto a los objetivos RT del proyecto. Luego, las herramientas MDE se usan para crear modelos de trazabilidad de requerimientos y ayudan a evolucionar modelos y metamodelos a proyectos específicos.

2.2.4 Etiquetado

La etiqueta es una palabra clave o término no jerárquico asignado a una pieza de información. Cada elemento en un artefacto debe tener un identificador único. Seguidamente esos identificadores se utilizan como etiquetas en los artefactos posteriores para identificar el trazado hacia atrás del artefacto predecesor (Moore, 1999). Ver Figura 3. En el trabajo de (Jacobsson, 2009) se realizó una serie de entrevistas con diferentes personas y roles de scrum. En base a los resultados de la entrevista, se identificó varios tipos de trazas. Por ejemplo, para las trazas entre requisitos y pruebas, propuso marcar una prueba con una ID de requisitos o agregar una ID de prueba a un requisito y usar ambos para una conexión de seguimiento bidireccional.

Figure 4: Example of Trace Tagging

Figura 3: Técnica de Etiquetado. (Moore, 1999)

2.2.5 Integración de documentos

Esta técnica fue desarrollada por (Kolla & Banka, 2014), en ella recolectaron datos por medio de una encuesta enfocada en compañías de software que asumen diferentes roles y responsabilidades en el campo de pruebas. Con base en su encuesta y hallaz-

gos de investigación, concluyeron fusionar requisitos y casos de prueba en un solo documento. Con este fin, han desarrollado una plantilla, ver Figura 4, y descrito el procedimiento paso a paso utilizado para crear esa plantilla, que se puede usar para los requisitos y las pruebas en un desarrollo de software ágil.

Table 23 Derived Template.

Requirement ID	Requirement Name	Source		
Title	Description	Dependencies		
Author	Status	Priority		
Test Case ID				
Tester ID				
Test Scenario	Test Case Description	Role performing Test Case		
Test Steps	Expected Result	Actual Result	Pass/Fail	Description of the Defect
Test Step where defect occurs	Defect Severity			

Figura 4: Plantilla de requerimientos y casos pruebas (Kolla & Banka, 2014)

2.2.6 Recuperación de información (IR)

Las técnicas de IR, como la indexación semántica latente (LSI), el modelo probabilístico (PM) y el modelo de espacio vectorial (VSM) indexan los documentos en un espacio de documento así como las consultas extrayendo información sobre las ocurrencias de términos dentro de ellos. A menudo, el resultado se describe mediante una matriz de término por documento. La entrada a_{ij} de esta matriz denota una medida del peso del i -ésimo término en el j -ésimo documento.

En su trabajo (Cleland-Huang et al., 2007), utiliza la técnica de recuperación de infor-

mación para crear automáticamente enlaces RT candidatos entre historias de usuario y código. La herramienta Just-In-Time Traceability (JITT) proporciona un entorno interactivo para recuperar códigos potencialmente relevantes. La herramienta devuelve una lista de clases candidatas y muestra la probabilidad de que la clase recuperada esté relacionada con la historia del usuario.

2.3 Modelo de trazabilidad

Para representar la trazabilidad resulta necesario contar con un modelo adecuado que permita expresar todas las relaciones entre los distintos objetos del ciclo de vida de software. Para la construcción de dicho modelo, resulta conveniente revisar los metamodelos existentes en busca de aquel que facilite la elaboración de un modelo propio que se adapte a las necesidades de la propuesta de este trabajo.

Se mencionan a continuación varias propuestas de metamodelos de trazabilidad que pueden ser útiles para esta tesis, por otro lado en un capítulo posterior se presenta el modelo desarrollado para soportar la propuesta de trazabilidad del presente trabajo.

2.3.1 Metamodelo de Ramesh y Jarke

Los aspectos esenciales de la trazabilidad pueden ser capturados en un modelo muy simple, como el que se presenta en la Figura 5.

Figura 5: Metamodelo de Trazabilidad - (Ramesh & Jarke, s. f.)

Cada entidad y enlace en el metamodelo pueden especializarse e instanciarse para crear modelos de trazabilidad específicos para cada proyecto.

Según (Ramesh & Jarke, s. f.) el metamodelo propuesto puede ser utilizado para las siguientes dimensiones de la información de trazabilidad.

- ¿Qué información se representa, incluidos los atributos destacados o las características de la información?
- ¿Quiénes son los INTERESADOS que desempeñan diferentes roles en la creación, mantenimiento y uso de varios OBJETOS y enlaces de trazabilidad a través de ellos?
- ¿Dónde está representado, en términos de FUENTES que "documentan" información de trazabilidad?
- ¿Cómo se representa esta información, tanto por medios formales como informales, y cómo se relaciona con otros componentes de la trazabilidad?
- ¿Por qué se creó, modificó o evolucionó un determinado OBJETO conceptual?
- ¿Cuándo esta información fue capturada, modificada y evolucionó?

2.3.2 *Metamodelo de Carniel y Pegoraro*

En esta propuesta, se identificaron los supuestos que guiaron la construcción del metamodelo para la trazabilidad de los requisitos a través de la dependencia entre las historias de los usuarios y el análisis del impacto en los métodos ágiles. Posteriormente, el metamodelo se construyó para cumplir con los supuestos definidos. En el siguiente paso, se analizó el metamodelo propuesto en cada una de sus relaciones y se confirmaron los supuestos correspondientes. Finalmente, el metamodelo propuesto se comparó con otros metamodelos existentes ubicados en la revisión de la literatura.

Figura 6: Metamodelo de trazabilidad en métodos ágiles - (Carniel & Pegoraro, 2018)

En la Figura 6, se puede observar el diagrama de clases correspondiente al modelo de trazabilidad desarrollado por (Carniel & Pegoraro, 2018).

El conjunto de supuestos definidos para construir el metamodelo propuesto se presenta a continuación:

1. Los requisitos se escriben en el formato de historias de usuarios: un requisito que se especifica en el formato de una historia de usuarios describe funcionalidades con valor comercial para las partes interesadas.
2. Las historias de los usuarios se documentan de manera estructurada: las historias de los usuarios tienen una estructura de documentación que permite identificar la información de los requisitos. Esta estructura generalmente se escribe en la forma:

Como <interesado>,
 Puedo <característica>,
 para que <valor de negocio>.

3. Las historias de usuarios tienen personas con roles involucrados, llamadas partes interesadas: las historias de usuarios son escritas, desarrolladas y validadas por miembros del equipo, clientes y otras personas llamadas partes interesadas, que tienen influencia en las historias de usuarios e interés en los requisitos. Las historias de los usuarios no contienen información técnica, ya que están escritas con términos de lenguaje simples, con el objetivo de representar la vista del usuario. En el método ágil Scrum, la parte interesada se llama Product Owner.
4. Las historias de los usuarios se escriben y desarrollan a lo largo de las iteraciones: las iteraciones o sprints son períodos de tiempo predefinidos por las partes interesadas, que determinan qué historias se desarrollarán durante la iteración. Una o más iteraciones pueden ser parte de una versión, que es una versión del producto entregada al cliente y comprende un conjunto de características. La lista de historias de usuarios evoluciona constantemente a lo largo del proyecto.
5. Las historias de usuario pueden tener dependencias entre ellas: las dependencias pueden estar relacionadas con el negocio o la técnica (proyecto o implementación).
6. Las historias de usuario generan tareas que también pueden tener dependencias: las historias de usuario generan tareas para desarrollar en una iteración. Las tareas son actividades de implementación, arquitectura o pruebas que producen resultados tangibles. Las tareas están intrínsecamente relacionadas y pueden depender unas de otras. Estas dependencias se producen entre tareas de la misma historia de usuario y entre diferentes tareas de historia.
7. Cambiar una historia de usuario podría afectar otras historias de usuario: el impacto puede ocurrir entre requisitos (reglas de negocio) y entre componentes (proyectos relacionados con la construcción del sistema, generación de

errores). Los cambios técnicos pueden conducir a la refactorización de código o arquitectura.

8. Las dependencias entre historias de usuarios y entre tareas pueden causar impacto: las historias de usuarios, así como las tareas, están intrínsecamente relacionadas entre sí. Estas relaciones pueden ser dependencias que impactan las historias cuando hay un cambio. Es posible, a través del análisis del usuario, asignar una intensidad de impacto a las dependencias entre historias y entre tareas.

2.3.3 Metamodelo de Letelier

Esta propuesta, (Letelier, 2002), pretende almacenar los siguientes tipos de información asociados a la trazabilidad de los requisitos y sus posibles usos:

Los vínculos de trazabilidad entre diferentes tipos de especificaciones permiten validar que la funcionalidad del sistema cubre las expectativas de los interesados, que no se ha implementado una funcionalidad superflua y realizar un análisis de impacto cuando los requisitos cambian.

Figura 7: Metamodelo para trazabilidad de requerimientos - (Letelier, 2002)

Los vínculos entre las partes interesadas y las especificaciones permiten mejorar la comunicación y la cooperación entre las partes interesadas, y garantizar que la contribución de cada persona sea considerada y registrada.

Las razones asociadas a las especificaciones, incluidas las alternativas, decisiones, suposiciones, etc. contribuyen a mejorar la comprensión y aceptación del sistema por parte de los interesados, y a mejorar la gestión del cambio evitando estudiar nuevamente las consideraciones ya excluidas. Esto es posible gracias a hacer accesibles las soluciones, sus fundamentos y las alternativas excluidas.

En la Figura 7, se puede observar el diagrama de clases de esta propuesta.

2.4 Trazabilidad & Ingeniería de Software

Se presenta a continuación una serie de actividades propias de un proyecto de software que pueden ser respaldadas por la trazabilidad de los requerimientos.

Estas actividades han sido enumeradas por (Bouillon et al., 2013) y conforman un grupo de 29 actividades típicas de los procesos de desarrollo, ver Tabla 1; las mismas están organizadas en las siguientes áreas:

1. Ingeniería de requerimientos y gestión,
2. Gestión de proyectos,
3. Demostración de cumplimiento,
4. Diseño e Implementación,
5. Prueba y
6. Mantenimiento y evolución

Tabla 1: Actividades respaldadas por la trazabilidad

1) Ingeniería y gestión de requerimientos	
a)	Encontrar el origen y la justificación de los requerimientos, es decir, la trazabilidad de los pre-requerimientos a regulaciones y a otras fuentes de requerimientos
b)	Refinamiento y detalle de requerimientos
c)	Documentar el historial de un requerimiento, es decir, para poder rastrear versiones anteriores de un requerimiento con el fin de conocer los cambios
d)	Identificar a las partes interesadas para el desarrollo continuo de los requerimientos.
e)	Análisis de calidad y madurez de los requerimientos.
f)	Análisis de impacto, que otras partes interesadas son importantes por un cambio en un requerimiento
2) Gestión de proyectos	
a)	Siguiendo el estado de requerimiento o implementación de tareas en detalle
b)	Planificación de entrega inicial
c)	Evaluación de progreso a nivel de proyecto o subproyecto para obtener una visión general de los requerimientos ya implementados
d)	Asignación de tareas a las partes interesadas, por ejemplo, asignación de un requerimiento a un desarrollador para su implementación
e)	Notificación a las partes interesadas sobre los cambios, por ejemplo, después de un cambio en un requerimiento, todos los propietarios de artefactos dependientes son informados automáticamente
f)	Ajuste del proyecto y plan de entrega, por ejemplo, en caso de que el límite de tiempo exceda

3) Demostración de cumplimiento

- a) Análisis de la cobertura de requerimientos en el código fuente, por ejemplo, para el cliente
- b) Documentación de trazabilidad para fines de certificación.
- c) Justificación de todo el código escrito basado en la especificación para propósitos de certificación

4) Diseño e Implementación

- a) Navegación entre especificación, diseño, prueba y código a través de trazas
- b) Navegación dentro de artefactos del mismo tipo, por ejemplo, dentro del código fuente
- c) Evaluación de diseño basada en métricas de trazabilidad, por ejemplo, para encontrar componentes que contengan demasiada funcionalidad y que deberían dividirse
- d) Comprensión de los artefactos de software, por ejemplo, familiarización del proyecto de los miembros del equipo de desarrollo.

5) Prueba

- a) Desarrollo de casos de prueba basados en requerimientos.
- b) Ubicación del defecto dentro del código fuente para casos de prueba fallidos
- c) Descubrir pruebas de regresión para ejecutar después del cambio de código
- d) Prueba de análisis de cobertura de especificación y código
- e) Identificación de las partes interesadas para comprender el comportamiento y resolver problemas complicados.

6) Mantenimiento y evolución

- a) Cambiar el análisis de impacto para determinar los artefactos afectados por una extensión de característica
- b) Cambiar la estimación del esfuerzo para extensiones de características
- c) Ubicación de características y soporte durante la implementación del cambio mediante el uso de trazas
- d) Reutilización de componentes de especificación y código, por ejemplo, una característica con toda su implementación
- e) Transferencia de conocimiento al equipo de mantenimiento, por ejemplo, en los casos en que un equipo realiza mantenimiento que no incluye ninguno de los miembros originales del equipo.

A continuación se realiza una breve introducción para cada área y posteriormente algunos hallazgos realizados por (Bouillon et al., 2013) referidos a las actividades que agrupa.

2.4.1 Ingeniería de requerimientos y gestión

La “Ingeniería de requerimientos es el conjunto de actividades que se ejecutan con el propósito de entender las necesidades de los usuarios y traducirlas en afirmaciones precisas y no ambiguas.” (Oliveros, 2014), mientras que la Gestión de Requerimientos es una actividad perteneciente a la Ingeniería de Requerimientos que incluye todas las actividades que mantienen la integridad, precisión y vigencia de los acuerdos de requerimientos a lo largo del proyecto. Las actividades centrales de la gestión de requi-

sitos se pueden agrupar en cuatro categorías principales: control de versiones, control de cambios, seguimiento del estado de los requisitos y seguimiento de los requisitos.

Según la encuesta realizada por (Bouillon et al., 2013), todas las actividades de esta área se utilizan en al menos el 70% de los proyectos informados, ya sea de forma regular u ocasional.

2.4.2 Gestión de proyectos

La gestión de proyectos de software es una actividad sombrilla dentro de la ingeniería de software. Comienza antes de iniciar cualquier actividad técnica y continúa a lo largo del modelado, construcción y despliegue del software de cómputo. (Pressman, 2010).

Los siguientes elementos tienen influencia sustancial sobre la administración del proyecto de software: personal, producto, proceso y proyecto. El personal debe organizarse en equipos eficaces, motivados para hacer trabajo de software de alta calidad, y coordinarse para lograr comunicación efectiva. Los requerimientos del producto deben comunicarse de cliente a desarrollador, dividirse (descomponerse) en sus partes constitutivas y ubicarse para su trabajo por parte del equipo de software. El proceso debe adaptarse al personal y al producto. Se selecciona un marco conceptual común al proceso, se aplica un paradigma de ingeniería de software adecuado y se elige un conjunto de tareas de trabajo para realizar el trabajo. Finalmente, el proyecto debe organizarse de forma que permita triunfar al equipo de software.

En su trabajo, (Bouillon et al., 2013), considera la gestión de proyectos como un área popular para la aplicación de la trazabilidad de requisitos. Todas las actividades del área se realizan en el 72% de los proyectos reportados, ya sea de forma regular u ocasional. Con un 75%, “Siguiendo el estado de requerimiento o implementación de tareas en detalle” es la actividad realizada con mayor regularidad de las 29 actividades evaluadas.

2.4.3 Demostración de cumplimiento

Desde el punto de vista de (Bouillon et al., 2013), la demostración del cumplimiento no es una parte tradicional de un proceso de desarrollo, sino más bien un beneficio citado importante de la trazabilidad de los requisitos. Además, la "demostración de la cobertura de requerimientos en el código fuente" se encuentra entre las actividades más utilizadas de los 29 evaluados.

2.4.4 Diseño e Implementación

El diseño se define en (Project Management Institute & Project Management Institute, 2013) como el proceso para definir la arquitectura, los componentes, los interfaces, y otras características de un sistema o un componente y el resultado de este proceso. Visto como proceso, el diseño del software es la actividad del ciclo de vida de la cual los requisitos del software se analizan para producir una descripción de la estructura interna del software que servirá como la base para su construcción.

La implementación del software es el proceso que transforma el comportamiento, las interfaces y las restricciones de implementación especificadas en acciones que crean un elemento del sistema implementado como un producto o servicio de software. Este proceso da como resultado un elemento de software que satisface los requisitos de diseño arquitectónico mediante la verificación y los requisitos de las partes interesadas mediante la validación. («The ISO/IEC & IEEE/EIA Standard 12207 - Standard for Information Technology - Software Life Cycle Processes», 2011).

Las actividades presentadas en esta área, se refieren a la navegación entre artefactos, a la evaluación del diseño de un sistema basado en la trazabilidad y a la familiarización del equipo de desarrollo con el proyecto mediante el uso de la trazabilidad. (Bouillon et al., 2013)

2.4.5 Prueba

Las pruebas del software consisten en verificar el comportamiento de un programa dinámicamente a través de un grupo finito de casos de prueba, debidamente seleccionados del, típicamente, ámbito de ejecuciones infinito, en relación al comportamiento esperado. (Project Management Institute & Project Management Institute, 2013)

En su trabajo, (Bouillon et al., 2013), encontró que las pruebas se citan regularmente como un área de desarrollo de software que se supone se beneficia enormemente de la trazabilidad de los requisitos establecidos. El “Desarrollo de casos de prueba basados en requerimientos” es la actividad más utilizada en ésta área (61%). Además menciona que la aplicación de la trazabilidad para las actividades de prueba está asociada con el tamaño y la duración del proyecto. Los proyectos más pequeños y más cortos aplican todos los escenarios con menos frecuencia, mientras que los proyectos más grandes y más largos los aplican con más frecuencia. Una razón para este hallazgo podría ser las diferencias en los procedimientos generales de prueba en proyectos más pequeños y más cortos en comparación con los más grandes y más largos.

2.4.6 Mantenimiento y evolución

El mantenimiento de software se define en (Pigoski, 2001), como la modificación de un producto de software después de la entrega para corregir fallas, mejorar el rendimiento u otros atributos, o adaptar el producto a un entorno modificado.

El mantenimiento de software sustenta el producto de software a lo largo de su ciclo de vida. Las solicitudes de modificación se registran y se realiza un seguimiento, se determina el impacto de los cambios propuestos, se modifica el código, se realizan pruebas y se lanza una nueva versión del producto de software.

Para (Bouillon et al., 2013) las tres actividades (6.a, 6.b y 6.c) que tratan con la implementación de cambios se utilizan en alrededor del 70% de los proyectos informados, pero su uso regular está detrás de los escenarios populares que respaldan la ingeniería y gestión de requisitos, la gestión de proyectos y las actividades de demostración de cumplimiento. Este hallazgo es interesante, ya que se debe asumir que la trazabilidad es especialmente útil cuando se trata de cambios y reevaluación de decisiones tomadas en el pasado.

3 Desarrollo Ágil

En el presente capítulo se presentan los elementos esenciales del Desarrollo Ágil, Concepto, Valores y Principios. Al final de este capítulo, se presentarán las técnicas de captura de requerimientos típicamente utilizadas en Metodologías Ágiles.

3.1 Definición

El Desarrollo Ágil es un conjunto de valores y una colección relacionada de métodos y técnicas para gestionar y entregar proyectos (*A Guide to Scrum and CMMI ®: Improving Agile Performance with CMMI*, 2016). Este enfatiza el desarrollo iterativo e incremental, la estrecha colaboración con el cliente y los equipos interfuncionales para crear software que funciona. Basado en un conjunto de valores y principios descritos en el Manifiesto Ágil, el desarrollo ágil incluye una serie de frameworks como Scrum y Extreme Programming (XP).

3.2 Valores

1. Individuos e interacciones sobre procesos y herramientas

Los procesos deben ser una ayuda y un soporte para guiar el trabajo. Deben adaptarse a la organización, a los equipos y a las personas; y no al revés. La defensa a ultranza de los procesos lleva a postular que con ellos se pueden conseguir resultados extraordinarios con personas mediocres, y lo cierto es que este principio es peligroso cuando los trabajos necesitan creatividad e innovación.

2. Software funcionando sobre documentación extensiva

El manifiesto no afirma que no hagan falta. Los documentos son soporte del software, permiten la transferencia del conocimiento, registran información histórica, y en muchas cuestiones legales o normativas son obligatorios, pero se resalta que son menos importantes que los productos que funcionan. Menos trascendentales para aportar valor al producto.

Los documentos no pueden sustituir, ni pueden ofrecer la riqueza y generación de valor que se logra con la comunicación directa entre las personas y a través de la interacción con los prototipos. Por eso, siempre que sea posible debe preferirse, y reducir al mínimo indispensable el uso de documentación, que genere trabajo que no aporta un valor directo al producto.

3. Colaboración con el cliente sobre negociación contractual

Para el desarrollo ágil el valor del resultado no es consecuencia de haber controlado una ejecución conforme a procesos, sino de haber sido implementado directamente sobre el producto. Un contrato no aporta valor al producto. Es una formalidad que establece líneas divisorias entre responsabilidades, que fija los referentes para posibles disputas contractuales entre cliente y proveedor.

En el desarrollo ágil el cliente es un miembro más del equipo, que se integra y colabora en el grupo de trabajo. Los modelos de contrato por obra no encajan.

4. Respuesta ante el cambio sobre seguir un plan

Para un modelo de desarrollo que surge de entornos inestables, que tienen como factor inherente el cambio y la evolución rápida y continua, resulta mucho más valiosa la capacidad de respuesta que la de seguimiento y aseguramiento de planes pre-establecidos. Los principales valores de la gestión ágil son la anticipación y la adaptación; diferentes a los de la gestión de proyectos ortodoxa: planificación y control para evitar desviaciones sobre el plan.

3.3 Principios

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

3.4 Desarrollo Ágil y Requerimientos

En el desarrollo ágil distintos artefactos son utilizados para realizar tareas de captura, especificación y gestión de los requerimientos, en esta sección buscamos identificar

cuáles son los principales artefactos empleados y particularmente, busquemos establecer el lugar que ocupan las historias de usuarios dentro de este grupo de artefactos.

En su trabajo, (Kassab, 2015), considera a las historias de usuario como una técnica para la captura de requerimientos que emerge con mucha fuerza. Ver Figura 8.

Figura 8: Uso de técnicas de captura de requerimientos 2003, 2008 y 2013.

Con el paso del tiempo el uso de las historias de usuario se ha incrementado, llegando a ocupar un primer lugar dentro de las metodologías ágiles, como lo menciona (Schön et al., 2017) en su revisión sistemática de la literatura. Ver en la Tabla 2.

Tabla 2: Artefactos en la Ingeniería de Requerimientos Ágiles

Artifact	Description	Reference	Percentage (N=27)
User Story	User story is a description of a feature written from the perspective of the person who needs this. It consists of a written text, conversation about it and acceptance criteria.	[56], [42], [57], [45], [58], [46], [47], [48], [49], [62], [50], [38], [53], [39], [40]	56%
Prototype	Prototype is a model of the software application that supports the evaluation of design alternatives and communication.	[41], [56], [58], [46], [49], [48], [36], [50], [52], [54], [39], [40], [55]	41%
Use Case	Use case describes an action or event steps, which are needed to achieve a goal.	[42], [58], [36], [62], [59], [54], [55]	26%
Scenario	Scenario is a textual representation of a problem and describes the interaction between user and system in a specific context.	[56], [42], [58], [49], [40], [55]	22%
Story Card	Story Card is a physical representation for the written text and details from a user story.	[57], [62], [37], [38], [39]	22%
Persona	Persona is a description of a fictitious person that represents a larger part of the target group.	[56], [45], [48], [49]	15%
Vision	Vision is an abstract description of the overarching goal that guides product development and aligns development, business people and other stakeholder.	[57], [45], [55]	11%
UML diagram	Unified Modeling Language (UML) provides a standard to visualize the design of a system.	[44], [52], [53]	11%
Storyboard	With a storyboard the workflow of the user is presented in a sequence of pictures.	[50], [54], [40]	11%
Task	One user story is split into more tasks, which describe more technical requirements.	[45], [40]	7%
Kanban board	Kanban board visualizes the progress of a requirement through the workflow of the development team.	[57], [45]	7%
UI pattern	UI pattern describes an abstract solution for recurring design problems and give inspiration to designer.	[53], [55]	7%
Essential use cases	Essential use case describes user tasks and is a simplified and generalized form of use cases.	[53], [55]	7%
Pictures	Picture is a visual representation (e.g. photograph, painting)	[41], [50]	7%
Videos	Video consists of a sequence of images processed electronically that are seen in a recording and displayed on a screen.	[41], [50]	7%
Mind Map	Mind map is a diagram used to visualize and organize information.	[47], [51]	7%
UI specification	Written specification that describes the UI of a system. The text is enriched by mock-ups, icons, etc.	[56], [49]	7%

4 Requerimientos & Historias de Usuario

En éste capítulo, se pretende establecer y profundizar en las diferencias y similitudes entre la idea clásica de requerimientos y las historias de usuario.

4.1 Requerimientos

Según el estándar (*IEEE Std. 610-1990 - IEEE Standard Computer Dictionary*, 1991):

1. Condición o capacidad que necesita el usuario para resolver un problema o alcanzar un objetivo
2. Condición o capacidad que debe satisfacer o poseer un sistema o una componente de un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto
3. Representación documentada de una condición o capacidad como en 1 o 2.

Por lo tanto, los requisitos incluyen no solo las necesidades del usuario, sino también las derivadas de los estándares generales de la organización, el gobierno y la industria. Claramente, como menciona (Aurum & Wohlin, 2005), un requerimiento es una colección de necesidades que surgen del usuario y de otras partes interesadas (organización general, comunidad, organismos gubernamentales y estándares de la industria), todo lo cual debe cumplirse. Idealmente, los requerimientos son independientes del diseño, mostrando "qué" debe hacer el sistema, en lugar de "cómo" debe hacerse.

4.1.1 Clasificación de los requerimientos

Comúnmente se clasifican los requerimientos en funcionales y no funcionales.

Los requerimientos funcionales son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema.

Los requerimientos no funcionales Son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se sue-

len aplicar al sistema como un todo, más que a características o a servicios individuales del sistema.

Además de la mencionada clasificación, (Aurum & Wohlin, 2005) , propone las siguientes formas posibles de catalogar los requerimientos.

Según el nivel:

- Requisitos de nivel de objetivo: relacionados con los objetivos comerciales
- Requisitos de nivel de dominio: relacionados con el área problemática
- Requisitos de nivel de producto: relacionados con el producto
- Requisitos de nivel de diseño: qué construir

Según la precedencia:

- Requisitos principales: obtenidos de las partes interesadas
- Requisitos derivados: derivados de requisitos primarios

Otra clasificación:

- Requisitos comerciales versus requisitos técnicos
- Requisitos del producto versus requisitos del proceso, es decir, necesidades del negocio versus cómo las personas interactuarán con el sistema
- Requisitos basados en roles, p. requisitos del cliente, requisitos del usuario, requisitos de TI, requisitos del sistema y requisitos de seguridad

4.1.2 Calidad de los requerimientos

Los requerimientos son especificados en un documento, éste adquiere el nombre de Especificación de Requerimientos de Software (ERS), resulta adecuado que los requerimientos en él expresados posean ciertas características de calidad, según (IEEE, 1998)

- Correctos: La ERS es correcta si y sólo si todo requisito que figura aquí (y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.
- No ambiguos: Cada requisito tiene una sola interpretación.
- Completos: Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos.
- Consistentes: Los requisitos no pueden ser contradictorios.

- Clasificados: Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito).
- Verificables: La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testeable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito.
- Modificables: La ERS es modificable si y sólo si se encuentra estructurada de forma que los cambios a los requisitos pueden realizarse de forma fácil, completa y consistente.
- Trazables: La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica qué componentes del sistema son los que realizan el requisito R.

4.2 Historias de usuario

Las historias de usuario se componen de tres elementos (Cohn, 2004):

- Una descripción escrita de la historia usada para la planificación y como un recordatorio.
- Conversaciones sobre la necesidad de la historia que sirven para profundizar en los detalles de la misma.
- Elementos de prueba (criterios de aceptación) que permiten determinar cuando las tareas que va a cubrir la historia está completa.

Descripción

La descripción se redacta en tarjetas o notas tipo post-it, y son pegadas en pizarrones o paredes para facilitar la planificación y discusión. Usualmente tiene el siguiente formato:

Como *<tipo de usuario>*, quiero *<realizar alguna función especificada>* para poder *<alguna razón>*.

Se puede colocar la razón al principio de la oración, sin embargo la descripción escrita tiene típicamente el formato mencionado.

Las historias demasiado grandes son denominadas épicas, estas pueden ser divididas en otras historias más pequeñas, pero se debe tener cuidado de no llegar a un nivel de detalle demasiado bajo.

Conversación

La descripción escrita sirve de guía a los desarrolladores, pero es en la conversación en donde están los detalles necesarios para avanzar en el desarrollo. Según (Cohn, 2004), “en lugar de escribir todos estos detalles como historias, el mejor enfoque es que el equipo de desarrollo y el cliente analicen estos detalles. Es decir, mantenga una conversación sobre los detalles en el momento en que los detalles se vuelven importantes.”

Test de aceptación

Debido a que las historias de los usuarios cambian el énfasis hacia hablar y no escribir, las decisiones importantes no se capturan en documentos que es poco probable que se lean.

En cambio, los aspectos importantes de las historias se capturan en las pruebas de aceptación automatizadas y se ejecutan con frecuencia.

4.2.1 Calidad de una Historia de Usuario

El acrónimo INVEST definido por (Wake, 2003), permite recordar un conjunto de criterios ampliamente aceptado, para evaluar la calidad de una historia de usuario.

Una buena historia debería ser:

- "I" Independiente (de todos los demás)
- "N" Negociable (no es un contrato específico para las características)
- "V" Valiosa (o vertical)
- "E" Estimable (a una buena aproximación)
- "S" Pequeña (para que quepa dentro de una iteración)
- "T" Testeable (en principio, incluso si todavía no hay una prueba)

Independiente:

Significa que la historia de usuario no se superponga conceptualmente con otra historia, que se pueda colocar en un cronograma y que las historias se puedan desarrollar en cualquier orden.

Muchas veces esta característica no puede cumplirse para el 100% de las historia. El objetivo que se debe buscar es cuestionarse si para cada historia de usuario se ha hecho todo lo posible para que sea independiente del resto de las historias.

Negociable:

Una historia no es un contrato explícito de características; más bien, los detalles serán creados conjuntamente por el cliente y el programador durante el desarrollo. Una buena historia captura la esencia, no los detalles.

El alcance de las historias (sus criterios de aceptación) podrían ser variables: pueden incrementarse o eliminarse con el correr del desarrollo y en función del feedback del usuario y/o la performance del equipo.

Valiosa (Vertical):

Una historia debe ser valiosa para el cliente. Los desarrolladores pueden tener legítimos intereses, pero estos serán expresados de una manera que el cliente los perciba como importantes.

Esto es especialmente un problema al dividir historias. Wake propone pensar (Wake, 2003) cada historia como un pastel de varias capas, por ejemplo, una capa de red, una capa de persistencia, una capa lógica y una capa de presentación. Cuando se divide una historia, se sirve solo una parte de esa torta. Se desea dar al cliente la esencia de todo el pastel, y la mejor manera es cortar verticalmente entre las capas. Los desarrolladores a menudo tienen una inclinación a trabajar solo en una capa a la vez (y hacerlo "correcto"); pero una capa de base de datos completa (por ejemplo) tiene poco valor para el cliente si no hay capa de presentación.

Estimable:

No es necesario una estimación exacta, pero sí lo suficiente para ayudar al cliente a clasificar y programar la implementación de la historia.

Es posible que el costo de una historia no pueda ser calculada con un nivel de precisión razonable, los motivos habituales por los que esto pueda ocurrir son:

- La Historia de Usuario es demasiado grande. En este caso la solución sería dividir la Historia de Usuario en historias más pequeñas que sean estimables.
- Falta de conocimiento funcional. En este caso la Historia de Usuario vuelve al Product Owner para bajar en detalle la Historia o inclusive (y recomendable) tener una conversación con el Equipo de Desarrollo.
- Falta de conocimiento técnico. Muchas veces el Equipo de Desarrollo no tiene el conocimiento técnico suficiente para realizar la estimación. En estos casos el Equipo de Desarrollo puede dividir la historia en 1) un time-box conocido como “spike” que le permita investigar la solución y proveer una estimación más certera y 2) la funcionalidad a desarrollar como parte de la Historia en si misma.

Pequeñas:

El tamaño de la historia sí importa porque si las historias son demasiado grandes o demasiado pequeñas, se dificulta utilizarlas para planificar. Una historia debe ser pequeña para caber en un sprint, pero lo suficientemente grande para tener sentido y agregar valor.

Las épicas son difíciles de trabajar porque con frecuencia contienen múltiples historias.

Las buenas historias tienden a ser pequeñas puesto que permiten obtener estimaciones más precisas. Las historias generalmente representan como mucho unas pocas personas-semanas de trabajo. (Algunos equipos los restringen a unos pocos días-persona de trabajo). Por encima de este tamaño, y parece ser demasiado difícil saber qué hay en el alcance de la historia. Decir, "me llevaría más de un mes" a menudo agrega implícitamente, "ya que no entiendo qué, todo lo que implicaría".

La determinación final de si una historia tiene el tamaño adecuado se basa en el equipo, sus capacidades y las tecnologías en uso.

Testeable:

Una buena historia es comprobable. Escribir una tarjeta de historia conlleva una promesa implícita: "Entiendo lo que quiero lo suficientemente bien como para escribir una prueba para ello". Varios equipos han informado que al requerir pruebas de los clientes antes de implementar una historia, el equipo es más productivo. La "capacidad de prueba" siempre ha sido una característica de los buenos requisitos; de hecho, escribir las pruebas temprano ayuda a saber si se cumple este objetivo.

Si un cliente no sabe cómo probar algo, esto puede indicar que la historia no es lo suficientemente clara, o que no refleja algo valioso para ellos, o que el cliente solo necesita ayuda para realizar las pruebas.

Un equipo puede tratar los requisitos no funcionales (como el rendimiento y la usabilidad) como elementos que deben probarse. Averiguar cómo poner en práctica estas pruebas ayudará al equipo a conocer las verdaderas necesidades.

4.2.2 Criterios de aceptación

Según (Wiegers & Beatty, 2013), son condiciones que debe cumplir un producto de software para ser aceptado por un usuario, cliente u otra parte interesada. En otras palabras son las condiciones bajo las cuales se cumple una historia de usuario.

Los criterios de aceptación son un conjunto de declaraciones, cada una con un claro resultado aprobado/no aprobado, que especifican requisitos funcionales y no funcionales. Estos requisitos representan "condiciones de satisfacción". No hay aceptación parcial: se cumple un criterio o no se cumple.

Los criterios de aceptación deben indicar el propósito, pero no una solución (por ejemplo, "Un gerente puede aprobar o desaprobado un formulario de auditoría" en lugar de "Un administrador puede hacer clic en un botón de Aprobar/Desaprobar para Aprobar un formulario de auditoría"). Los criterios deben ser independientes de la implementación: idealmente, la redacción debe ser la misma independientemente de la plataforma objetivo.

Criterios redactados concisamente ayudan a los equipos de desarrollo a evitar la ambigüedad sobre las demandas de un cliente y evitar la mala comunicación.

Existen varias formas de escribir los criterios de aceptación, las más populares son las orientadas a reglas y las orientadas a escenario. La primera es una lista de reglas que se deben cumplir.

4.3 Requerimientos vs Historias de usuario

Una primera diferencia es que un requerimiento posee mayor nivel de detalle que una historia de usuario sobre cómo deberá funcionar el software, ese nivel de detalle es lo que guía a los desarrolladores en su trabajo. En el caso de las historias de usuarios en ese espacio donde se manifiesta el detalle necesario para realizar el desarrollo del software.

Existen otras ventajas de las historias de usuario sobre los documentos, como lo menciona (Cohn, 2004):

- Los documentos escritos pueden hacer que suspenda el juicio.
- Con un documento escrito, no iteramos sobre el significado como lo haríamos en una conversación.
- Los documentos escritos disminuyen la responsabilidad de todo el equipo.

5 Léxico extendido del lenguaje

5.1 Definición

El LEL (Language Extended Lexicon) es un modelo que permite representar y documentar, un conjunto de símbolos que representan el lenguaje de la aplicación, sin necesidad de entender el problema (Leite & Franco, 1990) . Consiste en una descripción de los términos significativos del macrosistema, acotando el lenguaje externo con el uso de símbolos definidos en el mismo LEL y a su vez, minimiza el uso de símbolos externos al lenguaje de la aplicación.

5.2 Características

El LEL está compuesto por un conjunto de símbolos que identifican el lenguaje de la aplicación (Leite & Franco, 1990). Para generar el LEL se registran símbolos (palabras o frases) peculiares o relevantes del dominio. Cada entrada del léxico se identifica con un nombre (o más de uno en caso de sinónimos) y tiene dos tipos de descripciones. Una llamada Noción que describe la denotación del símbolo y la otra Impacto que describe la connotación del mismo. Las entradas se clasifican en cuatro tipos de acuerdo a su uso general en el UdeD. Estos tipos son: Sujeto, Objeto, Verbo y Estado (Kaplan et al., 2000).

La semántica de cada símbolo se representa con una o más nociones y uno o más impactos. El impacto puede no existir. La noción indica qué es el símbolo y el impacto, cómo repercute en el sistema. Por lo tanto, cada símbolo tiene un “nombre” que lo identifica, una “noción” y un “impacto” que lo describen (Leite, 1993).

En la Figura 9, podemos apreciar un ejemplo de un termino que forma parte de un LEL.

<p><u>Obra / Publicación</u></p> <p>Noción:</p> <ul style="list-style-type: none"> • Es un ítem de la <u>colección</u>. • Es un <u>libro</u>, <u>folleto</u>, <u>tesis</u>, <u>publicación-PUC</u> o <u>periódico</u>. • Puede ser una <u>obra de referencia</u>. • Puede ser una <u>obra de tapa roja</u>. • Puede tener más de un <u>ejemplar</u>. <p>Impacto:</p> <ul style="list-style-type: none"> • Después de la <u>adquisición</u>, el <u>bibliotecario</u> realiza el <u>registro</u> y el <u>procesamiento técnico</u>. • Después del <u>procesamiento técnico</u>, se puede <u>localizar</u>, <u>consultar</u>, <u>prestar</u>, <u>devolver</u>, <u>renovar</u>, <u>reservar</u> y <u>prestar para fotocopiar</u>.
--

Figura 9: Ejemplo de termino del LEL

5.3 Proceso de construcción

La construcción del LEL está regida por dos principios. El principio de circularidad y el principio de vocabulario mínimo. El principio de circularidad establece que en la descripción de la noción e impactos se debe maximizar el uso de signos definidos en el LEL. El principio de vocabulario mínimo, complementa el principio de circularidad, y establece que cuando se utilizan signos externos al LEL deben tener una representación matemática clara (por ejemplo: pertenencia, intersección, etc).

Ambos principios determinan que el conjunto de signos sea autocontenido. Es decir que un signo esté expresado en término de otros. De esta forma los signos están interrelacionados. Si cada signo se ve como un nodo de información, junto con la relación entre ellos se puede percibir un grafo. Ahora, como cada nodo está conformado por texto, el grafo no es otra cosa que un hipertexto

El LEL se obtiene del UdeD a través de la aplicación de un conjunto de heurísticas de construcción. Este proceso consta de seis actividades, algunas de las cuales pueden ocurrir simultáneamente (Kaplan et al., 2000):

1. Identificar fuentes de información
2. Identificar símbolos
3. Clasificar símbolos
4. Describir símbolos
5. Validar el LEL
6. Control del LEL

Identificar fuentes de información:

Las fuentes de información para la construcción de LEL son dos: las personas y los documentos. Cada uno tiene ventajas sobre el otro. Los documentos ofrecen un medio concreto, autocontenido e invariable. El ingeniero de requerimientos puede leer y volver a leer los documentos para una selección minuciosa de los términos. Los textos se pueden analizar en detalle. Por otra parte, las personas pueden aportar a través de entrevistas mucha mayor información. Una característica propia de la expresión oral de los seres humanos es que en forma inconsciente utilizan el principio de circularidad. En cambio en la descripción escrita, para una narrativa más entretenida se introducen muchos sinónimos, lo que puede dificultar el proceso de identificación de símbolos. Es aconsejable utilizar ambas fuentes de información. (Antonelli, 2003).

Identificar símbolos:

Una estrategia aconsejable para generar la lista de símbolos consiste en comenzar con una lectura de la documentación para hacer un análisis preliminar del lenguaje del dominio. Con este análisis se obtiene una lista inicial y se adquiere información para poder llevar a cabo entrevistas. Es aconsejable realizar entrevistas semiestructuradas y estructuradas para acotar el lenguaje. Ya que si se comenzara con entrevistas libres, el volumen de información sería muy difícil de manejar.

Los símbolos que se deben tomar para la lista inicial están dados por palabras o frases que se utilizan con mucha frecuencia, o aquellas que parezcan estar fuera de contexto. Los términos más utilizados son significativos en el dominio, por lo tanto deben estar definidos. En cambio los términos que parezcan estar fuera de contexto se deben a que tienen un significado propio en el dominio, distinto del tradicional. Estos términos, con más razón deben estar definidos.

Cabe destacar que los símbolos no tienen porque ser palabras individuales. Pueden ser palabras o frases. La razón es que en un lenguaje se puede encontrar un grupo de palabras, en donde cada una de ellas pueda tener cierto significado, pero si se las combina en una frase tienen otro significado (Antonelli, 2003).

Clasificar símbolos:

El objetivo de categorizar los símbolos es poder administrar mejor al conjunto. Además, cada categoría determina la forma en que se deba definir cada símbolo. Así se logra una definición consistente y uniforme.

Para clasificar los símbolos se parte de una clasificación general (Antonelli, 2003):

Sujeto: Elemento activo dentro del dominio que realiza acciones utilizando objetos. El sujeto (persona, maquina, dispositivo) puede llegar a pasar por distintos estados.

Verbo: Acción que realiza un sujeto, servicio que brinda un objeto, desencadenante para pasar de un estado a otro.

Objeto: Elemento pasivo con los cuales se realizan acciones que puede pasar por distintos estados.

Estado: Situación en la que se encuentra un sujeto o un objeto.

Describir símbolos:

Los símbolos se describen a partir del conocimiento obtenido de la lectura de la documentación y de las entrevistas. Cada símbolo se describe siguiendo las pautas establecidas en la clasificación de los términos. Esta descripción inicial luego será validada y controlada.

Algunas pautas generales para la descripción de los símbolos, independientemente de la clasificación a la que corresponden, son las siguientes:

- Un signo puede tener una o más nociones y cero o más impactos.
- Cada noción e impacto debe ser descrito con oraciones breves y simples.
- Las oraciones deben responder a los principios de circularidad y de vocabulario mínimo.
- Las nociones e impacto de un signo pueden representar diferentes puntos de vista o pueden ser complementarios.

Con la descripción de símbolos se facilita la tarea de encontrar sinónimos. Sin embargo, en la validación posterior, se despeja cualquier duda respecto de los sinónimos.

Validar el LEL:

Consiste en verificar la correctitud del LEL contra el usuario. Debido a la gran cantidad de símbolos que pueden estar definidos en el LEL, es impracticable realizar una validación completa y exhaustiva. Sin embargo, es posible mantener sesiones de entrevistas estructuradas para aclarar dudas (Antonelli, 2003).

Control del LEL:

El proceso de control es el que realiza el ingeniero de requerimientos por sus propios medios. El control no tiene que ver con la correctitud de la información del LEL, por el contrario está relacionado con la estructura:

- Todos los símbolos deben estar definidos. Aquellos de la lista inicial y también los que aparecen en la descripción de los signos de la lista inicial.
- Todos los símbolos deben estar dentro de la clasificación correspondiente. Debido a que la clasificación se realiza antes de la descripción, en esta última se puede descubrir que un símbolo se colocó en la categoría incorrecta, por lo cual se lo debe reubicar y verificar su definición.
- La descripción de los símbolos debe corresponderse con la de la categoría a la que pertenecen.
- El punto de vista para la descripción de los símbolos debe ser uniforme.
- No deben dejarse sinónimos definidos como signos independientes.

6 Construcción de LEL desde las Historias de Usuario

6.1 Introducción

En este capítulo se hace referencia al trabajo realizado por (Antonelli et al., 2017), en dicho trabajo se propone consolidar en el LEL conocimiento fraccionado en las distintas Historias de Usuario.

En la Figura 10, podemos observar cómo es creado el LEL y cómo crece incrementalmente después de cada Sprint.

Figura 10: Proceso de construcción del LEL - (Antonelli et al., 2017)

Para lograr construir el LEL propone un proceso conformado por los pasos presentados a continuación.

6.2 Pasos

A. Documentación de historias de usuario.

El Product Owner especifica un conjunto de Historias de Usuario que se apilan por prioridad en el backlog. En cada Sprint, durante la reunión de planificación, las Historias de Usuario son extraídas del backlog, estimadas por el equipo y enriquecidas con una mejor descripción y criterios de aceptación.

B. Normalización de historias de usuarios.

Las historias de usuario pueden estar escritas en voz pasiva, utilizando pronombres, etc. Sus descripciones deben estar normalizadas para posibilitar el procesamiento automático.

C. Requisito funcional del filtro.

Las historias de usuario pueden definir requisitos tanto funcionales como no funcionales. Dado que el enfoque solo se ocupa de los requisitos funcionales, este paso consiste en eliminar las historias de usuario que solo se refieren a los requisitos no funcionales. Este paso debe realizarse manualmente. Sin embargo, podría ser fácil implementar un herramienta que contiene un catálogo de términos de requisitos no funcionales para automatizar el filtrado.

D. Descripción de LEL de Historias de Usuario.

Cada Historia de Usuario se procesa y su información se extrae utilizando diferentes reglas para identificar los símbolos y describirlos.

Las reglas identifican tres categorías de símbolos: sujetos, objetos y verbos. Luego, cada símbolo se describe con dos atributos: noción y respuestas conductuales.

El enfoque consiste en describir las respuestas conductuales de las tres categorías de símbolos. Durante el proceso, cada noción y respuesta de comportamiento se etiqueta con la identificación del Sprint para contextualizar la definición en la vida útil del proyecto.

6.3 Reglas

Se mencionan a continuación las reglas que se deben aplicar para construir el LEL a partir de un conjunto de Historias de Usuario.

1. Identificación de los sujetos. De cada Historia de usuario, los roles después del atributo "Como" deben registrarse como Sujeto.
2. Identificación de las respuestas de comportamiento para los sujetos. Las metas o los deseos descritos en la cláusula "Quiero" representan respuestas de comportamiento. Las oraciones se reescriben de primera a tercera persona. Además, la cláusula "Quiero" tiene el formato sujeto verbo [objeto / sujeto] [otra expresión]. Solo el verbo sujeto [objeto / sujeto] debe considerarse como respuesta conductual.
3. Identificación de los verbos. Al considerar la cláusula "Quiero" (Respuestas de comportamiento del sujeto), los componentes del Verbo se extraen dando información sobre la funcionalidad del sistema.

4. Identificación del comportamiento de los verbos. El resto de la expresión que está presente en la oración "Quiero" que da origen a los símbolos verbales de la regla número 3 ([otra expresión]) debe usarse como respuesta conductual del símbolo verbal.
5. Identificación de nociones para los verbos. La noción del símbolo Verbo debe extraerse de la cláusula "so that".
6. Identificación del objeto. Una vez que se han definido los verbos, el elemento objeto que sigue al verbo se extrae y se registra como un símbolo.
7. Identificación conductual del objeto. A partir de las expresiones que dan lugar a los Objetos ("verbo [objeto]"), se debe agregar una nueva respuesta de comportamiento.
8. Respuestas de comportamiento adicionales de los sujetos. Cuando un verbo es seguido por un predicado, la oración se usa como respuesta de comportamiento del sujeto que coincide con el predicado.

7 Propuesta de trazabilidad

En este capítulo se presenta la técnica ad hoc desarrollada en el presente trabajo, también se detalla a continuación el proceso que permite vincular las Historia de Usuario¹ y el Léxico Extendido del Lenguaje (LEL) derivado de ellas, con fragmentos de código fuente. Finalmente se presenta el modelo de trazabilidad desarrollado que sustenta la propuesta.

7.1 Técnica de trazabilidad desarrollada

Se puede afirmar que la presente propuesta, se beneficia de ideas provenientes de más de una técnica. De la Referencia Cruzada se obtiene la idea de vincular la historia de usuario con diversos artefactos de software, inicialmente con fragmentos de código y posteriormente con casos de prueba. Del Etiquetado se obtiene el concepto de emplear un identificador único para referenciar a la historia de usuario en los fragmentos de código. Por otro lado los elementos del LEL, los símbolos, actúan como intermediarios entre los ítems y los fragmentos de código.

Figura 11: Vínculo entre ítems, símbolos y fragmentos de código

La idea central que subyace en esta propuesta consiste en establecer una relación como la que se ve en la Figura 11, entre ítems, símbolos y fragmentos de código. En secciones posteriores del presente capítulo se brindará detalle sobre la forma en que se crean las relaciones mencionadas.

¹En adelante se empleará de manera indistinta los términos historias de usuario/ítems de la pila del producto.

7.2 Proceso de registrar la trazabilidad

7.2.1 Visión general

Una visión sobre el desarrollo de software (Armour, Phillip, 2000) menciona que más que una actividad de manufactura de un producto es mas bien una actividad de adquisición de conocimiento. Siguiendo esta noción y pensando en la trazabilidad, podemos plantearnos ¿Cuándo? ¿Quién? ¿Dónde? ¿Cómo? puede realizar la tarea de gestionar las trazas entre los distintos artefactos de software.

La gestión de la trazabilidad es una tarea que se realiza a lo largo de cada iteración en el desarrollo del software, más precisamente al trabajar en cada ítem de la pila del producto. Inicia con la creación del ítem como un nuevo archivo dentro del proyecto.

En un proyecto ágil, típicamente es el dueño del producto (PO) quien redacta cada ítem, por lo tanto sólo es cuestión de brindarle los permisos en una carpeta del proyecto en donde se almacenaran los ítem como archivos independientes.

Posteriormente será el programador quien incorpore información relacionada con un ítem en particular pues es él quien esta involucrado en las conversaciones sobre las historias de usuario y posee el conocimiento sobre todos los fragmentos de código que están relacionados con la historia que está tratando.

Es fundamental que esta información sea almacenada en el mismo código fuente, pues es allí en definitiva el lugar por excelencia dónde reside el conocimiento propio del software. Además, como lo menciona (Turk et al., 2005) , "el código es la descripción más acertada y confiable de qué es lo que hace un sistema y cómo esta diseñado".

Para llevar a cabo la gestión de la trazabilidad se necesita una herramienta software que facilite la gestión de la información referente a las trazas entre ítem, símbolos del LEL y fragmentos de código.

Al observar la Figura 12: Proceso de trazabilidad propuesto, se puede tener una visión general del proceso propuesto.

Figura 12: Proceso de trazabilidad propuesto

Se describe a continuación el detalle del proceso de trazabilidad propuesto.

7.2.2 Redactar ítems

El proceso parte inicialmente de un conjunto de ítems del Product Backlog, cada uno de éstos ítems existen como archivos individuales de texto plano en el mismo repositorio donde se encuentra el código fuente del proyecto que se esta desarrollando.

Es fundamental la coexistencia de los ítem con el código fuente, pues esto facilita el establecimiento del vínculo entre ambos.

7.2.3 Extracción de símbolos

Para lograr esta tarea se utiliza como referencia el trabajo realizado en (Antonelli et al., 2017), que consiste fundamentalmente en la obtención de los símbolos pertenecientes a cada ítem a través de la aplicación de una serie de reglas sobre cada ítem previamente normalizado.

7.2.4 Vincular Ítems con símbolos

Posteriormente estos ítems son parseados y a partir de cada ítem, se construye una lista de símbolos asociada a dicho ítem (Figura 13: Relación entre ítems y símbolos). Cada uno de los símbolos es identificado unívocamente por medio del nombre y el tipo de símbolo.

Se puede notar que un ítem contiene varios símbolos, además se puede observar, que es posible que un mismo símbolo esté relacionado con mas de un ítem.

Figura 13: Relación entre ítems y símbolos

En la medida que se avanza con el procesamiento de cada ítem, se construye en definitiva un léxico extendido del lenguaje, LEL en adelante.

7.2.5 Vincular fragmentos de código

La siguiente tarea es la de vincular fragmentos de código con un determinado símbolo y con un ítem en particular (ver Figura 14).

La vinculación debe ser no invasiva al código fuente, es decir no se debe modificar la esencia del fragmento de código que se pretende vincular. Para que esto sea posible se pueden utilizar comentarios o anotaciones, esto dependerá del lenguaje empleado para en el desarrollo.

Figura 14: Relación entre ítem, símbolo y fragmento de código

Según el tipo de símbolo, se establece un tipo de fragmento de código con el que se puede relacionar, ver Tabla 3: Posibilidad de vincular Símbolos con Clases/Métodos. Por ejemplo si es de tipo verbo, se pueden vincular con métodos y no con clases. Para el caso del tipo de símbolo estado, resulta no aplicable por no haber identificado desde el paso anterior los símbolos que corresponden al tipo estado. En el caso de los símbolos de tipo objeto o sujeto, pueden ser correspondidos con vínculos con clases y/o métodos.

Tabla 3: Posibilidad de vincular Símbolos con Clases/Métodos

Tipo de Símbolo	Clases	Métodos
Objeto	Si	Si
Sujeto	Si	Si
Verbo	No	Si
Estado	No aplicable	No aplicable

Son dos las maneras de registrar las trazas entre símbolos y fragmentos de código, una manual y otra semiautomática.

7.2.5.1 Forma Manual

Al emplear esta forma el programador introduce una anotación o comentario en el código en el que está trabajando, indicando en la anotación que símbolos y qué ítem están relacionados con el fragmento de código.

7.2.5.2 Forma Semiautomática

En esta otra forma, se parte de una lista de símbolos y se busca cada uno de los mismos en el código, puntualmente en los nombres de clases y nombres de métodos, al encontrar algún nombre que lo contenga se permite anotar dicho símbolo.

Esta forma es semiautomática porque brinda al programador la posibilidad de indicar cuáles de las coincidencias que deben ser anotadas, evitando que el programador tenga que escribir la anotación que desea introducir al código.

El pseudocódigo de la Figura 15, permite tener una idea aproximada del procesamiento a realizar para generar la anotación adecuada que permita vincular un símbolo y una Historia de Usuario con una porción de código.

```

Vinculacion/Anotacion semiautomatica
USj/USj ∈ {listOfUserStories}
si/si ⊂ USj
si(symbol, symbolType) / symbolType ∈ {object, subject, verb}
for filek = 1 to allFilesInProject
  /* annotation is n/a when symbolType ∈ {object, subject, verb} * /
  if (symbolTypei = object ∨ symbolTypei = subject)
    if (findSymbolInClassName(filek, symboli, symbolTypei))
      generateClassAnnotation(symboli, symbolTypei, USj)
  if (symbolTypei = verb ∨ symbolTypei = object ∨ symbolTypei = subject)
    if (findSymbolInMethodName(filek, symboli, symbolTypei))
      generateMethodAnnotation(symboli, symbolTypei, USj)

```

Figura 15: Psuedocodigo de la generación de anotaciones

7.2.6 Buscar vínculos en el código

Finalmente, luego de haber anotado el código adecuadamente, se puede emplear un procedimiento automático que registra el vinculo de determinado símbolo con los fragmentos de código correspondientes y con un ítem en particular. Este procedimiento revisa el código en busca de anotaciones que informan el Símbolo, el Tipo de Símbolo y el ítem con el cual esta relacionada.

7.3 Modelo de trazabilidad desarrollado

Establecer un modelo adecuado para sustentar la información referente a las trazas de los requerimientos no es una tarea trivial, en este trabajo se optó por seguir el enfoque propuesto por (Taromirad & Paige, 2012). En el mismo, se considera adecuado a aquel modelo que respalda los objetivos de la trazabilidad y además resulta eficiente económicamente. En el presente trabajo el objetivo de trazabilidad significa establecer trazas que van desde las historias de usuario hasta el código fuente, utilizando como intermediario a los símbolos provenientes del LEL y se pretende que la eficiencia económica esté presente al significar poco trabajo adicional para el desarrollador utilizar la propuesta de trazabilidad aquí presentada.

En (Taromirad & Paige, 2012) se sugiere además que la forma de obtener el modelo adecuado debe ser incremental a través de iteraciones. Generalmente, en cada iteración, la información de trazabilidad se identifica con respecto a las necesidades actuales, el esquema de trazabilidad se extiende de acuerdo con los nuevos requisitos, y luego el esquema se usa para la trazabilidad. La información de trazabilidad generalmente se define de acuerdo con las necesidades de las partes interesadas, el ciclo de vida de desarrollo del proyecto o el proceso de desarrollo general para el dominio específico y los escenarios de uso (objetivos de trazabilidad). La ampliación del esquema existente podría necesitar proporcionar estrategias de evolución y cambio para mantener los modelos de rastreo existentes correctamente. Cada iteración termina cuando se identifica un cambio, ya sea por falta o por un nuevo requisito y, en consecuencia, comienza una nueva iteración. Este ciclo se repite durante el proyecto y el esquema de trazabilidad se extenderá cuando sea necesario. La Figura 16 muestra el modelo de proceso conceptual utilizado para definir el esquema de trazabilidad.

Inicialmente el modelo permite relacionar historias de usuario, denominadas ítems en el modelo, con documentos externos que contienen el texto original de las historias de usuario. Además, permite una representación y un almacenamiento textual de los requerimientos expresados por medio de historias de usuario.

Figura 16: Proceso de definición del modelo de trazabilidad - (Taromirad & Paige, 2012)

El modelo permite, luego de descomponer cada ítem, registrar cada uno de los símbolos encontrados y asociarlos con el ítem correspondiente.

Respecto de la trazabilidad, el modelo sustenta relaciones entre los ítems y las porciones de código, denominadas snippets. También soporta relaciones del tipo es-hijo-de entre ítems.

El modelo permite determinar el impacto de los cambios proveniente tanto desde los ítem hacia los snippets, como aquellos que se pueden generar desde los snippets hacia los distintos ítems.

A partir de localizar al interesado como un tipo de símbolo "sujeto", el modelo permite determinar relaciones entre interesados e ítems.

Finalmente, el modelo soportar actualmente análisis y procesamiento para JAVA y PHP, permitiendo realizar implementaciones para otros lenguajes.

Figura 17: Modelo de trazabilidad desarrollado

En la Figura 17, se puede ver un diagrama de clases del modelo propuesto en esta tesis.

8 Herramienta de Trazabilidad

8.1 Descripción general

Se desarrolló una herramienta para evaluar la propuesta de trazabilidad planteada. Esta herramienta consiste en una aplicación web cuyo principal objetivo es permitir el almacenamiento y la consulta de la información referente a las trazas entre Ítems, símbolos del LEL y fragmentos de código. Este software permite al usuario final llevar adelante los pasos detallados en el proceso propuesto en el Capítulo 7.

8.2 Arquitectura

Es una aplicación web desarrollada utilizando el framework Springboot², la misma sigue el patrón de arquitectura MVC, separando por capas y en diversos paquetes las clases correspondientes a cada componente del mencionado patrón, como se puede ver en la Figura 18.

Figura 18: Paquetes de la herramienta

² Versión 2.1.0

El paquete controllers agrupa las clases pertenecientes a la capa controlador de la aplicación. El paquete repository facilita el acceso a los datos, mientras que el paquete domain agrupa las clases referidas al dominio del negocio. En el paquete processor están las clases que permiten analizar el código fuente para llevar a cabo la vinculación entre código y elementos del LEL.

Como se mencionó previamente, la herramienta puede realizar el análisis y procesamiento para JAVA y PHP, permitiendo desarrollar implementaciones para otros lenguajes. Para que esta tarea sea algo sencillo de realizar, se empleó el patrón de diseño Strategy. Ver Figura 19.

Figura 19: Patrón Strategy para distintos lenguajes

8.3 Almacenamiento

Posee una Base de Datos en memoria H2³, que almacena las relaciones entre ítems, símbolos y fragmentos de código.

En la Figura 20 se pueden observar las entidades y cómo se relacionan entre sí.

³ H2 es un sistema administrador de bases de datos relacionales programado en Java. Puede ser incorporado en aplicaciones Java o ejecutarse de modo cliente-servidor. Una de las características más importantes de H2 es que se puede integrar completamente en aplicaciones Java y acceder a la base de datos lanzando SQL directamente, sin tener que pasar por una conexión a través de sockets. (H2 (DBMS) - Wikipedia, la enciclopedia libre, s. f.)

Figura 20: DER de la herramienta

8.4 Funcionalidad de la Herramienta

La funcionalidad de la herramienta acompaña el proceso de trazabilidad propuesto en el Capítulo 7, por lo que podemos separar la funcionalidad en dos áreas: establecer trazas y consulta de trazas.

8.4.1 Establecer trazas

Para realizar la tarea establecer trazas entre los diversos artefactos el menú de procesamiento cuenta con las opciones que se pueden ver en la Figura 21.

Figura 21: Opciones del menú procesamiento

La opción **Read Items & Parse Items** recorre la carpeta dónde están almacenados los ítems del proyecto y los procesa, cada uno de los ítems ha sido especificado en un archivo individual, la ubicación de los ítems ha sido indicada como un atributo del proyecto en cuestión.

El procesamiento de un ítem consiste en analizar dicho ítem y determinar los símbolos que lo componen, determinando adecuadamente el tipo de símbolo identificado. Para realizar este procesamiento se utilizó una librería⁴ proveniente del trabajo (Antonelli et al., 2017). Durante el procesamiento, se almacena en la base de datos H2 cada uno de los ítems, como así también los símbolos identificados.

La opción **Link Symbols with Items** vincula los símbolos detectados con el ítem correspondiente y almacena esta información en la base de datos para su posterior consulta.

La opción **Find [Objects|Subjects|Verbs]**, efectúa la búsqueda de los símbolos en el código fuente según el tipo de símbolo y permite indicar si se debe vincular el resultado de la búsqueda con los símbolos correspondientes, para llevar a cabo la búsqueda se siguen los criterios presentados en la Tabla 4.

⁴ LEL Parser, URL: https://bitbucket.org/lellifia/lel_agile/src/master/

Tabla 4: Criterios de búsqueda

Tipo de Símbolo	Clases	Métodos
Objeto	Si	Si
Sujeto	Si	Si
Verbo	No	Si

Por ejemplo al realizar la búsqueda de los símbolos de tipo objeto, se busca coincidencia en los nombres de las clases y en los nombres de los métodos. Mientras que en el caso del tipo verbo, se buscará solamente en los nombres de los métodos.

En la Figura 22, se puede ver el resultado de la búsqueda, **fileName** indica nombre del archivo en que se encontró la coincidencia, **target** hace referencia al nombre completo de la coincidencia, **typeTarget** indica si es de tipo método o clase, **insertString** indica la anotación generada que será insertada en el código, **fileNumber** hace referencia a la línea donde se encontró la coincidencia y **Select** es un checkbox para indicar si se insertará la anotación en el código fuente. Al final del listado existe un botón que permite confirmar las modificaciones que se realizarán en el código.

Code Fragment List

fileName	target	typeTarget	insertString	lineNumber	Select
/home/jguerra/proyectos/yupichay/src/main/resources/project/petclinic/src/main/java/org/springframework/samples/petclinic/owner/OwnerController.java	OwnerController	class	@unlp.figuerra.yupichay.annotations.Trace(notion = "owner", symboltype = "object")	39	☐
/home/jguerra/proyectos/yupichay/src/main/resources/project/petclinic/src/main/java/org/springframework/samples/petclinic/owner/Pet.java	Pet	class	@unlp.figuerra.yupichay.annotations.Trace(notion = "pet", symboltype = "object")	50	☒
/home/jguerra/proyectos/yupichay/src/main/resources/project/petclinic/src/main/java/org/springframework/samples/petclinic/owner/PetController.java	PetController	class	@unlp.figuerra.yupichay.annotations.Trace(notion = "pet", symboltype = "object")	35	☐

Figura 22: Resultado de Find Objects

Luego de realizar la vinculación entre un símbolo y una porción de código en particular, la siguiente vez que se efectúe la misma búsqueda, aquel símbolo vinculado no figurará entre los resultados ofrecidos, pues el mismo ya ha sido relacionado.

La opción **Find Symbol**, realiza la búsqueda de un símbolo en particular dentro del código fuente y permite efectuar la posterior vinculación entre símbolo y fragmento de código. En la Figura 23, se puede ver el resultado de efectuar la búsqueda del símbolo de tipo objeto Owner.

Code Fragment List

#	fileName	target	typeTarget	insertString	lineNumber	Select
1	/home/jguerra/proyectos/yupichay/src/main/resources/project/petclinic/src/main/java/org/springframework/samples/petclinic/owner/OwnerController.java	OwnerController	class	@unlp.figuerra.yupichay.annotations.Trace(notion = "owner", symboltype = "object")	39	☐
2	/home/jguerra/proyectos/yupichay/src/main/resources/project/petclinic/src/main/java/org/springframework/samples/petclinic/owner/OwnerController.java	initUpdateOwnerForm	method	@unlp.figuerra.yupichay.annotations.Trace(notion = "owner", symboltype = "object")	103	☐
3	/home/jguerra/proyectos/yupichay/src/main/resources/project/petclinic/src/main/java/org/springframework/samples/petclinic/owner/OwnerController.java	processUpdateOwnerForm	method	@unlp.figuerra.yupichay.annotations.Trace(notion = "owner", symboltype = "object")	110	☐
4	/home/jguerra/proyectos/yupichay/src/main/resources/project/petclinic/src/main/java/org/springframework/samples/petclinic/owner/OwnerController.java	showOwner	method	@unlp.figuerra.yupichay.annotations.Trace(notion = "owner", symboltype = "object")	127	☐
5	/home/jguerra/proyectos/yupichay/src/main/resources/project/petclinic/src/main/java/org/springframework/samples/petclinic/owner/PetController.java	findOwner	method	@unlp.figuerra.yupichay.annotations.Trace(notion = "owner", symboltype = "object")	52	☐
6	/home/jguerra/proyectos/yupichay/src/main	initOwnerBinder	method	@unlp.figuerra.yupichay.annotations.Trace(notion =	57	☐

Figura 23: Resultado de buscar Owner en el código

La opción **Find Snippets**, realiza un recorrido por el código fuente en búsqueda de métodos y clases que estén vinculados con símbolos y/o ítems. En la Figura 24, se puede ver el una lista de los fragmentos de código resultantes. Al finalizar la búsqueda se actualiza la información correspondiente en la base de datos. Es decir se agregan los nuevos fragmentos y se crean las relaciones con los símbolos e ítems correspondientes.

Snippet List

Id	Qualified Name	Simple Name	Position	Source	View
2	Syllus\Bundle\ShippingBundle\SyllusShippingBundle	SyllusShippingBundle	22	use Syllus\Bundle\ResourceBundle\SyllusResourceBundle; use Syllus\Bundle\ShippingBundle\DependencyInjection\Compiler\RegisterCalculatorsPass; use Syllus\Bundle\ShippingBundle\DependencyInjection\Compiler\RegisterShippingMethodsResolversPass; use Symfony\Component\DependencyInjection\ContainerBuilder; // @unlp.figuerra.yupichay.annotations.Trace(notion = "shipping", symboltype = "subject", item = "US484") final class SyllusShippingBundle extends AbstractResourceBundle { /** * (@inheritdoc)	View
3	buildView	buildView	39	} /** * (@inheritdoc) */ // @unlp.figuerra.yupichay.annotations.Trace(notion = "view", symboltype = "verb") public function buildView(FormView \$view, FormInterface \$form, array \$options): void { \$view->vars['remote_criteria_type'] = 'contains'; \$view->vars['remote_criteria_name'] = 'phrase'; }	View

Figura 24: Resultado de realizar la búsqueda de Snippets

Existe una forma más en que la herramienta permite realizar la vinculación entre un símbolo y un fragmento de código. En este caso se accede primero a un ítem en parti-

cular y desde allí se realiza la búsqueda del símbolo que se desea vincular. Por ejemplo en la Figura 25, se visualiza información correspondiente al ítem US2.

Item detail

Id: US2

Text: As an administrator i want to View information pertaining to a pet owner

Location:

Symbols List

Id	Notion	Symbol Type	View	Find
2	administrator	subject	View	Find
4	view	verb	View	Find
5	information	object	View	Find
6	pertaining	object	View	Find
7	owner	object	View	Find

Figura 25: Información del ítem US2

Luego de hacer clic en la opción **Find** del símbolo owner, se obtiene como resultado todos los nombres de métodos y nombres de clases en que se puede encontrar el texto owner. Ver Figura 26. Existe en este listado un checkbox que permite indicar cuáles son los renglones a modificar, al final del listado existe un botón que permite confirmar las modificaciones que se realizarán en el código.

Code Fragment List

#	fileName	target	typeTarget	insertString	lineNumber	Select
1	/home/jguerra/proyectos/yupichay/src/main/resources/project/petclinic/src/main/java/org/springframework/samples/petclinic/owner/OwnerController.java	OwnerController	class	@unlp.fguerra.yupichay.annotations.Trace(notion = "owner", symboltype = "object", item = "US2")	39	☐
2	/home/jguerra/proyectos/yupichay/src/main/resources/project/petclinic/src/main/java/org/springframework/samples/petclinic/owner/OwnerController.java	initUpdateOwnerForm	method	@unlp.fguerra.yupichay.annotations.Trace(notion = "owner", symboltype = "object", item = "US2")	103	☐
3	/home/jguerra/proyectos/yupichay/src/main/resources/project/petclinic/src/main/java/org/springframework/samples/petclinic/owner/OwnerController.java	processUpdateOwnerForm	method	@unlp.fguerra.yupichay.annotations.Trace(notion = "owner", symboltype = "object", item = "US2")	110	☐

Figura 26: Resultado de buscar owner desde US2

En este caso el valor propuesto para **insertString** incluye como información el ítem al cual corresponde el código que se desea vincular, para el ejemplo actual corresponde

el ítem US2, como se puede ver en la Figura 27. Al incluir esta información se agrega una traza que va desde el fragmento de código hacia un ítem.

target	typeTarget	insertString
OwnerController	class	@unlp.figuerra.yupichay.annotations.Trace(notion = "owner", symboltype = "object", item = "US2")

Figura 27: Detalle de insertString

8.4.2 Consultar trazas

Una vez que la información referida a los ítems, símbolos, fragmentos de código y las relaciones entre ellos ha sido registrada, la herramienta permite consultar las trazas existentes utilizando como punto de partida un ítem, un símbolo o bien un fragmento de código.

Por ejemplo, para acceder de un listado de los ítems recolectados se debe al hacer clic en la opción de menú **Get Items**. Ver Figura 28.

En el listado se presenta un ítem por fila y se puede ver información general de cada de ellos, un identificador único, el texto que describe dicho ítem y al final de cada fila existe un link **View** que permite acceder información detallada de cada ítem.

Figura 28: Opción de menu Get Items

Existe además en esta pantalla, una opción para reducir los ítems que se están visualizando. Al ingresar valor en el campo **Search for ...**, presente en la parte superior del

listado, y hacer clic en el botón Search, se filtran los ítems que muestran en el listado incluyendo en él solo aquellos que coinciden con el criterio especificado. Ver Figura 29.

Search for ...

Item List

Id	Text	View
US89	As a Customer I want to be able to see shipping fee so that I can be aware of shipping fee applied for my shipment	View
US95	As a Customer I want to be able to see shipping fee in channel's base currency so that I can be aware of shipping fee applied for my shipment	View
US216	As an Administrator I want orders with no undelivered shipments to have shipping state shipped so that I can have coherent shipping states of all orders	View
US219	As an Administrator I want to be able to ship a shipment so that I can confirm shipping of an order	View
US229	As an Administrator I want to have new shipments after my customer's checkout so that I can correctly process customer's shipments	View
US475	As an Administrator I want to browse all shipments in the system so that I can manage all shipments regardlessly of orders	View
US476	As an Administrator I want to be able to ship a shipment from shipments list so that I can manage all shipments status	View
US477	As an Administrator I want to browse all shipments with a chosen state so that I can filter shipments by state	View
US478	As an Administrator I want to be able to access order's page from shipments index so that I can make shipments and orders management more fluent	View
US479	As an Administrator I want to browse all shipments in the system so that I can manage all shipments regardlessly of orders	View
US480	As an Administrator I want to be able to filter shipments from a specific channel on the list so that I can browse only relevant shipments	View
US483	As a Customer I want to have shipping fee applied based on chosen shipping method so that I can decide on amount paid for shipment	View
US521	As a Customer I want to be able to view details of my shipment status so that I can be aware whether my order has already been shipped	View
US495	As an Administrator I want to be prevented from adding it without specifying amount so that I can avoid making mistakes in a flat rate per shipment calculator of shipping method	View

Figura 29: Lista de ítems filtrados según criterio ingresado

En el detalle de cada ítem se puede observar información específica de cada ítem, además del identificador y el texto que lo conforma, se puede ver la ubicación del ítem dentro del proyecto como así también los símbolos y fragmentos asociados a dicho ítem. Ver Figura 30.

Item detail

Id: US484

Text: As an Administrator I want to be prevented from adding it without specifying amount so that I can avoid making mistakes in a flat rate per unit calculator of shipping method

Location: ./src/main/resources/project/Sylius/src/Sylius/Bundle/ShippingBundle

Symbols List

Id	Notion	Symbol Type	View	Find
2	administrator	subject	View	Find
5	be	verb	View	Find
22	avoid	verb	View	Find
74	amount	object	View	Find
77	method	subject	View	Find
86	shipping	subject	View	Find
108	rate	subject	View	Find
373	unit	subject	View	Find
374	calculator	subject	View	Find

Snippets List

Id	Qualified Name	Simple Name	Position	Source	View
2	Sylius\Bundle \ShippingBundle \SyliusShippingBundle	SyliusShippingBundle	22	use Sylius\Bundle\ResourceBundle\SyliusResourceBundle; use Sylius\Bundle\ShippingBundle \DependencyInjection\Compiler\RegisterCalculatorsPass; use Sylius\Bundle\ShippingBundle \DependencyInjection\Compiler\RegisterShippingMethodsResolversPass; use Symfony\Component \DependencyInjection\ContainerBuilder; // @unip.figuerra.yupichay.annotations.Trace(notion = "shipping", symboltype = "subject", item = "US484") final class SyliusShippingBundle extends AbstractResourceBundle { /** * {@inheritdoc}	View

Figura 30: Detalle del ítem US484

9 Validación de la propuesta de trazabilidad

Para validar el proceso propuesto, se utilizaron dos proyectos existentes. En el primer caso, el proyecto Petclinic⁵, fue empleado principalmente como medio para validar el correcto funcionamiento de la herramienta desarrollada.

El segundo proyecto elegido se denomina Sylius⁶, en esta oportunidad, se busco identificar el grado de utilidad de la propuesta en un proceso de desarrollo gestionado mediante el uso de tickets.

9.1 Proyecto Petclinic

Es una aplicación desarrollada en JAVA cuya finalidad es explicar mediante un ejemplo el uso y la funcionalidad principal del Framework Spring.

A continuación brindamos información específica del proyecto para lograr una idea más completa del contexto en el cual se desarrollo la herramienta. En la Figura 31 , se puede observar cantidad de tipos de archivos que posee el proyecto.

Figura 31: Cantidad de archivos por tipo. Petclinic.

⁵ <https://github.com/spring-projects/spring-petclinic>

⁶ <https://github.com/Sylius/Sylius>

Este proyecto cuenta con una lista de 9 características, las mismas han sido modificadas en su texto para que sean consistentes con el formato de historia de usuario. Ver Figura 32.

Item List

Id	Text	View
US1	As an administrator i want to View a list of veterinarians and their specialties	View
US2	As an administrator i want to View information pertaining to a pet owner	View
US3	As an administrator i want to Update the information pertaining to a pet owner	View
US4	As an administrator i want to Add a new pet owner to the system	View
US5	As an administrator i want to View information pertaining to a pet	View
US6	As an administrator i want to Update the information pertaining to a pet	View
US7	As an administrator i want to Add a new pet to the system	View
US8	As an administrator i want to View information pertaining to a pet's visitation history	View
US9	As an administrator i want to Add information pertaining to a visit to the pet's visitation history	View

Figura 32: Lista de items de Petclinic

Luego de procesar los ítem correspondientes al proyecto, se identificaron 13 símbolos, agrupados de la siguiente forma: 1 símbolos de tipo sujeto, 3 símbolos de tipo verbo y 9 símbolos de tipo objeto.

Con el proyecto Petclinic se probaron los procesos esenciales de la propuesta de trazabilidad:

1. Procesamiento de ítems
2. Procesamiento de símbolos
3. Procesamiento de anotaciones

La herramienta desarrollada permite anotar correctamente los métodos y clases en los cuales se logra identificar un símbolo en particular. Es decir se realiza la búsqueda de los métodos y clases, empleando el criterio definido en el proceso propuesto en este trabajo, ver Tabla 4, página 55. Posteriormente se genera la anotación correspondiente, indicando símbolo, tipo de símbolo e ítem.

Finalmente se modifica el archivo elegido, insertando la anotación antes del elemento correspondiente, un nombre de clase o un nombre de método. Se comprueba la existencia de anotaciones existentes antes de insertar una nueva anotación.

9.2 Proyecto Sylius

Es una plataforma de comercio electrónico open source desarrollada en PHP utilizando el Framework Symfony.

Figura 33: Cantidad de archivos por tipo. Sylius.

Este proyecto posee un conjunto de 495 ítems agrupados por características. Ver Apéndice I: Lista de ítems del proyecto Sylius.

Luego de procesar los ítem correspondientes al proyecto, se identificaron 406 símbolos, agrupados de la siguiente forma: 145 símbolos de tipo sujeto, 122 símbolos de tipo verbo y 139 símbolos de tipo objeto.

En la Figura 34, se puede observar una lista parcial de los ítem del proyecto.

Item List

Id	Text	View
US1	As an Administrator I want to be able to delete multiple tax rates at once so that I can remove test, obsolete or incorrect tax rates in an efficient way	View
US2	As an Administrator I want to be prevented from adding two tax rates with same code so that I can uniquely identify tax rates	View
US3	As an Administrator I want to be able to delete a tax rate so that I can remove test, obsolete or incorrect tax rates	View
US4	As an Administrator I want to add a new tax rate to the registry so that I can apply different tax rates to various tax categories	View
US5	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a tax rate	View
US6	As an Administrator I want to be able to edit a tax rate so that I can change taxes applied to products	View
US7	As a Visitor I want to have correct taxes applied to my order when it has a discount so that I can pay proper amount when buying goods	View
US8	As a Visitor I want to have correct taxes applied to my order so that I can pay proper amount when buying goods from the same tax category	View
US9	As a Visitor I want to have correct taxes applied to my order with a discount so that I can pay proper amount when buying goods	View
US10	As a Visitor I want to have correct taxes applied to my order with a discount and tax rates are included in products prices so that I can pay proper amount when buying goods	View
US11	As a Visitor I want to have correct taxes applied to my order so that I can pay proper amount when buying goods from different tax categories	View
US12	As a Visitor I want to have correct taxes applied to my order with a discount and tax rates are included in products prices so that I can pay proper amount when buying goods	View
US13	As a Visitor I want to have correct taxes applied to my order when it has a discount and tax rates are included in products prices so that I can pay proper amount when buying goods	View
US14	As a Customer I want to have correct taxes applied to my order so that I can pay proper amount when buying goods	View

Figura 34: Lista parcial de Items de Sylius

9.2.1 Enfoque de validación

Para estimar de alguna forma el grado de utilidad de la herramienta, en la presente sección se busca responder la pregunta Q1: **¿El enfoque propuesto puede asistir en la resolución de tickets?**

Para responder Q1, se establece el siguiente protocolo:

1. Seleccionar un ticket con estado cerrado.
2. Seleccionar símbolos relevantes desde el texto del ticket elegido.
3. Identificar los archivos de los commits asociados al ticket elegido.
4. Realizar la búsqueda de los símbolos elegidos en los archivos del proyecto utilizando la herramienta desarrollada.
5. Comparar los resultados obtenidos en los pasos 3 y 4.

En la Tabla 5 se pueden observar los distintos tickets que han sido procesados, para cada uno de ellos se indican los símbolos seleccionados.

Tabla 5: Tickets revisados

Nro	Tipo	Texto	Símbolo buscado (Tipo)
10790	Bug	Do not allow to specify shipping charge below 0	shipping (O)
10781	Bug	Allow no default tax zone in channel fixtures	tax (S) zone (O) channel (O)
10679	UX	Change channel code to name	channel (O)
10695	Bug	Fix displayed stocks on product show page	stock (O) product (O)
10658	Feature	Filtering payments by channel	payment (O)
10603	Bug	Promotion integrity checker fix	promotion (O) checkout (V)
10806	Bug	Include order promotion adjustments in order promotion total	promotion (O)
10793	Bug	Validating default locale for a channel	channel(O)

La búsqueda de símbolos del paso 4, se realiza según los criterios establecidos previamente en la Tabla 4. Al realizar la búsqueda de los símbolos de tipo objeto o sujeto, se busca coincidencia en los nombres de las clases y en los nombres de los métodos. Mientras que en el caso del tipo verbo, se buscará solamente en los nombres de los métodos.

Para cada caso evaluado, se calcula la eficacia de la búsqueda realizada de la siguiente manera:

$$\text{Eficacia} = \text{Total de archivos coincidentes} / \text{Total de archivos válidos}$$

Total de archivos coincidentes: Son los casos en los cuales el símbolo elegido ha sido encontrado en el interior del archivo.

Total de archivos válidos: Es la totalidad de archivos en los cuales se puede realizar la búsqueda, para el caso de Sylius, son todos los archivos con extensión php, pues es allí se encuentran los nombres de clases y métodos que pretendemos asociar.

9.2.2 Análisis de tickets

Para el caso del ticket 10790, se puede observar que en los commit asociados al ticket hay incluidos archivos con extensión feature, yml y php, solamente son considerados aquellos con extensión php al momento de efectuar la búsqueda del símbolo **shipping** de tipo **object**.

Tabla 6: Ticket 10790

Archivos	Símbolo (Tipo)
	shipping (object)
shipping_method_flat_rate_per_shipment_validation.feature	n/a
shipping_method_flat_rate_per_unit_validation.feature	n/a
ManagingShippingMethodsContext.php	Si
FlatRateConfigurationType.php	No
PerUnitRateConfigurationType.php	No
validators.en.yml	n/a
% Eficacia	33 %

Si se verifican los archivos en los que no se encontró el símbolo **shipping**, FlatRateConfigurationType.php y PerUnitRateConfigurationType.php, y se revisan las novedades del commit, se puede observar que los cambios se refieren a la utilización de un nuevo rango de validación de la aplicación. Ver Figura 35 y Figura 36.

```

src/Sylius/Bundle/ShippingBundle/Form/Type/Calculator/FlatRateConfigurationType.php
18 18 use Symfony\Component\Form\FormBuilderInterface;
19 19 use Symfony\Component\OptionsResolver\OptionsResolver;
20 20 use Symfony\Component\Validator\Constraints\NotBlank;
21 21 + use Symfony\Component\Validator\Constraints\Range;
22 22 use Symfony\Component\Validator\Constraints\Type;
23 23
24 24 final class FlatRateConfigurationType extends AbstractType
25 25 {
26 26     /**
27 27      * {@inheritdoc}
28 28      */
29 29     public function buildForm(FormBuilderInterface $builder, array $options): void
30 30     {
31 31         $builder
32 32             ->add('amount', MoneyType::class, [
33 33                 'label' => 'sylius.form.shipping_calculator.flat_rate_configuration.amount',
34 34                 'constraints' => [
35 35                     new NotBlank(['groups' => ['sylius']]),
36 36 +                     new Range(['min' => 0, 'minMessage' => 'sylius.shipping_method.calculator.min', 'groups' => ['sylius']]),
37 37                     new Type(['type' => 'integer', 'groups' => ['sylius']]),
38 38                 ],
39 39                 'currency' => $options['currency'],

```

Figura 35: FlatRateConfigurationType.php

```

src/Sylius/Bundle/ShippingBundle/Form/Type/Calculator/PerUnitRateConfigurationType.php
18 18 use Symfony\Component\Form\FormBuilderInterface;
19 19 use Symfony\Component\OptionsResolver\OptionsResolver;
20 20 use Symfony\Component\Validator\Constraints\NotBlank;
21 21 + use Symfony\Component\Validator\Constraints\Range;
22 22 use Symfony\Component\Validator\Constraints\Type;
23 23
24 24 final class PerUnitRateConfigurationType extends AbstractType
25 25 {
26 26     /**
27 27      * {@inheritdoc}
28 28      */
29 29     public function buildForm(FormBuilderInterface $builder, array $options): void
30 30     {
31 31         $builder
32 32             ->add('amount', MoneyType::class, [
33 33                 'label' => 'sylius.form.shipping_calculator.per_unit_rate_configuration.amount',
34 34                 'constraints' => [
35 35                     new NotBlank(['groups' => ['sylius']]),
36 36 +                     new Range(['min' => 0, 'minMessage' => 'sylius.shipping_method.calculator.min', 'groups' => ['sylius']]),
37 37                     new Type(['type' => 'integer', 'groups' => ['sylius']]),
38 38                 ],
39 39                 'currency' => $options['currency'],

```

Figura 36: PerUnitRateConfigurationType.php

Por lo que resulta correcto el cálculo del valor de la Eficacia en 33%.

En el caso del ticket 10781, se realizó la búsqueda de tres símbolos, **tax**, **zone** y **channel**. Ver Tabla 7. En esta oportunidad no se encontraron resultados para los dos primeros, mientras que para el último hubo coincidencia con uno de los archivo referenciado en el commit, el archivo ChannelExampleFactory.php.

Tabla 7: Ticket 10781

Archivos	Símbolo (Tipo)	Símbolo (Tipo)	Símbolo (Tipo)
	tax (S)	zone (O)	channel (O)
ChannelExampleFactory.php	No	No	Si
LazyOption.php	No	No	No
% Eficacia	0 %	0 %	50 %

Si se observa con detalle los cambios realizados en el commit, ver Figura 37, se puede inferir que la clase LazyOption definida en el archivo LazyOption.php es una clase auxiliar.

```

src/Sylus/Bundle/CoreBundle/Fixture/Factory/ChannelExampleFactory.php
@@ -136,7 +136,7 @@ protected function configureOptions(OptionsResolver $resolver): void
136 136     ->setAllowedTypes('skipping_payment_step_allowed', 'bool')
137 137     ->setDefault('account_verification_required', true)
138 138     ->setAllowedTypes('account_verification_required', 'bool')
139 139     ->setDefault('default_tax_zone', LazyOption::randomOne($this->zoneRepository))
139 139     +>setDefault('default_tax_zone', LazyOption::randomOneOrNull($this->zoneRepository))
140 140     ->setAllowedTypes('default_tax_zone', ['null', 'string', ZoneInterface::class])
141 141     ->setNormalizer('default_tax_zone', LazyOption::findOneBy($this->zoneRepository, 'code'))
142 142     ->setDefault('tax_calculation_strategy', 'order_items_based')

src/Sylus/Bundle/CoreBundle/Fixture/OptionsResolver/LazyOption.php
@@ -51,7 +51,7 @@ public static function randomOne(RepositoryInterface $repository): \Closure
51 51     };
52 52     }
53 53
54 54     - public static function randomOneOrNull(RepositoryInterface $repository, int $chanceOfRandomOne): \Closure
54 54     + public static function randomOneOrNull(RepositoryInterface $repository, int $chanceOfRandomOne = 100): \Closure
55 55     {
56 56         return function (Options $options) use ($repository, $chanceOfRandomOne) {
57 57             if (random_int(1, 100) > $chanceOfRandomOne) {

```

Figura 37: Commit asociado al ticket 10781

En el caso del ticket 10679 el resultado de la búsqueda fue positivo, ver Tabla 8.

Tabla 8: Ticket 10679

Archivos	Símbolo (Tipo)
	channel (O)
ManagingShipmentsContext.php	Si
channel.html.twig	n/a
% Eficacia	100 %

Para el caso del ticket 10695, ver Tabla 9, ninguno de los archivos posee la extensión php, por lo que no pudo realizar la búsqueda en los mismos.

Tabla 9: Ticket 10695

Archivos	Símbolo (Tipo)	Símbolo (Tipo)
	stock (O)	product (O)
_details.html.twig	n/a	n/a
_variants.html.twig	n/a	n/a
% Eficacia		

En el caso del ticket 10658, ver Tabla 10, se identificaron cuatro archivos con extensión php.

Tabla 10: Ticket 10658

Archivos	Símbolo (Tipo)
	payment (O)
filtering_payments_by_channel.feature	n/a
ManagingPaymentsContext.php	Si
IndexPage.php	Si
IndexPageInterface.php	Si
payment.yml	n/a
OrderContext.php	No
managing_payments.yml	n/a
% Eficacia	75 %

Si se revisa el archivo OrderContext.php, Figura 38, se puede observar que no se encuentra el símbolo **payment** ni en los cambios efectuados ni en ninguna otra parte de código.

9. Validación de la propuesta de trazabilidad

```

@@ -320,9 +320,9 @@ public function theCustomerChosePayment(PaymentMethodInterface $paymentMethod)
320 320     * @Given the customer bought a single :product
321 321     * @Given I bought a single :product
322 322     */
323 - public function theCustomerBoughtSingleProduct(ProductInterface $product)
323 + public function theCustomerBoughtSingleProduct(ProductInterface $product, ?ChannelInterface $channel = null)
324 324     {
325 - $this->addProductVariantToOrder($this->variantResolver->getVariant($product), 1);
325 + $this->addProductVariantToOrder($this->variantResolver->getVariant($product), 1, $channel);
326 326
327 327     $this->objectManager->flush();
328 328     }
@@ -412,15 +412,21 @@ public function iHaveAlreadyPlacedOrderNthTimes(
412 412
413 413     /**
414 414     * @Given there is an :orderNumber order with :product product
415 + * @Given there is an :orderNumber order with :product product in this channel
416 + * @Given there is an :orderNumber order with :product product in :channel channel
417 417     * @Given there is a :state :orderName order with :product product
418 418     */
419 - public function thereIsAOrderWithProduct(string $orderNumber, ProductInterface $product, string $state = null): void
419 + {
420 - $order = $this->createOrder($this->createOrProvideCustomer('amba@fatima.org'), $orderNumber);
420 + public function thereIsAOrderWithProduct(
421 + string $orderNumber,
422 + ProductInterface $product,
423 + string $state = null,
424 + ?ChannelInterface $channel = null
425 + ): void {
426 - $order = $this->createOrder($this->createOrProvideCustomer('amba@fatima.org'), $orderNumber, $channel);
426 + $order = $this->createOrder($this->createOrProvideCustomer('amba@fatima.org'), $orderNumber, $channel);
427 427
428 428     $this->sharedStorage->set('order', $order);
429 429
430 - $this->theCustomerBoughtSingleProduct($product);
430 + $this->theCustomerBoughtSingleProduct($product, $channel);
431 431
432 432     $this->createShippingPaymentMethodsAndAddress();
@@ -702,13 +702,16 @@ private function applyTransitionOnOrder(OrderInterface $order, string $transiti
702 702     *

```

Figura 38: Cambios en OrderContext.php

Para el ticket 10603, ver Tabla 11, se realizó la búsqueda de los símbolos **promotion** y **checkout**.

Tabla 11: Ticket 10603

Archivos	Símbolo (Tipo)
	promotion (O)
order_promotion_integrity_validation.feature	n/a
CheckoutCompleteContext.php	Si
CompletePage.php	Si
CompletePageInterface.php	Si
OrderIntegrityChecker.php	No
OrderPromotionIntegrityChecker.php	No
OrderTotalIntegrityChecker.php	No
listeners.xml	n/a
SyliusShopExtensionTest.php	No
OrderIntegrityCheckerSpec.php	Si

9. Validación de la propuesta de trazabilidad

OrderPromotionIntegrityCheckerSpec.php	No
OrderTotalIntegrityCheckerSpec.php	No
% Eficacia	40 %

Para el ticket 10806, ver Tabla 12, se realizó la búsqueda de los símbolos **promotion**. En esta oportunidad se logró encontrar todos los símbolos en los archivos modificados en el commit.

Tabla 12: Ticket 10806

Archivos	Símbolo (Tipo)
	promotion (O)
Order.php	Si
OrderSpec.php	Si
% Eficacia	100 %

En el caso del ticket 10793, ver Tabla 13, se realizó la búsqueda del símbolo **locale** y **channel**. Se localizó el símbolo **channel** en 5 de 7 archivos asociados al commit, mientras que **locale** se halló en la totalidad de los archivos asociados al commit. Al revisar los archivos CreatePage.php y UpdatePage.php, se puede notar que hubiese resultado deseable localizar el símbolo **channel**, en los mismos, puesto que hacen referencia a cierta configuración del canal utilizado por defecto al momento de la creación y su posterior modificación. Ver Figura 39.

Tabla 13: Ticket 10793

Archivos	Símbolo (Tipo)	Símbolo (Tipo)
	locale (O)	channel (O)
Selecting_locales_available_for_channel.feature		n/a
LocaleContext.php	Si	Si
ManagingChannelsContext.php	Si	Si
CreatePage.php	Si	No
UpdatePage.php	Si	No
Validators.xml	n/a	n/a
Channel.xml	n/a	n/a
Validators.en.yml	n/a	n/a
ChannelDefaultLocaleEnabled.php	Si	Si

9. Validación de la propuesta de trazabilidad

ChannelDefaultLocaleEnabledValida- tor.php	Si	Si
ChannelDefaultLocaleEnabledValida- torSpec.php	Si	Si
% Eficacia	100 %	71 %

Figura 39: Cambios en CreatePage.php UpdatePage.php

9.2.3 Limitaciones

Mientras se simuló de resolución de tickets, se logró identificar una serie de restricciones. La primera se refiere a los tipos de archivos en los cuales se realiza la búsqueda. Como se mencionó previamente, la herramienta soporta el procesamiento de archivos con extensiones PHP y JAVA, propios de cada lenguaje. Sin embargo los proyectos evaluados proyectos están compuestos por una amplia variedad de tipos de archivos, HTML, YML, TWIG, XML, JSON, etc. En el proyecto Sylius se puede ver que la cantidad de archivos con extensión PHP es similar a la cantidad de archivos con extensión YML. Esta situación, permite afirmar que sería muy conveniente que la herramienta soporte el procesamiento de distintos tipos de archivo. Como consecuencia de este planteo, sería necesario adaptar el modelo de trazabilidad propuesto debido a que

esta diseñado para soportar trazas entre ítems, elementos del LEL y elementos del paradigma orientado a objetos (clases, atributos o métodos).

La segunda situación que se logra advertir como poco favorable, al utilizar la herramienta como medio para asistir en la resolución de tickets, es que en algunas oportunidades el código al que se busca hacer referencia o que se debe modificar está dentro de clases auxiliares, en estos casos resulta habitual que no se haga mención a ningún símbolo en nombres de métodos o nombres de clases, por lo tanto la búsqueda resulta infructuosa.

9.2.4 Beneficios

Como resultado favorable del uso de la herramienta, se observa que la búsqueda de símbolos en los nombres de clases y nombres de métodos se realiza adecuadamente, no se obtuvieron falsos positivos. Al realizar la búsqueda no estaban incluidos como parte del resultado aquellos archivos en los cuales figuraba el símbolo como parte de un comentario o como parte de una invocación al método.

9.2.5 Resultado

Como respuesta a la pregunta efectuada al inicio de éste capítulo, Q1: **¿El enfoque propuesto puede asistir en la resolución de tickets?**. Basados en la variedad de tickets revisados (errores, interfaz de usuario, nuevas características); se puede decir que efectivamente, el enfoque propuesto puede asistir en la resolución de tickets. Por otro lado sería beneficioso para esta tarea de resolución de tickets, llevar a cabo mejoras en la herramienta desarrollada, por ejemplo soportar el procesamiento de una mayor variedad de formatos de archivos.

10 Discusión sobre la propuesta de trazabilidad

En este capítulo se realizará una revisión cualitativa del proceso de trazabilidad propuesto, para ello se empleará como criterio las actividades enumeradas por (Bouillon et al., 2013). En la primera parte de este capítulo se realizará una discusión utilizando el criterio presentado. En la segunda parte de este capítulo, se mencionarán a modo de resumen, los resultados observados de la discusión planteada previamente. Para finalizar este capítulo se presentan algunos beneficios y dificultades encontrados durante la implementación de la propuesta de trazabilidad.

10.1 Actividades revisadas

A continuación se revisará si las actividades presentadas por (Bouillon et al., 2013) son soportadas por el proceso de trazabilidad propuesto. Al finalizar cada sección, se presentará una tabla a modo de resumen, en la misma se indicará el grado⁷ de soporte ofrecido por la propuesta de trazabilidad para cada actividad.

10.1.1 Ingeniería y gestión de requerimientos

a) Encontrar el origen y la justificación de los requerimientos, es decir, la trazabilidad de los pre-requerimientos a regulaciones y a otras fuentes de requerimientos

Esta actividad no es apoyada por la propuesta de trazabilidad.

b) Refinamiento y detalle de requerimientos

Para el caso que un ítem sea una historia de usuario épica, éste puede ser refinado en otros ítems. Además, al completar la información correspondiente a cada símbolo existente en el LEL, como ser la noción y el impacto, se consigue un mayor detalle de los requerimientos.

c) Documentar el historial de un requerimiento, es decir, para poder rastrear versiones anteriores de un requerimiento con el fin de conocer los cambios

⁷ S: Soportada, PS: Parcialmente Soportada, NS: No Soportada

En la propuesta se define que los ítem deben existir como archivos individuales que coexistan con el código fuente, esto último junto con la utilización de un sistema de control de versiones permite rastrear versiones anteriores de cada ítem e identificar cambios en los requerimientos.

d) Identificar a las partes interesadas para el desarrollo continuo de los requerimientos

La estructura propia de cada ítem de poseer secciones "Como ...", "Quiero ...", "Para poder ..."; propicia la identificación de algunos interesados. Además al ser un proyecto ágil, se cuenta con la presencia de algún responsable del negocio o alguien que desempeña dicho rol, 4^{to} principio del manifiesto ágil.

e) Análisis de calidad y madurez de los requerimientos

El uso de LEL como diccionario de términos ayuda a resolver problemas de ambigüedad, completitud, comprensión y consistencia que pudieran surgir.

f) Análisis de impacto, que otras partes interesadas son importantes por un cambio en un requerimiento

Como se menciona en (Urbietta et al., 2020), otros interesados pueden ser identificados luego de acceder a los símbolos que forman parte de un ítem en particular.

Tabla 14: Soporte de la propuesta a las actividades de Ingeniería y Gestión de Requerimientos

Ingeniería y gestión de requerimientos	
a) Encontrar el origen y la justificación de los requerimientos, es decir, la trazabilidad de los pre-requerimientos a regulaciones y a otras fuentes de requerimientos	NS
b) Refinamiento y detalle de requerimientos	S
c) Documentar el historial de un requerimiento, es decir, para poder rastrear versiones anteriores de un requerimiento con el fin de conocer los cambios	S
d) Identificar a las partes interesadas para el desarrollo continuo de los requerimientos.	S
e) Análisis de calidad y madurez de los requerimientos.	PS
f) Análisis de impacto, que otras partes interesadas son importantes por un cambio en un requerimiento	S

10.1.2 *Gestión de proyectos*

a) Seguimiento del estado de un requerimiento o implementación de tareas en detalle

Se puede emplear un estado para indicar la situación de cada ítem. Además se puede determinar rápidamente si un ítem posee fragmentos de código vinculados a él y de esta manera saber si dicho ítem posee algún grado de avance.

b) Planificación de entrega inicial

Al utilizar un sistema de control de versiones para resguardar los ítems a lo largo del tiempo, se puede emplear el mismo sistema de control de versiones para agrupar los ítems bajo algún rótulo, de esta manera considerar que ciertos ítems pertenecen a una determinada entrega.

c) Evaluación de progreso a nivel de proyecto o subproyecto para obtener una visión general de los requerimientos ya implementados

Se puede incorporar el uso de rotular ítems para agruparlos en un mismo sprint, junto con la gestión de estados permitiría tener una visión global del proyecto.

d) Asignación de tareas a las partes interesadas, por ejemplo, asignación de un requerimiento a un desarrollador para su implementación

Se puede utilizar algún sistema de gestión de tickets integrado con el repositorio y de esta manera permitir asignar tareas asociadas a un ítem a ciertos miembros del equipo de desarrollo. Sin embargo este aspecto no está considerado como parte de la propuesta de trazabilidad.

e) Notificación a las partes interesadas sobre los cambios, por ejemplo, después de un cambio en un requerimiento, todos los propietarios de artefactos dependientes son informados automáticamente

Como se menciona en (Urbietta et al., 2020), se podría notificar de manera automática a los usuarios que poseen el rol de un ítem luego de efectuarse cambios en el mismo.

De manera análoga, al contar con un sistema de control de versiones, podría notificar al desarrollador del código vinculado al ítem que se modificó.

f) Ajuste del proyecto y plan de entrega, por ejemplo, en caso de que se exceda el límite de tiempo

Se podría registrar el ajuste en el alcance de la entrega suprimiendo el rótulo asociado al ítem.

Tabla 15: Soporte de la propuesta a las actividades de Gestión de Proyectos

Gestión de proyectos	
a) Siguiendo el estado de requerimiento o implementación de tareas en detalle	PS
b) Planificación de entrega inicial	PS
c) Evaluación de progreso a nivel de proyecto o subproyecto para obtener una visión general de los requerimientos ya implementados	S
d) Asignación de tareas a las partes interesadas, por ejemplo, asignación de un requerimiento a un desarrollador para su implementación	NS
e) Notificación a las partes interesadas sobre los cambios, por ejemplo, después de un cambio en un requerimiento, todos los propietarios de artefactos dependientes son informados automáticamente	PS
f) Ajuste del proyecto y plan de entrega, por ejemplo, en caso de que el límite de tiempo exceda	PS

10.1.3 Demostración de cumplimiento

a) Análisis de la cobertura de requerimientos en el código fuente, por ejemplo, para el cliente

Esta actividad se puede efectuar a través del análisis de los fragmentos de código vinculados a cada ítem.

b) Documentación de trazabilidad para fines de certificación

Al aplicar la propuesta de trazabilidad definida en este trabajo, se puede emplear la herramienta desarrollada como medio de consulta para generar la documentación de trazabilidad.

c) Justificación de todo el código escrito basado en la especificación para propósitos de certificación

Se podría emplear un IDE junto la herramienta desarrollada para determinar aquellos fragmentos de código que no estén vinculados de manera directa o indirecta con ningún ítem.

Tabla 16: Soporte de la propuesta a las actividades de Demostración de cumplimiento

Demostración de cumplimiento	
a) Análisis de la cobertura de requerimientos en el código fuente, por ejemplo, para el cliente	S
b) Documentación de trazabilidad para fines de certificación.	S
c) Justificación de todo el código escrito basado en la especificación para propósitos de certificación	S

10.1.4 Diseño e Implementación

a) Navegación entre especificación, diseño, prueba y código a través de trazas

A cada ítem del proyecto se pueden vincular fragmentos de código que respondan a alguna funcionalidad, como así también fragmentos de código que se correspondan con distintos casos de prueba. Sin embargo no resulta aplicable para navegar hacia especificaciones de diseño.

b) Navegación dentro de artefactos del mismo tipo, por ejemplo, dentro del código fuente

Al disponer de un LEL construido con símbolos provenientes de los ítems y que dichos símbolos estén vinculados con su correspondiente fragmento de código, se puede entonces a partir de un símbolo en particular navegar entre ítems o bien entre fragmentos de código que tengan el mencionado símbolo en común.

c) Evaluación de diseño basada en métricas de trazabilidad, por ejemplo, para encontrar componentes que contengan demasiada funcionalidad y que deberían dividirse

A partir del análisis de la cantidad de vinculaciones realizadas en los fragmentos de código, se puede tener una noción de componentes sobrecargados

d) Comprensión de los artefactos de software, por ejemplo, familiarización del proyecto de los miembros del equipo de desarrollo

Se puede emplear la propuesta de trazabilidad para lograr una comprensión del software de dos maneras, la primera es utilizando como punto de entrada cada ítem y desde allí tener un panorama de los fragmentos vinculados a él; la segunda manera es mediante el uso de LEL, en este caso se emplean los símbolos como punto de acceso para revisar los fragmentos de código a él relacionados.

Tabla 17: Soporte de la propuesta a las actividades de Diseño e Implementación

Diseño e Implementación	
a) Navegación entre especificación, diseño, prueba y código a través de trazas	PS
b) Navegación dentro de artefactos del mismo tipo, por ejemplo, dentro del código fuente	S
c) Evaluación de diseño basada en métricas de trazabilidad, por ejemplo, para encontrar componentes que contengan demasiada funcionalidad y que deberían dividirse	S
d) Comprensión de los artefactos de software, por ejemplo, familiarización del proyecto de los miembros del equipo de desarrollo.	S

10.1.5 Prueba

a) Desarrollo de casos de prueba basados en requerimientos

Al contar con los ítems en archivos separados se puede implementar BDD⁸ ampliando el contenido del ítem y registrando luego los casos de prueba relacionados con dicho ítem. Una alternativa posible es vincular los archivos de casos de prueba directamente con los ítems y símbolos correspondientes.

b) Ubicación del defecto dentro del código fuente para casos de prueba fallidos

Sería necesario en este punto pensar en relaciones entre fragmentos de código, esta posibilidad no está contemplada en la actual propuesta.

⁸ El desarrollo guiado por el comportamiento o BDD es un enfoque del proceso de desarrollo de software que proporciona a los equipos de gestión y desarrollo de software herramientas compartidas y un proceso compartido para colaborar en el desarrollo de software.

c) Descubrir pruebas de regresión para ejecutar después del cambio de código

Luego de realizar las pruebas correspondientes a un ítem en particular, se puede definir algún criterio para ejecutar las pruebas asociadas a otros ítems, por ejemplo por tener en común el símbolo sujeto o símbolo objeto asociados, o bien por poseer un determinado rótulo asociado a un sprint en particular.

d) Prueba de análisis de cobertura de especificación y de código

Esta actividad no es apoyada por la propuesta de trazabilidad.

e) Identificación de las partes interesadas para comprender el comportamiento y resolver problemas complicados

Se puede emplear un enfoque similar al empleado con la actividad d) de las actividades de Ingeniería y Gestión de Requerimientos. Es decir aprovechar el formato propio de la historias de usuario para identificar partes interesadas.

Tabla 18: Soporte de la propuesta a las actividades de Prueba

Prueba	
a) Desarrollo de casos de prueba basados en requerimientos.	PS
b) Ubicación del defecto dentro del código fuente para casos de prueba fallidos	NS
c) Descubrir pruebas de regresión para ejecutar después del cambio de código	S
d) Prueba de análisis de cobertura de especificación y código	NS
e) Identificación de las partes interesadas para comprender el comportamiento y resolver problemas complicados.	S

10.1.6 Mantenimiento y evolución

a) Análisis de impacto del cambio para determinar los artefactos afectados por una extensión de característica

Una vez registrada la relación entre fragmentos e ítems, se puede determinar fácilmente qué artefactos se verían afectados ante posibles cambios

b) Estimación del esfuerzo del cambio para extensiones de características

Se puede analizar la cantidad de fragmentos que están vinculados con un ítem, esto puede brindar una idea de la magnitud del cambio. Si se cuenta además con el tiempo de realización del ítem que se desea modificar, la estimación que se realice puede ser bastante acertada.

c) Ubicación y soporte de las características durante la implementación del cambio mediante el uso de trazas

La vinculación entre ítem y porciones de código se puede realizar desde el inicio de la implementación, es decir se pueden vincular con clases y métodos que aún están vacíos. Por consiguiente se puede lograr una gestión de la trazabilidad durante la implementación del cambio.

d) Reutilización de componentes de especificación y código, por ejemplo, una característica con toda su implementación

Un mismo fragmento de código se puede corresponder con más de ítem o con más de un elemento de LEL, por lo que la reutilización de código está soportada.

e) Transferencia de conocimiento al equipo de mantenimiento, por ejemplo, en los casos en que un equipo realiza mantenimiento que no incluye ninguno de los miembros originales del equipo

La utilización de LEL como elemento mediador entre ítems y fragmentos de código, facilita la adquisición de conocimiento sobre el dominio de la aplicación, como así también una idea de cual es la importancia de cada elemento del LEL al considerar todos los fragmentos con los que se relaciona.

Tabla 19: Soporte de la propuesta a las actividades de Mantenimiento y evolución

Mantenimiento y evolución	
a) Cambiar el análisis de impacto para determinar los artefactos afectados por una extensión de característica	S
b) Cambiar la estimación del esfuerzo para extensiones de características	S
c) Ubicación de características y soporte durante la implementación del cambio mediante el uso de trazas	S
d) Reutilización de componentes de especificación y código, por ejemplo, una característica con toda su implementación	S
e) Transferencia de conocimiento al equipo de mantenimiento, por ejemplo, en los casos en que un equipo realiza mantenimiento que no incluye ninguno de los miembros originales del equipo	S

10.2 Resultado

Luego de revisar las actividades que pueden ser soportadas por la trazabilidad se elabora el gráfico que se puede ver en la Figura 40. Dicho gráfico facilita la determinación de algunas observaciones.

Figura 40: Actividades que pueden ser soportadas por la trazabilidad de requerimientos

1. Todas las actividades referidas al área de demostración de cumplimiento y las que pertenecen al área de mantenimiento son soportadas por la propuesta de trazabilidad.
2. Son 18 las actividades soportadas por la propuesta de trazabilidad, sobre un total de 29.

10.3 Beneficios

Un beneficio observable es el que proviene del empleo de la propuesta como medio para la resolución de tickets. En este caso el uso de LEL como elemento intermediario entre ítems y código fuente, da la posibilidad de contar con símbolos derivados de las historias de usuario, dichos símbolos resultan significativos y pueden utilizarse al menos de dos maneras. La primera forma de empleo, como parte de un índice para con-

sultar información sobre ítems o sobre fragmentos de código. La segunda forma de empleo, es como término de búsqueda en el interior del código fuente, como parte de nombres de clase o nombres de métodos.

Un aspecto que se considera positivo es la facilidad de identificación de los fragmentos de código pertenecientes a una historia de usuario una vez que los distintos artefactos de software han sido debidamente vinculados.

Otro aspecto positivo es el hecho de almacenar en un mismo repositorio el código y las historias de usuario, esto colabora en la simplicidad de la propuesta y facilita el control de la coherencia entre requerimientos.

10.4 Desventajas

Mientras se lleva adelante la gestión de la trazabilidad empleando la propuesta aquí definida, se puede observar que existe una carga adicional de trabajo para quien escribe el código fuente de la aplicación. Esta carga proviene de incorporar la información referida a la traza entre el código que escribe y su correspondiente símbolo e historia de usuario. Sin embargo, apoyados en el trabajo realizado por, (Ji et al., 2015), se puede afirmar que el costo de crear y, lo que es más importante, de mantener el modelo de trazabilidad es insignificante comparado con el costo de buscar la información referida al código perteneciente a una historia de usuario.

Otra dificultad que surge de la utilización de la propuesta, es la que implica aprender los conceptos empleados para gestionar la trazabilidad, como así también familiarizarse con el uso de la herramienta desarrollada.

Otra dificultad observada al implementar la propuesta en el proyecto Sylus, es el tiempo que la herramienta demora en procesar las historias de usuario y vincular con ellas los símbolos encontrados. Para la lectura y procesamiento de 575 historias de usuario se emplearon 28 segundos. Para realizar la vinculación entre historias de usuario y símbolos, se demoró 27 segundos.

11 Conclusión

En el presente trabajo se ha definido un proceso para facilitar la gestión de la trazabilidad entre las Historia de Usuario (ítems), LEL y el código fuente.

Junto con el proceso que permite la vinculación, se desarrolló una herramienta software para gestionar la información referida a las trazas entre los elementos de software mencionados.

Posteriormente se implementó el proceso propuesto en un proyecto real, evaluando los resultados provenientes de utilizar la herramienta desarrollada como medio para asistir en la resolución de tickets. Como resultado de dicha evaluación se pueden realizar las siguientes afirmaciones, la primera es que resulta beneficioso el empleo de la herramienta construida en la tarea de resolver tickets; la segunda es que agregando cierta funcionalidad adicional (procesamiento de otros tipos de archivos) se lograría mejorar el desempeño logrado al utilizar la herramienta desarrollada para resolver tickets.

También se hizo una revisión de una lista actividades propias de un proyecto de software que pueden ser respaldadas por la trazabilidad de los requerimientos y cómo la propuesta de trazabilidad desarrollada puede colaborar en cada una de ellas. A partir del análisis y discusión realizados se puede observar que más del 60% de las actividades están soportadas por la propuesta de trazabilidad, particularmente las actividades referidas al mantenimiento y evolución junto con las pertenecientes a la demostración de cumplimiento están soportadas por la propuesta de trazabilidad planteada.

Finalmente se mencionan algunos beneficios y limitaciones que han surgido del trabajo realizado. Por lo que se afirma que el proceso propuesto puede colaborar en la tarea del desarrollo de software, facilitando el acceso a información de trazabilidad entre los requerimientos, expresados como historias de usuario y el código propio de la aplicación. No obstante es preciso mencionar que sería conveniente probar la propuesta de trazabilidad en tareas de mantenimiento de un proyecto real, para poder evaluar el beneficio de contar con la información referida a la trazabilidad.

11.1 Consideraciones finales

Se considera como un aspecto novedoso de este trabajo la incorporación de LEL como elemento intermediario entre las historias de usuario y el código fuente.

El segundo elemento distintivo aquí presentado es la idea de conservar en un mismo repositorio tanto los requerimientos expresados en el formato de historias de usuario, como la información referente a la trazas que vinculan los requerimientos con el código fuente. Como se menciona en el Capítulo 3.4, las historias de usuarios son hoy en día una técnica ampliamente utilizada en las metodologías ágiles.

11.2 Trabajos futuros

Durante la realización de este trabajo de tesis han surgido interrogantes de distinta índole que originan diversos trabajos a futuro, se mencionan a continuación los mismos. Como se dijo previamente, resultaría conveniente probar la propuesta en tareas de mantenimiento, esto se podría realizar comparando el tiempo que demanda efectuar cambios en el código utilizando la propuesta de trazabilidad con el tiempo necesario para efectuar idéntica tarea sin emplear la propuesta.

Un posible trabajo futuro radica en la incorporación de BDD como parte de las historias de usuario y consecuentemente conectar escenarios con fragmentos de código.

Consideramos que una mejora sustancial a este trabajo sería la posibilidad de realizar la vinculación automática entre porciones código y símbolos e ítems, tal vez utilizando algún nivel de IA.

Otro punto de interés es la posibilidad de incorporar la gestión de trazabilidad de los requerimientos no funcionales. Recordemos que un requerimiento no funcional puede afectar a más de un requerimiento funcional, esto hace que sea necesario contar con un modelo adecuado para gestionar este tipo de información de trazabilidad.

Desde un punto de vista más técnico y considerando la herramienta desarrollada, vemos adecuado que la misma pueda realizar el procesamiento de distintos tipos de archivos para un mismo proyecto.

12 Trabajos relacionados

Existen algunos autores que han contribuido en esta área del conocimiento, a continuación se hace referencia a aquellos que se consideran mas importantes y para cada caso se menciona que elementos lo diferencian del trabajo de tesis acá presentado.

Por ejemplo (Delater, 2013) apunta directamente a resolver el problema de trazabilidad entre requerimientos y código y de qué manera los desarrolladores utilizan estos vínculos durante su trabajo. Para ello, propone un enfoque que consta de tres partes. La primera parte comprende un modelo de información de trazabilidad (TIM) que consta de artefactos de tres áreas diferentes, ingeniería de requisitos, gestión de proyectos y código. El TIM también incluye los vínculos de trazabilidad entre ellos. La segunda parte presenta tres procesos para capturar vínculos de trazabilidad entre requisitos, elementos de trabajo y código durante el desarrollo. La tercera parte define un algoritmo que infiere automáticamente vínculos de trazabilidad entre los requisitos y el código en función de los elementos de trabajo interconectados.

El enfoque de trazabilidad se implementa como una extensión de la herramienta CASE basada en modelos UNICASE, que se llama UNICASE Trace Client.

A diferencia de la propuesta presentada en esta tesis, (Delater, 2013) , almacena toda la información de la trazabilidad, artefactos de trazabilidad y trazas entre dichos elementos en un ambiente distinto al repositorio del código fuente.

Un trabajo que intenta resolver la trazabilidad utilizando anotaciones embebidas en el código es el realizado por (Ji et al., 2015), en él se se construye un modelo de características y cada característica es vinculada con nuevos archivos denominados “activos” que son creados en el repositorio donde se almacena el código fuente.

Sumado a esto, todo el trabajo de creación y mantenimiento de la información referida a la trazabilidad se realiza de forma manual.

A diferencia de la presente tesis, no se realizan trazas hacia historias de usuario.

Por otro lado, (Wijesinghe et al., 2014) ante el problema de mantener la consistencia entre los artefactos de software que son creados durante el desarrollo, plantea como solución mantener los vínculos de trazabilidad entre dichos artefactos y actualizar esos artefactos en consecuencia. Para lograr esto, proporciona una herramienta para establecer vínculos de trazabilidad y visualización de artefactos.

Los distintos artefactos de software son traducidos en archivos XML de manera colaborativa con el usuario de la aplicación. Después del proceso de extracción de artefactos, las relaciones entre los artefactos se construyen mediante un proceso semiautomático. Los usuarios pueden proporcionar retroalimentación para las relaciones mientras que los algoritmos de aprendizaje pueden extraer patrones específicos involucrados con un entorno específico. Estos patrones extraídos se utilizan para obtener relaciones en iteraciones posteriores.

Después de crear los archivos XML de relaciones, las relaciones se almacenan en una base de datos de grafos. Mientras ocurre el desarrollo incremental, se analizan los artefactos y se proporciona información sobre posibles violaciones de relaciones y la posible corrección.

Este trabajo no es adecuado al problema presentado en esta tesis debido a que los diversos artefactos que se pueden vincular existen fuera del código fuente.

En su trabajo, (Spanoudakis et al., 2004), presenta un enfoque basado en reglas para respaldar la generación automática de relaciones de trazabilidad entre documentos que especifican los requerimientos y casos de uso (expresados en formas estructuradas de lenguaje natural) y modelos de objetos del análisis. La generación de tales relaciones se basa en reglas de trazabilidad de dos tipos diferentes. La primera, reglas de requisito a modelo de objeto, para rastrear los requisitos y documentos de especificación de casos de uso a un modelo de objeto de análisis. La segunda, son reglas de trazabilidad entre requisitos, para rastrear los documentos de requisitos y especificaciones de casos de uso entre sí.

Para implementar y demostrar este enfoque, se ha implementado un sistema prototipo de trazabilidad. Este sistema asume documentos de especificación de casos de uso y requisitos y modelos de objetos de análisis representados en XML. También utiliza re-

glas de trazabilidad que están representadas en un lenguaje de marcado de reglas basado en XML que ha desarrollado para este propósito.

En este trabajo se establecen vínculos entre requerimientos, casos de uso y modelos de objeto del análisis, por lo tanto no se logra una trazabilidad que permite llegar desde los requerimientos hasta el código.

En su trabajo, (Rahimi & Cleland-Huang, 2018) presentan una solución para automatizar la evolución de enlaces de rastreo bidireccionales entre clases de código fuente o métodos y requisitos de lenguaje natural no estructurado.

La solución propuesta toma como entrada un conjunto inicial de requisitos, código fuente y enlaces de seguimiento, así como una versión revisada de requisitos y código fuente. Luego, detecta un conjunto de escenarios de cambio de alto nivel que se han producido entre las dos versiones. Cada escenario de cambio está asociado con un conjunto correspondiente de heurísticas de enlaces de seguimiento que especifican exactamente qué enlaces deben crearse y/o eliminarse para desarrollar enlaces de seguimiento. Mediante el uso de la heurística de evolución de enlaces de seguimiento, se desarrollan automáticamente enlaces de seguimiento en respuesta a los escenarios de cambio detectados.

Este trabajo sólo ha sido utilizado en proyectos Java, debido a la madurez de las herramientas de refactoring empleadas en el mismo.

Este trabajo no resulta adecuado para resolver el problema planteado en esta tesis puesto que no se utilizan historias de usuario como artefacto durante el desarrollo, en el mismo sentido tampoco se almacena la información referida a la trazabilidad en un mismo repositorio que el código fuente.

13 Apéndices

Apéndice I: Lista de ítems del proyecto Sylius

Id	Texto
US1	As an Administrator I should be able to translate an entity in all locales, not just the channel ones so that I can translate my application properly
US2	As a Customer I want to view texts in my preferred locale so that I can fully understand the texts on the shop
US3	As a Customer I want to be able to switch locales so that I can browse shop in my preferred locale
US4	As an Administrator I want to add a new locale to the registry so that I can provide service to my customers in their preferred language
US5	As an Administrator I want to be prevented from adding a locale with an existing code so that I can avoid making mistakes when managing locales
US6	As an Administrator I want to be prevented from adding a locale without a code so that I can avoid making mistakes when managing locales
US7	As a Customer I want to browse channels with a valid locale only so that I can see right prices
US8	As an Administrator I want to see overall statistics on my admin dashboard in a specific channel so that I can have an overview of my sales
US9	As an Administrator I want to see overall statistics on my admin dashboard so that I can have an overview of my sales
US10	As a Visitor I want to be able to log in to the store so that I can view my orders
US11	As an Administrator I want to be able to switch locales so that I can see the panel in my preferred language
US12	As a logged out Administrator I need to be redirected on login page so that I can access on my admin dashboard
US13	As an Administrator I want to impersonate a shop user so that I can provide a top-notch customer support
US14	As an Administrator I want to be able to place an order as an impersonated shop user so that I can provide a customer support
US15	As a Customer I want to be able to have my cart cleared after logging out so that I can not allow to use my cart by anybody
US16	As a Visitor I want to have proper product image displayed in the cart In order be aware of how the product that I'm buying looks like
US17	As a Visitor I want to be able to change quantity of an item in my cart so that I can buy chosen quantity of a specific product
US18	As a Visitor I want to have total value of my cart items rounded properly so that I can pay proper value of my cart
US19	As a Visitor I want to be able to add products with selected options to cart so that I can select specific variant of product for purchase
US20	As a Visitor I want to be able to clear my cart so that I can quick start shopping again
US21	As a Customer I want to be only be charged for shipping when it is necessary so that I can pay the correct amount for my order
US22	As a Customer I want to add products to my cart with prices dependent on visited channel so that I can buy products in correct prices
US23	As a Visitor I want to be able to see my cart summary with the correct shipping method based on variants so that I can see details about my order
US24	As a Visitor I want to be able to remove cart item so that I can delete some unnecessary cart items
US25	As an Administrator I want to have expired carts automatically deleted so that I can remove started but not finished orders after 2 days of idleness
US26	As a Visitor I want to be able to add products with selected variants to cart so that I can select products for purchase
US27	As a Visitor I want to see different carts in different channels so that I can see details about my order in selected channel
US28	As a Visitor I want to be able to have my cart maintained so that I can not lose cart after authorization
US29	As a Visitor I want to be able to add a simple product with stated quantity to the cart so that I can buy multiple items at once
US30	As a visitor I want to be able to see my cart summary so that I can see details about my order
US31	As a Visitor I want to be able to add simple products to cart so that I can select products for purchase
US32	As a Visitor I want to be able to have my cart previous maintained so that I can retrieve my previous unfinished order
US33	As a Visitor I want to see every cash amount rounded to my chosen currency so that I can know estimated price for foreign currencies
US34	As a Customer I want to be redirected directly to checkout complete so that I can not select payment method if its unnecessary
US35	As a Customer I want to not see payment instructions if order total is zero so that I can not being wrongly informed about my order's payments
US36	As an Administrator I want user to place orders in channel's base currency so that I can be sure how much should I pay for my cart
US37	As an Administrator I want user to place orders in channel's base currency so that I can be sure how much should I pay for my cart
US38	As an Administrator I want user to place orders in channel's base currency so that I can be sure how much should I pay for my cart
US39	As a Customer I want to be able to see all remaining steps so that I can see all remaining checkout steps before confirmation
US40	As a Customer I want to be redirected directly to checkout summary page so that I can not select shipping and payment method if its unnecessary
US41	As a Customer I want to be prevented from finishing the payment step with no method selected so that I can be prevented from finishing payment step without selecting a method
US42	As a Customer I want to be prevented from selecting not available payment methods so that I can pay for my order properly

US43	As a Customer I want to be able prevent placing order with invalid order total In order inform the customer about any changes that affect order total
US44	As a Guest I want to see correct details about my order on changing the payment method page so that I can verify that I am changing the payment method of correct order
US45	As a Customer I want to have valid order so that I can have valid payment method
US46	As a Customer I want to be able to complete checkout process without paying so that I can pay with cash or by external means
US47	As a Customer I want to have them already sorted so that I can see the most suitable payment methods first
US48	As a Customer I want to be able to go back to addressing or shipping step from shipping step so that I can modify my order after addressing and selecting shipping method
US49	As a Customer I want to be able to pay with PayPal Express Checkout so that I can buy products
US50	As a Customer I want to be able to choose a payment method so that I can pay for my order in a convenient way for me
US51	As a Guest I want to be able to complete checkout process without paying so that I can pay with cash or by external means
US52	As a Customer I want to be able to access the order payment page from my account panel so that I can complete my unpaid orders
US53	As a Customer I want to be informed about payment instructions for chosen payment method so that I can know how to pay for my order
US54	As a Customer I want to be prevented from paying for the cancelled order so that I can not be able to buy order that is already cancelled
US55	As a Guest I want to be able to change the method after order confirmation so that I can try different payment methods
US56	As a Customer I want all prices to be updated after each step of the checkout process so that I can keep in sync with current exchange rate throughout checkout
US57	As a Customer I want my cart to be cleared after I place an order so that I can start a new order after purchase
US58	As a Customer I want to be able to leave order notes for administrators to see In for the administrators to know about my special requests
US59	As a Visitor I want to receive the order confirmation email so that I can receive proof that my order has been confirmed
US60	As an Administrator I want to have customer's IP address assigned to their orders so that I can know from which IP address a new order has been places
US61	As a Customer I want to be able to go back to addressing, shipping and payment step from order summary page so that I can modify my order after addressing, selecting shipping method and payment method
US62	As a Customer I want to access this order from the thank you page so that I can check my order right after I have placed it
US63	As a Customer I want new addresses provided during checkout to be saved in my address book so that I can ease my address management
US64	As a Visitor I want to be able to return to the addressing step and change previously typed data so that I can correct my address data
US65	As a Customer I want to be able to specify a province for selected country manually so that I can specify a province for a country without default provinces
US66	As a Guest I want to be able to sign in so that I can sign during the addressing step
US67	As an Customer I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when addressing an order
US68	As a Customer I want to be able to fill addressing details so that I can address an order
US69	As a customer I want to be able to sign in and fill addressing details so that I can address an order
US70	As a Customer I want to have my default address preselected on addressing step so that I can speed up checkout process
US71	As a Customer I want to be able to choose it from my address book so that I can quickly fill in my address information during checkout
US72	As a Guest I want to be able to fill addressing details so that I can address an order
US73	As a Customer I want to be able to fill addressing details choosing country with provinces so that I can address an order properly
US74	As a Customer I want to have only available country preselected so that I can not be forced to select country when only one country is available
US75	As a Customer I want to be able to go back to shop homepage from checkout addressing step so that I can return to shopping after checkout started
US76	As a Customer I want to have information about promotion changes on my order so that I can have valid promotions applied on my order
US77	As a Customer I want to be redirected directly to payment selection so that I can not select shipping method if its unnecessary
US78	As a Customer I want to be prevented from accessing checkout with empty cart so that I can proceed through the checkout correctly
US79	As an Visitor I want to be able to create an account in the store after placing an order so that I can make future purchases with ease
US80	As a Customer I want to be redirect directly to order summary page after shipping selection so that I can not select payment method when it is unnecessary
US81	As a Customer I want to be redirected directly to order summary page after shipping selection when my order total is zero In order not to select payment method when it is unnecessary
US82	As a Customer I want to be able to finish checkout for zones in different scopes so that I can deliver my purchase to different zone than my tax zone
US83	As a Customer I want to be able to see all remaining steps so that I can be sure that the shipping discount was applied to my order
US84	As a Customer I want to be able to modify these steps so that I can have possibility to change remaining steps
US85	As an Visitor I want to have account registration form prefilled after placing an order so that I can make future purchases with ease
US86	As a Customer I want to have enabled products in my order so that I can have valid products
US87	As a Customer I want to be redirected directly to payment selection so that I can not select shipping method if its unnecessary

US88	As a Visitor I want to have valid order so that I can have valid shipping method on my order
US89	As a Customer I want to be able to see shipping fee so that I can be aware of shipping fee applied for my shipment
US90	As a Customer I want to be able to go back to addressing step from shipping step so that I can readdress already addressed order
US91	As a Customer I want to not be able to choose not available shipping methods so that I can ship my order properly
US92	As a Customer I want to be prevent from finish shipping step without selected method so that I can be prevent from finish shipping step without selected method
US93	As a Customer I want to be able to choose only shipping methods that match shipping category of all my items so that I can select correct shipping method for my order
US94	As a Customer I want to be able to choose a shipping method so that I can ship my order properly
US95	As a Customer I want to be able to see shipping fee in channel's base currency so that I can be aware of shipping fee applied for my shipment
US96	As a Customer I want to be able to choose shipping method which category is not same as any category of all my units so that I can select correct shipping method for my order
US97	As a Customer I want to be able to choose shipping method which based on my units categories so that I can select correct shipping method for my order
US98	As a Customer I want to have them already sorted so that I can see the most suitable shipping methods first
US99	As a Customer I want to be able to choose only shipping methods that match shipping category of all my items so that I can select correct shipping method for my order
US100	As a Customer I want to be able to finish checkout process so that I can get back to the right checkout step after leaving
US101	As a Customer I want to be able to see shipping total on order summary page In order be certain about shipping total
US102	As a Customer I want to be able to see shipping promotion on the order summary so that I can be sure that the shipping discount was applied to my order
US103	As a Customer I want to be able to proceed to the summary page with order containing a free product so that I can be able to buy free products
US104	As a Customer I want to be able to see addresses on the order summary page so that I can be certain about shipping and billing address
US105	As a Customer I want to be able to see promotion total on the order summary page so that I can see what kind of discounts I received
US106	As a Customer I want to be able to see province on the order summary page so that I can be certain about province which I created manually
US107	As a Customer I want to see discounts of specific order item so that I can be aware of discounts applied to an order
US108	As a Customer I want to be able to see order locale on the order summary page In order be certain what is the order locale
US109	As a Customer I want to be able to see order items on the order summary page In order be certain about order content
US110	As a Customer I want to be able to see tax total on the order summary page so that I can be certain about tax total
US111	As a Customer I want to be able to see all details of chosen shipping method and payment method so that I can be certain about a shipping method and payment method
US112	As a store owner I want to be able to set theme per channel so that I can allow customizing channels' appearance
US113	As an Visitor I want to see a different user interface on each one so that I can easily distinguish store pages provided by Sylius plugin
US114	As an Visitor I want to see a different user interface on each one so that I can easily distinguish stores
US115	As an Administrator I want to be able to browse list of them so that I can have a overview of all defined channels
US116	As an Administrator I want to be able to select available currencies so that I can give possibility to pay in different currencies on my stores website
US117	As an Administrator I want to add a new channel to the registry so that I can sell through multiple websites or mobile applications
US118	As an Administrator I want to be able to select default tax zone so that I can give an opportunity of choosing different default taxes logic for different channels
US119	As an Administrator I want to be able to delete a channel so that I can remove test, obsolete or incorrect channels
US120	As an Administrator I want to add a new channel with shop billing data to the registry so that I can sell through multiple channels of distributions with specific billing data
US121	As an Administrator I want to be able to switch state of channel between enable and disable so that I can change channels which are available to my customers
US122	As an Administrator I want to be able to delete multiple channels at once so that I can remove test, obsolete or incorrect channels in an efficient way
US123	As an Administrator I want to be prevented from adding or editing it without specifying required fields so that I can avoid making mistakes when managing a channel
US124	As an Administrator I want to be prevented from deleting last available channel so that I can continuously provide access to store
US125	As an Administrator I want to be able to edit shop billing data on a channel so that I can have proper shop billing data on shop-related documents
US126	As an Administrator I want to be able to edit a channel so that I can change channel details
US127	As an Administrator I want to be prevented from adding two channels with same code so that I can uniquely identify channels
US128	As an Administrator I want to be able to select available locales so that I can present stores website in different languages
US129	As an Administrator I want to be prevented from adding disabled channel when no other exist so that I can allow users to access to my store
US130	As an Administrator I want to be able to select tax calculation strategy so that I can use different tax strategies on different channels
US131	As an Administrator I want to be able to switch state of channel between enable and disable so that I can change channels which are available to my customers
US132	As a Developer I want to change between channels with ease so that I can see how my store looks on a different channels

US133	As a Visitor I want to have promotion's discounts applied to my cart only if the given promotion is valid so that I can pay proper amount after using the promotion
US134	As a Customer I want to receive discount for my purchase so that I can pay less while buying goods from promoted taxons
US135	As a Visitor I want to have promotions applied in prioritized fashion so that I can pay proper amount while buying promoted goods
US136	As a Customer I want to receive discount for my purchase so that I can pay less while buying goods
US137	As a Customer I want to receive discount on each unit of promoted product so that I can buy specific product with a discount
US138	As a Customer I want to receive discount for my purchase so that I can pay proper amount while buying promoted goods
US139	As a Visitor I want to have promotions applied to my cart so that I can pay proper amount while buying promoted goods in different channels
US140	As a Visitor I want to have only available promotions applied to my cart so that I can place a valid order
US141	As a Customer I want to receive discount for my purchase so that I can pay less while buying goods from promoted taxons
US142	As a Customer I want to receive discount for my purchase so that I can pay less while buying goods from specific price range
US143	As a Customer I want to receive discount for my purchase so that I can pay proper amount while buying promoted goods
US144	As a Customer I want to have shipping promotion applied to my cart so that I can pay decreased amount for shipping
US145	As a Customer I want to receive discount for my purchase so that I can pay less while buying goods from specific price range
US146	As a Visitor I want to have promotions applied to my cart so that I can pay proper amount while buying promoted goods
US147	As a Customer I want to receive discount on each unit of promoted product so that I can buy specific product with a discount
US148	As a Customer I want to receive a discount for my purchase so that I can pay less for my order during a promotion period
US149	As a Customer I want to have proper discount applied after every operation on my cart so that I can receive proper discount for my order
US150	As a Visitor I want to receive a discount for my purchase so that I can pay less while placing an order
US151	As a Visitor I want to have promotion applied to my cart when my cart quantity qualifies so that I can pay decreased amount for my order during promotion
US152	As a Customer I want to receive a discount for my purchase so that I can pay less while an placing order
US153	As a Visitor I want to have a discount applied when I belong to a specific customer group so that I can apply discount only for selected customer group
US154	As a Customer I want to receive discount for my purchase so that I can pay less while buying goods with required total from promoted taxon
US155	As a Customer I want to receive discount for my purchase so that I can pay less while buying some goods
US156	As a Visitor I want to have promotion applied when my items total is qualified so that I can pay decreased amount for my order during promotion
US157	As a Customer I want to receive discount for my purchase so that I can pay less while buying goods from promoted taxons
US158	As a Visitor I want to have promotion coupon's discounts applied to my cart when it is a promotion that provides 100% discount so that I can pay proper amount after using the promotion coupon
US159	As a Customer I want to have promotion coupon's discounts applied to my cart only if the given promotion coupon is valid so that I can pay proper amount after using the promotion coupon
US160	As a Visitor I want to have promotion coupon's discounts applied to my cart so that I can pay proper amount after using the promotion coupon
US161	As a Visitor I want to have promotion coupon's discounts applied to my cart only if the given promotion coupon is valid so that I can pay proper amount after using the promotion coupon
US162	As a Customer I want to have promotion coupon's discounts applied to my cart only if the given promotion coupon is valid so that I can pay proper amount after using the promotion coupon
US163	As a Visitor I want to have promotion coupon's discounts applied to my cart only if the given promotion coupon is valid so that I can pay proper amount after using the promotion coupon
US164	As an Administrator I want to have a promotion usage increased or unchanged after order placement so that I can checking promotion usage
US165	As an Administrator I want to have a promotion coupon usage decreased after order cancellation so that I can exclude cancelled orders from promotion coupon usage
US166	As an Administrator I want to have a promotion coupon usage increased after order placement so that I can track usage of promotion coupon
US167	As an Administrator I want to have a promotion usage increased after order placement so that I can limit promotion usage
US168	As an Administrator I want to have a promotion usage decreased after order cancellation so that I can properly limit promotion usage
US169	As an Administrator I want to be prevented from generating it without specifying required fields so that I can avoid making mistakes when generating coupons for my promotions
US170	As an Administrator I want to be able to generate coupons for promotion so that I can quickly create specific number of coupons for my promotions
US171	As an Administrator I want to be able to delete multiple coupons at once from the registry so that I can remove test, obsolete or incorrect coupons in an efficient way
US172	As an Administrator I want to be able to delete a coupon from the registry so that I can remove test, obsolete or incorrect coupons
US173	As an Administrator I want to be able to generate promotion coupons with prefix and suffix so that I can quickly create specific number of tailored coupons for my promotions
US174	As an Administrator I want to be able to edit coupon so that I can change promotion coupon usage limits or expires date
US175	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a coupon
US176	As an Administrator I want to browse coupons of specific promotion so that I can see all promotion coupons
US177	As an Administrator I want to add a new coupon so that I can create coupons for my promotions
US178	As an Administrator I want to be prevented from adding two coupons with the same code so that I can uniquely identify coupons

US179	As an Administrator I want to be prevented from deleting a used coupon so that I can maintain proper payment history
US180	As a Customer I want to have previously applied promotion reverted so that I can get discount only on specific case
US181	As an Administrator I want to be able to delete a promotion from the registry so that I can remove test, obsolete or incorrect promotions
US182	As an Administrator I want to add a new promotion with rule configured in different channels to the registry so that I can give possibility to pay less for some goods based on specific configuration
US183	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a promotion
US184	As an Administrator I want to be prevented from deleting a promotion which has been applied to an order so that I can maintain proper order history
US185	As an Administrator I want to sort promotions by their priority so that I can change the order by which promotions are used
US186	As an Administrator I want to add a new promotion with rule to the registry so that I can give possibility to pay less for some goods based on specific configuration
US187	As an Administrator I want to add a new promotion with action to the registry so that I can give possibility to pay specifically less price for some goods
US188	As an Administrator I want to add a new promotion with filtered action to the registry so that I can give possibility to define which product should be affected by promotion
US189	As an Administrator I want to browse existing promotions so that I can see all promotion
US190	As an Administrator I want to be prevented from adding two promotions with the same code so that I can uniquely identify promotions
US191	As an Administrator I want to be able to edit a promotion so that I can change promotion details
US192	As an Administrator I want to add a new promotion so that I can sell more by creating discount incentives for customers
US193	As an Administrator I want to access the coupons management directly from the promotion show page so that I can facilitate work with the management of coupons
US194	As an Administrator I want to be able to delete multiple promotions at once from the registry so that I can remove test, obsolete or incorrect promotions in an efficient way
US195	As an Administrator I want to add a new promotion with action configured in different channels to the registry so that I can give possibility to pay specifically less price for some goods
US196	As an Administrator I want to be able to access an edit page of a promotion with a rule that contains a removed taxon so that I can change promotion details
US197	As an Administrator I want to be prevented from defining filters without specifying required fields so that I can avoid making mistakes when managing a promotion with filters
US198	As an Administrator I want to be able to edit a country so that I can enable or disable countries
US199	As an Administrator I want to be able to edit a country and its provinces so that I can add or remove provinces in existing countries
US200	As an Administrator I want to be prevented from adding two provinces with the same code so that I can uniquely identify provinces
US201	As an Administrator I want to add a new country with its provinces to the store so that I can sell my goods to different countries and provinces
US202	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a province
US203	As an Administrator I want to add a new country to the store so that I can sell my goods to different countries
US204	As an Administrator I want to be prevented from adding a new country with an existing code so that I can avoid making mistakes when managing countries
US205	As an Administrator I want to be able to delete multiple zones at once so that I can remove test, obsolete or incorrect zones in an efficient way
US206	As an Administrator I want to be able to edit a zone so that I can change my my tax and shipping configuration
US207	As an Administrator I want to add a new zone so that I can apply taxes and allow shipping to geographical areas
US208	As an Administrator I want to be prevented from adding two zones with same code so that I can uniquely identify zones
US209	As an Administrator I want to browse zones so that I can see all zones in the store
US210	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a zone
US211	As an Administrator I want to be able to delete a zone so that I can remove test, obsolete or incorrect zones
US212	As an Administrator I want to be able to see orders' total in it's currency so that I can know in which currency a customer will pay
US213	As a Administrator I want to be able to see province on the order summary page so that I can be certain about province which was created manually by customer
US214	As an Administrator I want orders which have been paid and shipped to be fulfilled so that I can know which orders don't need further monitoring
US215	As an Administrator I want to be able to see taxes in order so that I can know shipping and product taxes
US216	As an Administrator I want orders with no undelivered shipments to have shipping state shipped so that I can have coherent shipping states of all orders
US217	As an Administrator I want to be able to manage new orders made by customers so that I can know about new orders
US218	As an Administrator I want to be able to track changes on order's addresses so that I can be aware of order's addresses changes on order
US219	As an Administrator I want to be able to ship a shipment so that I can confirm shipping of an order
US220	As an Administrator I want to have a promotion still applied after cancelling order even if the promotion is no longer valid so that I can maintain correct order history
US221	As an Administrator I want to be able to delete an order with all it's details from the registry so that I can remove test, obsolete or incorrect orders
US222	As an Administrator I want orders with no unpaid payments to have payment state paid so that I can have coherent payment states of all orders

US223	As an Administrator I want to see an item price with all promotion discounts so that I can be aware of the amount of product promotions in an order
US224	As an Administrator I want to be able to finalize payment so that I can mark order's payment state as complete
US225	As an Administrator I want to see aggregated taxes of a specific order so that I can be aware of taxes in an order
US226	As an Administrator I want to be able to list items so that I can see ordered products
US227	As an Administrator I want to be able to see shipping states so that I can get to know the state of shipping
US228	As an Administrator I want to be able to filter orders on the list so that I can filter orders in specific period of time
US229	As an Administrator I want to have new shipments after my customer's checkout so that I can correctly process customer's shipments
US230	As an Administrator I want to see the taxes value of a specific order so that I can be aware of the amount of product taxes in an order
US231	As an Administrator I want to see the exact payment amount including all additional fees so that I can see how much a customer has to pay for his order
US232	As an Administrator I want to have order numbers generated sequentially so that I can have unique order numbers
US233	As an Administrator I want to have unpaid orders automatically cancelled so that I can get rid of completed orders after 5 days of being unpaid
US234	As an Administrator I want to see the discount value of a specific order so that I can be aware of the amount of discount applied to an order
US235	As an Administrator I want to have new payments after my customer's checkout so that I can correctly process customer's payments
US236	As an Administrator I want to see aggregated discount of a specific order so that I can be aware of discounts applied to an order
US237	As an Administrator I want to be able to see payment state as paid when order total was zero so that I can know that the payment is always paid if order total is zero
US238	As an Administrator I want to be able to view basic information about an order so that I can see details of a specific order
US239	As an Administrator I want to be able to modify a customer's shipping address without changing an order's total so that I can ship an order to a correct place
US240	As an Administrator I want to be able to modify a customer's shipping address without changing an order's total so that I can ship an order to a correct place
US241	As an Administrator I want to be prevented from removing required fields from customer's shipping address so that I can avoid making mistakes when modifying a customer's shipping address
US242	As an Administrator I want to be able to modify a customer's shipping address without changing the order's total so that I can ship an order to a correct place
US243	As an Administrator I want to be able to modify a customer's shipping address after an order has been placed so that I can ship an order to a correct place
US244	As an Administrator I want to be able to list items so that I can see ordered products with proper names
US245	As an Administrator I want to see detailed data about items in order so that I can be aware of the every important information about each order item
US246	As an Administrator I want to be prevented from removing required fields from customer's billing address so that I can avoid making mistakes when modifying a customer's billing address
US247	As an Administrator I want to be able to modify a customer's billing address without changing an order's total so that I can ship an order's bill to a correct place
US248	As an Administrator I want to be able to modify a customer's billing address after an order has been placed so that I can ship an order's bill to a correct place
US249	As an Administrator I want to be able to modify a customer's billing address without changing an order's total so that I can ship an order's bill to a correct place
US250	As an Administrator I want to be able to modify a customer's billing address without changing the order's total so that I can ship an order's bill to a correct place
US251	As an Administrator I want to be able to see orders' total in it's currency so that I can know how much, and in which currency a customer placed an order
US252	As an Administrator I want to be able to see shipping total so that I can get to know the cost of shipping
US253	As an Administrator I want to be able to filter orders by their total and currency so that I can filter orders with total in a specific range and currency
US254	As an Administrator I want to be able to cancel an order so that I can mark order state as cancelled
US255	As an Administrator I want to be able to view basic information about an order so that I can see details of a specific order which does not have to be shipped
US256	As an Administrator I want to be able to see shipping fees so that I can get to know the cost of shipping
US257	As an Administrator I want to be able to filter orders on the list so that I can filter orders by a specific channel
US258	As an Administrator I want to be able to sort orders by number so that I can faster find new orders
US259	As an Administrator I want to see included in price taxes value of a specific order so that I can be aware of the amount of product taxes in an order
US260	As an Administrator I want to see every price in order's currency so that I can be aware what currency a specific order has been placed in
US261	As an Administrator I want to be able to browse new orders so that I can process orders
US262	As an Administrator I want to be able to mark order's payment as refunded so that I can refund order's payment
US263	As a Customer I want to be able to send a message to the store's support so that I can receive help from the store's support
US264	As a Customer I want to be prevented from making mistakes in my contact request so that I can avoid making mistakes when requesting contact
US265	As a Developer I want to be informed about available plugins so that I can be aware of available plugins
US266	As a Visitor I want to have correct taxes applied to my order so that I can pay proper amount when buying goods without tax category
US267	As a Visitor I want to have correct taxes applied to my order so that I can pay proper amount when buying goods from the

	same tax category
US268	As a Visitor I want to have correct taxes applied to my order when it has a discount and tax rates are included in products prices so that I can pay proper amount when buying goods
US269	As a Customer I want to have correct taxes applied to my order so that I can pay proper amount when buying goods having different tax rates for different zones
US270	As a Visitor I want to have correct taxes applied to my order so that I can pay proper amount when buying goods with tax rate included in price
US271	As a Visitor I want to have correct taxes applied to my order with a discount so that I can pay proper amount when buying goods
US272	As a Visitor I want to have correct taxes applied to my order with a discount so that I can pay proper amount when buying goods
US273	As a Visitor I want to have correct taxes applied to my order when it has a discount so that I can pay proper amount when buying goods
US274	As a Visitor I want to have correct taxes applied to my order so that I can pay proper amount when buying goods from different tax categories
US275	As a Visitor I want to have correct taxes applied to my order with a discount and tax rates are included in products prices so that I can pay proper amount when buying goods
US276	As a Visitor I want to have correct taxes applied to my order with a discount and tax rates are included in products prices so that I can pay proper amount when buying goods
US277	As a Visitor I want to have correct taxes applied to my order so that I can avoid taxes amount inaccuracy
US278	As a Visitor I want to have correct taxes applied to my order so that I can buy goods with correct taxes applied
US279	As a Customer I want to have correct taxes applied to my order so that I can pay proper amount when buying goods
US280	As a Visitor I want to have correct taxes applied to my order so that I can pay proper amount when buying goods with variants from different tax categories
US281	As an Administrator I want to be able to delete multiple tax rates at once so that I can remove test, obsolete or incorrect tax rates in an efficient way
US282	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a tax rate
US283	As an Administrator I want to be able to delete a tax rate so that I can remove test, obsolete or incorrect tax rates
US284	As an Administrator I want to be prevented from adding two tax rates with same code so that I can uniquely identify tax rates
US285	As an Administrator I want to add a new tax rate to the registry so that I can apply different tax rates to various tax categories
US286	As an Administrator I want to be able to edit a tax rate so that I can change taxes applied to products
US287	As an Administrator I want to be prevented from adding it without specifying its code or name so that I can avoid making mistakes when managing a tax category
US288	As an Administrator I want to be able to delete a tax category so that I can remove test, obsolete or incorrect tax categories
US289	As an Administrator I want to be prevented from adding two tax categories with same code so that I can uniquely identify tax categories
US290	As an Administrator I want to add a new tax category to the registry so that I can apply different taxes to various products
US291	As an Administrator I want to be able to delete multiple tax categories at once so that I can remove test, obsolete or incorrect tax categories in an efficient way
US292	As an Administrator I want to be able to edit tax category so that I can change tax classification of my products
US293	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a product variant
US294	As an Administrator I want to see option values when editing product variant so that I can edit product variant
US295	As an Administrator I want to choose a channel in which I will be viewing it so that I can check a product in shop in all channels it is available in
US296	As an Administrator I want to be prevented from generating it without specifying required fields so that I can avoid making mistakes when generating variants
US297	As an Administrator I want to generate all possible product variants so that I can sell different variations of a single product
US298	As an Administrator I want to be able to delete multiple product variants at once from the product catalog so that I can remove test, obsolete or incorrect product variants in an efficient way
US299	As an Administrator I want to be prevented from deleting purchased product variant so that I can maintain proper order history
US300	As an Administrator I want to add a new product variant to the shop so that I can sell different variations of a single product
US301	As an Administrator I want to get product variants according to my specified phrase so that I can get hints when looking for named product variants
US302	As an Administrator I want to sort product variants from a product by their positions so that I can change the order by which product variants from a product are displayed
US303	As an Administrator I want to browse product variants of specific product so that I can see all product variants
US304	As an Administrator I want to toggle the inventory tracking so that I can have the inventory tracked in my shop
US305	As an Administrator I want to be prevented from generating of product variants from options without any values so that I can be prevented from invalid variant generation
US306	As an Administrator I want to be able to delete product variant from the product catalog so that I can remove test, obsolete or incorrect product variants
US307	As an Administrator I want to be able to edit a product option so that I can change which product option details
US308	As an Administrator I want to be able to delete multiple product options at once so that I can remove test, obsolete or incorrect product options in an efficient way
US309	As an Administrator I want to be able to edit a product option and its option values so that I can add or remove option values in existing product options
US310	As an Administrator I want to sort product options by their positions so that I can change the order by which product options are displayed
US311	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a product option

US312	As an Administrator I want to be prevented from adding two product options with the same code so that I can uniquely identify product options
US313	As an Administrator I want to be able to browse product options so that I can see all product options in the store
US314	As an Administrator I want to be able to add a new product option to the registry so that I can sell various options of the same product
US315	As an Administrator I want to sort product attributes by their positions so that I can change the order by which product attributes are displayed
US316	As an Administrator I want to add a new textarea product attribute so that I can show specific product's parameters to customer
US317	As an Administrator I want to be prevented from adding it without specify required fields so that I can avoid making mistakes when managing text product attributes
US318	As an Administrator I want to be able to delete multiple product attributes at once so that I can remove test, obsolete or incorrect product attributes in an efficient way
US319	As an Administrator I want to add a new text product attribute so that I can show specific product's parameters to customer
US320	As an Administrator I want to be able to edit a text product attribute so that I can change text product attributes applied to products
US321	As an Administrator I want to be able to delete a text product attribute so that I can remove test, obsolete or incorrect text product attribute
US322	As an Administrator I want to be able to edit a select product attribute so that I can change select product attributes applied to products
US323	As an Administrator I want to add a new checkbox product attribute so that I can show specific product's parameters to customer
US324	As an Administrator I want to add a new integer product attribute so that I can show specific product's parameters to customer
US325	As an Administrator I want to browse product attributes so that I can see all product attributes available in the store
US326	As an Administrator I want to add a new percent product attribute so that I can show specific product's parameters to customer
US327	As an Administrator I want to add a new select product attribute so that I can show specific product's parameters to customer
US328	As an Administrator I want to be prevented from adding it with wrong configuration so that I can avoid making mistakes when managing select product attributes
US329	As an Administrator I want to be prevented from adding a new product attribute with taken code so that I can uniquely identify product attributes
US330	As a Customer I want to be able to add product review so that I can share my opinion about product with other customers
US331	As a Visitor I want to be able to add product review so that I can share my opinion about product with other customers
US332	As a Visitor I want to be prevented from submitting invalid review so that I can avoid making mistakes while submitting a new product review
US333	As an Administrator I want to be able to move to edit page directly from product show page so that I can edit product in the simple way
US334	As an Administrator I want to choose a channel in which I will be viewing it so that I can check a product in shop in all channels it is available in
US335	As an Administrator I want to be able to view product show page so that I can view detailed product information
US336	As an Administrator I want to be able to view product show page so that I can view detailed product information
US337	As a Administrator I want to be able to see product's attributes in all locales so that I can see product's specification in all locales
US338	As an Administrator I want to have product's average rating recalculated after review rate change so that I can have my product's average rating properly calculated
US339	As an Administrator I want to be able to delete multiple product reviews at once so that I can remove test, obsolete or incorrect product reviews in an efficient way
US340	As an Administrator I want to be able to delete a product review so that I can remove test, obsolete or incorrect product reviews
US341	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a product review
US342	As an Administrator I want to edit a product review so that I can correct customer's opinion about my products
US343	As an Administrator I want to be able to reject a product review so that I can have control over the opinions about my products
US344	As an Administrator I want to browse product reviews so that I can know customer's opinion about my products
US345	As an Administrator I want to be able to accept a product review so that I can give the customers insight into others opinions about a product
US346	As a Customer I want to read all product reviews on product's reviews page so that I can know other customer's opinion about product
US347	As a Customer I want to read product reviews on product's details page so that I can know other customer's opinion about product
US348	As a Visitor I want to be able to view an image of a single product so that I can see images of a product
US349	As a Visitor I want to be able to view products from taxon and from taxon children so that I can browse products from general taxons
US350	As a Visitor I want to be able to see a proper price for each product variant so that I can see product variant price
US351	As a Customer I want to read product's average rating so that I can know product reviews summary
US352	As a Visitor I want to be able to see a proper price for each product variant so that I can see product variant price
US353	As a Visitor I want to be able to search products by name within a specific taxon so that I can browse products which names contains specific text
US354	As a Visitor I want to be able to view products from a specific taxon so that I can browse products that interest me most
US355	Background: Given the store operates on a single channel in "United States"
US356	As a Visitor I want to be able to see product's select attributes in a chosen locale so that I can see product's specification in different locales

US357	As a visitor I want to be able to have access to product page by permalink so that I can see products detailed page
US358	As a Visitor I want to be able to view a single product price so that I can see products price
US359	As a Visitor I want to be able to view a single product so that I can see products detailed information
US360	As a visitor I want to be able to see product's select attributes so that I can see product's specification
US361	As a Visitor I want to be able to view products from a specific taxon and a channel so that I can browse products that interest me most
US362	As a visitor I want to be able to see product's attributes so that I can see product's specification
US363	As a Visitor I want to see related products when viewing product details so that I can quickly navigate to other products associated with the one I'm currently viewing
US364	As a Customer I want to be able to sort products so that I can change the order in which products are displayed
US365	As a Visitor I want to be able to view a single product price on products list so that I can see the prices of listed products
US366	As a Visitor I want to be able to see product's attributes in a chosen locale and a channel so that I can see product's specification in different locales and channels
US367	As a Visitor I want to be able to view products from taxon children so that I can browse products from general taxons
US368	As an Administrator I want to sort products from a taxon by their positions so that I can change the order by which products from a taxon are displayed
US369	As a Customer I want to be able to sort products with multiple variants by the price of their first variant so that I can change the order in which products are displayed
US370	As a Customer I want to be aware of product's variants names so that I can differentiate product's variants by names
US371	As a visitor I want to be able to be informed that a product does not exists so that I can have a good navigation
US372	As a Visitor I want to be able to see product's attributes in chosen locale so that I can see product's specification in different than default locale
US373	As an Administrator I want to add a new product association type to the store so that I can connect products together in many contexts
US374	As an Administrator I want to be able to edit the product association type so that I can change information about a product association type
US375	As an Administrator I want to be able to filter product association types on the list so that I can filter product association types
US376	As an Administrator I want to be able to delete multiple product association types at once so that I can remove test, obsolete or incorrect product association types in an efficient way
US377	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a product association type
US378	As an Administrator I want to browse product association types so that I can see all product association types in the store
US379	As an Administrator I want to be able to delete a product association type so that I can remove test, obsolete or incorrect product association types
US380	As an Administrator I want to be prevented from adding two product association types with the same code so that I can uniquely identify product association types
US381	As an Administrator I want to be able to edit a product so that I can change product details
US382	As an Administrator I want to sort products from a taxon by their positions so that I can change the order by which products from a taxon are displayed
US383	As an Administrator I want to add a new product to the shop so that I can extend my merchandise
US384	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a product
US385	As an Administrator I want to add a new product to the shop with different prices for each channel so that I can define products for each channel
US386	As an Administrator I want to add a new product with text attribute to the shop so that I can extend my merchandise with more complex products
US387	As an Administrator I want to be able to delete multiple products at once from the product catalog so that I can remove test, obsolete or incorrect products in an efficient way
US388	As an Administrator I want to be able to edit product's attributes so that I can modify product details
US389	As an Administrator I want to be able to edit product's slug so that I can manage access path to product page
US390	As an Administrator I want to be able to remove images from an existing product so that I can get rid of obsolete images of my products
US391	As an Administrator I want to be able to delete products from the product catalog so that I can remove test, obsolete or incorrect products
US392	As an Administrator I want to add a new product with an integer attribute to the shop so that I can extend my merchandise with more complex products
US393	As an Administrator I want to add a new product to the shop so that I can extend my merchandise
US394	As an Administrator I want to be able to delete products from the product catalog so that I can remove test, obsolete or incorrect products
US395	As an Administrator I want to be able to change associations of an existing product so that I can change associations of my product
US396	As an Administrator I want to sort products so that I can change the order by which products are displayed
US397	As an Administrator I want to be able to add new images to a taxon so that I can change images of my product
US398	As an Administrator I want to be able to edit product's slug in multiple locales so that I can manage access path to product page in many languages
US399	As an Administrator I want to be able to add attributes to an existing product so that I can extend product details
US400	As an Administrator I want to add text attributes in different locales to a product so that I can extend my merchandise with more complex products
US401	As an Administrator I want to be able to browse products so that I can manage my shop merchandise
US402	As an Administrator I want to be able to browse products so that I can manage my shop merchandise
US403	As an Administrator I want to have product's price removed after corresponding channel deletion so that I can have product's

	prices specified only for existing channels
US404	As an Administrator I want to add a new product with associated products so that I can associate my product with others
US405	As an Administrator I want to add a new product with a percent attribute to the shop so that I can extend my merchandise with more complex products
US406	As an Administrator I want to be able to select taxons for a new product so that I can specify in which taxons a product is available
US407	As an Administrator I want to be able to access the variants management directly from the product edit page so that I can facilitate work with the management of variants
US408	As an Administrator I want to be able to select taxons for an existing product so that I can specify in which taxons a product is available
US409	As an Administrator I want to associate new products to an existing product so that I can associate my product with others
US410	As an Administrator I want to be able to change images of an existing product so that I can change images of my product
US411	As an Administrator I want to be prevented from deleting purchased products so that I can maintain proper order history
US412	As an Administrator I want to choose a channel in which I will be viewing it so that I can check a product in shop in all channels it is available in
US413	As an Administrator I want to toggle the inventory tracking so that I can have the inventory tracked in my shop
US414	As an Administrator I want to buy selected products so that I can be sure that products I buy were not bought by another customer
US415	As a Customer I want to be prevented from placing an order with products that are out of stock so that I can be sure that products I buy are available
US416	As a Visitor I want to be prevented from adding products which are not available in the inventory so that I can buy only available products
US417	As a Visitor I want to be prevented from adding items over the available amount so that I can buy only available items
US418	As a Customer I want to be notified that requested item quantity cannot be handled so that I can not be able to add more items than available
US419	As an Administrator I want to see inventory tab only on simple product edit page so that I can manage product inventory
US420	As an Administrator I want to be prevented from decreasing inventory below on hold's value so that I can be prevented from setting incorrect inventory amount
US421	As an Administrator I want to be able to browse tracked product variants so that I can see all tracked product variants
US422	As an Administrator I want to be able to filter tracked product variants on the list so that I can filter tracked product variants by code
US423	As an Administrator I want to be able to filter tracked product variants on the list so that I can filter tracked product variants by name
US424	As an Administrator I want to be able to see the on hand quantity of a selected product variant so that I can see how many variants of selected product are available on hand
US425	As an Administrator I want to be able to see the correct quantity of a specific product variant in stock after order cancellation so that I can be certain that reserved inventory is back to stock correctly
US426	As an Administrator I want to browse administrators so that I can see all administrators in the store
US427	As an Administrator I want to be able to delete other administrator account so that I can get rid of deprecated administrators
US428	As an Administrator I want to be able to edit the administrator so that I can change information about an administrator
US429	As an Administrator I want to add an avatar to an administrator account so that I can visually identify the account
US430	As an Administrator I want to add a administrator to the store so that I can create new administrator account
US431	As an Administrator I want to remove an administrator avatar of an account so that I can properly identify the account
US432	As an Administrator I want to be able to update an avatar of an existing administrator so that I can always have an actual photo
US433	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing administrators
US434	As an Administrator I want to be able to delete multiple administrator accounts at once so that I can get rid of deprecated administrators in an efficient way
US435	As an Administrator I want to be prevented from adding two administrators with the same email or name so that I can uniquely identify administrators
US436	As an Administrator I want to be prevented from adding a new customer group with an existing code so that I can avoid making mistakes when managing customer groups
US437	As an Administrator I want to be able to delete a customer group so that I can remove test, obsolete or incorrect customer groups
US438	As an Administrator I want to be able to edit the customer group so that I can change information about a customer group
US439	As an Administrator I want to be able to delete multiple customer groups at once so that I can remove test, obsolete or incorrect customer groups in an efficient way
US440	As an Administrator I want to browse customer groups so that I can see all customer groups in the store
US441	As an Administrator I want to add a new customer group to the store so that I can categorize my customers
US442	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing customer groups
US443	As an Administrator I want to be able to see specific customer's page with provinces in the addresses so that I can see a country's province defined by customer on their details page
US444	As an Administrator I want to be prevented from adding two customers with the same email so that I can uniquely identify customer
US445	As an Administrator I want to add a customer to the store so that I can track information about my customers
US446	As an Administrator I want to add a customer with an user account so that I can allow my customers to sign in
US447	As an Administrator I want to be able to see orders statistics of customer on their details page so that I can know how many orders a customer has placed, and what's their total value
US448	As an Administrator I want to be able to disable and enable their user account so that I can control whether the customer can

	login or not
US449	As an Administrator I want to be able to edit the customer's email so that I can be able to help a customer when he wants to change his email
US450	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing customers
US451	As an Administrator I want not see create account option if the customer account already exists so that I can correctly administrate customers
US452	As an Administrator I want to browse all orders of a customer so that I can see all orders of a specific customer
US453	As an Administrator I want to be able to show specific customer's page so that I can see customer's details in the store
US454	As an Administrator I want to be able to edit the customer so that I can change information about a customer
US455	As an Administrator I want to be able to add a customer with an user account after failed creation action so that I can be able to create customer with account even if the creation failed
US456	As an Administrator I want to be able to reset customer's password so that I can help customers to retrieve access to theirs accounts
US457	As an Administrator I want to browse customers so that I can see all customers in the store
US458	As an Administrator I want to be able to remove customer account details from the system so that I can delete customer account on demand
US459	As an Administrator I want to be able to change shop user password so that I can modify shop user credentials
US460	As a Visitor I want to be able to view a latest product list so that I can be up-to-date with the newest products
US461	As an Administrator I want to browse exchange rates so that I can manage exchange rates used in the store
US462	As an Administrator I want to add a new exchange rate to the store so that I can specify exchange rates between different currencies in my store
US463	As an Administrator I want to be able to delete an exchange rate so that I can remove obsolete exchange rates
US464	As an Administrator I want to be able to filter exchange rates in the store so that I can filter exchange rates by a specific currency
US465	As an Administrator I want to be able to delete multiple exchange rates at once so that I can remove obsolete exchange rates in an efficient way
US466	As an Administrator I want to be prevented from adding an exchange rate with same source-target currency pair so that I can have only unique exchange rates
US467	As an Administrator I want to be able to edit an exchange rate so that I can modify an exchange rate's configuration
US468	As an Administrator I want to be prevented from adding exchange rates without specifying required fields so that I can avoid making mistakes when managing exchange rates
US469	As an Administrator I want to be prevented from adding two currency with the same code so that I can uniquely identify currency
US470	As an Administrator I want to be able to edit a currency so that I can change currency configuration
US471	As an Administrator I want to browse currencies so that I can see all currencies in the store
US472	As an Administrator I want to add a new currency to the store so that I can sell products in different currencies
US473	As a Customer I want to browse channels with a valid currency only so that I can see right prices
US474	As a Customer I want to be able to switch currencies so that I can see prices in my preferred currency
US475	As an Administrator I want to browse all shipments in the system so that I can manage all shipments regardlessly of orders
US476	As an Administrator I want to be able to ship a shipment from shipments list so that I can manage all shipments status
US477	As an Administrator I want to browse all shipments with a chosen state so that I can filter shipments by state
US478	As an Administrator I want to be able to access order's page from shipments index so that I can make shipments and orders management more fluent
US479	As an Administrator I want to browse all shipments in the system so that I can manage all shipments regardlessly of orders
US480	As an Administrator I want to be able to filter shipments from a specific channel on the list so that I can browse only relevant shipments
US481	As a Customer I want to have correct shipping fees and taxes applied to my order so that I can pay proper amount for shipping
US482	As a Customer I want to have correct shipping fees and all taxes applied to my order so that I can pay proper amount for shipping and product taxes
US483	As a Customer I want to have shipping fee applied based on chosen shipping method so that I can decide on amount paid for shipment
US485	As an Administrator I want to sort shipping methods by their positions so that I can change the order by which shipping methods are displayed
US486	As an Administrator I want to sort shipping methods so that I can change the order by which shipping methods are displayed
US487	As an Administrator I want to be able to edit a shipping method so that I can change shipping method details
US488	As an Administrator I want to archive shipping methods so that I can hide no longer available shipping methods from the list and customers' use
US489	As an Administrator I want to be prevented from deleting used shipping method so that I can maintain proper order history
US490	As an Administrator I want to be able to delete a shipping method so that I can remove test, obsolete or incorrect shipping methods
US491	As an Administrator I want to be able to delete multiple shipping methods at once so that I can remove test, obsolete or incorrect shipping methods in an efficient way
US492	As an Administrator I want to be able to switch state of shipping method between enable and disable so that I can change shipping methods which are available to my customers
US493	As an Administrator I want to add a new shipping method to the registry so that I can deliver goods in different ways
US494	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a shipping method
US496	As an Administrator I want to be able to browse list of them so that I can have a overview of all defined shipping methods
US497	As an Administrator I want to be prevented from adding two shipping methods with same code so that I can uniquely identify

	shipping methods
US498	As an Administrator I want to be prevented from adding it with invalid code so that I can avoid making mistakes when managing a shipping method
US499	As an Administrator I want to be able to delete a shipping category so that I can remove test, obsolete or incorrect shipping categories
US500	As an Administrator I want to add a new shipping category to the registry so that I can deliver goods according to their shipping categories
US501	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a shipping category
US502	As an Administrator I want to be able to delete multiple shipping categories at once so that I can remove test, obsolete or incorrect shipping categories in an efficient way
US503	As an Administrator I want to be able to edit a shipping category so that I can change shipping category details
US504	As an Administrator I want to be prevented from adding two shipping categories with same code so that I can uniquely identify shipping categories
US505	As an Administrator I want to be able to browse list of them so that I can have a overview of all defined shipping categories
US506	As a Visitor I want to be prevented from creating an account without required fields so that I can avoid making mistakes when registering account
US507	As a Visitor I need to be able to reset my password so that I can login to my account when I forgot my password
US508	As a User I want to be able to verify my email address so that I can improve security of my account
US509	As a Customer I want to be able to change my password so that I can enhance the security of my account
US510	As a Customer I want to be able to edit my name and email so that I can manage my personal information
US511	As a Customer I want to be redirected to account panel dashboard when accessing the login page so that I can be aware that I am already logged in
US512	As a Customer I want to be able to subscribe to the newsletter so that I can be up-to-date with products and promotions
US513	As a Customer I want to be prevented from entering incorrect values so that I can avoid making mistakes when changing my personal information
US514	As a Customer I want to see the payment amount of my order with all fees so that I can know how much I have to pay for my order
US515	As a Customer I want to be able to see province in the order history so that I can be certain about province which I created manually
US516	As a Customer I want to be able to track my placed orders so that I can follow my orders
US517	As an Customer I want to be able to view list of my orders so that I can browse orders placed before registration
US518	As an Customer I want to be able to view items with proper names of my placed order so that I can check some details of my placed order
US519	As an Customer I want to be able to view details of my placed order so that I can check some details of my placed order
US520	As a Customer I want to be able to track my placed orders from current channel so that I can follow my orders
US521	As a Customer I want to be able to view details of my shipment status so that I can be aware whether my order has already been shipped
US522	As a Customer I want to be able to track only my placed orders so that I can follow my orders
US523	As a Customer I want to be able to change a payment method of an order so that I can buy products
US524	As an Customer I want to be charged in a base currency of channel so that I can pay exact amount of money
US525	As a Customer I want to see payment's status of my order so that I can know whether an order has already been paid, or not
US526	As an Customer I want to be able to view details of my placed order so that I can check some details of my placed order
US527	As a Customer I want to be prevented from entering incorrect password so that I can avoid making mistakes when changing my password
US528	As a Customer I want to be able to remove an address so that I can have only relevant addresses in my address book
US529	As a Customer I want the province which I chose to still be on the form after I make a mistake in the form so that I can not specify the province each time I make a mistake
US530	As a Customer I want to be able to browse my address book so that I can see all my addresses
US531	As a Customer I want to be able to edit my addresses so that I can keep my address information up to date
US532	As a Customer I want to be able to browse my address book so that I can see all my addresses
US533	As a Customer I want to be able to add a new address to address book so that I can have saved addresses on my account
US534	As a Customer I want to be prevented from adding invalid addresses so that I can be sure that the address I'm trying to add is correct
US535	As a Customer I want to mark an address as default so that I can have one of my addresses set as default
US536	As a Customer I want to be able to remove an address so that I can have only relevant addresses in my address book
US537	As a Visitor I want to be able to register again with the same e-mail so that I can set up a new account after I deleted it from the system
US538	As a Visitor I need to be prevented from making mistakes in my address email so that I can avoid making mistakes when re-setting password
US539	As a Visitor I want to be prevented from signing in with bad credentials so that I can avoid making mistakes when signing in to the store
US540	As a Visitor I need to be able to create an account in the store so that I can make future purchases with ease
US541	As a Visitor I need to be able to create an account with possibility to subscribe to the newsletter so that I can be up-to-date with products and promotions
US542	As a Visitor I want to be able to log in to the store so that I can view my orders
US543	As an Administrator I want to browse all payments in the system so that I can manage all payments regardlessly of orders
US544	As an Administrator I want to filter payments by state so that I can browse all payments with a chosen state
US545	As an Administrator I want to be able to filter payments from a specific channel on the list so that I can browse only relevant payments

US546	As an Administrator I want to be able to access order's page from payments index so that I can make payment and orders management more fluent
US547	As an Administrator I want to be able to complete a payment from payments list so that I can easily manage all payments statuses
US548	As an Administrator I want to be able to delete multiple payment methods at once so that I can remove test, obsolete or incorrect payment methods in an efficient way
US549	As an Administrator I want to be able to edit payment method so that I can change which payment methods are available in my store
US550	As an Administrator I want to be able to delete a payment method so that I can remove test, obsolete or incorrect payment methods
US551	As an Administrator I want to sort the payment methods so that I can change the order by which payment methods are displayed
US552	As an Administrator I want to be able to browse payment methods so that I can see all payment methods in the store
US553	As an Administrator I want to be prevented from adding two payment methods with the same code so that I can uniquely identify payment methods
US554	As an Administrator I want to be prevented from adding it without specifying required fields so that I can avoid making mistakes when managing a payment method
US555	As an Administrator I want to sort payment methods by their positions so that I can change the order by which payment methods are displayed
US556	As an Administrator I want to be prevented from deleting used payment method so that I can maintain proper order history
US557	As an Administrator I want to add a new payment method to the registry so that I can pay for orders in different ways
US558	As a Developer I want to run a command that loads sample data so that I can have sample data in Sylius
US559	As a Developer I want to run an installation command so that I can install Sylius via CLI
US560	As an Administrator I want to browse taxons so that I can see all taxons in the store
US561	As an Administrator I want to be able to edit a taxon so that I can change categorization of my merchandise
US562	As an Administrator I want to add a new taxon to the registry so that I can categorize my merchandise
US563	As an Administrator I want to get taxons according to my specified phrase so that I can get hints when looking for taxons
US564	As an Administrator I want to be able to delete a taxon so that I can remove test, obsolete or incorrect taxons
US565	As an Administrator I want to be able to edit taxon's slug so that I can manage access path to taxon page
US566	As an Administrator I want to browse taxons so that I can see all taxons in the store
US567	As an Administrator I want to be able to remove images from an existing taxon so that I can remove images of my categories
US568	As an Administrator I want to be able to changing images of an existing taxon so that I can change images of my categories
US569	As an Administrator I want to be able to edit taxon's slug in multiple locales so that I can manage access path to taxon page in many languages
US570	As an Administrator I want to be able to add new images to a taxon so that I can change images of my categories
US571	As an Administrator I want to browse ordered taxons so that I can see all ordered taxons in the store
US572	As an Administrator I want to be prevented from adding two taxons with same code so that I can uniquely identify taxons
US573	As an Administrator I want to be prevented from adding any without specifying required fields so that I can avoid making mistakes when managing taxons
US574	As an Administrator I want to add a new taxon for specific parent so that I can categorize my merchandise
US575	As an Administrator I want to add a new taxon to the registry so that I can categorize my merchandise
US484	As an Administrator I want to be prevented from adding it without specifying amount so that I can avoid making mistakes in a flat rate per unit calculator of shipping method
US495	As an Administrator I want to be prevented from adding it without specifying amount so that I can avoid making mistakes in a flat rate per shipment calculator of shipping method

Apéndice II: Proyecto Yupichay

La herramienta Yupichay se encuentra alojada en Gitlab

<https://gitlab.com/fernandojavierguerra/yupichay>

A) Libreria necesaria:

LELParser

1. Descargar proyecto LELParser y compilar

```
git clone git@gitlab.com:fernandojavierguerra/lel_agile.git
```

```
cd lel_agile/LELParser
```

```
mvn package
```

2. Instalar .jar en repositorio local

desde el directorio del proyecto LELParser:

```
mvn install:install-file -Dfile=target/LELParser-0.0.1-SNAPSHOT.jar -
```

```
DpomFile=pom.xml
```

B) Ejecutar yupichay

```
git clone git@gitlab.com:fernandojavierguerra/yupichay.git
```

```
cd yupichay
```

```
mvn package
```

```
java -jar target/yupichay-0.0.1-SNAPSHOT.jar
```

Acceso aplicacion web yupichay: <http://localhost:8080/>

Acceso yupichay DB: <http://localhost:8080/console>

C) Carga Datos

1. Acceder a BD yupichay a la URL <http://localhost:8080/console>

2. Cambiar JDBC URL por jdbc:h2:mem:testdb

3. From H2 console Execute: RUNSCRIPT FROM 'bd/datos.sql';

14 Bibliografía

- A Guide to Scrum and CMMI ®: Improving Agile Performance with CMMI*. (2016).
- Antonelli, L., Urbietta, M., Tonelli Juan, & Rossi, G. (2017). *Improving Requirements Management in Agile Development using the Domain Language*.
- Armour, Phillip. (2000). The Five Orders of Ignorance. *The Five Orders of Ignorance*, 17. <https://doi.org/10.1145/352183.352194>
- Aurum, A., & Wohlin, C. (Eds.). (2005). *Engineering and managing software requirements*. Springer.
- Badreddin, O., Sturm, A., & Lethbridge, T. C. (2014). Requirement traceability: A model-based approach. *2014 IEEE 4th International Model-Driven Requirements Engineering Workshop (MoDRE)*, 87-91. <https://doi.org/10.1109/MoDRE.2014.6890829>
- Bouillon, E., Mäder, P., & Philippow, I. (2013). A survey on usage scenarios for requirements traceability in practice. *International Working Conference on Requirements Engineering: Foundation for Software Quality*, 158-173.
- Brooks, F. (1986). *No Silver Bullet – Essence and Accident in Software Engineering*. 16.
- Carniel, C. A., & Pegoraro, R. A. (2018). Metamodel for Requirements Traceability and Impact Analysis on Agile Methods. En V. A. dos Santos, G. H. L. Pinto, & A. G. Serra Seca Neto (Eds.), *Agile Methods* (Vol. 802, pp. 105-117). Springer International Publishing. https://doi.org/10.1007/978-3-319-73673-0_9
- Cleland-Huang, J. (2012). Traceability in Agile Projects. En J. Cleland-Huang, O. Gotel, & A. Zisman (Eds.), *Software and Systems Traceability* (pp. 265-275). Springer London. https://doi.org/10.1007/978-1-4471-2239-5_12
- Cleland-Huang, J., Berenbach, B., Clark, S., Settими, R., & Romanova, E. (2007). Best practices for automated traceability. *Computer*, 40(6).

- Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley.
- Delater, A. (2013). *Tracing Requirements and Source Code During Software Development*. 198.
- Gotel, O. C., & Finkelstein, C. W. (1994). An analysis of the requirements traceability problem. *Requirements Engineering, 1994., Proceedings of the First International Conference on*, 94-101.
- Greenspan, S. J., & McGowan, C. L. (1978). Structuring software development for reliability. *Microelectronics Reliability*, 17(1), 75-83.
[https://doi.org/10.1016/0026-2714\(78\)91140-X](https://doi.org/10.1016/0026-2714(78)91140-X)
- H2 (DBMS)—*Wikipedia, la enciclopedia libre*. (s. f.). Recuperado 19 de mayo de 2020, de [https://es.wikipedia.org/wiki/H2_\(DBMS\)](https://es.wikipedia.org/wiki/H2_(DBMS))
- Heindl, M., & Biffl, S. (2005). *A case study on value-based requirements tracing*. 60.
<https://doi.org/10.1145/1081706.1081717>
- IEEE. (1998). *IEEE recommended practice for software requirements specifications: Approved 25 June 1998*. IEEE.
- IEEE Std. 610-1990—IEEE Standard Computer Dictionary*. (1991). IEEE.
- Jacobsson, M. (2009). *Implementing Traceability In Agile Software Development*. 55.
- Ji, W., Berger, T., Antkiewicz, M., & Czarnecki, K. (2015). Maintaining feature traceability with embedded annotations. *Proceedings of the 19th International Conference on Software Product Line - SPLC '15*, 61-70.
<https://doi.org/10.1145/2791060.2791107>
- Kannenbergh, A., & Saiedian, D. H. (2009). *Why Software Requirements Traceability Remains a Challenge*. 6.
- Kassab, M. (2015). The changing landscape of requirements engineering practices over the past decade. *2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*, 1-8.
<https://doi.org/10.1109/EmpiRE.2015.7431299>

- Kodali, M. (2015). *TRACEABILITY OF REQUIREMENTS IN SCRUM SOFTWARE DEVELOPMENT PROCESS*. 34.
- Kolla, M., & Banka, M. (2014). *Merging Functional Requirements with Test Cases*. 92.
- Letelier, P. (2002). *A Framework for Requirements Traceability in UML-based Projects*. 10.
- Moore, J. W. (Ed.). (1999). *Customer and terminology standards: This volume contains the customer interaction standards of the SESC collection ; for convenience, standards relating to terminology have also been included in this volume*. IEEE.
- Oliveros, A. (2014). *Requerimientos e Ingeniería de Requerimientos*.
- Pigoski, T. M. (2001). *CHAPTER 6 SOFTWARE MAINTENANCE*. 16.
- Pressman, R. S. (2010). *Ingeniería del software: Un enfoque práctico*. McGraw-Hill.
- Principios del Manifiesto Ágil*. (2001). <http://agilemanifesto.org/iso/es/principles.html>
- Project Management Institute & Project Management Institute. (2013). *Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK)*.
- R. Watkins, y M. Neal. (1994). *Why and How of Requirements Tracing*.
- Rahimi, M., & Cleland-Huang, J. (2018). Evolving software trace links between requirements and source code. *Empirical Software Engineering*, 23(4), 2198-2231. <https://doi.org/10.1007/s10664-017-9561-x>
- Ramesh, B., & Edwards, M. (1992). *Issues in the development of a requirements traceability model*. 256-259. <https://doi.org/10.1109/ISRE.1993.324849>
- Ramesh, B., & Jarke, M. (s. f.). *TOWARDS REFERENCE MODELS FOR REQUIREMENTS TRACEABILITY*. 54.
- Ratanotayanon, S., Sim, S. E., & Gallardo-Valencia, R. (2009). *Supporting Program Comprehension in Agile with Links to User Stories*. 26-32. <https://doi.org/10.1109/AGILE.2009.66>

- Schön, E.-M., Thomaschewski, J., & Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces*, 49, 79-91. <https://doi.org/10.1016/j.csi.2016.08.011>
- Spanoudakis, G., Zisman, A., Pérez-Miñana, E., & Krause, P. (2004). Rule-based generation of requirements traceability relations. *Journal of Systems and Software*, 72(2), 105-127. [https://doi.org/10.1016/S0164-1212\(03\)00242-5](https://doi.org/10.1016/S0164-1212(03)00242-5)
- Taromirad, M., & Paige, R. F. (2012). *Agile requirements traceability using domain-specific modelling languages*. 45-50. <https://doi.org/10.1145/2467307.2467316>
- The ISO/IEC & IEEE/EIA Standard 12207—Standard for Information Technology—Software Life Cycle Processes. (2011). En C. McClure, *Software Reuse*. IEEE. <https://doi.org/10.1109/9781118156681.ch2>
- Turk, D., France. Robert, & Rumpe, B. (2005). Assumptions Underlying Agile Software-Development Processes: *Journal of Database Management*, 16(4), 62-87. <https://doi.org/10.4018/jdm.2005100104>
- Urbieto, M., Antonelli, L., Rossi, G., & Leite, J. (2020). The impact of using a Domain Language for an Agile Requirement Management. *Information and Software Technology*, 27.
- Wake, B. (2003, agosto 17). INVEST in Good Stories, and SMART Tasks. *XP123 Exploring Extreme Programming*. <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>
- Wieggers, K. E., & Beatty, J. (2013). *Software requirements* (Third edition). Microsoft Press, s division of Microsoft Corporation.
- Wijesinghe, D. B., Kamalabalan, K., Uruththirakodeeswaran, T., Thiyaalingam, G., Perera, I., & Meedeniya, D. (2014). *Establishing traceability links among software artefacts*. 55-62. <https://doi.org/10.1109/ICTER.2014.7083879>
- Winkler, S., & von Pilgrim, J. (2010). A survey of traceability in requirements engineering and model-driven development. *Software & Systems Modeling*, 9(4), 529-565. <https://doi.org/10.1007/s10270-009-0145-0>