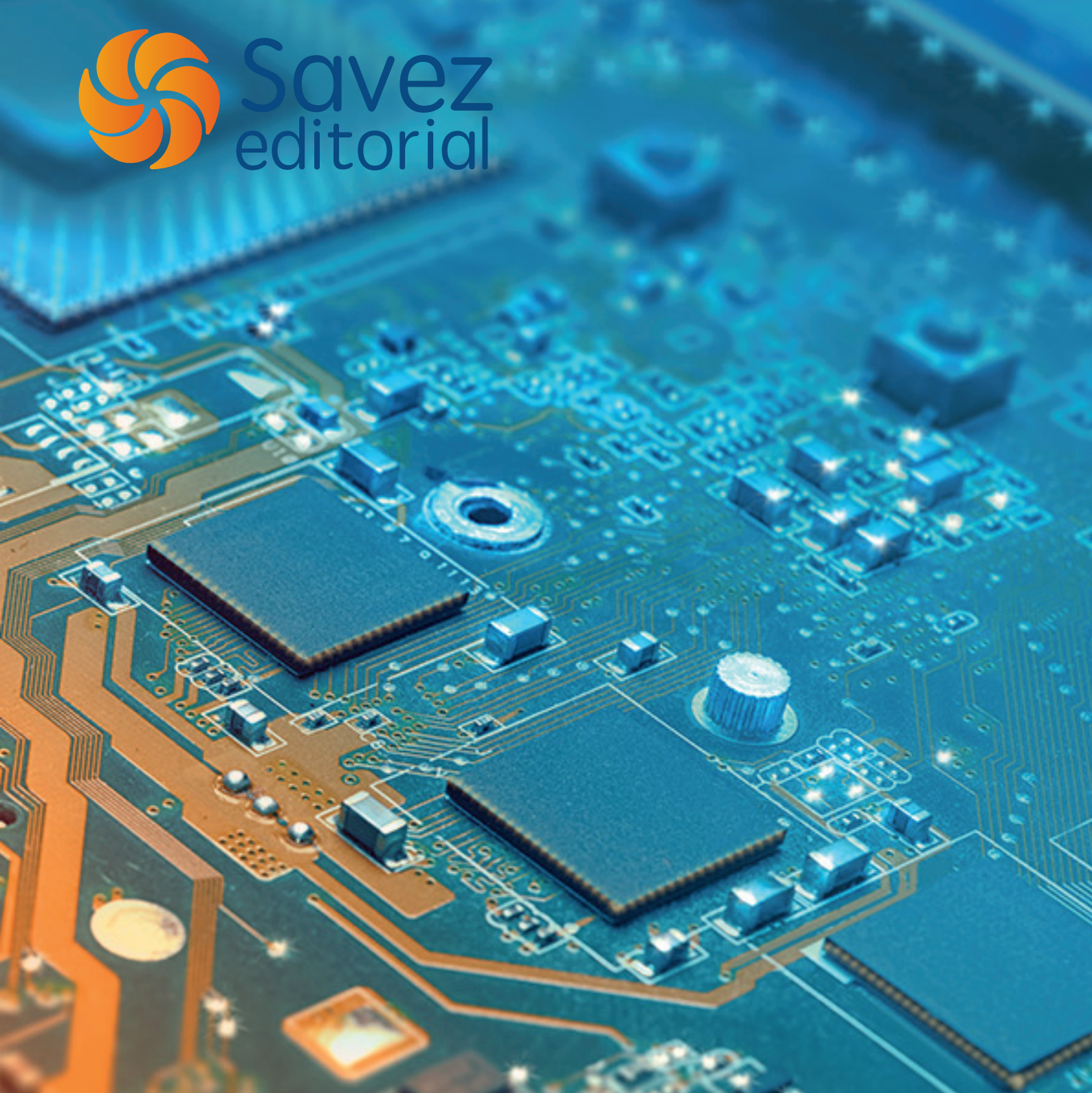




Savez
editorial



INGENIERÍA DE REQUERIMIENTOS un enfoque práctico

Julissa Elizabeth Reyna González
Freddy Ronald Huapaya Condori
Roberto Sixto Perales Flores
Denny John Fuentes Adrianzén



INGENIERÍA DE REQUERIMIENTOS

Un enfoque práctico

INGENIERÍA DE REQUERIMIENTOS

Un enfoque práctico

Julissa Elizabeth Reyna González
Freddy Ronald Huapaya Condori
Roberto Sixto Perales Flores
Denny John Fuentes Adrianzén



Julissa Elizabeth Reyna González
Freddy Ronald Huapaya Condori
Roberto Sixto Perales Flores
Denny John Fuentes Adrianzén

INGENIERÍA DE REQUERIMIENTOS. Un enfoque práctico
ISBN: 978-9942-8951-5-8

Savez editorial

Título: INGENIERÍA DE REQUERIMIENTOS. Un enfoque práctico

Primera Edición: July 2021

ISBN: 978-9942-8951-5-8

Obra revisada previamente por la modalidad doble par ciego, en caso de requerir información sobre el proceso comunicarse al correo electrónico

editor@savezeditorial.com

Queda prohibida la reproducción total o parcial de esta obra por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros), sin la previa autorización por escrito del titular de los derechos de autor, bajo las sanciones establecidas por la ley. El contenido de esta publicación puede ser reproducido citando la fuente.

El trabajo publicado expresa exclusivamente la opinión de los autores, de manera que no compromete el pensamiento ni la responsabilidad del Savez editorial

PRESENTACIÓN

Comunidad Universitaria, tengo a bien presentarles el libro titulado: "INGENIERÍA DE REQUERIMIENTOS. Un enfoque práctico", el cual tiene como finalidad dar a conocer los principios básicos a tener en cuenta en la Ingeniería de Requerimientos, el texto académico está dirigido a todos los profesionales de la ingeniería de sistemas, ingeniería de software, estudiantes de maestría y estudiantes universitarios que estén interesados en temas relacionados a Fundamentos de la Ingeniería de Requerimientos, Elicitación de Requerimientos, Análisis de Requerimientos, Especificación y Validación de requerimientos, distribuidos en IV capítulos, teniendo en cuenta contenidos teóricos y prácticos del curso. El presente libro constituye una herramienta útil e importante en el logro del proceso de enseñanza-aprendizaje virtual de hoy en día.

Cordialmente,

Mg. Julissa E. Reyna González
Ingeniera de Sistemas

PRÓLOGO

Con mucho agrado, después de 30 años de experiencia profesional en relación con el desarrollo de Sistemas de Información en sus distintas etapas, para la sistematización de instituciones públicas y empresas privadas, como parte de la optimización de los procesos a partir de los Sistemas de la Información, y hace 20 años en roles de Gestión Empresarial liderando equipos humanos algunos relacionados con Proyectos de Tecnologías de la Información y Comunicación como una herramienta parte de la Gerencia Estratégica y bajo el Enfoque Sistémico, y sobre todo, cuando estuve liderando una Escuela Académico Profesional de Ingeniería de Sistemas en pregrado que incluía cursos de análisis y diseño, y la especificación de requisitos, o en docencia en Escuelas de Posgrados en cursos relacionados con Ingeniería y Sistemas de Información, como Sistemas de Información Gerencial en la Universidad Nacional de Trujillo, en donde tuve que manejar competencias relacionadas con “requerimientos”, por lo cual, asumo me hubiera servido mucho este libro denominado: “Ingeniería de Requerimientos. Un enfoque práctico”, pues brinda los principios básicos y/o fundamentos acerca de los requerimientos, elicitación, análisis, especificación y validación de los mismos, a partir de un contenido teórico y ejemplos prácticos, dirigidos a los profesionales de Ingeniería de Sistemas, Ingeniería de Software, entre otros.

Los que hemos desarrollado Sistemas de Información, sabemos que, una limitación es la determinación de la problemática y sobre todo interpretar los requerimientos del cliente, entender sus necesidades de información sabiendo todo lo que el Sistema de Información puede brindarle, es allí donde el analista – diseñador debe comprender la problemática de la organización y sobre todo de los grupos de interés del Sistema de Información, durante el proceso de desarrollo, así como en su implementación y mantenimiento del mismo.

Por todo ello, y mucho más; es que después de haber revisado – en primicia, este bonito trabajo de investigación es que me place poder hablar de él, por lo práctico de su enfoque, facilitando el proceso para aquellos que desean salir airoso en el desarrollo de sistemas.

Tengo la certeza que, los autores, nos entregan una investigación, técnica y fácil de comprender, una herramienta que será de mucha utilidad.

Finalmente, quiero comentarles que ha sido un honor revisar este importante libro, lo cual agradezco, y recomiendo a la comunidad académica pueda beneficiarse con su utilización.

Atentamente;

Dr. Ing. Grover Eduardo Villanueva Sánchez – MBA.

DEDICATORIAS

A mi familia, por todo el amor, cariño y comprensión.

¡A mi hermano Henry, que nos dejó como consecuencia de la pandemia Covid19, a ti hermano por tus palabras de aliento, por tus ocurrencias, ahora estas en el cielo, gozando de vida eterna!

¡A la comunidad universitaria, adelante siempre adelante!

Con afecto, Julissa Reyna González.

A la memoria de mi padre Rigoberto Huapaya Mercado, símbolo eterno de trabajo, honradez y generosidad.

A mi madre Rosa por su constante apoyo incondicional y a mis hermanos por su apoyo moral.

Freddy Ronald Huapaya Condori

A mi esposa Carmela, a mis hijos Roberto y Milka, por su inmenso amor y apoyo incondicional.

Roberto Sixto Perales Flores

A mis padres Oscar y Rosa, mis hermanos Janet y Carlos Eduardo por su amor y paciencia y en especial a mi amado hijo Oscar Rodrigo porque eres lo más especial de mi vida.

Denny John Fuentes Adrianzén

ÍNDICE

PRESENTACIÓN.....	2
PRÓLOGO.....	3
INTRODUCCIÓN.....	10
INGENIERÍA DE REQUERIMIENTOS.....	12
Un enfoque práctico.....	12
CAPÍTULO I.....	13
FUNDAMENTOS DE LA INGENIERÍA DE REQUERIMIENTOS	13
Análisis de la Problemática.....	16
Casos de estudio.....	17
Ingeniería de Requerimientos (IR).....	18
Procesos de la Ingeniería de Requerimientos.....	21
Método Controlled Requirements Expression (CORE).....	22
Estudio de Factibilidad.....	26
Obtención y Análisis de requerimientos.....	28
Stakeholders o interesados.....	29
Identificación de Stakeholders.....	30
Lista de interesados.....	32
El Negocio detrás del sistema (NDS).....	32
Validación de Requerimientos.....	35
Principios de la Ingeniería de Requerimientos.....	36
Clasificación de Requerimientos (Dominios).....	37
Clasificación de Requerimientos.....	38
Dominio de la Ingeniería de Requerimientos.....	40
Ejercicios.....	41

Casos de estudio	42
CAPÍTULO II	49
ELICITACIÓN DE REQUERIMIENTOS.....	49
Metodología para la Elicitación de Requerimientos	49
Técnicas	52
La entrevista	52
Casos de estudio	55
Características de los Casos de uso	60
Diagramas de Casos de uso	62
Modelado del Sistema.....	63
Casos de estudio	66
Representación de Casos de Uso.....	75
Caso de estudio.....	78
Referencias	80
CAPÍTULO III.....	82
ANÁLISIS DE REQUERIMIENTOS	82
Casos de Uso.....	82
Características de los casos de uso	83
Relación entre actores y casos de uso.....	85
Proceso de análisis de requerimientos.....	85
Ejemplos.....	105
Caso de estudio.....	110
Referencias	121
CAPÍTULO IV	122
ESPECIFICACIÓN Y VALIDACIÓN DE REQUERIMIENTOS..	122
Validación de Requerimientos.....	122

Consideraciones para un documento de requerimientos	125
Consideraciones para la Validación de Requerimientos Funcionales.....	126
Chequeo de Requerimientos Funcionales.....	127
Consideraciones para la Validación de Requerimientos Funcionales.....	128
Referencias:	129
Apéndice	131

INTRODUCCIÓN

La presente obra titulada: Ingeniería de Requerimientos. Enfoque práctico, es el acierto a muchas noches de desvelo e investigación cuando nos propusimos escribir este libro. Los requerimientos para un sistema se basan en descripciones de lo que el sistema debe hacer, es decir atender las necesidades de los clientes. La Ingeniería de Requerimientos (IR), es el conjunto de fases que se debe tener en cuenta para comprender el problema del usuario de forma acertada y a partir de ello construir el sistema. La IR es un proceso basado en descubrir, analizar, documentar y verificar tanto los servicios como restricciones que tendrás en cuenta para el sistema.

Esta obra pretende dar a conocer a sus lectores los principios básicos fundamentados de la IR, está estructurado en cuatro capítulos claves para que comprendas desde un inicio el proceso de la IR, al final de cada capítulo se presentan una serie de ejemplos, ejercicios y casos de estudio que fortalecen los contenidos teóricos abordados.

A continuación, se describen los capítulos:

Capítulo I: FUNDAMENTOS DE LA INGENIERÍA DE REQUERIMIENTOS, en este capítulo se han abordado temas introductorios de la IR desde cómo comprender el problema, definiciones, procesos de la IR, el Negocio detrás del sistema, stakeholders, casos de estudio y ejercicios.

Capítulo II: ELICITACIÓN DE REQUERIMIENTOS, en este capítulo se aborda sobre la metodología, tareas y técnicas para realizar con éxito la elicitación de requerimientos y se describe el Documento de Requisitos del Sistema (DRS), basado en el estándar de la IEEE.

Capítulo III: ANÁLISIS DE REQUERIMIENTOS, en esta parte se aborda la etapa del proceso de ingeniería de requerimientos adquisición y análisis de requerimientos.

CAPÍTULO IV: ESPECIFICACIÓN Y VALIDACIÓN DE REQUERIMIENTOS, en este capítulo se aborda la especificación y validación de requisitos, casos de usos, actividades y ejemplos.

INGENIERÍA DE REQUERIMIENTOS

Un enfoque práctico

“La correcta obtención de los requisitos es uno de los aspectos más críticos de un proyecto software, independientemente del tipo de proyecto que se trate, dado que una mala captura de los mismos es la causa de la mayor parte de los problemas que surgen a lo largo del ciclo de vida”

(Johnson, 1995)

“El coste de un cambio en los requisitos, una vez entregado el producto, es entre 60 y 100 veces superior al coste que hubiera representado el mismo cambio durante las fases iniciales de desarrollo”

(Pressman, 2010)

“La parte más difícil al construir un sistema de software es decidir qué construir. Ninguna parte del trabajo invalida tanto al sistema resultante si ésta se hace mal. Nada es más difícil de corregir después.”

(Fred Brooks)

CAPÍTULO I

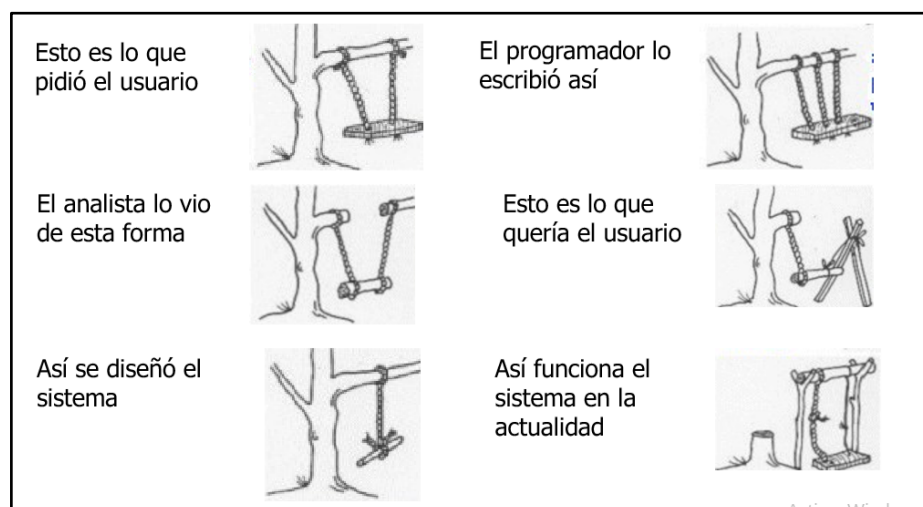
FUNDAMENTOS DE LA INGENIERÍA DE REQUERIMIENTOS

Uno de las dificultades que se presentan al determinar la problemática del sistema es comprender lo que requiere el cliente, es entender y comprender en un lenguaje natural lo que el usuario ha pedido que haga el sistema, es decir lo se espera de éste. Sin embargo, al momento de comprender la problemática por la cual atraviesa la empresa no se interpreta como debería ser, es decir una parte de los problemas que surten a lo largo del proceso de desarrollo se deben a la carencia de un proceso adecuado de definición y entendimiento del problema y a la definición poco clara de las necesidades del cliente.

Es así como visualizando la figura 1, analizamos la problemática descrita en la imagen.

Figura 1.

Problemática



Nota. En la figura se representa diversas situaciones respecto a lo que se desea construir, entender y analizar. Tomado de Universidad de Salamanca – Dpto. de Informática y Automática.

El proceso de indagación del problema, pareciera muy fácil, sin embargo, es allí donde surge el "problema", entender, comprender y analizar *¿cuál es el problema?*, es un aspecto clave que lo deberías considerar antes del desarrollo de software. Para ello Christel y Kang [Cri92] identificaron cierto número de problemas que se encuentran cuando ocurre la indagación:

Problemas de alcance. La frontera de los sistemas está mal definida o los clientes o usuarios finales especifican detalles

técnicos innecesarios que confunden, más que clarifican, los objetivos generales del sistema.

Problemas de entendimiento. Los clientes o usuarios no están completamente seguros de lo que se necesita, comprenden mal las capacidades y limitaciones de su ambiente de computación, no entienden todo el dominio del problema, tienen problemas para comunicar las necesidades al ingeniero de sistemas, omiten información que creen que es "obvia", especifican requerimientos que están en conflicto con las necesidades de otros clientes o usuarios, o solicitan requerimientos ambiguos o que no pueden someterse a prueba.

Problemas de volatilidad. Los requerimientos cambian con el tiempo. (Pressman, 2010, p. 103)

Recuerda que la meta es identificar el problema, proponer elementos de la solución, negociar distintos enfoques y especificar un conjunto preliminar de requerimientos de la solución en una atmósfera que favorezca el logro de la meta. (Pressman, 2010, p. 109)

Análisis de la Problemática

- a. ¿Qué entiendes por la problemática planteada en la figura?
- b. ¿El programador escribió lo que pidió el usuario?
¿Porqué?
- c. ¿El analista vio lo que el usuario quería? ¿Porqué?
- d. ¿Qué problemas se presentaron al concluir con el diseño del software?

En muchas ocasiones es importante entender lo que el usuario requiere del sistema, por lo observado en la figura 1, el resultado no fue el esperado. Es importante definir los requerimientos del sistema., aunque no es fácil, pero si lo haces te permitirá entender mejor lo que quieres hacer en el sistema.

Casos de estudio

Caso 1: Fundación Frederick Ebert (Extraído de la Universidad Politécnica de Nicaragua –UPOLI, Ingeniería de Software)

Actualmente la fundación Frederick Ebert Stiffurd (www.fesamericacentral.org) está pretendiendo automatizar sus actividades, a fin de manejar la gestión de fondos asignados a los diversos programas que ejecutan en Nicaragua. Lo prioritario es la agenda de eventos (lugar, fecha, cantidad de participantes, invitaciones, confirmaciones y llenado de datos, expositores, temas, documentos, materiales de apoyo, refrigerio o alimentos, entre otros recursos requeridos), posteriormente realizan informes y resúmenes de los temas, incorporan fotos y lo publican a fin de establecer comentarios al respecto y con ellos hacer encuestas y gráficos.

A continuación, te pido contestes las preguntas planteadas:

- a. ¿Cuál es la problemática planteada?
- b. ¿Qué requiere la fundación Frederick Ebert Stiffurd?
- c. ¿Cuál es el objetivo del negocio?

Ingeniería de Requerimientos (IR)

Aquel proceso que permite recopilar, analizar y verificar lo que requiere el cliente para un sistema de software es llamado *Ingeniería de Requerimientos (IR)*. El fin de la ingeniería de requerimientos es entregar una *especificación* de requerimientos de software correcta y completa. La IR permite comprender y mejorar las necesidades del cliente, de tal forma que podamos entender los sistemas de software complejos. He considerado diversos aportes referentes a la definición de requerimientos:

“La IR es un proceso sistemático mediante el cual se determinan los servicios que el software como producto debe suministrar y las restricciones sobre las cuales operará”.

Definición (1): La IR es, por tanto, una actividad esencialmente de interacción con los interesados en el sistema. Es incorrecto y extremadamente riesgoso que los Ingenieros de Software establezcan los requerimientos del sistema (a menos que la estrategia de la empresa sea forzar a los potenciales clientes a adecuarse al sistema tal cual está, como es el caso de los programas de uso masivo) sin haber mantenido numerosas reuniones con diferentes representantes del cliente, sin haberles mostrado prototipos del sistema, sin haberles hecho una y otra vez las mismas preguntas, sin haber comprendido el negocio del cliente. Pero, ¿qué es un requerimiento? Si bien no es vital dar una definición muy formal de este concepto, creemos que las siguientes pueden ayudar a comprender el tema. (Cristiá,2014)

Definición (2): Al proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se le llama IR. (Sommerville, 2011, p. 83)

Los ingenieros de software no originan los requerimientos; su función es convertir pedidos de los usuarios o clientes en requerimientos. Luego deben proveer un sistema que los implemente.

Es importante mencionarles, estimados lectores que no es lo mismo un pedido o deseo de un usuario o cliente que un *requerimiento*. No todos los pedidos o deseos de un usuario o cliente se convierten necesariamente en requerimientos, pero si todos los requerimientos se originan en un pedido o deseo de un usuario o cliente. Para que un pedido o deseo de un usuario o cliente se convierta en requerimiento, este debe ser documentado apropiadamente y el solicitante debe validarlo.

Los *requerimientos* siempre están en el entorno del sistema que se va a desarrollar, nunca dentro de él. Por consiguiente, palabras tales como tabla, XML, clase, subrutina, entre otros, no pueden formar parte de un requerimiento. (Cristiá, 2014)

Definición (3): De acuerdo al estándar IEEE 830 un *requerimiento* es una condición o capacidad que debe estar presente en un sistema o componentes de un sistema para satisfacer un contrato, estándar, especificación o cualquier otro documento formal. Una representación documentada de una condición o necesidad de un sistema.

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer, el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información. (Sommerville, 2011)

Procesos de la Ingeniería de Requerimientos

Es importante establecer el proceso a realizar en tal sentido en este capítulo abordaremos a autores como Pressman (2010), manifiesta que en el proceso de análisis de requerimientos del software se puede identificar cinco tareas o etapas fundamentales:

1. Reconocimiento del problema. Se deben de estudiar inicialmente las especificaciones del sistema y el plan del proyecto del software. Realmente se necesita llegar a comprender el software dentro del contexto del sistema. En esta etapa la función primordial del analista en todo momento es reconocer los elementos del problema tal y como los percibe el usuario.
2. Evaluación y síntesis. En esta etapa el analista debe centrarse en el flujo y estructura de la información, definir

las funciones del software, determinar los factores que afectan el desarrollo del sistema, establecer las características de la interfaz del sistema y descubrir las restricciones del diseño.

3. Modelización. Durante la evaluación y síntesis de la solución, se crean modelos del sistema que servirán al analista para comprender mejor el proceso funcional, operativo y de contenido de la información.

4. Especificación. La tarea asociada con la especificación intenta proporcionar una representación del software.

5. Revisión. Una vez que se han descrito la información básica, se especifican los criterios de validación que han de servir para demostrar que se ha llegado a un buen entendimiento de la forma de implementar con éxito el software.

Método Controlled Requirements Expression (CORE)

A su vez, el Método Controlled Requirements Expression conocido como CORE está basada en el principio de definir el problema a ser analizado (definición del

problema), y luego dividirlo en unidades o puntos de vista a considerar.

El método (CORE) es un conjunto de notaciones textuales y gráficas, con guías especificadas para la captura y validación de requerimientos del sistema, en las etapas iniciales del diseño del sistema. CORE. Está compuesto de siete etapas:

- a. Definición del problema. El propósito de la definición del problema es identificar los límites del mismo. Contiene detalles de los objetivos de la empresa de los usuarios del sistema, la base para la necesidad de un nuevo sistema, limitaciones de costo y tiempo, y quién va a ser el responsable de la revisión y aceptación de los resultados finales.
- b. Estructuración del punto de vista. El propósito de esta etapa es descomponer el ambiente del sistema en los elementos para que el sistema propuesto pueda ser analizado desde los puntos de vista de todas las entidades que se comunican con él, la más importante de las cuales son los usuarios. Durante esta etapa, todas las entidades que son fuentes potenciales de información deben ser identificadas.

- c. Colección tabular. Esta etapa es cuando la información sobre los flujos de datos entre los puntos de vista y el procesamiento de éstos son reunidos. Esto ayuda a establecer la totalidad y consistencia.
- d. Estructuración de datos. En la etapa previa, los elementos de información que pasan entre los puntos de vista son referidos por sus nombres generales. En esta etapa, se da una vista más cercana al contenido, a la estructura y a la derivación de datos, al producir diagramas de estructura de datos.
- e. Modelación individual de puntos de vista. Esta etapa puede dividirse en dos partes. Lo único concerniente a la primera es convertir las TCF'S en una notación diferente para producir los diagramas individuales del modelo de punto de vista. La segunda parte se refiere a agregar alguna información nueva perteneciente a flujos de datos internos, control de acciones y tiempo de acciones.
- f. Modelación combinada de punto de vista. Esta etapa facilita el análisis de una secuencia de eventos de más de un punto de vista. Cada diagrama de modelo combinado de punto de vista producido durante esta etapa es una representación del procesamiento de información que ocurre entre puntos de vista.

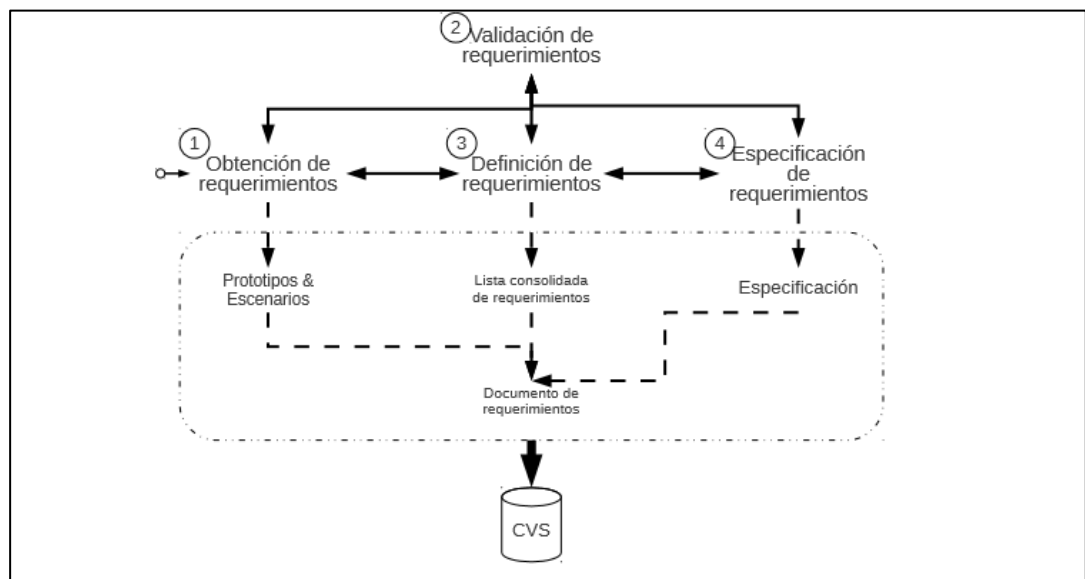
g. Análisis de restricciones. En esta etapa, se consideran restricciones adicionales tales como desempeño y seguridad. Éstas pueden afectar los diagramas de puntos de vista ya producidos. Las restricciones se documentan en una especificación de restricción del sistema.

Sin embargo, los métodos analizados, constituyen sólo un aporte teórico a la práctica, considero que los requerimientos se deben ajustar teniendo en cuenta el problema abordado, debe ser entendido y comprendido por el especialista (Ingeniero de Software).

El proceso de *Ingeniería de Requerimientos* establece actividades que se enfocan en valorar si el sistema es útil para la empresa (estudio de factibilidad), descubrir requerimientos (obtención y análisis), convertir dichos requerimientos en alguna forma estándar (especificación) y comprobar que los requerimientos definan realmente el sistema que quiere el cliente (validación). En tal sentido se presenta a continuación la figura los procesos y productos de la IR.

Figura 2.

Procesos y productos de la Ingeniería de Requerimientos



Nota. En la figura se aprecia el proceso de IR, los números indican la secuencia más común entre las etapas. Tomado de Cristiá, M., (2014).

Estudio de Factibilidad

Un estudio de factibilidad es un breve estudio enfocado que debe realizarse con oportunidad en el proceso de IR. Para todos los sistemas nuevos, el proceso de IR empieza con un estudio de factibilidad. (Sommerville, 2011, p.100)

Entrada: Descripción resumida del sistema y cómo se utilizará dentro de una organización.

Salida: Es un informe que recomienda si es conveniente llevar a cabo la IR y el proceso de desarrollo del sistema.

Un estudio de factibilidad es corto y orientado a resolver las interrogantes:

- ¿El sistema contribuye a los objetivos generales de la organización?
- ¿El sistema se puede implementar utilizando la tecnología actual y con las restricciones de
- costo y tiempo?
- ¿El sistema puede integrarse a otros que existen en la organización?

Cuando la información del estudio de factibilidad está terminada se prepara el *Informe del Estudio de Factibilidad* que debe contener:

- Recomendaciones de cuándo debe continuar el desarrollo del sistema.
- Propuesta de cambios en el alcance y el presupuesto.
- Calendarización del sistema.
- Sugerencias sobre requerimientos adicionales de alto nivel.

Obtención y Análisis de requerimientos

Etapa en la que los Ingenieros de Software trabajarán con los Stakeholders (involucrados o interesados), se entiende como:

- Clientes y los usuarios finales del sistema,
- Ingenieros que desarrollan o dan mantenimiento a otros sistemas,
- Administradores del negocio,
- Expertos en el dominio del sistema,
- Representantes de los trabajadores, entre otros.

Con la finalidad de descubrir el dominio de aplicación, qué servicios debe proporcionar el sistema, el desempeño requerido, las restricciones de hardware entre otros a tener en cuenta, con la finalidad de establecer los *requerimientos*.

En la **Obtención de Requerimientos** se determinan los requerimientos del sistema a través de la observación de sistemas existentes y del entorno donde se instalará el sistema, reuniones con los interesados, generación de prototipos, definición de escenarios, etc. Estas técnicas deben aplicarse una y otra vez hasta que finalice la etapa.

Es crucial que antes de comenzar esta etapa el cliente tenga claro lo que se denomina el negocio detrás del sistema (NDS). Si el negocio no está claro es muy probable que la construcción del sistema sea un fracaso más que nada para el cliente (lo que en muchos casos termina por afectar al desarrollador). Por lo tanto, es una responsabilidad de los Ingenieros de Software lograr que el cliente pueda definir con precisión el NDS.

Resulta igualmente importante que los ingenieros de requerimientos tengan un conocimiento del dominio de aplicación suficiente como para poder comprender el vocabulario del cliente, caso contrario el proceso de IR puede tornarse largo, costoso y el cliente puede perder confianza en la empresa desarrolladora. (Cristiá, 2014)

Stakeholders o interesados

Los *stakeholders* son todas aquellas personas, grupos y entidades que tienen intereses de cualquier tipo en una empresa y se ven afectados por sus actividades. Son interesados, directos o indirectos, en que la empresa funcione ya en caso contrario les afectaría directamente. Es así como Somerville (2011), sostiene que un stakeholder es aquella persona que está directa o indirectamente

relacionada con el sistema, y ésta puede ser parte de la organización, cliente o usuario final.

Stakeholders es un término en inglés que define a los interesados en un proyecto de desarrollo de software (esta es una particularización del concepto ya que es igualmente válido para cualquier tipo de proyecto).

¿Qué es un interesado? Son todas aquellas personas o departamentos que participando o no directamente en la realización de los trabajos se ven afectados por el resultado de los mismos.

La identificación de los stakeholders resulta por tanto esencial en un proyecto, ya que es necesario que de alguna u otra forma tengan participación en el mismo definiendo el rol que deben tener en la ejecución de los trabajos y determinando las expectativas que tienen respecto a los resultados del proyecto.

Identificación de Stakeholders

Para poder determinar quiénes son las personas interesadas en el proyecto, se debe conocer su influencia e interés en el mismo y dividirlos en:

Stakeholders directos. Quienes están directamente involucrados en el ciclo de vida del proceso, se verán afectados y tienen interés en la finalización exitosa del mismo.

– Stakeholders indirectos. Quienes tienen un nivel de interés o influencia bajo y muestran cierta preocupación por el proyecto.

Otra de las clasificaciones para la identificación de stakeholders es:

Stakeholders primarios. Son aquellas personas indispensables para el correcto funcionamiento de la organización y que tienen una relación económica directa con la empresa. Estos pueden ser sus socios, clientes y accionistas.

Stakeholders secundarios. Son los entes que no participan directamente de la compañía, pero también son afectados por sus resultados. En este círculo se encuentran los competidores, el mercado y las personas en general.

Lista de interesados

Se define como la lista de categorías de posibles candidatos a ser incluidos en la lista de interesados; en cada proyecto esta lista debería ser usada como *checklist*. Cada categoría debería particionarse en categorías más detalladas y de cada una de ellas se deberían listar tantos interesados como sea posible. Obviamente tal cantidad de interesados no puede tener la misma visión, necesidades y deseos sobre el sistema por lo que necesariamente se generan conflictos entre ellos mismos y con la empresa desarrolladora. Uno de los principales roles de los ingenieros de requerimientos, y de aquí la necesidad que sean personas con gran capacidad de comunicación, es la de resolver esos problemas. La única forma de resolver un conflicto es expresarlo con claridad y sin preconceptos y discutirlo entre TODOS los afectados hasta alcanzar un consenso. (Cristiá, 2014)

El Negocio detrás del sistema (NDS)

¿Por qué quiere desarrollar el sistema? Esta debe ser la primera pregunta que haga el ingeniero de requerimientos a quien paga por el sistema (que es uno de los interesados clave). Si la respuesta no está directa y claramente

relacionada *con maximizar las ganancias o minimizar los costos*, entonces el proceso de la IR debe detenerse inmediatamente. Si en la respuesta el cliente no incluye una cuantificación de sus expectativas respecto de los costos y beneficios que lo impulsan a construir el sistema, entonces será muy difícil determinar los requerimientos. Estos datos se llaman habitualmente el negocio detrás del sistema. (Cristiá, 2014)

El *negocio detrás del sistema* (NDS) es el criterio último y definitivo para determinar:

- Cada uno de los requerimientos.
- Si algo que un interesado pide un requerimiento o no. O dicho de otra forma si un requerimiento es correcto o no. Es decir, el negocio detrás del sistema brinda un criterio para determinar qué es un requerimiento y qué no lo es.
- La prioridad de cada requerimiento.

El *NDS* permite encontrar el camino hacia los interesados y el dominio de aplicación y, a partir de estos, los requerimientos y el criterio para priorizarlos. Claramente, el objetivo de la inversión indicara de forma directa o indirecta los sectores de la organización que se verán

afectados por el sistema, lo que permite determinar los interesados y el dominio de aplicación. Por ejemplo, si uno de los objetivos de la inversión dice algo como “pensamos en un nuevo sistema de ventas que nos permita incrementar las ventas en un 15% anual” entonces los interesados más directos serán la gerencia de ventas y los vendedores, y el dominio de aplicación los productos y/o servicios que se venden, las formas en que se venden, el plan de negocios de la empresa, los clientes que compran esos productos, etc.

Según la teoría de Jackson (1995), los requerimientos hablan sobre el dominio de aplicación y no sobre la máquina que eventualmente los implementara. Es crucial alcanzar una comprensión del dominio de aplicación en tanto el problema a resolver se define más en términos de la estructura y propiedades del dominio de aplicación, que en términos de la naturaleza de la máquina. En este sentido el dominio de aplicación es el material que le es dado al ingeniero y los requerimientos lo que el ingeniero debe hacer con ese material. (Cristiá, 2014)

Determinar el *dominio de aplicación* (DA) no siempre es sencillo. Parecería natural comenzar investigando los

requerimientos para el sistema mirando al viejo sistema (siempre hay un sistema anterior, sea manual o automático, a menos que la organización sea completamente nueva). Sin embargo, esto puede llevar a confusión. Por ejemplo, si se está desarrollando un nuevo sistema de sueldos, el ingeniero puede caer fácilmente en la idea de que el dominio de aplicación es la gerencia de recursos humanos o sueldos. Eso significaría que los requerimientos son sobre los documentos que pasan de empleado a empleado en esa sección. Si bien eso puede ser correcto, en ese caso el sistema no sería un sistema de sueldos sino más bien un sistema de procesamiento de documentos relacionados con el pago de sueldos. Es mucho más probable que si se trata de un sistema de sueldos los requerimientos sean sobre los empleados, sus escalas salariales, sus horarios de trabajo y las horas extras, sus ausencias, feriados, jubilaciones, seguros médicos, etc. Estas son las cosas que se deberían escribir en la *Lista consolidada de requerimientos*.

Validación de Requerimientos

Los requerimientos solo pueden ser validados por el cliente. Por lo tanto, una vez que se ha obtenido un

requerimiento (preguntando al cliente) se le debe preguntar si ese es el requerimiento que expresó. Esto no necesariamente es o da lugar a una tautología. El "cliente" no es una sola persona, como, ni lo que un ingeniero de software escribe es exactamente lo que el "cliente" dijo. Por lo tanto, la validación de requerimientos trata esencialmente de preguntar a un representante del cliente si lo que un ingeniero de requerimientos puso por escrito, a instancias de otro o el mismo representante del cliente, es correcto o no. (Cristiá, 2014).

Principios de la Ingeniería de Requerimientos

Es muy importante resaltar los principios que nunca deben olvidarse a la hora de desarrollar esta actividad, y en particular a la hora de concebir y aplicar técnicas o herramientas para ayudar en la tarea.

- El negocio detrás del sistema es el último y fundamental criterio para incluir, rechazar y priorizar cada requerimiento particular.
- Los requerimientos están en el dominio de aplicación y se los debe describir con ese vocabulario (Jackson, 1995)

- Nunca ir a una reunión con un cliente sin un prototipo. (Cristiá, 2014)

Clasificación de Requerimientos (Dominios)

Los requerimientos para un sistema se refieren a los servicios proporcionados por el sistema y sus restricciones operativas. Debido a la diversidad de los requerimientos, Sommerville (2011), sugiere organizarlos en tres dominios o niveles:

- Requerimientos del usuario
- Requerimientos del sistema
- Especificaciones del diseño de software

a. Requerimientos del usuario

Son declaraciones en lenguaje natural y en diagramas informales, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar. Acerca de qué servicios esperan los usuarios del sistema, y de las restricciones con las cuales éste debe operar.

b. Requerimientos del sistema

Establecen con detalle las funciones, servicios y restricciones operativas del sistema. El documento de requerimientos del sistema debe ser preciso. (Se incluye en el contrato)

Se utiliza para designar la descripción detallada de lo que el sistema debe hacer.

c. Especificaciones del diseño de software

Se obtienen de las especificaciones del sistema. Están definidas de manera formal en los documentos de Especificaciones de Requerimientos del Software.

Clasificación de Requerimientos

Según sus tipos se clasifican en:

a. Requerimientos Funcionales (RF).

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también

pueden declarar explícitamente lo que el sistema no debe hacer. Representación: Lenguaje natural, Modelos visuales, Métodos formales.

Los RF son enunciados de servicios que el sistema debe proporcionar tanto de cómo reaccionaría a sus entradas y cómo reaccionaría a situaciones específicas.

Es decir, los RF definen:

- ¿Cuáles entradas debe aceptar el sistema?
- ¿Cuáles salidas debe producir el sistema?
- ¿Qué datos debe almacenar el sistema que utilizarán otros sistemas?
- ¿Qué operaciones debe realizar el sistema?

b. Requerimientos No Funcionales (RNF).

Son restricciones de los servicios ofrecidos por el sistema. Sin embargo, es importante mencionarles que cuando se realiza un análisis a profundidad debemos tomar el sistema como un todo para determinar los RNF.

Los RNF surgen a través de necesidades del usuario, debido a restricciones presupuestales, políticas de la organización, necesidad de interoperabilidad con otro

software o sistemas de hardware, o factores externos como regulaciones de seguridad o legislación sobre privacidad. (Sommerville, 2011, p. 87)

c. Requerimientos de Dominio (RD)

Son aquellos que derivan del dominio de la aplicación del sistema. Pueden ser funcionales o no funcionales y derivan de las necesidades del sistema.

Dominio de la Ingeniería de Requerimientos

Algunas funcionalidades del dominio están relacionadas con: Identificar usuarios, Registrar la sesión de trabajo, ofrecer respuesta en cada interacción, ingresar, modificar o eliminar nuevo contenido, emitir reporte de usuarios, registrar estadísticas de uso, entre otros.

Es importante en el momento de definir los requerimientos del sistema, no separar los distintos niveles de descripción, como menciona Sommerville, (2011), en su texto se distinguen "requerimientos del usuario" para representar los requerimientos abstractos de alto nivel; y "requerimientos del sistema" para caracterizar la descripción detallada de lo que el sistema debe hacer. (p.83)

El documento de requerimientos del sistema (llamado en ocasiones especificación funcional) tiene que definir con exactitud lo que se implementará. Es importante definirlo en la parte del contrato entre el comprador del sistema y los desarrolladores del software. (Sommerville, 2011).

Ejercicios

Ejercicio 1. Considere una farmacia que se plantea los siguientes objetivos para su negocio:

Abrir dos sucursales en la ciudad.

Diferenciar la atención entre clientes con personería jurídica y clientes con persona natural

Agilizar la autorización de recetas con personería jurídica.

Suponga que la farmacia desea comprar o desarrollar un sistema informático que le ayude a cumplir los objetivos.

Entonces:

1. Escriba el negocio detrás del sistema alineado con los objetivos descritos.
2. Determine una lista preliminar de interesados.
3. Escriba la lista estructurada de requerimientos funcionales (no menos de 10 requerimientos).

Ejercicio 2. Considere un restaurante que se plantea los siguientes objetivos para su negocio:

Establecer una cadena de locales a nivel provincial.
Evitar diferencias entre comandas y facturas.
Mejorar control de stock y pedidos a proveedores.

Suponga que el restaurante desea comprar o desarrollar un sistema informático que le ayude a cumplir los objetivos.

Entonces:

1. Escriba el negocio detrás del sistema alineado con los objetivos descritos.
2. Determine una lista preliminar de interesados.
3. Escriba la lista estructurada de requerimientos funcionales (no menos de 10 requerimientos).

Casos de estudio

Caso 2: La compañía Nueva Inglaterra (Poveda, s/f)

La compañía de artículos para mantenimiento Nueva Inglaterra es un distribuidor al mayoreo y menudeo de productos de limpieza. Compra grandes cantidades de

artículos y los vende a sus clientes en pequeños lotes que van desde unos cuantos artículos hasta varias cajas, hecho que depende del tipo de artículo. La compañía inició sus operaciones hace veinte años, es rentable y se encuentra bien administrada.

El propietario de la compañía piensa desarrollar un sistema basado en computadora para administrar el inventario del almacén y estar al tanto de los artículos solicitados en cada pedido. Así mismo también desea desarrollar un sistema automatizado para procesar las ordenes de pedido de sus clientes sin importar donde se originen estas, ya sea que se den en el mostrador, por vía telefónica, correo o a través de los representantes de ventas de la compañía.

El dueño no tiene planes para ampliar las instalaciones de la compañía o el territorio de ventas, el cual abarca gran parte del área de Nueva Inglaterra. Sin embargo, sus planes actuales incluyen un aumento en el volumen de ventas con poco cambio en la línea de productos ofrecidos por la compañía.

El propietario lo contrata como analista de sistemas y acuerda reunirse con usted para discutir el sistema deseado. Usted recibe solo la información antes

mencionada y debe prepararse para su primera reunión con el dueño.

Lee detenidamente y contesta las preguntas:

- a. ¿Describa la problemática encontrada en la compañía Nueva Inglaterra?
- b. ¿Qué preguntas debe formular usted para averiguar más detalles relacionados con la compañía, sus clientes y los procedimientos actuales de inventario y procesamiento de órdenes?

Caso 3: Sistema de Administración de Inversiones

(Poveda, s/f)

Un analista de sistemas ha desarrollado un nuevo sistema para administrar las inversiones de cierta compañía en las bolsas de valores. Por regla general, la compañía tiene inversiones en bonos y acciones por 100 millones de dólares y da empleo a varios gerentes de inversión cuya única responsabilidad es administrar estos fondos. Los gerentes están autorizados para crear, vender y negociar acciones cuando lo juzguen necesario para aumentar el valor de la inversión o evitar pérdidas cuando cambien las condiciones del mercado.

Todos los gerentes de inversión de la compañía están suscritos a varios boletines informativos y servicios de la bolsa de valores que les proporcionan información sobre las tendencias actuales del mercado y seguridades específicas. Sin embargo, la mayor parte de la información que utilizan los gerentes para decidir cómo administrar las inversiones se obtiene por medio de contactos personales o de opiniones e investigaciones muy cuidadosas y detalladas. Aunque los gerentes reconocen que su trabajo los presiona mucho y los lleva a efectuar gran cantidad de cálculos aritméticos, les agrada bastante.

El nuevo sistema automatizado fue desarrollado para proporcionarles ayuda en sus actividades de inversión. Los analistas de sistemas y los corredores de bolsa creen que será difícil utilizar el nuevo sistema porque no se ajusta a sus patrones de análisis y pensamiento actuales. Por otra parte, el método utilizado por el nuevo sistema computarizado necesita una cuantificación de bonos y acciones específicos, hecho que se aleja de la forma normal de análisis basada en la experiencia e intuición del inversionista.

Lee atentamente y responde a las preguntas planteadas:

- a. ¿Qué factores se deben considerar al formular un conjunto de recomendaciones relacionadas con la posibilidad de implantar el nuevo sistema?
- b. ¿Qué recomendaciones formularía usted? ¿por qué?

Referencias:

Capability Maturity Model Integrated. (CMMI).
<http://www.sei.cmu.edu/cmmi/>.

Cristiá, M. (20014). *Introducción a la Ingeniería de Software*. Universidad Nacional de Rosario. Argentina.

Jackson, M. (1995). *Software requirements & specifications: a lexicon of practice, principles and prejudices*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.

Londoño, L., Anaya, R., & Tabares, M. (2008). Análisis de la ingeniería de requisitos orientada por aspectos según la industria del software. *Revista EIA*, (9), 43-52. Retrieved June 21, 2021, from http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1794-12372008000100004&lng=en&tlng=es.

Pressman, R. (2010). *Ingeniería del Software*. Ed. McGraw Hill. México. ISBN: 9786071503145.

Poveda, J. (s/f). *Ingeniería de Software*. Universidad Nacional de Ingeniería (UNI). <https://jmpovedar.wordpress.com/ingenieria-de-software-ii/>

Robertson, S. (2004). Requirements and the business case. *IEEE Softw.*, 21:93–95, September 2004.

Sommerville, I. (2011). *Ingeniería del Software*. Pearson 9a. Edición. Extraído del 21 de Junio del 2021 de: https://www.academia.edu/40713382/Libro_Somerville. ISBN: 978-607-32-0603-7

Singleton, J. (2007). *Stakeholder Identification and Management*. Lower Colorado River Authority. Disponible en: <http://nt1.adventuresports.com/canoe/whitewatercoursesandparks/2007presentations/StakeholderIdentificationManagementJeffSingleton.pdf>

CAPÍTULO II

ELICITACIÓN DE REQUERIMIENTOS

Metodología para la Elicitación de Requerimientos

El objetivo de esta metodología es la definición de las tareas a realizar, los productos a obtener y las técnicas a emplear durante la actividad de elicitación de requisitos de la fase de ingeniería de requisitos del desarrollo de software. En esta metodología se distinguen dos tipos de productos: los productos entregables y los productos no entregables o internos. Los productos entregables son aquellos que se entregan oficialmente al cliente como parte del desarrollo en fechas previamente acordadas, mientras que los no entregables son productos internos al desarrollo que no se entregan al cliente. El único producto entregable definido en esta metodología es el Documento de Requisitos del Sistema (DRS).

Las tareas recomendadas para obtener los productos descritos en esta metodología son las siguientes:

Tarea 1: Obtener información sobre el dominio del problema y el sistema actual.

Tarea 2: Preparar y realizar las reuniones de elicitación/negociación.

Tarea 3: Identificar/revisar los objetivos del sistema.

Tarea 4: Identificar/revisar los requisitos de información.

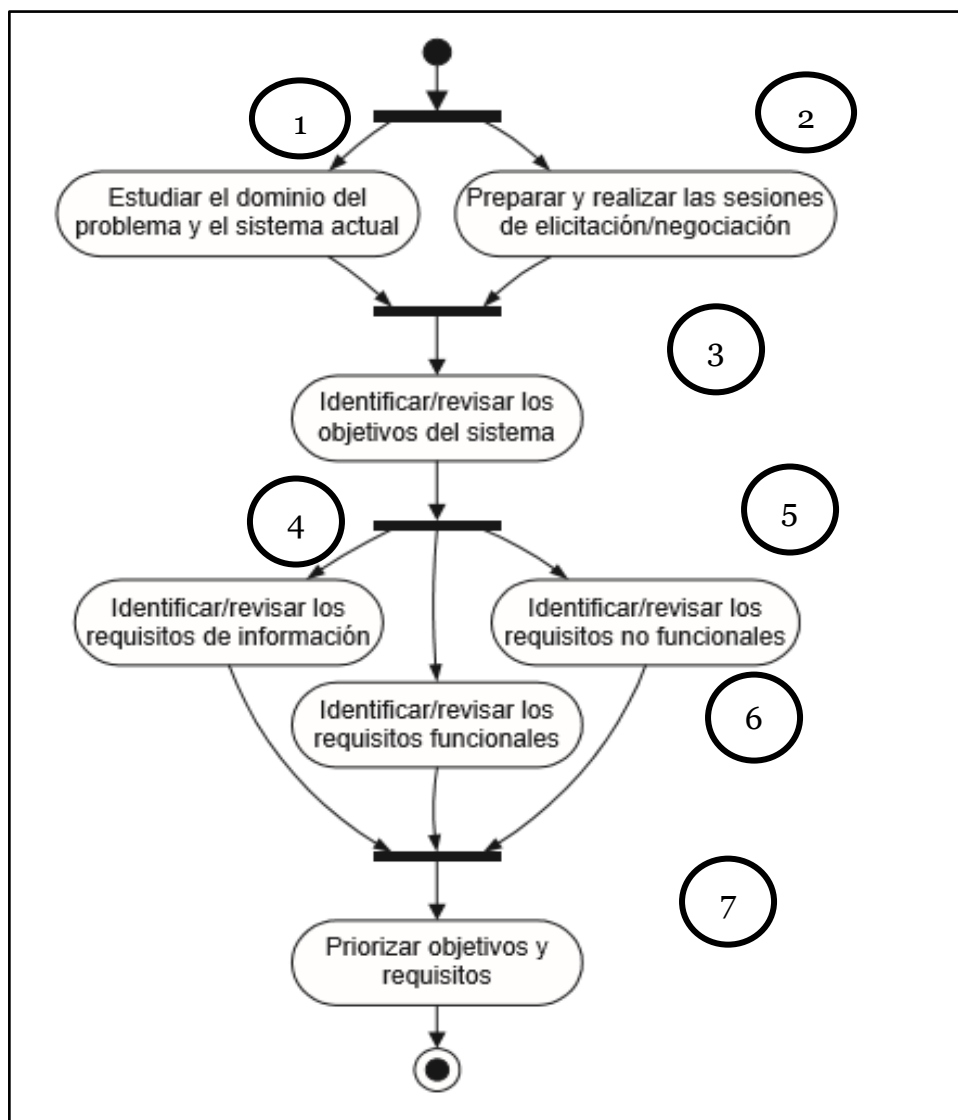
Tarea 5: Identificar/revisar los requisitos funcionales.

Tarea 6: Identificar/revisar los requisitos no funcionales.

Tarea 7: Priorizar objetivos y requisitos. (Durán, y Bernárdez,2002)

Figura 3.

Tareas de elicitación de requisitos.



Nota. La figura representa el conjunto de tareas que se deben realizar para la elicitación de requerimientos. Adaptado de Durán y Bernárdez (2002).

El orden recomendado de realización para estas tareas es: 1 a 7, aunque las tareas 4, 5, y 6 pueden realizarse simultáneamente o en cualquier orden que se considere oportuno. La tarea 1 es opcional y depende del conocimiento previo que tenga el equipo de desarrollo de IR sobre el dominio del problema y el sistema actual.

Técnicas

A continuación, se describen algunas de las técnicas que se proponen en esta metodología para obtener los productos de las tareas que se han descrito. Las técnicas más habituales en la elicitación de requisitos son las entrevistas, el Joint Application Development (JAD) o Desarrollo Conjunto de Aplicaciones, el brainstorming o tormenta de ideas y la utilización de escenarios (Weidenhaput et al. 1998, Rolland et al. 1998), más conocidos como casos de uso (Jacobson et al. 1993, Booch et al. 1999).

La entrevista

Es la técnica de elicitación más utilizada, y de hecho son prácticamente inevitables en cualquier desarrollo ya que son una de las formas de comunicación más naturales entre personas. En las entrevistas, y en otras técnicas de

interacción, se pueden identificar tres fases: preparación, realización y análisis (Piattini et al. 1996).

Las entrevistas formales o informales con participantes del sistema son una parte de la mayoría de los procesos de ingeniería de requerimientos, es el equipo de IR quienes formulan preguntas a los participantes sobre el sistema que actualmente usan y el sistema que se va a desarrollar. (Sommerville, 2011, p. 104)

Las entrevistas pueden ser de dos tipos: abiertas o cerradas.

Las entrevistas abiertas no se cumple un protocolo definido. El equipo de IR realiza las preguntas dejando más libertad para responder al entrevistado.

Las entrevistas cerradas el equipo de IR establece un conjunto de preguntas ya establecidas.

La técnica denominada JAD (Joint Application Development, Desarrollo Conjunto de Aplicaciones), desarrollada por IBM en 1977, es una alternativa a las entrevistas individuales que se desarrolla a lo largo de un conjunto de reuniones en grupo durante un periodo de 2 a 4 días. En estas reuniones se ayuda a los clientes y

usuarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo. Esta técnica se base en cuatro principios (Raghavan et al. 1994): dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación (diagramas, transparencias, multimedia, herramientas CASE, etc.), mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que se ves lo que se obtiene), por la que durante las reuniones se trabaja directamente sobre los documentos a generar. El JAD tiene dos grandes pasos, el JAD/Plan cuyo objetivo es elicitar y especificar requisitos, y el JAD/Design, en el que se aborda el diseño del software. En este documento sólo se verá con detalle el primero de ellos.

El brainstorming o tormenta de ideas es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios (Gause y Weinberg 1989, Raghavan et al. 1994). Las sesiones de brainstorming suelen estar formadas por un número de cuatro a diez participantes, uno de los cuales es el jefe de la sesión, encargado más de comenzar la sesión que de

controlarla. Como técnica de elicitación de requisitos, el brainstorming puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de elicitación, cuando los requisitos son todavía muy difusos.

Casos de estudio

Caso 1: STRONGBODIES

Strongbodies, una importante cadena local de clubes deportivos, ha experimentado un crecimiento espectacular en los últimos cinco años. Los directivos quieren depurar su proceso de toma de decisiones para la compra de un nuevo equipo de fisicoculturismo. Actualmente, los gerentes escuchan a los clientes, asisten a las exposiciones profesionales, examinan la publicidad y solicitan nuevas compras de equipo con base en sus percepciones subjetivas. Posteriormente, estas compras son aprobadas o rechazadas por Henry Mussels.

Henry es la primera persona que usted entrevistará. Es un gerente de división, de 37 años, que dirige cinco clubes del área. Viaja por toda la ciudad para llegar a las dispersas instalaciones de los clubes. Henry cuenta con una oficina

en las instalaciones del Este, aunque está allí menos de una cuarta parte de su tiempo.

Además, cuando Henry está en el club, está ocupado contestando llamadas telefónicas relacionadas con los negocios, resolviendo al instante problemas que le presentan los gerentes o interactuando con los miembros del club. Su tiempo es breve y, para compensarlo, se ha vuelto un gerente de división eficiente y extremadamente bien organizado. El no puede concederle demasiado tiempo para la entrevista. Sin embargo, la aportación que puede hacer es importante y siente que él sería el principal beneficiado con el sistema propuesto.

¿Qué tipo de pregunta podría ser el más adecuado para su entrevista con Henry? ¿Por qué este tipo es el más apropiado? ¿Cómo afectará su elección del tipo de pregunta la cantidad de tiempo que empleará en prepararse para entrevistar a Henry? Escriba 10 preguntas de este tipo.

Caso 2: SURECHECK DAIRY

Usted acaba de concluir una visita guiada a las instalaciones de la empresa de productos lácteos SureCheck Dairy y está a punto de salir cuando otro miembro de su equipo de análisis se sistema lo llama para decirle que no puede asistir a la cita para entrevistar al gerente de la planta porque está enfermo. El gerente de la planta se encuentra sumamente ocupado, y usted quiere que éste conserve el interés por el proyecto haciendo las cosas como se habían planeado. También comprende que, sin los datos de la entrevista inicial, el resto de la recopilación de datos se atrasará. Aunque no tiene planeada ninguna pregunta para la entrevista, decide seguir adelante y entrevistar al gerente de la planta inmediatamente.

Usted sabe que SureChek está interesado por procesar sus propios datos sobre las cantidades y tipos de productos lácteos vendidos con el fin de que su gerente pueda usar esa información para tener un mejor control de la producción de la gran línea de productos de la compañía (que incluye leche entera, descremada, al 2% y al 1%, leche en polvo, queso mozzarella, yogurt y helados). Los gerentes

de venta envían actualmente sus cifras de ventas a las oficinas centrales, a 950 kilómetros de distancia, y el procesamiento de estos datos parece lento. Usted basará sus preguntas improvisadas en lo que recién ha descubierto en el paseo.

Poco antes de que empiece la entrevista, escoja una pregunta embudo, pirámide o diamante. En un párrafo explique porque procedería con la estructura de la entrevista que ha escogido basado en el contexto poco común de esta entrevista. Escriba una serie de 10 a 15 preguntas y organícelas en la estructura que ha escogido.

Casos de Uso

Los *casos de uso*, son una técnica para la definición de requerimientos funcionales propuesta inicialmente por (Jacobson et al. 1993) y que actualmente forma parte de la propuesta de UML. Los casos de uso se han convertido en una característica fundamental del modelado de lenguaje unificado (UML). En su forma más sencilla, un caso de uso identifica a los actores implicados en una interacción, y nombra el tipo de interacción. (Sommerville, 2011, p. 107)

Un *caso de uso* es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra y en la que los actores obtienen resultados observables.

Los *casos de uso* tienen una representación gráfica en los denominados **diagramas de casos de uso**. En estos diagramas, **los actores** se representan en forma de pequeños monigotes y los casos de uso se representan por *elipses* contenidas dentro de un rectángulo que representa al sistema. La participación de los actores en los casos de uso se indica por una flecha entre el actor y el caso de uso que apunta en la dirección en la que fluye la información.

Los *casos de uso* son una técnica de descubrimiento de requerimientos que se introdujo por primera vez en el método Objectory (Jacobson et al., 1993). Ahora se ha convertido en una característica fundamental del modelado de lenguaje unificado. En su forma más sencilla, un caso de uso identifica a los actores implicados en una interacción, y nombra el tipo de interacción. (Sommer,2010, p. 107)

Características de los Casos de uso

Los casos de uso son una técnica para la especificación de requisitos funcionales propuesta inicialmente por Jacobson [Jacobson, 1987], [Jacobson et al. 1992] e incorporada a UML. Modelan la funcionalidad del sistema tal como la perciben los agentes externos, denominados actores, que interactúan con el sistema desde un punto de vista particular.

Sus componentes principales son:

- Sujeto: sistema que se modela.
- Casos de uso: unidades funcionales completas.
- Actores: entidades externas que interactúan con el sistema.

Actores

Un actor es un clasificador que modela un tipo de rol que juega una entidad que interacciona con el sujeto pero que es externa a él.

Los **actores**, en el contexto de los casos de uso, un actor es un interesado u otro sistema, se representan como figuras sencillas. (Sommerville, 2011, p. 107)

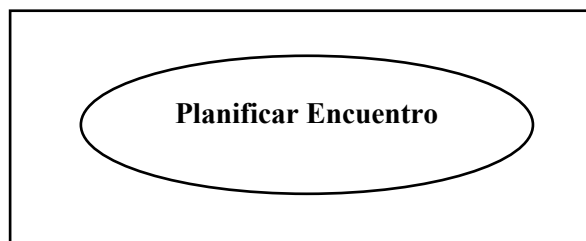
- Un actor puede tener múltiples instancias físicas
- Una instancia física de un actor puede jugar diferentes papeles. Los actores se comunican con el sujeto intercambiando mensajes (señales, llamadas o datos).

Notación

- Elipse con el nombre del caso de uso dentro o debajo de ella. Se puede colocar algún estereotipo encima del nombre y una lista de propiedades debajo.
- La representación alternativa es la del símbolo del clasificador con una elipse pequeña en la esquina superior derecha.

Figura 4.

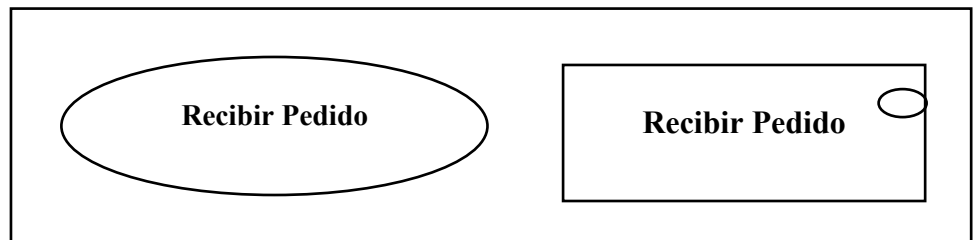
Notaciones usadas para la representación de Casos de Uso



Nota. En la figura la representación del Caso de uso Planificar encuentro.

Figura 5.

Notaciones usadas para la representación de Casos de Uso



Nota. En la figura la representación del Caso de uso Recibir Pedido.

Puedes tener en cuenta estas notaciones usadas para la representación de casos de uso.

Diagramas de Casos de uso

Los **diagramas de casos** de uso sirven para proporcionar una visión global del conjunto de casos de uso de un sistema, así como de los actores y los casos de uso en los que éstos intervienen. Las interacciones concretas entre los actores y el sistema no se muestran en este tipo de diagramas.

Como fruto de la experiencia de su utilización, para algunos campos de las plantillas se han identificado frases "estándar" que son habituales en las especificaciones de requisitos y que se han parametrizado. Estas frases, a las

que hemos denominado patrones lingüísticos, o abreviadamente patrones-L, pueden usarse para rellenar los campos de las plantillas dándole valores a los parámetros con la información oportuna.

Ambos aspectos, la estructuración de la información en forma de plantilla y la propuesta de frases "estándar", facilita la redacción de los requisitos, permitiendo a los participantes en las actividades de elicitación centrarse en expresar sus necesidades y no en cómo expresarlas. En la notación usada para describir los patrones-L, las palabras o frases entre < y > deben ser convenientemente reemplazadas, las palabras o frases que se encuentren entre { y } y separadas por comas representan opciones de las que se debe escoger una y las palabras entre [y] son opcionales, es decir, pueden aparecer o no. (Durán y Bernárdez,2002).

Modelado del Sistema

En este punto conoceremos algunos tipos de modelo de sistema que pueden desarrollarse, como parte de la ingeniería de requerimientos y los procesos de diseño del sistema. El Modelado del Sistema se define como el

proceso para elaborar, diseñar modelos abstractos, cada modelo presenta una visión o perspectiva diferente de dicho sistema. (Sommerville, 2011, p. 119)

Los modelos se usan durante el proceso de ingeniería de requerimientos para ayudar a derivar los requerimientos de un sistema, durante el proceso de diseño para describir el sistema a los ingenieros que implementan el sistema, y después de la implementación para documentar la estructura y la operación del sistema. (Sommerville, 2011, p. 119)

Es así como los modelos que desarrollamos en la IR se basan en representaciones gráficas en el Lenguaje de Modelado Unificado (UML). En este capítulo abordaremos el modelado gráfico.

Un modelo es una abstracción del sistema a estudiar. De manera ideal, una representación de un sistema debe mantener toda la información sobre la entidad a representar. (Sommerville, 2011, p. 119)

Un estudio realizado por Erickson y Siau, (2007) mostró que la mayoría de los usuarios del UML consideraban que cinco

tipos de diagrama podrían representar lo esencial de un sistema.

1. *Diagramas de actividad*, que muestran las actividades incluidas en un proceso o en el procesamiento de datos.

2. *Diagramas de caso de uso*, que exponen las interacciones entre un sistema y su entorno.

3. *Diagramas de secuencias*, que muestran las interacciones entre los actores y el sistema, y entre los componentes del sistema.

4. *Diagramas de clase*, que revelan las clases de objeto en el sistema y las asociaciones entre estas clases.

5. *Diagramas de estado*, que explican cómo reacciona el sistema frente a eventos internos y externos. (Sommerville, 2011, p. 120)

En este capítulo se centrará el Modelado de Casos de Uso, con el desarrollo de problemas basados en el tema.

Casos de estudio

Problema 1:

Se desea desarrollar un sistema de encuentros virtuales (parecido a un chat). Cuando se conecta al servidor, un usuario puede entrar o salir de un encuentro. Cada encuentro tiene un manager. El manager es el usuario que ha planificado el encuentro (el nombre del encuentro, la agenda del encuentro y el moderador del encuentro)

a. Descripción del Sistema actual

Actualmente la idea de la creación de encuentros virtuales, frente la situación que atraviesa el mundo entero, o por la distancia entre los actores del encuentro es cada vez más común, frente a ello se crean plataformas virtuales, o las mismas redes sociales que hoy en día conocemos. En consideración a lo planteado inicialmente, el desarrollo se puede suscitar en muchos ámbitos de implementación tales como: temas laborales, temas sociales, temas amorosos, temas educacionales, etc. Al plantearnos un Moderador, el deseo de este sistema es la formalidad y la fluidez de un agradable encuentro virtual.

Desde que la pandemia del Covid 19. Apareció en el mundo, se implantaron normas de bioseguridad en cada uno de los países, teniendo en cuenta la magnitud de contagios, como el aislamiento social obligatorio, uso de mascarilla, entre otros, lo que imposibilitó en la mayoría de casos un trabajo presencial, es así que tanto el trabajo remoto y la educación presencial se tornó virtual, es así como en Perú, se inicia con un proceso de adaptación hacia la virtualización y surgen los encuentros virtuales.

b. Formulación del Problema

¿Cómo funcionaría un sistema integrado de encuentros virtuales formales y fluidos?

c. Presentación del Sistema

El sistema integrado de Encuentros Virtuales ya tendrá designados a los Managers y Moderadores, para que de esa manera los integrantes del encuentro virtual, solo se dignen a asistir al encuentro.

d. Objetivos del Sistema

Brindar Formalidad y fluidez en el desarrollo de los encuentros virtuales.

Ofrecer Facilidad de planeación de los encuentros virtuales.

Reducción de tiempo en el proceso de identificación de quienes llevarán a cabo el encuentro virtual (Managers, Moderadores.)

e. Negocio detrás del Sistema (NDS)

Generar una alternativa de un encuentro virtual diferente, a las plataformas ya existentes.

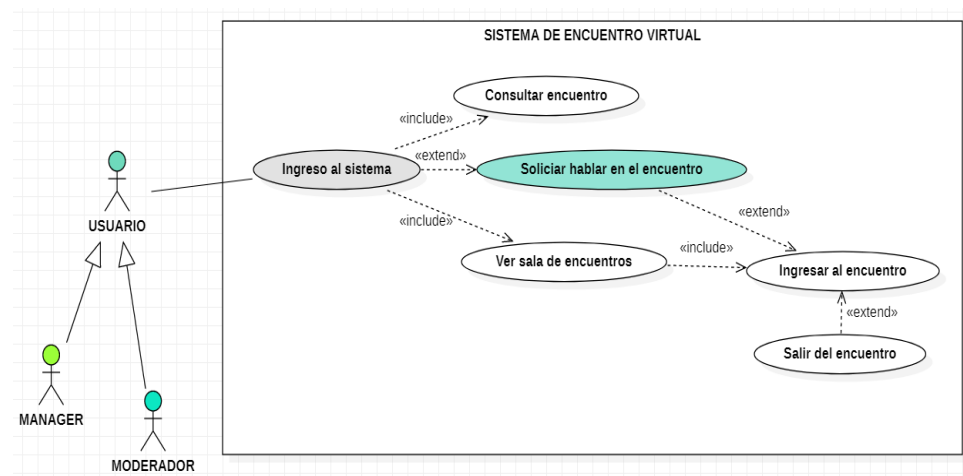
Ganar usuarios para el uso del método de trabajo actual en los encuentros virtuales.

f. Identificación de actores y casos de uso

Actores

Usuarios: Manager, Moderador.

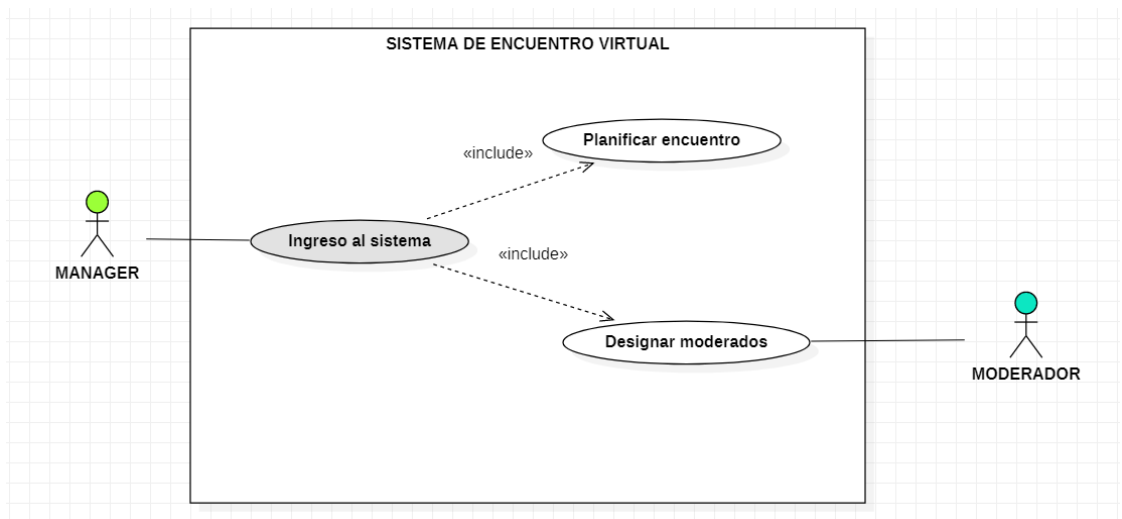
Figura 6.
Caso de uso de usuarios



Nota. La figura representa los casos de uso de usuarios del Sistema Encuentro Virtual, donde se aprecia como usuarios a manager y moderador.

Figura 7.

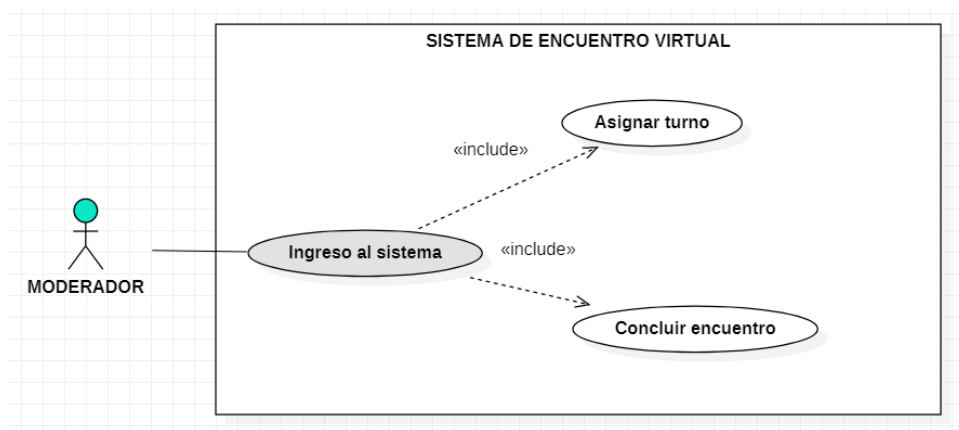
Caso de uso Manager



Nota. La figura representa el Caso de uso del usuario Manager del Sistema Encuentro Virtual, donde se aprecia que el manager puede realizar las acciones de Planificar encuentro y designar moderador.

Figura 8.

Caso de Uso Moderador



Nota. La figura representa el Caso de uso del usuario Moderador del Sistema Encuentro Virtual, donde se aprecia que el moderador puede realizar las acciones de asignar turno y concluir encuentro.

Problema 2:

Se pide desarrollar un Sistema para realizar compras por Internet.

a. Descripción del Sistema actual

El comercio electrónico o e-commerce supone el desarrollo de una nueva forma de comercio vinculada a una tecnología, la red de Internet. El comercio electrónico ha venido en alza, puesto que los consumidores ya no temen comprar sus productos por internet ya que esta es una forma rápida, segura y eficaz de adquirir bienes y servicios evitando de esta forma los contagios que podrían darse al comprar de forma presencial.

Generalmente en un proceso de compras online representa a los usuarios brindar información de los datos de la tarjeta de crédito, tener problemas para devolver un producto y no poder tocar lo que se quiere comprar, son algunos de los recelos más comunes. Al realizar las comprar por internet los clientes, aseguran que estos temores son infundados y recuerdan que la tecnología y la legislación son muy exigentes en materia de protección.

b. Formulación del Problema

¿Cómo desarrollar un sistema para realizar compras por Internet?

c. Presentación del Sistema

El Sistema para realizar compras por Internet, permite hacer comprar on line, brindando seguridad y confianza al cliente, permite lograr una mayor demanda en el mercado, con el fin mejorar la forma tradicional de realizar compras presenciales.

d. Objetivos del Sistema

Desarrollar un sistema de compras por internet buscando automatizar las distintas tareas que realizan.

e. Negocio detrás del Sistema (NDS)

El propósito es determinar el requerimiento del sistema ventas web.

- Maximizar las ganancias de la tienda.
- Mejorar el desempeño y eficiencia de los trabajadores de la tienda.
- Atraer clientes para minimizar los costos.

f. Identificación de actores

Actor	Administrador	Identificador: 001
Descripción	Acceso total al sistema web.	
Características	Iniciar sesión, visualizar pedidos, visualizar pedidos cancelados, visualizar pedidos entregados, visualiza estadísticas, visualizar pedidos entregados, ver detalle del pedido entregado, administrar usuarios, administrar categorías, agregar imágenes al producto, generar reporte del pedido, llamar al comprador, enviar reporte del pedido al comprador, cambiar estado del pedido, pedidos entregados, cancelar pedido.	
Relación	Relación entre el sistema web.	

Actor	Cliente	Identificador: 002
Descripción	Usuario que interactúa con el sistema.	
Características	Navegar por la tienda, ver detalles del producto, gestionar carrito, registrarse, ver productos disponibles.	
Relación	Existe relación entre el sistema web (Plaza Vea).	

Actor	Cliente registrado	Identificador: 003
Descripción	Usuario cliente registrado interactúa con el sistema web (Plaza Vea).	
Características	Navegar por la tienda, ver detalles del producto, gestionar carrito, ver productos disponibles, iniciar sesión, realizar compra, visualizar pedidos, visualizar datos, actualizar datos, cambiar contraseña, llamar al vendedor, descargar reporte de compra.	
Relación	Existe relación entre el sistema web (Plaza Vea). Pero con más privilegios.	

Representación de Casos de Uso

Figura 9.

Caso de Usos Procesos de la Tarjeta de Crédito

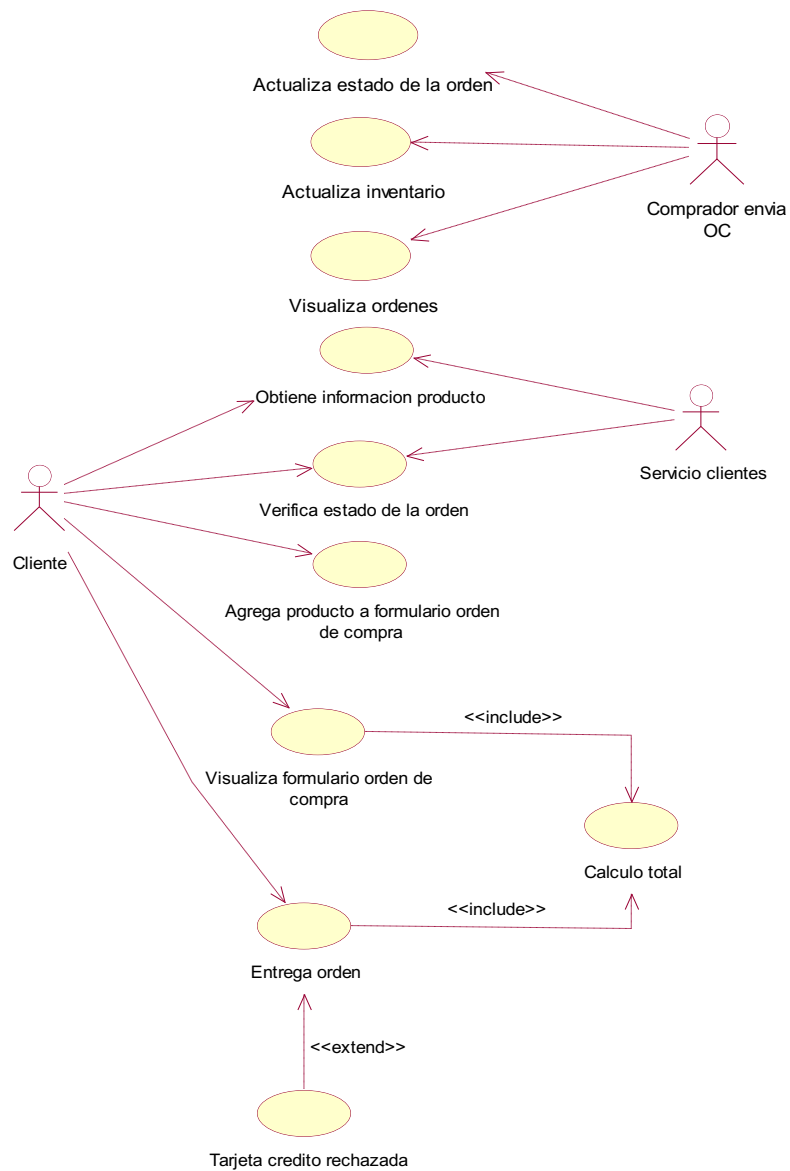
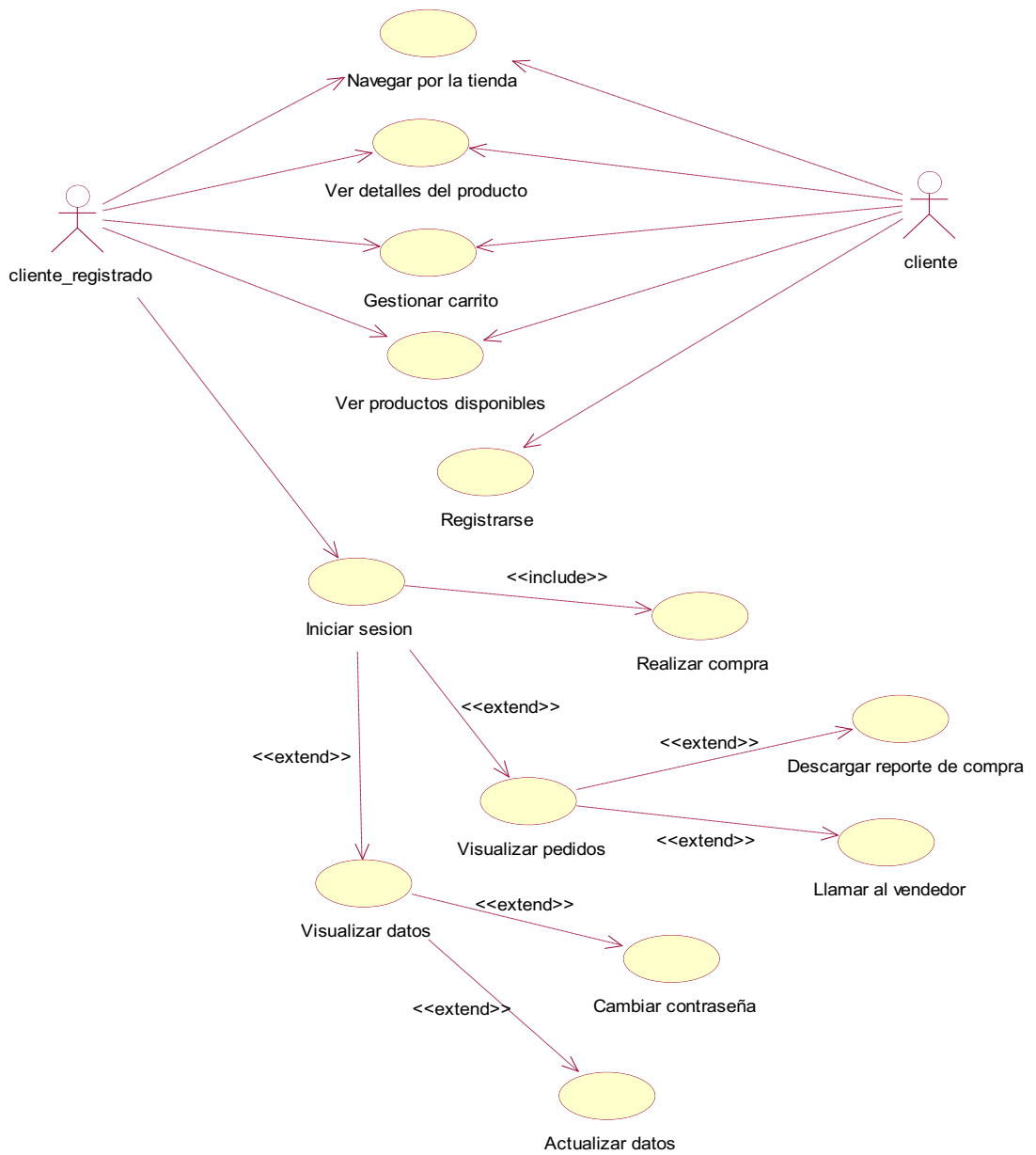


Figura 10.

Casos de uso Clientes del sistema



Caso de estudio

Ejercicio 1: SureCheck Dairy

Usted acaba de concluir una visita guiada a las instalaciones de la empresa de productos lácteos SureCheck Dairy y está a punto de salir cuando otro miembro de su equipo de análisis se sistema lo llama para decirle que no puede asistir a la cita para entrevistar al gerente de la planta porque está enfermo. El gerente de la planta se encuentra sumamente ocupado, y usted quiere que éste conserve el interés por el proyecto haciendo las cosas como se habían planeado. También comprende que, sin los datos de la entrevista inicial, el resto de la recopilación de datos se atrasará. Aunque no tiene planeada ninguna pregunta para la entrevista, decide seguir adelante y entrevistar al gerente de la planta inmediatamente.

Usted sabe que SureChek está interesado por procesar sus propios datos sobre las cantidades y tipos de productos lácteos vendidos con el fin de que su gerente pueda usar esa información para tener un mejor control de la producción de la gran línea de productos de la compañía (que incluye leche entera, descremada, al 2% y al 1%, leche en polvo, queso mozzarella, yogurt y helados). Los gerentes de venta envían actualmente sus cifras de ventas a las oficinas centrales, a 950 kilómetros de distancia, y el procesamiento de estos datos

parece lento. Usted basará sus preguntas improvisadas en lo que recién ha descubierto en el paseo.

Poco antes de que empiece la entrevista, escoja una pregunta embudo, pirámide o diamante. En un párrafo explique porque procedería con la estructura de la entrevista que ha escogido basado en el contexto poco común de esta entrevista. Escriba la lista de actores, tabla de tareas, descripción de los casos de uso y su modelado.

Referencias

- Cristiá, M. (20014). *Introducción a la Ingeniería de Software*. Universidad Nacional de Rosario. Argentina.
- Durán, A. y Bernárdez B. (2002). Metodología para la Elicitación de Requisitos de Sistemas Software. Informe Técnico LSI-2000-10.
- Pressman, R. (2010). Ingeniería del Software. Ed. McGraw Hill. México. ISBN: 9786071503145.
- Piattini, M. G., Calvo-Manzano, J. A., Cervera, J. y Fernández, L. (1996). *Análisis y Diseño detallado de aplicaciones informáticas de gestión*. Ed.RA-MA.
- Jackson, M. (1995). *Software requirements & specifications: a lexicon of practice, principles and prejudices*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.
- Schneider, G. y Winters J. (2001). *Applying Use Cases: A Practical Guide*, 2nd Edition. Pearson.
<https://www.pearson.com/us/higher-education/program/Schneider-Applying-Use->

Cases-A-Practical-Guide-2nd-
Edition/PGM108695.html?tab=authors

Sommerville, I. (2011). *Ingeniería del Software*. Pearson 9a.
Edición. Extraído del 21 de Junio del 2021 de:
[https://www.academia.edu/40713382/Libro_Somer
ville](https://www.academia.edu/40713382/Libro_Sommerville). ISBN: 978-607-32-0603-7

CAPÍTULO III

ANÁLISIS DE REQUERIMIENTOS

En este capítulo se aborda la etapa del proceso de ingeniería de requerimientos adquisición y análisis de requerimientos. En esta actividad, los ingenieros de software trabajan con clientes y usuarios finales del sistema para descubrir el dominio de aplicación, qué servicios debe proporcionar el sistema, el desempeño requerido de éste, las restricciones de hardware, entre otros.

Casos de Uso

Un caso de uso se define como un conjunto de acciones realizadas por el sistema que dan lugar a un resultado observable. El caso de uso especifica un comportamiento que el sujeto puede realizar en colaboración con uno o más actores, pero sin hacer referencia a su estructura interna. El caso de uso puede contener posibles variaciones de su comportamiento básico incluyendo manejo de errores y excepciones. Una instanciación de un caso de uso es un escenario que representa un uso particular del sistema (un camino).



Un caso de uso es un artefacto que define una secuencia de acciones que da lugar a un resultado de valor observable.




Características de los casos de uso

- Un caso de uso se inicia por un actor.
- Los casos de uso proporcionan valores a los actores.
- La funcionalidad de un caso de uso debe ser completa El comportamiento de un caso de uso se puede describir mediante interacciones, actividades, máquinas de estado.

Tabla 1

Relaciones de los casos de uso

Relación	Descripción	Notación
Asociación	Línea de comunicación entre un actor y un caso de uso en el que participa.	
Generalización	Una relación entre un caso de uso general y un caso	

	de uso específico, que hereda y añade propiedades al caso de uso base.	
Inclusión	Describe explícitamente la inserción.	<code><<include >></code> 
Extensión	Inserción de comportamiento adicional.	<code><<extend >></code> 
Realización	Establece relación entre el caso de uso y los diagramas que describen la funcionalidad del caso de uso.	

Nota. En la tabla 1, se muestra las relaciones de los casos de uso, descripción y notación.

Los casos de uso pueden tener asociaciones y dependencias con otros clasificadores.

Relación entre actores y casos de uso

- Asociación Relaciones entre casos de uso.
- Generalización: Un caso de uso también se puede especializar en uno o más casos de uso hijos.
- Inclusión: Un caso de uso puede incorporar el comportamiento de otros casos de uso como fragmentos de su propio comportamiento.
- Extensión: Un caso de uso también se puede definir como una extensión incremental de un caso de uso base Relación entre un caso de uso y una colaboración
- Realización: Se establece la relación entre el caso de uso y los diagramas que muestran la funcionalidad del caso de uso.

Proceso de análisis de requerimientos

Podríamos considerar las actividades señaladas:

A. Identificar actores

Se identifican los diferentes tipos de usuarios a los que el sistema ha de dar soporte.

B. Identificar escenarios

Se desarrolla el conjunto de escenarios para la funcionalidad proporcionada por el sistema.

C. Identificar casos de uso

Obtención de los casos de uso que representan el sistema

D. Refinar casos de uso

Garantizar que la especificación del sistema esté completa. Se describe el comportamiento del sistema en presencia de errores o condiciones de excepción

E. Identificar relaciones entre casos de uso

Se consolida el modelo de casos de uso eliminando redundancias

F. Identificar requisitos no funcionales

Aspectos relacionados con la funcionalidad: restricciones de rendimiento, documentación, recursos, seguridad, calidad.

A continuación, se describen cada uno de las actividades indicadas.

A. Actividad: Identificar actores (i)

Representan entidades externas que interaccionan con el sistema: Cualquier cosa que está "enlazada" al sistema: persona, sistema software, dispositivos hardware, almacenes de datos, redes de comunicaciones.

Son abstracciones de rol y no necesariamente se corresponden con personas.

Cada entidad externa puede estar representada por varios actores.

Si una persona física desempeña diferentes roles respecto al sistema se representa por varios actores.

Sirven para definir los límites del sistema y para buscar todas las perspectivas desde las que los desarrolladores tienen que considerar el sistema.

Actividad: Identificar actores (ii)

Guía

¿Quién usa el sistema?

¿Qué grupo de usuarios están soportados por el sistema para llevar a

cabo su trabajo?

¿Qué grupo de usuarios ejecutan las principales funciones del sistema?

¿Qué grupo de usuarios realiza las funciones auxiliares, tales como

¿mantenimiento y administración?

¿Quién inicial el sistema?

¿Quién instala el sistema?

¿Quién apaga el sistema?

¿Interaccionará el sistema con algún sistema software o hardware externo?

¿Existen otros sistemas que utilicen el sistema?

¿Quién obtiene información del sistema?

¿Quién proporciona información a nuestro sistema?

¿Ocurre algo de forma automática en un tiempo predeterminado?

B. Actividad: Identificar escenarios (i)

Descripción informal, concreta y orientada de una característica del sistema desde el punto de vista de un actor.

Los escenarios pueden tener diferentes utilidades a lo largo del ciclo de vida. Existen los siguientes tipos de escenarios.

As-is escenarios: Describen la situación actual.

Visionary escenarios: Describen un sistema futuro. Pueden utilizarse en la obtención de requisitos y en la representación de diseño. Pueden considerarse como prototipos baratos.

Evaluation escenarios: Describen las tareas de usuario contra las que se evaluará el sistema.

Training escenarios: Son tutoriales utilizados para introducir a los nuevos usuarios al sistema. Son instrucciones paso a paso diseñados para llevar de la mano al usuario a través de las tareas comunes.

En la elicitación de requisitos los desarrolladores y los usuarios escriben y refinan una serie de escenarios para conseguir un entendimiento común acerca de lo que debe ser el sistema.

Actividad: Identificar escenarios (ii)

Guía

¿Cuáles son las tareas que el actor quiere que realice el sistema?

¿A qué información quiere acceder el actor?

¿Quién crea los datos?

¿Puede ser modificada o eliminada? ¿Por quién?

¿Acerca de que cambios externos tiene el actor que informar al

sistema? ¿Con qué frecuencia? ¿Cuándo?

¿Acerca de qué eventos ha de ser informado el actor por parte del

sistema? ¿Con qué periodicidad?

La respuesta a estas preguntas se obtiene de la documentación del

dominio de la aplicación

Los escenarios han de escribirse utilizando el lenguaje del dominio

Los escenarios se formalizan en los casos de uso.

C. Actividad: Identificar casos de uso (i)

La identificación de actores y escenarios permiten la comprensión del dominio de la aplicación y la definición del sistema correcto.

El siguiente paso es la formalización de los escenarios en casos de uso.

Un caso de uso especifica todos los escenarios posibles para una funcionalidad dada.

Un caso de uso se inicia por un actor.

Una vez iniciado el caso de uso puede interactuar con otros actores.

Cada caso de uso es una secuencia completa de eventos iniciada por un actor y especifica la interacción que tiene lugar entre un actor y el sistema.

Un caso de uso es una forma concreta de utilizar un sistema haciendo uso de alguna parte de su funcionalidad.

Agrupar una familia de escenarios de uso según un criterio funcional

Es un proceso iterativo en el que se debe incluir, refinar, describir, eliminar los casos de uso.

No hay que olvidar los casos "raros" o el manejo de excepciones.

D. Actividad: Identificar casos de uso (ii)

-Se identifican los casos de uso de cada actor.

Un caso de uso es un comportamiento del sistema que produce un resultado medible y valioso para un actor.

Describe las cosas que los actores quieren del sistema.

Debe ser una tarea completa desde la perspectiva del actor.

En UML un caso de uso siempre se inicializa por un actor.

Pueden existir situaciones en las que se inicie de forma interna al sistema.

-Guía para encontrar casos de uso.

¿Qué funciones desea el actor del sistema?

¿Almacena el sistema información? ¿Qué actores la crean, leen, actualizan o borran?

¿Necesita el sistema comunicar los cambios en su estado interno a algún actor? ¿Existen eventos externos que el sistema tenga que conocer? ¿Qué actores notifican estos eventos al sistema?

¿Cuáles son las operaciones de mantenimiento del sistema?

Los casos de uso se determinan observando y precisando, actor por actor, las secuencias de

interacción (los escenarios) desde el punto de vista del usuario.

E. Actividad: Elaborar el modelo de casos de uso inicial

Una vez identificados los actores, escenarios y casos de uso se construye el **modelo inicial de casos de uso.**

Se establecen las relaciones de comunicación entre los actores y los casos de uso

Estas relaciones representan el flujo de información del caso de uso

El actor que inicia el caso de uso se distingue del resto de actores con los que se puede comunicar el caso de uso

Estas relaciones se van identificando a medida que se identifican los casos de uso.

Se construye el Diagrama de Casos de Uso

Se describen los casos de uso

Descripción del escenario básico.

F. Actividad: Documentar los casos de uso (i)

Cada caso de uso tiene que describir **qué** hace un sistema.

Un caso de uso debe ser simple, inteligible, claro y conciso.

Hay que considerar:

Funcionalidad básica – Flujo de eventos principal (escenario principal).

Alternativas – Flujo de eventos excepcional (escenario alternativo).

Condiciones de error – Flujo de eventos excepcional.

Precondiciones y poscondiciones – Qué tiene que ser cierto antes y después del caso de uso.

La descripción caso de uso puede contener condiciones, bifurcaciones e iteraciones.

El flujo de eventos se describe inicialmente de forma textual.

Cuando la comprensión de los requisitos ha avanzado se utilizan diagramas para especificar los escenarios.

Diagramas de interacción.

Conviene separar el flujo principal del flujo alternativo.

Se comienza describiendo el flujo principal (escenario básico) al que se le irán añadiendo alternativas y excepciones.

G. Actividad: Documentar los casos de uso (ii)

Escenario básico

Se escribe como si todo transcurriese de forma correcta.

Debe existir un escenario básico para cada caso de uso.

Serie de sentencias declarativas sin ramificaciones ni alternativas.

Escenarios alternativos

Describen secuencias distintas a la que fue utilizada en el camino básico.

Documentan las alternativas, situaciones de error o cancelación.

Dos métodos de localización de caminos alternativos.

Revisión de cada sentencia del escenario básico.

¿Hay alguna acción adicional que pueda hacerse en este punto?

¿Hay algo que pueda fallar en este punto?

¿Hay algún comportamiento que pueda suceder en cualquier momento?

Utilización de categorías

Un actor [aborta la aplicación / cancela una operación particular / solicita ayuda / proporciona mal los datos]

El sistema [falla / no está disponible]

Cada camino alternativo necesita un nombre y/o una descripción breve.

Se documenta en una sección de caminos alternativos o en documentación aparte.

H. Actividad: Documentar los casos de uso (iii)

Flujo de eventos

Es una serie de sentencias declarativas que "relatan" los pasos de un escenario desde la perspectiva del actor

Ha de indicar cómo se inicia.

Sentencia del tipo "El caso de uso comienza cuando..."

Ha de indicar cómo finaliza.

Sentencia del tipo "El caso de uso finaliza con..."

Las alternativas se muestran con una sentencia **if**

Se pueden indicar repeticiones.

- Hay que indicar claramente donde empieza y finaliza la repetición.
- Hay que indicar la condición de finalización.
- Utilizar **for** o **while**.

Cada paso tiene que ser una sentencia declarativa simple.

Por defecto, los pasos deberán estar ordenados temporalmente. En caso contrario ha de especificarse.

No debe incluirse demasiado detalle. Se están recogiendo los requisitos, no realizando análisis y diseño.

Los requisitos de tipo no funcional o se ponen en un documento aparte o se añaden como coletilla al final de la descripción.

I. Actividad: Documentar los casos de uso (iv)

Existen múltiples propuestas para la utilización concreta de los casos de uso como técnica tanto de obtención como de especificación de los requisitos funcionales del sistema.

Para la descripción concreta de los casos de uso se proponen plantillas, en las que las interacciones se numeran siguiendo diversas propuestas y se describen usando lenguaje natural.

Algunas de estas propuestas son:

[Schneider y Winters, 2001]

[Cockburn, 2000]

[Durán y Bernárdez, 2002]

J. Actividad: Documentar los casos de uso (v)

Los casos de uso describen cómo interactúan los actores externos con el sistema software a crear.

Sería deseable aislar e ilustrar las operaciones que un actor externo solicita a un sistema.

Un *diagrama de secuencia del sistema* (DSS) muestra, para un escenario específico de un caso de uso, los eventos que generan los actores externos, el orden y los eventos entre los sistemas.

Todos los sistemas se tratan como cajas negras.

Los diagramas destacan los eventos que cruzan los límites del sistema desde los actores a los sistemas.

Debería hacerse un DSS para el escenario principal del caso de uso, así como para los escenarios alternativos complejos o frecuentes [Larman, 2002]

K. Actividad: Documentar los casos de uso (vi)

En este texto se ha considerado utilizar el Esquema del Informe de Requerimientos del Sistema (DRS), según el estándar de IEEE 29148:2011 (antes el IEEE 830:1998). UML define varios modelos para la representación de los sistemas que pueden verse y manipularse mediante un conjunto de diagramas diferentes:

- Diagramas de estructura
- Diagrama de clases
- Diagrama de estructuras compuestas
- Diagrama de componentes
- Diagrama de despliegue
- Diagrama de objetos
- Diagrama de paquetes
- Diagramas de comportamiento

- Diagrama de casos de uso
- Diagrama de actividad
- Diagramas de interacción
- Diagrama de secuencia
- Diagrama de comunicación o colaboración
- Diagrama de visión global de la interacción
- Diagrama de tiempo
- Diagrama de máquina de estados

L. Actividad: Identificar relaciones entre casos de uso (i)

Incluso en sistemas de tamaño medio pueden aparecer un número elevado de casos de uso.

Una vez elaborado el modelo inicial de casos de uso hay que organizarlos.

Pueden existir similitudes en varios casos de uso que se pueden abstraer en un caso de uso común.

- Se utilizará la relación «include» para reducir la redundancia entre casos de uso.

- Se puede desear extender un caso de uso sin cambiar la descripción original.
- Se utilizará la relación «extend» para separar los flujos de eventos comunes de los excepcionales.

También se pueden encontrar similitudes en actores.

En resumen, se utilizan las relaciones de generalización, inclusión y extensión para factorizar el comportamiento común y las variantes.

M. Actividad: Identificar relaciones entre casos de uso

(ii)

Si se repiten bloques de comportamiento entre casos de uso, puede indicar que existe algo genérico que puede reutilizarse.

Se puede abstraer el comportamiento común en una relación de inclusión.

Factorizar el comportamiento común de los casos de uso presenta grandes ventajas, incluyendo descripciones más cortas y menor redundancia.

El comportamiento **únicamente** puede ser factorizado en un caso de uso separado si es compartido por dos o más casos de uso.

- La fragmentación excesiva de la especificación del sistema puede conducir a una especificación confusa.
 - Procedimiento
 - Identificar los pasos de los escenarios que se quieren utilizar en diferentes sitios.
 - Agruparlos en un caso de uso y darle nombre.
- El caso de uso incluido **no puede** tener dependencias con respecto a ningún caso de uso que lo incluya.

N. Actividad: Identificar relaciones entre casos de uso

(iii)

Un caso de uso extiende a otro caso de uso si el caso de uso extendido puede incluir el comportamiento de la extensión bajo ciertas condiciones

Modela la parte de un caso de uso que el usuario puede ver como un comportamiento opcional del sistema.

Se utiliza para extender de **forma condicionada** el comportamiento de un caso de uso existente.

Es una forma de añadir comportamiento a un caso de uso sin modificarlo.

Se utiliza cuando se trabaja con una versión posterior de un producto existente o para indicar los sitios donde un producto puede adaptarse o personalizarse.

El caso de uso se actualiza **añadiendo** puntos de extensión.

Las extensiones se describen al igual que los casos de uso. Debe existir una condición de entrada (inicio) y de salida.

El caso de uso extendido **no cambia**.

O. Actividad: Identificar relaciones entre casos de uso (iv)

La generalización indica que un caso de uso es una versión especializada de otro caso de uso.

El caso de uso general puede ser únicamente una descripción, los flujos se especifican en los casos de uso especializados.

Es posible agregar comportamiento al caso de uso hijo añadiendo pasos a la secuencia de comportamiento heredada del padre, así como

declarando relaciones de extensión y de inclusión para el hijo.

Si el padre es abstracto, su secuencia de comportamiento puede tener secciones que sean explícitamente incompletas en el padre y que debe proporcionar el hijo.

El hijo puede redefinir pasos heredados del padre.

En la secuencia heredada del padre se pueden intercalar pasos adicionales.

El uso de la generalización múltiple en casos de uso requiere la especificación explícita de la forma en la que se intercalan las secuencias de comportamiento de los padres para crear la secuencia correspondiente al hijo.

El caso de uso base (A) es autosuficiente. El caso de uso hijo (B) necesita al caso de uso base (obtiene la funcionalidad base de A) y controla qué es ejecutado desde A y qué cambia.

Ejemplos

Diagrama de Casos de Uso

Ejemplo 1: Sistema de citas médicas en línea

Descripción del problema

Problema General: ¿En qué medida la implementación del Sistema mejorará las citas médicas?

Objetivos del Sistema

Objetivo General: Implementar un Sistema para mejorar las de citas médicas en línea.

Negocio detrás del sistema

Optimizar el tiempo de atención de pacientes con respecto a las citas médicas online para ofrecer un servicio rápido, de calidad, fiable y con el menor costo posible.

Identificación de actores y casos de uso

Actores

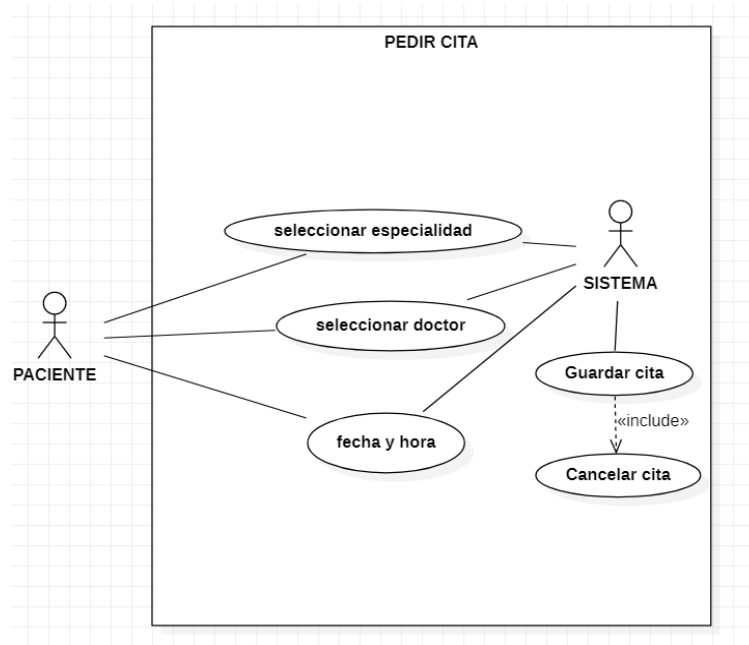
Paciente: Son todos aquellos usuarios pertenecientes al sistema, que soliciten una cita en el sistema.

Médico o Personal de Salud: Son aquellos usuarios que formen parte del personal sanitario de la entidad prestadora de salud.

Administrador: Es aquella persona que desempeñe las funciones de registro de nuevos usuarios y de gestión del calendario de citas.

A continuación, se muestra los casos de uso del ejemplo propuesto.

Figura 11.
Caso de Uso Pedir cita



Nota. En la figura se aprecia el caso de uso pedir cita del Sistema Citas médicas en línea. Se observa que el paciente podrá seleccionar la especialidad que necesite atenderse, seleccionar el doctor, la fecha y hora de la cita. El Sistema podrá guardar la cita o cancelar cita, también se observa la relación incluye (inclusión) entre guardar y cancelar cita.

Ejemplo 2: Sistema de Ventas por catálogo on line

Descripción del problema

En un Sistema de Venta por Catálogo on line los clientes hacen pedidos que recibe el departamento comercial y la empresa los atiende lo antes posible; y además ellos también pueden devolver productos y cancelar pedidos.

Objetivos del sistema

- Establecer los medios de pago para que el cliente pueda cancelar los pedidos.
- Mejorar el control de pagos para que haiga menos devoluciones.
- Mejorar control de stock y pedidos por catálogo.

Negocio detrás del sistema

- Incrementar las ventas en la empresa.

Identificación de Actores y Casos de usos

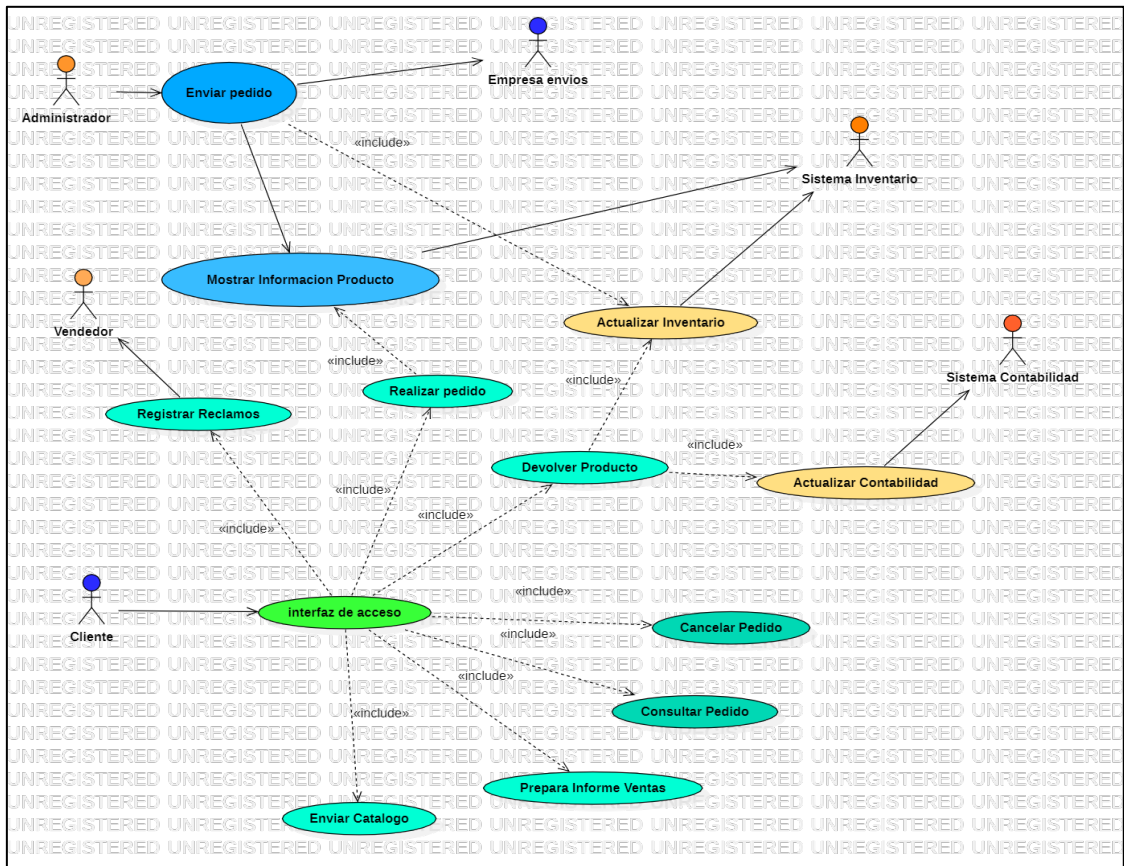
Actores

Internos: Administrador, Vendedor, Sistema de Inventario, Sistema contable.

Externos: Cliente, empresa de envíos.

Figura 12.

Caso de usos del Sistema Ventas por Catálogo on line



Nota. En la figura se aprecia el caso de uso propuesto para el Sistema de Ventas por catálogo on line, se observa la representación de los actores administrador, vendedor, cliente, sistema de inventario, sistema contabilidad, a su vez se muestra la relación <<include >> entre la interfaz de acceso y realizar pedido, devolver producto, cancelar pedido, consultar pedido, informe de ventas, enviar catálogo.

Caso de estudio

Caso 1: Telecomix Lagones TV

La empresa TELECOMIX, S. A., ha obtenido, por concurso público, la concesión para la realización de emisiones de televisión por cable, ante lo cual ha procedido a la apertura del LAGONESS TV.

Ante esta situación TELECOMIX, S. A., ha encargado a su departamento de informática que diseñe el sistema informático necesario para gestionar las operaciones que se pueden realizar en el canal.

Este sistema informático será utilizado por las siguientes personas:

El jefe de programación. Es el responsable de la elaboración de la parrilla de programación de LAGONESS TV. Utilizará el sistema informático cada vez que se defina o haya cambios en la parrilla del canal. Es decir, utilizará el sistema una cantidad de tiempo que suele suponer la cuarta parte de su jornada laboral.

El responsable de los abonados. Utilizará el sistema una cantidad de tiempo que suele suponer la cuarta parte de su jornada laboral. Es el responsable de la gestión de las suscripciones al canal LAGONESS TV, así como de la supervisión de las compras por parte de los abonados de los diferentes programas (retransmisiones deportivas, películas, conciertos, etc.) de pago. Además, será el encargado de recibir las reclamaciones que los diferentes abonados puedan realizar.

El director general de LAGONESS TV, el cual utilizará el sistema informático como ayuda a la toma de sus decisiones. Para ello será necesaria la obtención de estadísticas de visualización de programas ordenados por tipo, de ventas por duración y de número de errores en la retransmisión de programas.

Abonados. Los abonados, para poder acceder al canal LAGONESS TV, utilizarán desde sus casas un terminal que estará formado por una pantalla de televisión (para visualizar los programas y los menús) y un mini teclado inalámbrico para poder realizar las operaciones necesarias.

Se debe suponer que este tipo de usuarios no tiene ningún conocimiento previo en sistemas informáticos, que no va a recibir ninguna formación específica y que utilizará el sistema de una manera intensiva.

El sistema, dependiendo del tipo de usuario que lo utilice (director general, jefe de programación, responsable de abonados) ha de aparecer de distinta manera, es decir, la pantalla principal y los menús que son accesibles para un tipo de usuario no han de ser visibles ni accesibles para el resto. Por lo tanto, para que uno de estos usuarios pueda hacer sus operaciones, previamente deberá identificarse.

Una vez que se han descrito los principales usuarios del sistema se describirá, de una manera más detallada, cada una de las funcionalidades del sistema.

Determinación de programas

El jefe de programación debe determinar, en primer lugar, el tipo de operación que desea realizar: altas, bajas o modificaciones de un programa.

En caso de que se realice el alta de un nuevo programa, el jefe de programación deberá indicar el nombre del programa, su precio de emisión y al tipo al que pertenece.

Posteriormente, y en caso de que sea posible colocar en la programación, se mostrará un mensaje confirmando la operación. En caso de que haya un error se deberá indicar mediante un mensaje identificativo.

Para realizar una modificación es necesario indicar el nombre del programa. En caso de que éste exista, se podrá modificar su fecha de emisión, pero en caso de que no exista se deberá emitir un mensaje de error.

El funcionamiento de las bajas debe ser parecido, pero cuando se haya encontrado el programa a eliminar se pedirá la confirmación de la operación y, sólo en caso afirmativo, se realizará la eliminación del programa.

Gestionar abonados

La gestión de abonados abarca el alta, las modificaciones y la baja de los mismos. Una vez, que el responsable de los abonados se dispone a introducir los datos de una nueva suscripción, debe indicar los datos imprescindibles del mismo (NTF, nombre y apellidos, dirección, teléfono y datos bancarios). En caso de que el alta se haya ejecutado correctamente, se mostrará un mensaje de aceptación de

la operación, en caso contrario se mostrará un mensaje informando del problema.

Para poder modificar los datos de un abonado es necesario introducir su NIF. En caso de que se encuentre el abonado, se mostrarán sus datos, permitiendo modificar todos menos el NIF. Si el abonado que se desea modificar no existe, se deberá mostrar un mensaje informando de esta circunstancia.

El funcionamiento de las bajas debe ser parecido al de las modificaciones, pero cuando se haya encontrado el abonado a eliminar, se pedirá la confirmación de la operación y, sólo en caso afirmativo, se realizará la eliminación del programa.

Comprar emisiones

La suscripción de un contrato de abonado ofrece, al usuario que lo adquiere, el derecho a comprar las emisiones de los diferentes programas que realiza LAGONESS TV.

Para ello, los abonados, cuando accedan a la utilidad de adquisición de programas, podrán ver una lista con todos

los programas pendientes por emitirse, junto con el día y la hora de emisión.

Para comprar la emisión es importante que se pueda hacer fácilmente, es decir, seleccionando el programa y pulsando un botón para ordenar la operación de compra. Sería conveniente que se pidiera al usuario una confirmación de la operación para que no hubiera lugar a confusiones.

Ver programas de televisión

Una de las principales funcionalidades del terminal de los abonados es la emisión de los programas de televisión contratados por el abonado. Para ello deberá pulsar un botón que permita acceder a los programas comprados y disponibles en esa fecha. Una vez seleccionado el citado programa bastará con apretar un botón para comenzar la emisión del programa.

**Se pide identificar los actores, tabla de tareas, descripción de los casos de uso y sus diagramas.

Ejercicio 2: Gestión de Fincas e Inmuebles

Se desea desarrollar una aplicación de gestión de fincas e inmuebles. La aplicación deberá cubrir todos los aspectos

relacionados con dicho tema, teniendo en cuenta la siguiente dinámica de funcionamiento:

Una empresa gestiona un conjunto de inmuebles, que administra en calidad de propietaria. Cada inmueble puede ser bien un local (local comercial, oficinas,), un piso o bien un edificio que a su vez tiene pisos y locales. Como el número de inmuebles que la empresa gestiona no es un número fijo, la empresa propietaria exige que la aplicación permita tanto introducir nuevos pisos o locales con sus datos correspondientes (planta, letra,), darlos de baja, modificarlos y hacer consultas sobre ellos.

Cualquier persona que tenga una nómina, un aval bancario, un contrato de trabajo o venga avalado por otra persona puede alquilar el edificio completo o alguno de los pisos o locales que no estén ya alquilados, y posteriormente desalquilarlo. Por ello deberán poderse dar de alta, si son nuevos inquilinos, con sus datos correspondientes (nombre, DNI, edad, sexo, fotografía,), poder modificarlos, darlos de baja, consultar, etc. (para la realización de cualquiera de estas operaciones es necesaria la identificación por parte del inquilino).

Por otra parte, cada mes el secretario de la empresa pedirá la generación de un recibo para cada uno de los pisos y locales, el cual lleva asociado un número de recibo que es único para cada piso y para cada local y que no variará a lo largo del tiempo, indicando el piso o local a que pertenece, la fecha de emisión, la renta, el agua, la luz, la actualización del IPC anual, portería, IGV, etc. Y otros conceptos, teniendo en cuenta que unos serán opcionales (sólo para algunos recibos) y otros obligatorios (para todos los recibos). Además, para cada recibo se desea saber si está o no cobrado.

Con vistas a facilitar la emisión de recibos cada mes, la aplicación deberá permitir la generación de recibos idénticos a los del mes anterior, a excepción de la fecha. Además, deberán existir utilidades para inicializar los conceptos que se desee de los recibos a una determinada cantidad y también debe ser posible modificar recibos emitidos en meses anteriores al actual. La aplicación también deberá presentar los recibos en formato impreso, pero teniendo en cuenta que en un recibo nunca aparecerán aquellos conceptos cuyo importe sea igual a cero.

De igual forma, el secretario debe poder gestionar los movimientos bancarios que se producen asociados a cada edificio, piso o local. Un movimiento bancario siempre estará asociado a un banco y a una cuenta determinada de ese banco. En esa cuenta existirá un saldo, acreedor o deudor, que aumentará o disminuirá con cada movimiento. Para cada movimiento se desea saber también la fecha en que se ha realizado. Un movimiento bancario puede ser de dos tipos: un gasto o un ingreso.

Si el movimiento bancario es un gasto, entonces estará asociado a un inmueble determinado, y se indicará el tipo de gasto al que pertenece entre los que se tienen estipulados. Ejemplos de gastos son el coste de la reparación del ascensor del inmueble que pertenece a un gasto de reparación, el sueldo de la señora de la limpieza, etc. Si el movimiento bancario es un ingreso entonces estará asociado a un piso de un inmueble determinado o a un local y también se indicará el tipo de ingreso al que pertenece, como en el caso de los gastos. Ejemplos de ingresos son precisamente los recibos que se cobran cada mes a los inquilinos.

Basándose en los gastos e ingresos que se deducen de los movimientos bancarios, la aplicación deberá ser capaz de ocuparse de la gestión económica generando los informes que facilitan la realización de la declaración de la renta.

Por último, la aplicación deberá ser capaz de proporcionar el acceso, de forma estructurada, a toda la información almacenada en el sistema, generando para ello los listados necesarios que requiera el secretario.

Ejemplos de listados: el listado de todos los inquilinos ordenados por fecha, el listado de inquilinos que han pagado o no en un determinado intervalo de tiempo, el listado de todos los inmuebles, el listado de todos los pisos y locales de cada edificio, el listado de todos los recibos de cobro de un determinado intervalo de tiempo, etc.

**Se le pide realizar las siguientes tareas, teniendo en cuenta el ejercicio planteado:

Tarea 1: Obtener información sobre el dominio del problema y el sistema actual.

Tarea 2: Preparar y realizar las reuniones de elicitación/negociación.

Tarea 3: Identificar/revisar los objetivos del sistema.

Tarea 4: Identificar/revisar los requisitos de información.

Tarea 5: Identificar/revisar los requisitos funcionales.

Tarea 6: Identificar/revisar los requisitos no funcionales.

Tarea 7: Priorizar objetivos y requisitos.

Referencias

- Cristiá, M. (20014). *Introducción a la Ingeniería de Software*. Universidad Nacional de Rosario. Argentina.
- Durán, A. y Bernárdez B. (2002). *Metodología para la Elicitación de Requisitos de Sistemas Software*. Informe Técnico LSI-2000-10.
- García-Peñalvo, J. y Vázquez-Ingelmo A. (2019). *Ingeniería de Requisitos*. Eds., Salamanca, España: Grupo GRIAL, Universidad de Salamanca.
- Pressman, R. (2010). *Ingeniería del Software*. Ed. McGraw Hill. ISBN: 9786071503145.
- Sommerville, I.. (2011). *Ingeniería del Software*. Ed. Pearson Educación. ISBN: 9788478290741.

CAPÍTULO IV

ESPECIFICACIÓN Y VALIDACIÓN DE REQUERIMIENTOS

Para la verificación de requisitos se deben añadir criterios de aceptación por cada requisito, una tarea de la calidad es asegurarse de que cada requisito cumple con los criterios asignados, este criterio es una medida del requisito que lo hace entendible y con capacidad de ser probado.

Validación de Requerimientos

Objetivo

Esta actividad tiene como objetivo realizar la Validación de todos los requerimientos del sistema a construir.

Estos requerimientos pueden ser tanto funcionales como no funcionales.

Descripción

Los Analistas se reúnen con el Cliente y/o usuario del sistema para validar los requerimientos que se relevaron y especificaron, es decir que estas especificaciones reflejan realmente lo que los usuarios necesitan.

De la reunión de Validación de Requerimientos pueden surgir cambios en los requerimientos que impliquen que se realicen nuevamente las actividades Especificación de Requerimientos y Priorización de Requerimientos.

Entrada

- Especificación de Requerimientos
- Requerimientos Suplementarios
- Modelo de Casos de Uso
- Glosario
- Requerimientos Candidatos

Salida

- Acta de Reunión de Requerimientos

Rol responsable

- Analista

Roles involucrados

- Cliente y/o Usuario
- Administrador
- Responsable de SQA

- Responsable de Verificación
- Arquitecto

Los requisitos se recogen en documentos técnicos que reciben el nombre genérico de ERS (Especificación de Requisitos del Software)

- Este documento debe contemplar tanto los requisitos–C como los requisitos–D.

Hay metodologías que abogan por la separación de estas representaciones en dos documentos diferentes:

- DRS (Documento de Requisitos del Sistema), también denominado catálogo de requisitos, donde se recogen los requisitos–C.
- ERS propiamente dicha, donde se recogen los requisitos–D
- En la realización de una ERS participan: Ingenieros de software (analistas), Clientes y usuarios.

Catálogo de Requisitos: El catálogo de requisitos recoge todos los requisitos del sistema, de forma que éste quede definido con precisión. El catálogo se elabora y se actualiza a lo largo de las etapas de análisis y diseño.

Matriz de Rastreabilidad: La **matriz de trazabilidad de requerimientos del proyecto** es el instrumento base para el diseño y ejecución de la ingeniería de requisitos.

En este texto se ha considerado utilizar el Esquema del Informe de Requerimientos del Sistema (DRS), según el estándar de IEEE 29148:2011 (antes el IEEE 830:1998), también se han diseñado documentos para los requerimientos.

Consideraciones para un documento de requerimientos

Lo primero y más importante en esta parte es registrar los requerimientos considerando lo solicitado por el cliente.

Al plantear las especificaciones debes considerar que toda especificación debe ser válida, no ambigua, completa, consistente, agrupada y ordenada por importancia, verificable, modificable, trazable, factible, entendible.

El Estándar 29148 también describe tres documentos asociados a dichos procesos:

1. *Documento de especificación de requerimientos de stakeholder*: describe los requerimientos de negocio y la motivación tras la construcción del sistema (en este

contexto el sistema implica no solo software, sino también hardware, infraestructura y personal). Incluye descripciones de procesos, políticas y reglas de negocio que serán afectados por él. 2. *Documento de especificación de requerimientos de sistema*: describe el sistema, su entorno y las formas en las cuales interactuará con los usuarios y otros sistemas. Incluye elementos asociados a restricciones, requerimientos no funcionales (usabilidad, desempeño, seguridad, etc.).

3. *Documento de especificación de requerimientos de software*: es el documento de uso más común y se centra exclusivamente en la descripción del software, sus restricciones, operaciones y requerimientos.

Consideraciones para la Validación de Requerimientos Funcionales

Como se ha indicado la validación es el proceso el cual se determina si los requerimientos son consistentes con las necesidades del cliente. Para lo cual se ha de tener en cuenta;

Verificación — determina si el producto de alguna actividad de desarrollo cumple los requisitos (hacer las cosas bien).

Validación — evalúa si un producto satisface las necesidades del cliente (hacer la cosa correcta).

Además, se debe tener en cuenta:

Planificar quién (qué stakeholder) va a validar qué (artefacto) cómo (técnica).

Ejecutar

Registrar

Reporte de validación / Firma

Chequeo de Requerimientos Funcionales

Validez— ¿El sistema provee las funciones que mejor soportan las necesidades del cliente? Consistencia— ¿Hay algún requisito en conflicto?

Completitud— ¿Están incluidas todas las funciones requeridas por el cliente?

Realismo— ¿Los requisitos pueden ser implementados con el presupuesto y la tecnología disponible?

Verificabilidad— ¿Los requisitos pueden ser chequeados?

Consideraciones para la Validación de Requerimientos Funcionales

Son difíciles de validar.

Se deben expresar de manera cuantitativa utilizando métricas que se puedan probar de forma objetiva (esto es ideal).

Por ejemplo, tener en cuenta las siguientes propiedades al momento de la validación: rapidez, tamaño, fiabilidad, portabilidad y facilidad de uso.

Referencias:

- Cristiá, M. (20014). *Introducción a la Ingeniería de Software*. Universidad Nacional de Rosario. Argentina.
- Durán, A. y Bernárdez B. (2002). *Metodología para la Elicitación de Requisitos de Sistemas Software*. Informe Técnico LSI-2000-10.
- García-Peñalvo, J. y Vázquez-Ingelmo A. (2019). *Ingeniería de Requisitos*. Eds., Salamanca, España: Grupo GRIAL, Universidad de Salamanca.
- Pressman, R, (2010). *Ingeniería de Software. Un enfoque práctico*. Mc Graw Hill. 7ma. Edición. México. ISBN: 978-607-15-0314-5
- Schneider, G. y Winters J. (2001). *Applying Use Cases: A Practical Guide*, 2nd Edition. Pearson.
<https://www.pearson.com/us/higher-education/program/Schneider-Applying-Use-Cases-A-Practical-Guide-2nd-Edition/PGM108695.html?tab=authors>

Sommerville, I. (2011). *Ingeniería del Software*. Pearson 9a. Edición. Extraído del 21 de Junio del 2021 de: https://www.academia.edu/40713382/Libro_Somerville. ISBN: 978-607-32-0603-7

Apéndice

Según el estándar 29148:2011 (antes el IEEE 830:1998),

Documento de Requisitos del Sistema (DRS)

Introducción

Propósito

[En esta sección se debe especificar el propósito de este documento de **Reunión de Requerimientos**.]

Alcance

[En esta sección incluya una breve descripción del alcance de la reunión indicando todo aquello que es afectado o influenciado por este documento de **Reunión de Requerimientos**.]

Definiciones, siglas y abreviaturas

[Esta sección debe proporcionar las definiciones de todos los términos, las siglas, y abreviaciones requeridas para interpretar apropiadamente el documento **Reunión de Requerimientos**. Esta información puede proporcionarse por la referencia al Glosario del proyecto.]

Referencias

[Esta sección debe proporcionar una lista completa de todos los documentos a los que se hace referencia en este documento Reunión de Requerimientos. Cada documento debe identificarse por el título, número del informe (si se

aplica), fecha, y organización que lo publica. Especifique las fuentes de las que pueden obtenerse las referencias. Esta información puede proporcionarse por la referencia a un apéndice o a otro documento. Entre éstas referencias pueden estar documentos que establecen la operativa del usuario o cliente, normas de calidad para los procesos que usa el cliente o usuario, documentos estándares relativos a los temas de la reunión que usa el cliente o usuario, etc.]

Establecer el perfil del usuario

Nombre: _____ Empresa/Sector: _____

Trabajo que realiza: _____

Responsabilidades principales: _____

Producto que produce: _____ Para quién?: _____

¿Cómo mide el éxito de su trabajo?: _____

¿Qué problemas interfieren con el éxito de su trabajo? _____

¿Qué elementos, si existen, hacen su trabajo más fácil o más difícil? _____

Evaluar el problema

¿Para qué problema de [tipo de aplicación] le faltan las soluciones buenas? _____

¿Cuáles son los problemas? ¿Hay algo más? _____

Para cada problema preguntar: _____

¿Por qué existe el problema? _____

¿Cómo lo resuelva ahora?

¿Cómo le gustaría resolverlo?

¿Qué prioridad le asigna a este problema?

Entender el entorno del usuario

¿Quiénes son los usuarios?

¿Cuál es su nivel educativo?

¿Cuál es su conocimiento en computación?

¿Tienen experiencia en el tipo de aplicación?

¿Qué plataformas hay en uso?

¿Qué planes tiene sobre plataformas futuras?

¿Qué otra aplicación usa que necesiten comunicarse con la que está en discusión?

¿Qué expectativas tiene sobre la usabilidad del producto?

¿Qué expectativas tiene sobre el tiempo de capacitación?

¿Qué tipo de documentación, en papel o en línea, necesita?

Reafirmar el entendimiento

Me has dicho [lista de los problemas descritos por el usuario o cliente en sus propias palabras]:

- [Problema 1]

- [Problema 2]

¿Esto representa los problemas que usted está teniendo con su solución existente?

¿Qué otro problema está experimentando?

Problemas adicionales

[Lista de todas las necesidades o problemas adicionales que piensa que conciernen al involucrado o usuario, o problemas que surgen si no se logra resolver alguno de los planteados previamente.]

- [Necesidad o problema adicional 1]
- [Necesidad o problema adicional 2]

Pregunte por cada problema sugerido:

¿Es este un problema real?

¿Cuáles son las razones para este problema?

¿Cómo resuelve usted el problema actualmente?

¿Cómo le gustaría resolver el problema?

¿Qué prioridad le daría a este problema comparado con otros usted ha mencionado?

Evaluar la solución (si aplica)

Si usted pudiera:

[Si ya evaluó soluciones posibles y desea plantearlas al cliente o usuario escriba un resumen de las capacidades importantes de su solución propuesta, para solicitar al usuario o involucrado que asigne importancia o prioridad a las mismas.]

- [Capacidad 1]

○ [Capacidad 2]

○ ...

¿Cómo los organizaría según la importancia de los mismos?

Evaluar la oportunidad

¿Quién necesita esta aplicación en la organización?

☒ ¿Cuántos de estos tipos de usuarios usaría la aplicación?

¿Cómo valoraría una solución exitosa?

Evaluar fiabilidad, performance y necesidad de apoyo

¿Cuáles son sus expectativas para la fiabilidad?

¿Cuáles son sus expectativas para la performance?

¿Usted hará el soporte del producto, u otros lo harán?

¿Usted tiene necesidades especiales de soporte?

¿Qué necesidades hay sobre el mantenimiento y acceso de servicio?

¿Cuáles son los requisitos de seguridad?

¿Cuáles son los requisitos de instalación y de configuración?

¿Cuáles son los requisitos de autorizaciones especiales?

¿Cómo se quiere distribuir el software?

¿Qué requisitos hay sobre el etiquetado y empaquetando?

Otros requerimientos

¿Se debe soportar algún requisito regulador o normas?

¿Usted piensa que hay cualquier otro requisito que nosotros debemos saber?

Cierre

¿Hay alguna otra pregunta que yo debería hacerle?

¿Si yo necesito preguntarle algo, puedo llamarle?

¿Le gustaría participar en una revisión de los requerimientos?

Resumen del Analista

[Resuma debajo los tres o cuatro problemas de prioridad más alta para este usuario involucrado]

[Problema 1]

[Problema 2]

[Problema 3]



Julissa Elizabeth Reyna González
Universidad Nacional Hermilio Valdizan
<https://orcid.org/0000-0001-9970-9025>
jreyna@unheval.edu.pe

Ingeniera de Sistemas, con Maestría en Docencia y Gestión Educativa, Maestría en Administración y Marketing, estudiante del doctorado en Educación, pertenece al sistema de investigadores del Perú-CONCYTEC, y miembro vitalicio del Colegio de Ingenieros del Perú (CIPLL), miembro activo de la IEEE. Desde 2005 es catedrática en diferentes Universidades del Perú, inició sus actividades como docente en la Universidad Señor de Sipan en la Facultad de Ingeniería, desde el 2010 fue docente en la Escuela de Posgrado de la UCV filial Trujillo. Ha sido docente en la Facultad de Ciencias Empresariales, Escuela de Marketing y Dirección de Empresas de la Universidad Cesar Vallejo y actualmente docente en la Facultad de Ingeniería, Escuela de Ingeniería de Sistemas de la Universidad Nacional Hermilio Valdizan, Huánuco, Perú. Asesora y Jurado de Tesis en diversas universidades de la región. Ponente Internacional.



Freddy Ronald Huapaya Condori
Universidad Nacional Hermilio Valdizan
<https://orcid.org/0000-0003-4783-3803>
fhuapaya@unheval.edu.pe

Bachiller en Ingeniería de Sistemas de la Universidad de Huánuco – (UDH), Ingeniera de Sistemas e Informática de la Universidad de Huánuco – (UDH) cuenta con Maestría en Gestión de Proyectos, pertenece al sistema de investigadores del Perú-CONCYTEC. Ingeniero colegiado, con orientación en Gestión y Tecnología de Información y comunicación. Más de 12 años de experiencia en Manejo de empresas y áreas administrativas, capacidad en la toma de decisiones trascendentales, facilidad de palabras, trabajo en equipo, AUDITOR Y ASESOR en sistemas e Informática, dominio de técnicas para pruebas previas al ambiente de producción. Desde 2014 es catedrático en diferentes Universidades del Perú, inició sus actividades como docente en la Universidad de Huánuco en la Facultad de Ingeniería, del 2015 fui docente en la Universidad Alas Peruanas en la facultad de ingeniería, desde 2018 es docente en la Facultad de Ingeniería, Escuela Ingeniería de Sistemas de la Universidad Nacional Hermilio Valdizan.



Roberto Sixto Perales Flores
Universidad Nacional Hermilio Valdizan
<https://orcid.org/0000-0002-2502-9593>
rperales@unheval.edu.pe

Bachiller en Ingeniería Mecánica y Eléctrica de la Universidad Nacional de Ingeniería – (UNI); Ingeniero Electrónico de la Universidad Nacional de Ingeniería – (UNI); Magíster en Educación con mención en Gestión y Planeamiento Educativo de la Universidad Nacional Hermilio Valdizán - (UNHEVAL) y Doctor en Medio Ambiente y Desarrollo Sostenible de la Universidad Nacional Hermilio Valdizán – (UNHEVAL). Miembro del Sistema de Investigadores del Perú –CONCYTEC, Registro Nacional de Investigadores – RENACYT; así como, miembro vitalicio del Colegio de Ingenieros del Perú - Huánuco (CIP-Huánuco). Docente a tiempo completo y dedicación exclusiva desde 1980, con más de 40 años de experiencia académica en la Universidad Nacional Hermilio Valdizán – (UNHEVAL).



Denny John Fuentes Adrianzén
Universidad Nacional Pedro Ruiz Gallo
<https://orcid.org/0000-0003-4864-1352>
dfuentesad@unprg.edu.pe

Ingeniero Informático y de Sistemas, Maestro en Administración con Mención en Gerencia Empresarial, Estudios Concluidos de Doctorado en Ciencias de la Computación y Sistemas. Además, con Estudios Concluidos en la Maestría en Gestión Pública por EUCIM-USMP. Docente adscrito al Departamento Académico de Computación y Electrónica de la Universidad Nacional Pedro Ruiz Gallo de Lambayeque, Perú.