



UNIVERSIDAD JOSÉ CARLOS MARIÁTEGUI

VICERRECTORADO DE INVESTIGACIÓN

**FACULTAD DE INGENIERÍA Y
ARQUITECTURA**

**ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMAS E INFORMÁTICA**

TRABAJO DE SUFICIENCIA PROFESIONAL

**PROCESO UNIFICADO DE DESARROLLO
DE SOFTWARE RUP**

PRESENTADO POR:

BACHILLER OSMAR CATALÁN CALIZAYA

ASESOR:

ING. WALTER COAYLA MAMANI

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS E INFORMÁTICA**

**MOQUEGUA – PERÚ
2019**

CONTENIDO

PORTADA	Pág.
Página de jurado.....	i
Dedicatoria.....	ii
Contenido.....	iii
Índice de figuras.....	v

CAPÍTULO I

Introducción.....	1
-------------------	---

CAPÍTULO II

OBJETIVOS

Objetivo general.....	3
Objetivos específicos.....	3

CAPÍTULO III

DESARROLLO DEL TEMA

3.1. Marco teórico.....	4
3.1.1. Proceso unificado de desarrollo de software (RUP).....	4
3.1.2. Lenguaje de modelado unificado (UML).....	8
3.1.3. Diagramas de UML.....	12
3.1.4. Orientado a objetos.....	11
3.2. Caso práctico.....	17
3.2.1. Información de la empresa.....	17
3.2.2. Planteamiento del problema.....	17
3.2.3. Análisis y modelamiento.....	18
3.3. Representación de resultados.....	27

3.3.1. Diagrama de actividad.....	28
3.3.2. Diagrama de secuencia.....	35
3.3.3. Diagrama de colaboración.....	41
3.3.4. Diagrama de estado.....	46
3.3.5. Diagrama de paquete.....	48
3.3.6. Diagrama de clases.....	48

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones.....	49
4.2 Recomendaciones.....	49
REFERENCIAS BIBLIOGRÁFICAS.....	50

ÍNDICE DE FIGURAS

	Pág.
Figura 1. Inicios de RUP.....	5
Figura 2. Modelo de clase.....	10
Figura 3. Modelo de objeto.....	11
Figura 4. Modelo de caso de uso.....	11
Figura 5. Modelo de diagrama de caso de uso.....	11
Figura 6. Modelo de diagrama de estado.....	12
Figura 7. Modelo de diagrama de actividad.....	13
Figura 8. Modelo de diagrama de secuencia.....	13
Figura 9. Modelo de diagrama de colaboración.....	14
Figura 10. Gestión de personal usuario - diagrama de CU.....	19
Figura 11. Esquema CU-001 iniciar sesión.....	19
Figura 12. Esquema CU-002 validar datos.....	20
Figura 13. Gestión de salida pedido de artículo - diagrama de CU.....	20
Figura 14. Esquema CU-003 buscar artículo.....	21
Figura 15. Esquema CU-004 solicitar pedido.....	21
Figura 16. Esquema CU-005 entregar pedido.....	22
Figura 17. Gestión de abastecimiento de artículo - diagrama de CU.....	22
Figura 18. Esquema CU-006 hacer pedido faltante.....	23
Figura 19. Esquema CU-007 ingresar artículo faltante.....	23
Figura 20. Esquema CU-008 buscar artículo.....	24
Figura 21. Gestión de artículo - diagrama de CU.....	24
Figura 22. Esquema CU-009 agregar artículo.....	25
Figura 23. Esquema CU-010 actualizar datos y stock.....	25
Figura 24. Esquema CU-011 buscar artículo.....	26
Figura 25. Gestión de informes - diagrama de CU.....	26
Figura 26. Esquema CU-012 solicitar informe.....	27
Figura 27. Esquema CU-013 generar reportes.....	27
Figura 28. Diagrama de actividad para iniciar sesión.....	28
Figura 29. Diagrama de actividad para validar datos.....	28

Figura 30. Diagrama de actividad para buscar artículo.....	29
Figura 31. Diagrama de actividad para solicitar pedido.....	29
Figura 32. Diagrama de actividad para entregar pedido.....	30
Figura 33. Diagrama de actividad para hacer pedido faltante.....	30
Figura 34. Diagrama de actividad para ingresar artículo faltante.....	31
Figura 35. Diagrama de actividad para buscar artículo.....	31
Figura 36. Diagrama de actividad para agregar artículo.....	32
Figura 37. Diagrama de actividad para actualizar datos y stock.....	33
Figura 38. Diagrama de actividad para buscar artículo.....	33
Figura 39. Diagrama de actividad para solicitar informe actual.....	34
Figura 40. Diagrama de actividad para generar reportes.....	34
Figura 41. Diagrama de secuencia para iniciar sesión.....	35
Figura 42. Diagrama de secuencia para validar datos.....	35
Figura 43. Diagrama de secuencia para buscar artículo.....	36
Figura 44. Diagrama de secuencia para solicitar pedido.....	36
Figura 45. Diagrama de secuencia para entregar pedido.....	37
Figura 46. Diagrama de secuencia para hacer pedido faltante.....	37
Figura 47. Diagrama de secuencia para ingresar artículo faltante.....	38
Figura 48. Diagrama de secuencia para buscar artículo.....	38
Figura 49. Diagrama de secuencia para agregar artículo.....	39
Figura 50. Diagrama de secuencia para actualizar datos y stock.....	39
Figura 51. Diagrama de secuencia para buscar artículo.....	40
Figura 52. Diagrama de secuencia para solicitar informe actual.....	40
Figura 53. Diagrama de secuencia para generar reportes.....	41
Figura 54. Diagrama de colaboración para iniciar sesión.....	41
Figura 55. Diagrama de colaboración para validar datos.....	42
Figura 56. Diagrama de colaboración para buscar artículo.....	42
Figura 57. Diagrama de colaboración para solicitar pedido.....	42
Figura 58. Diagrama de colaboración para entregar pedido.....	43
Figura 59. Diagrama de colaboración para hacer pedido faltante.....	43
Figura 60. Diagrama de colaboración para ingresar artículo faltante.....	43

Figura 61. Diagrama de colaboración para buscar artículo.....	44
Figura 62. Diagrama de colaboración para agregar artículo.....	44
Figura 63. Diagrama de colaboración para actualizar datos y stock.....	44
Figura 64. Diagrama de colaboración para buscar artículo.....	45
Figura 65. Diagrama de colaboración para solicitar informe actual.....	45
Figura 66. Diagrama de colaboración para generar reportes.....	45
Figura 67. Diagrama de estado para pedido de artículo.....	46
Figura 68. Diagrama de estado para búsqueda de datos.....	46
Figura 69. Diagrama de estado para registro de datos.....	47
Figura 70. Diagrama de estado para informes.....	47
Figura 71. Diagrama de paquete para la gestión de almacén.....	48
Figura 72. Diagrama general de clases para la gestión del almacén.....	48

RESUMEN

En el desarrollo de mi trabajo, con el título “Proceso Unificado de Desarrollo de Software RUP”, donde se desarrolló para la empresa NCK Ingenieros E.I.R.L. en el área de almacén; primeramente se ha enfocado en el levantamiento de la información, requerimientos y necesidades, el proceso que se hizo, fue a través de visitas en situ, haciendo preguntas al personal sobre el manejo de dicha área.

Posteriormente a través del análisis a más detallado, se pudo conocer sus actividades principales y cuál era el proceso de las operaciones en la oficina, se logró concretar los tipos de caso de uso más comunes entre el encargado del almacén y el personal operativo, y todo esto se ha representado en los diagramas y figuras de esta metodología RUP.

Este nuevo modelamiento de proyecto será referencial para su futura implementación a través del código y programación.

Agradeciendo todo el apoyo y las facilidades de acceso, por parte de la gerencia de la empresa, donde se ha podido cumplir con los objetivos del presente trabajo.

Palabras clave: Aplicación de metodología RUP, proceso unificado de software, análisis y diseño, área de almacén, empresa NCK Ingenieros E.I.R.L.

ABSTRACT

In the development of my work, with the title “RUP software Development Unified Process”, where it was developed for the company NCK Ingenieros E.I.R.L. in the warehouse area; first of all, it has focused on the gathering of information, requirements and needs, the process that was carried out was through on-site visits, asking questions of the personnel about the management of said area.

Later, through a more detailed analysis, it was possible to know their main activities and what was the process of operations in the office, it was possible to specify the most common types of use case between the warehouse manager and the operational staff, and all this has been represented in the diagrams and figures of this RUP methodology.

This new project modeling will be a reference for its future implementation through code and programming.

Thanking all the support and access facilities, from the company’s management, where it has been possible to meet the objectives of this work.

Keywords: Application of RUP methodology, unified software process, analysis and design, warehouse area, company NCK Ingenieros E.I.R.L.

CAPÍTULO I

INTRODUCCIÓN

Desde tiempos remotos, el ser humano ha basado su aprendizaje mediante la practica constante y la perseverancia; todo este trabajo se da, al momento de *repetir sus procesos o procedimientos van mejorando sus métodos de trabajo, y consecutivamente se van perfeccionando*; partiendo de estos conceptos y enseñanzas, se ha desarrollado los conceptos fundamentales de la metodología “proceso unificado de software (RUP)”.

Para la aplicación de esta metodología, lo trataremos dentro de una organización o empresa tercerizadora de servicios NCK INGENIEROS E.I.R.L. en el área de almacenes; donde existen diferentes tipos de actividades, como los ingresos y salidas de materiales, herramientas y maquinarias; se generan pedidos de abastecimiento, manejo de stock de materiales, se requiere información técnica de maquinaria y equipo para realizar algún tipo de operación; como se puede ver estas son las actividades más importantes dentro del almacén, y toda la información actual se relaciona directamente con la gerencia y administracion de forma semanalmente y/o mensual, se hace la presentación de informes o reportes de los activos fijos y los stock de los materiales.

En el contexto teórico y definiciones, el proceso unificado de software (RUP) y la poderosa herramienta UML para el modelado, donde nos apoyaremos para efectuar el análisis y diseño que usaremos a las notaciones, los gráficos y casos de usos para las actividades comunes, diagramas de interacción para dar con la secuencia y movimiento de las acciones de los actores principales, se representará en el diseño los diagramas de clases e identificar los objetos. Todo este trabajo nos ayudará a mejorar y ver posibles problemas que el futuro se pudiera generar.

Es muy necesario contar con algún tipo de sistema de información para la empresa NCK INGENIEROS E.I.R.L. para el área de almacén, ya que es un sector empresarial que viene en crecimiento y cada vez sus operaciones de trabajo vienen teniendo demoras por falta de integridad de la información, en la gestión y la toma de decisiones.

CAPÍTULO II

OBJETIVOS

2.1. Objetivo General

Desarrollar el análisis y diseño con la metodología RUP en el área de almacenes para la gestión de una empresa tercerizadora de servicios NCK INGENIEROS E.I.R.L.

2.2. Objetivos específicos

Desarrollar los conceptos fundamentales de la metodología RUP y UML.

Aplicar el análisis y diseño a través de la metodología RUP para las actividades que desempeñan en el entorno de trabajo de la empresa contratista NCK INGENIEROS E.I.R.L.

CAPÍTULO III

DESARROLLO DEL TEMA

3.1. Marco teórico

3.1.1. Proceso unificado de desarrollo de software (RUP)

3.1.1.1. Inicios.

En sus comienzos, esta metodología de Proceso Unificado Racional (RUP) se da con el antecedente más importante, que se ubica en 1967 con la metodología de Erickson (Ericsson Approach) elaborada por Ivar Jacobson, una aproximación de desarrollo basado en componentes, que introdujo el concepto de caso de uso. Entre los años de 1987 a 1995 Ivar Jacobson fundó la compañía Objectory AB y lanza el proceso de desarrollo Objectory (Jacobson, Booch y Rumbaugh, 2000).

Posteriormente en 1995 Rational Software Corporation adquiere entre 1995 y 1997 y se desarrolla Rational Objectory Process (ROP) a partir de Objectory 3.8 y el Enfoque Racional (Rational Approach) adoptando UML como lenguaje de modelado. Desde ese entonces, y a la cabeza de Grady Booch, Ivar Jacobson y Rumbaugh, Rational Software desarrolló e incorporó diversos elementos para expandir ROP, destacándose especialmente el flujo de trabajo conocido como modelado del negocio. En junio de 1998 se lanza Rational Unified Process como una nueva versión del producto (Jacobson et al., 2000).

El cambio de nombre refleja de hecho, de que la unificación ha tenido lugar en muchas dimensiones, la unificación de técnicas de desarrollo, a través de UML y el trabajo de muchas metodologías, no solo Rational sino también en las oficinas los cientos de clientes que llevan utilizando el proceso por muchos años (Jacobson et al., 2000).

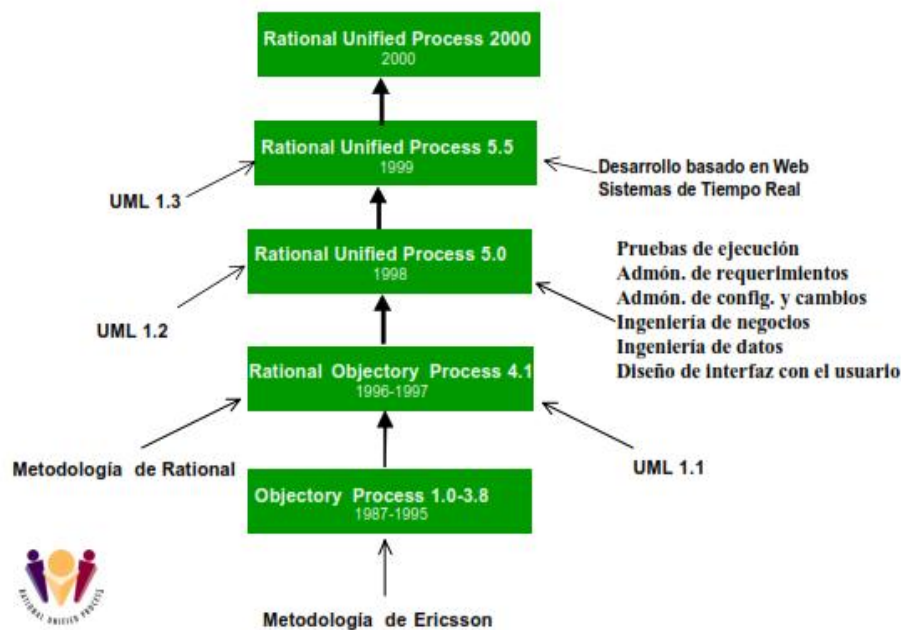


Figura 1. Inicios de RUP
 Fuente: Jacobson, Booch y Rumbaugh, 2000

3.1.1.2. Concepto General.

Dentro las definiciones que se le puede atribuir al *proceso unificado de desarrollo de software* es un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo, es más que un simple proceso, ese un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto (Jacobson et al., 2000).

El proceso unificado está basado en componentes, utiliza el lenguaje unificado de modelado (UML) para preparar todos los esquemas de un sistema de software. De hecho UML, es una parte esencial del proceso unificado (Jacobson et al., 2000).

3.1.1.3. Características.

(Jacobson et al., 2000) Los autores de RUP que tiene 3 características esenciales:

- Guiado por casos de uso: Los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.
- Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.

Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

3.1.1.4. Ventajas.

(Jacobson et al., 2000) Se ha podido identificar las ventajas más sobresalientes con la aplicación de esta metodología, las cuales son:

- Permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de los riesgos.
- Permite la producción de software que cumpla con las necesidades de los usuarios, a través de la especificación de los requisitos, con una agenda y costo predecible.
- Enriquece la productividad en equipo y proporciona prácticas óptimas de software a todos sus miembros.

- Permite llevar a cabo el proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades críticas.
- Unifica todo el equipo de desarrollo de software y mejora la comunicación al brindar a cada miembro del mismo una base de conocimientos un lenguaje de modelado y un punto de vista de cómo desarrollar software.
- Esta es una metodología completa en sí misma, con un énfasis en la documentación precisa.
- El tiempo de desarrollo menor debido a la reutilización de componentes.
- Aumento de la productividad, en la exactitud en la entrada y el entendimiento de los datos.
- Disminución del tiempo y costo de formación y entrenamiento, se reduce los costos de desarrollo y mantenimiento.

3.1.1.5. Desventajas.

(Jacobson et al., 2000) También se ha evidenciado que hay ciertas limitaciones o desventajas al momento de utilizar este proceso unificado y son las siguientes:

- Los miembros del equipo deben ser expertos en su campo para desarrollar un software bajo esta metodología.
- El proceso de desarrollo es demasiado complejo y desorganizado.
- En la reducción de los proyectos de vanguardia que utilizan las nuevas tecnologías, la reutilización de componentes no será posible.
- La integración en el proceso de desarrollo de software, en teoría parece una buena cosa. Pero en particular los grandes proyectos de desarrollo con flujos múltiples solo servirán para aumentar la confusión y causar más problema durante las etapas de prueba.

- Es una metodología muy rígida y compleja, ya que se necesita contar con personal altamente especializado y con conocimiento para su elaboración y construcción en las diferentes etapas.

En esta metodología se toma mucho tiempo para su desarrollo e implementación en comparación que otras metodologías.

3.1.2. Lenguaje de modelado unificado (UML)

3.1.2.1. Origen.

El UML es la creación de Grady Booch, James Rumbaugh e Ivar Jacobson. Estos caballeros, apodados recientemente “los tres amigos”, trabajaban en empresas distintas durante la década de los años ochenta y principios de los noventa y cada uno diseñó su propia metodología para el análisis y diseño orientado a objetos.

Sus metodologías pre-dominaron sobre las de sus competidores. A mediados de los años noventa empezaron a intercambiar ideas entre si y decidieron desarrollar su trabajo en conjunto (Schmuller, 2001).

En 1994, Rumbaugh ingresó a Rational Software Corporation, donde ya trabajaba Booch. Jacobson ingresó a Rational un año después, y el resto como dicen ya es historia (Schmuller, 2001).

Los anteproyectos del UML empezaron a circular en la industria del software y las reacciones resultantes trajeron consigo considerables modificaciones. Conforme diversos corporativos vieron que el UML era útil a sus propósitos, se conformó un consorcio del UML (Schmuller, 2001).

Entre los miembros se encuentran DEC, Hewlett Packard, Intellicorp, Microsoft, Oracle, Texas Instrumentos y Rational. En 1997 el consorcio produjo la versión

1.0 del UML y lo puso a consideración del OMG (Grupo de administración de objetos) como respuesta a su propuesta para un lenguaje de modelado estándar (Schmuller 2001).

El consorcio aumentó y generó la versión 1.1, misma que se puso nuevamente a consideración del OMO. El grupo adoptó esta versión a finales de 1997. El OMO se encargó de la conservación del UML, y produjo otras dos revisiones en 1998. El UML ha llegado a ser el estándar de facto en la industria del software, y su evolución continua (Schmuller, 2001).

3.1.2.2. Definiciones básicas.

Del acrónimo en inglés UML (Lenguaje de modelado unificado) es una herramienta más emocionante en el mundo actual del desarrollo de sistemas. Esto se debe a que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicárselas a otras personas (Schmuller, 2001).

El UML es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas. Está pensado para usarse en todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar (Booch, Jacobson y Rumbaugh, 2000).

a. Clase.

Este término se le conoce como al conjunto de objetos que están constituidos por sus características y operaciones. Por ejemplo tenemos la clase de equipo electrodoméstico “Lavadora” donde están determinadas por atributos de marca, modelo, número de serie, etc.; y sus acciones vendrían a ser “agregar ropa”, “agregar detergente”, “lavar ropa”, “secar”, entre otras funciones que puedan tener (Schmuller 2001).

Lo podemos denotar gráficamente con un rectángulo subdividido en tres partes:



Figura 2. Modelo de clase

Fuente: Schmuller/Prentice Hall, 2001

Los diagramas de clases son un conjunto que tiene similitudes con el modelo de datos (entidad-relación) recuerde el que las clases no solo muestra la estructura de la información, sino que describen también comportamiento. El propósito es definir una base para otros diagramas (Schmuller, 2001).

b. Objeto.

Un objeto se representa de la misma forma que una clase. En el comportamiento superior aparece el nombre del objeto junto con el nombre de la clase subrayado (Schmuller, 2001).

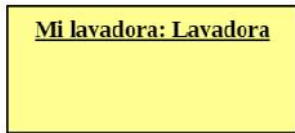


Figura 3. Modelo de objeto
Fuente: Schmuller/Prentice Hall, 2001

c. Caso de uso.

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Esta es una herramienta valiosa, ya que es una técnica de acierto y errores para obtener los requerimientos del sistema desde el punto de vista del usuario. Esto es importante si la finalidad es crear un sistema que pueda ser utilizado por la gente en general (Schmuller, 2001).



Figura 4. Modelo de caso de uso
Fuente: Schmuller/Prentice Hall, 2001

d. Diagrama de caso de uso.

Para los desarrolladores del sistema, esta es una herramienta valiosa, ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema desde el punto de vista del usuario (Schmuller, 2001).

Estos gráficos se puede graficar de la siguiente manera.

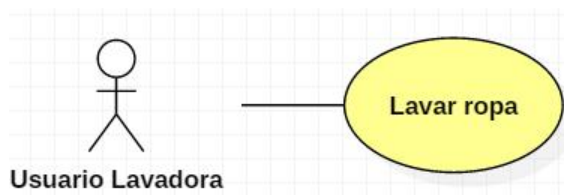


Figura 5. Modelo de diagrama de caso de uso
Fuente: Schmuller/Prentice Hall, 2001

3.1.3. Diagrama de UML

3.1.3.1. Diagrama de estado.

Es típicamente un complemento de la descripción de una clase. Muestra todos los estados posibles que los objetos de la clase pueden tener, y que eventos causan un cambio de estado (Schmuller, 2001).

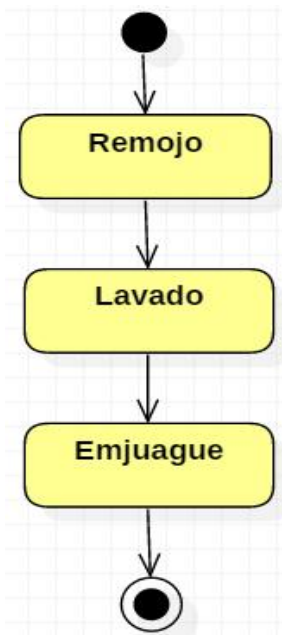


Figura 6. Modelo de diagrama de estado

Fuente: Schmuller/Prentice Hall, 2001

3.1.3.2. Diagrama de actividad.

Este tipo muestra el flujo secuencial de las actividades. El diagrama de actividades es utilizado típicamente para describir las actividades realizadas en una operación, aunque puede ser también utilizado para describir otros diagramas, tal como un caso de uso o una interacción. El diagrama de actividades consiste de estados de acción, los cuales contienen una especificación de la actividad que va a ser realizada (Schmuller 2001).

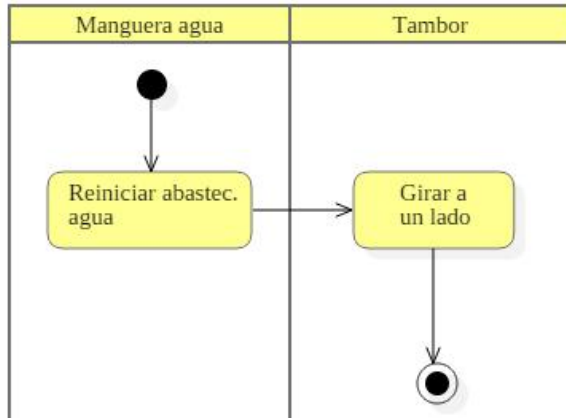


Figura 7. Modelo de diagrama de actividad
Fuente: Schumler/Prentice Hall, 2001

3.1.3.3. Diagrama de secuencia.

Este diagrama muestra una colaboración dinámica entre una serie de objetos. Los diagramas consisten en una serie de objetos mostrados con líneas verticales. El tiempo pasa descendentemente en el diagrama y la muestra un intercambio de mensajes entre los objetos a medida que pasa el tiempo en secuencia o función (Schumler 2001).

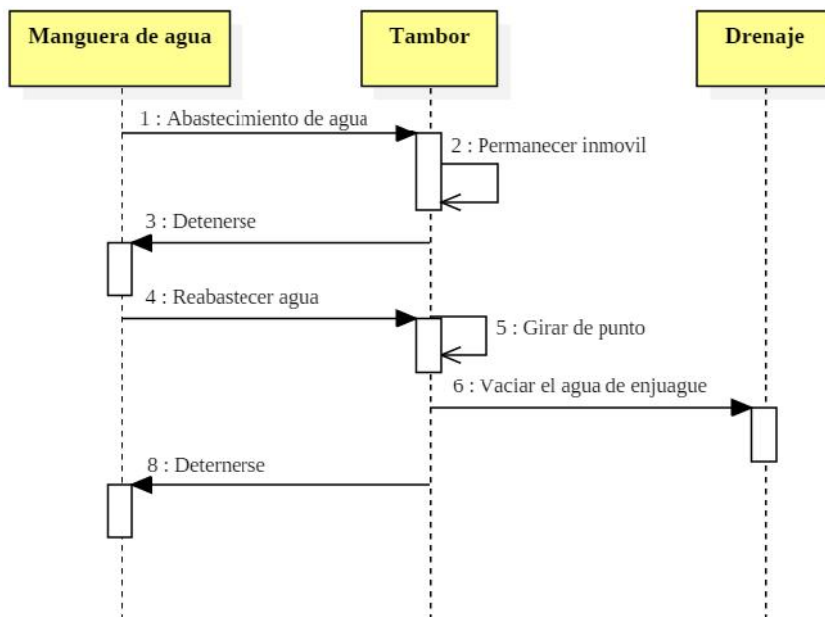


Figura 8. Modelo de diagrama de secuencia
Fuente: Schumler/Prentice Hall, 2001

3.1.3.4. Diagrama de colaboración.

Los elementos de un sistema trabajan en conjunto para cumplir con los objetivos del sistema, y un lenguaje de modelo deberá contar con una forma de representar esto (Schmuller 2001).

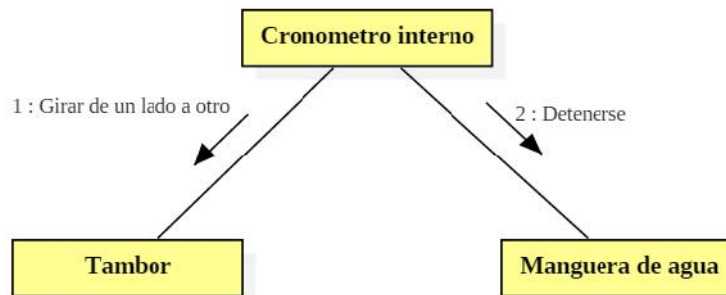


Figura 9. Modelo de diagrama de colaboración
Fuente: Schmuller/Prentice Hall, 2001

3.1.3.5. Diagrama de componente.

Un diagrama de componente muestra la estructura física del código en términos de los componentes de código. Un componente puede ser un componente de código fuente, un componente binario o un componente ejecutable (Schmuller, 2001).

3.1.4. Orientado a objetos

3.1.4.1. Definiciones.

Esta nueva tecnología de la ingeniería de software ha marcado nuevas formas para el análisis y diseño del mundo real, de manera que primero se genera un sistema mediante un conjunto de objetos, luego podrá ampliar el sistema agregándole funcionalidad a los componentes que ya había generado o agregándole nuevos componentes, y finalmente podrá volver a utilizar los objetos que generó para el sistema cuando cree uno nuevo, con lo cual reducirá sustancialmente el tiempo de desarrollo de un sistema (Schmuller 2001).

3.1.4.2. Características.

a. Abstracción.

Al buscar abstracciones es necesario concentrarse en los detalles importantes o incluso en las similitudes ignorando las diferencias por el momento, para poder describir y especificar algunos detalles del sistema, y pasar por desapercibidos, los no relevantes. Un concepto puede ser una abstracción solo si puede ser descrito, entendido y analizado independientemente de los mecanismos utilizados para cumplir sus propósitos.

Una abstracción denota características esenciales de un objeto, distinguibles de otros tipos de objetos y provee límites conceptuales a la perspectiva del observador (Schmuller 2001).

b. Encapsulamiento.

Mientras la abstracción permite pensar el que hacer, la encapsulación se dirige en cómo se va a realizar, permitiendo cambiar programas en forma confiable, con un mínimo de esfuerzo por esta barrera. La abstracción y encapsulación son conceptos complementarios, la abstracción se enfoca en el funcionamiento observable de un objeto, mientras la encapsulación se enfoca en la implementación que da origen a ese funcionamiento (Schmuller 2001).

c. Modularidad.

El segmentar un programa en componentes individuales permite reducir su complejidad, además de agrupar un conjunto de componentes con algún propósito o característica común, permitiendo localizar eficazmente sus componentes. La modularización se realiza para dividir un sistema en módulos compilables separadamente, pero con ciertas conexiones mínimas de dependencias con otros

módulos. La modularidad se encarga de empaquetar abstracciones en unidades discretas, llamadas módulos, es decir, la modularización empaqueta físicamente las clases y objetos del diseño de la estructura lógica.

Al modularizar se aventaja la reducción del costo del software al permitir diseñar y revisar módulos independientes, e incluso reutilizar módulos básicos para cualquier sistema (Schmuller 2001).

d. Herencia.

La herencia es muy importante en la jerarquía de generalización/especialización por definir una relación entre clases que tiene como objetivo en la reutilización y la extensibilidad de sus propios recursos, en donde una clase comparte la estructura o funcionamiento definido en una o más clases (herencia simple y múltiple, respectivamente).

La herencia representa una jerarquía de abstracciones en donde una subclase (clase hijo) hereda de una o más superclases (clases padre). La estructura y funcionamiento común de diferentes clases pueden formar una superclase, así la herencia es una jerarquía de generación/especialización, en donde la superclase representa abstracciones generalizadas y la subclase representa la especialización de métodos de la superclase, los cuales son agregados, modificados y ocultados.

La herencia es uno de los mecanismos más usados, a través de esta característica se pueden crear nuevas clases, jerarquías u objetos, y en relación para cada objeto creado va estar identificado y diferenciado a través de sus atributos y métodos (Schmuller, 2001).

3.2. Caso Práctico

3.2.1. Información de la empresa

- **Razón Social:** NCK INGENIEROS E.I.R.L.

- **Ruc:** 20412979304

- **Ubicación:** Parque Industrial Mz Q Lote 9 y 10 Ilo, Moquegua.

- **Rubro:** Es una empresa dedicada a la fabricación, construcción y mantenimiento de componentes de maquinarias pesadas, trabajos en desmontaje y montaje de todo tipo de estructuras metálicas como chutes, ductos, tanques, escaleras con grating, soportería con barandas, puertas enrollables entre otros, mantenimiento de bombas hidráulicas, microfiltrados, gatas hidráulicas, así como comercialización de herramientas accesorios y equipos para los sectores productivos de minería, pesca, construcción e industria en general.

- **Misión:** Ofrecer servicios del sector industrial, con la asesoría técnica y comercial con el buen manejo de nuestro capital humano y la tecnología, orientándonos siempre a un crecimiento productivo sostenido, reduciendo sus costos de operación e incrementando la disponibilidad operativa de sus equipos mediante un adecuado manejo de operaciones seguras de trabajo.

- **Visión:** Liderar en el campo de ingeniería de mantenimiento en el sector minero e industrial con gran proyección en la región sur del Perú.

3.2.2. Planteamiento del problema

El contexto actual de la empresa contratista se ha podido observar, que no cuenta con un sistema informático que se desempeñe el manejo de la información en forma eficiente y ordenada, tan solo han optado por llevar su información en programa de escritorio en hojas de cálculo y/o office.

Durante las visitas del levantamiento de información, se ha podido visualizar que presenta diferentes problemas por dar una serie de soluciones, en el área de almacén, podemos identificar que no se puede precisar con exactitud la cantidad en stock de materiales, no se sabe cuál es su ubicación de las herramientas y cuantas hay, a veces se necesita información del equipo y maquinaria para realizar algún tipo de servicio a sus nuestros clientes y no se cuenta este tipo información al alcance, o sea no se lleva ningún tipo de control para mejorar la administración de estos materiales y bienes, solo se hace de manera muy empírica y de memoria, uno de los problemas es la falta impartir directivas y políticas de organización en el manejo de los almacenes, y esto se refleja en el desorden del control físico de almacenamiento, cuando la gerencia y administración requiere algún tipo de informe o reporte de los activos fijos y los stocks de los materiales no hay forma de tenerlo listo inmediatamente.

A partir de este análisis referencial, se iniciará con la propuesta del presente trabajo donde se hará el diseño y modelamiento con la asistencia del Proceso Unificado de Rational (RUP) a fin de tener prototipos para su futura implementación como un sistema informático.

3.2.3. Análisis y modelamiento

A través de los conceptos de UML, narrados anteriormente, daremos inicio a la ejecución para cada tipo de diagrama o elemento de diseño, estará basado según los requerimientos o problema propuesto.

3.2.3.1. Diagrama de caso de uso.

Para iniciar con el análisis nos apoyaremos en estos diagramas de caso de uso, nos enfocaremos de manera rápida según las actividades más relevantes en área del almacén de la organización.

a. Gestión de personal usuario.

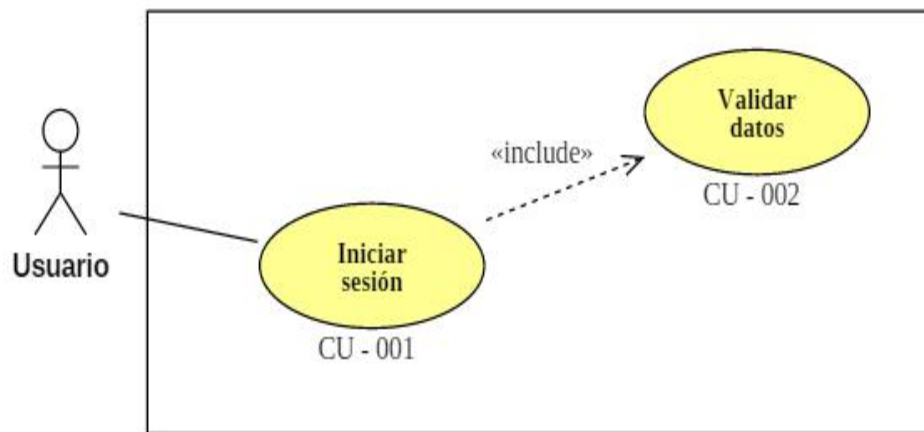


Figura 10. Gestión de personal usuario - diagrama de CU.

CU-001	Iniciar sesión	
VERSIÓN	1.0	
DESCRIPCIÓN	El personal de almacén puede acceder al sistema	
SECUENCIA	PASO	DESCRIPCION
NORMAL	1	Se procede con el ingreso al programa.
	2	En la ventana login se pide su nombre de usuario y contraseña.
	3	El personal de almacén ingresa la información solicitada.
	4	Se procede a validar los campos del login.
		4.1 Si están vacíos, regresa al paso 2.
		4.2 Si están llenos, se va al CU-002.
FRECUENCIA	Diariamente	
IMPORTANCIA	Muy importante	
URGENCIA	Inmediatamente	
COMENTARIOS	Ninguno	

Figura 11. Esquema CU-001 iniciar sesión

CU-002	Validar datos	
VERSIÓN	1.0	
DESCRIPCIÓN	El usuario envía sus datos al sistema para que este los valide.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	Se hace la comprobación de la información ingresada.
		1.1 Si son correctos, da la bienvenida y muestra la ventana principal.
		1.2 Si están errados, regresa al CU-001.
FRECUENCIA	Diariamente.	
IMPORTANCIA	Muy importante.	
URGENCIA	Inmediatamente.	
COMENTARIOS	Ninguno.	

Figura 12. Esquema CU-002 validar datos

b. Gestión de salida pedido de artículo.

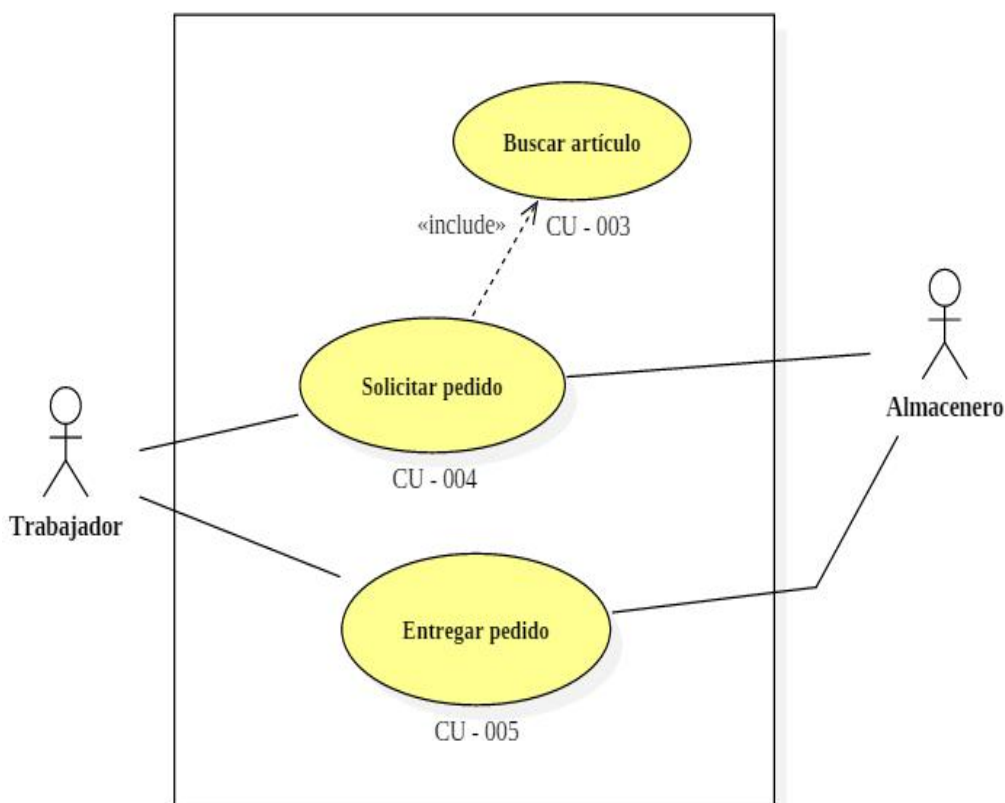


Figura 13. Gestión de salida pedido de artículo – diagrama de CU

CU-003	Buscar artículo	
VERSIÓN	1.0	
DESCRIPCIÓN	El almacenero desea buscar datos sobre la existencia de un determinado artículo.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	El almacenero accede a la ventana de búsqueda.
	2	Mediante el sistema se puede visualizarse el formato de la sección de búsqueda.
	3	Mientras se digita, el sistema se identifica y filtra una muestra de los datos del artículo.
	4	El almacenero selecciona el artículo buscado y nos dirigimos al CU-004.
FRECUENCIA	Diariamente.	
IMPORTANCIA	Muy importante.	
URGENCIA	Inmediatamente.	
COMENTARIOS	Ninguno.	

Figura 14. Esquema CU-003 buscar artículo

CU-004	Solicitar pedido	
VERSIÓN	1.0	
DESCRIPCIÓN	El trabajador requiere un determinado artículo en el área del almacén.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	El trabajador hace un nuevo pedido de artículo.
	2	El almacenero solicita datos.
	3	El trabajador proporciona datos.
	4	El almacenero ingresa datos en el sistema.
	5	El sistema devuelve la información solicitada.
	6	El almacenero guarda los datos del pedido. Termina el proceso.
FRECUENCIA	Diariamente.	
IMPORTANCIA	Muy importante.	
URGENCIA	Inmediatamente.	
COMENTARIOS	Ninguno.	

Figura 15. Esquema CU-004 solicitar pedido

CU-005	Entregar pedido	
VERSIÓN	1.0	
DESCRIPCIÓN	El almacenero realiza la entrega el pedido al trabajador.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	Almacenero ubica en almacén el artículo.
	2	Realiza entrega del pedido.
	3	El trabajador verifica si el pedido es conforme.
		3.1 Si es correcto el pedido “es conforme”.
		3.2 Si no es, se procede a la devolución.
	4	En el sistema se anula el pedido.
FRECUENCIA	Diariamente.	
IMPORTANCIA	Muy importante.	
URGENCIA	Inmediatamente.	
COMENTARIOS	Ninguno.	

Figura 16. Esquema CU-005 entregar pedido

c. Gestión de abastecimiento de artículo.

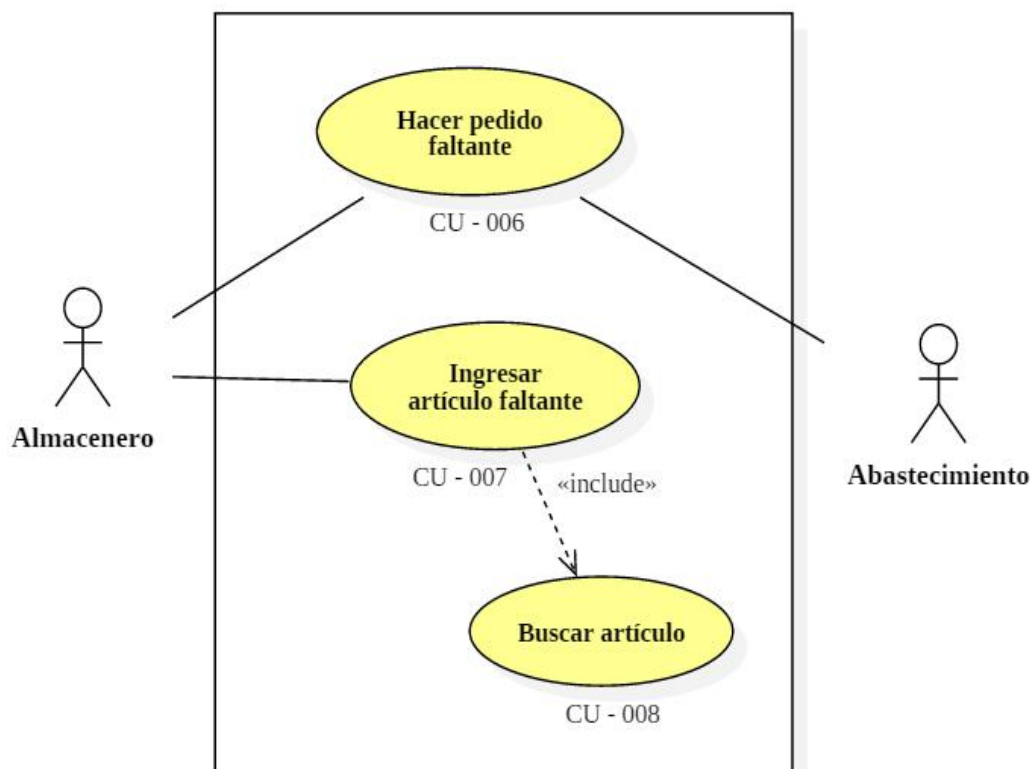


Figura 17. Gestión de abastecimiento de artículo - diagrama de CU

CU-006	Hacer pedido faltante	
VERSIÓN	1.0	
DESCRIPCIÓN	El almacenero solicita el requerimiento de artículos faltantes para el almacén.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	Aperturamos en la ventana de solicitud de artículo faltante.
	2	Seleccionamos y llenado en los campos código, descripción, stock del artículo.
	3	Se genera la solicitud, y enviamos la información al área de abastecimiento. Y concluimos con este procedimiento.
FRECUENCIA	1 a 2 veces por semana.	
IMPORTANCIA	Muy importante.	
URGENCIA	Regular.	
COMENTARIOS	Ninguno.	

Figura 18. Esquema CU-006 hacer pedido faltante

CU-007	Ingresar artículo faltante	
VERSIÓN	1.0	
DESCRIPCIÓN	El almacenero registra el ingreso de un artículo que requiere ser abastecido.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	Primero nos vamos al CU-008. En esta ventana, Mientras escribimos se va filtrando las posibles opciones de búsqueda.
	2	Una vez seleccionado el artículo. Volvemos al formulario presente.
	3	Terminamos con el llenado de los demás datos y guardamos.
FRECUENCIA	1 a 2 vez por semana.	
IMPORTANCIA	Importante.	
URGENCIA	Regular.	
COMENTARIOS	Ninguno.	

Figura 19. Esquema CU-007 ingresar articulo faltante

CU-008	Buscar artículo	
VERSIÓN	1.0	
DESCRIPCIÓN	El almacenero desea buscar datos sobre la existencia del artículo en almacén.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	El almacenero requiere información del artículo, a fin iniciar la búsqueda.
	2	En la ventana de búsqueda, el sistema se visualiza los datos del artículo.
	3	El almacenero mientras digita, el sistema identifica y filtra una muestra los datos.
	4	El almacenero selecciona el artículo buscado y nos dirigimos al CU-007.
FRECUENCIA	1 a 2 veces por semana	
IMPORTANCIA	Importante.	
URGENCIA	Regular.	
COMENTARIOS	Ninguno.	

Figura 20. Esquema CU-008 buscar artículo

d. Gestión de artículo

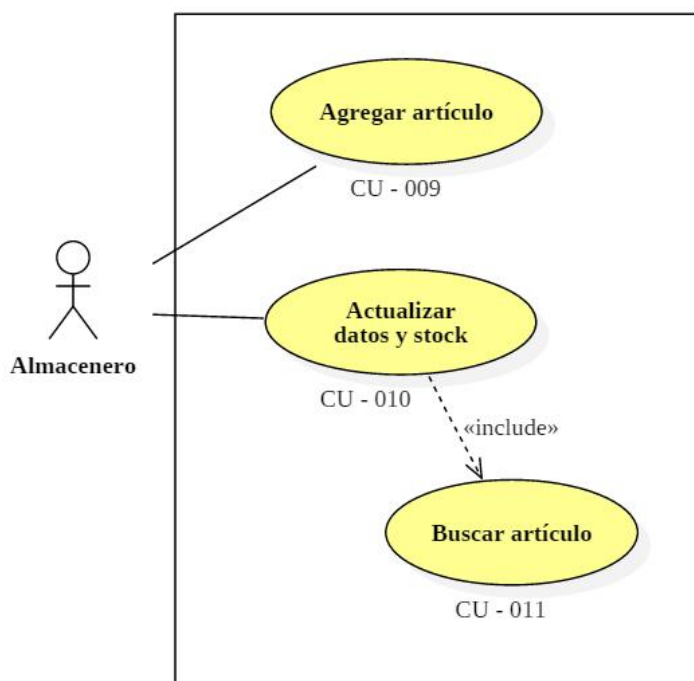


Figura 21. Gestión de artículo - diagrama de CU

CU-009	Agregar artículo	
VERSIÓN	1.0	
DESCRIPCIÓN	El almacenero desea buscar datos sobre la existencia de un determinado artículo.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	Se debe registrar un nuevo artículo en sistema.
	2	El sistema nos muestra la ventana de ingreso del artículo.
	3	El almacenero mientras digita los campos requeridos: código, descripción, tipo, marca y cantidad.
	4	Se verifica que los datos ingresados sean correctos.
	5	Terminando la verificación el sistema procede a guardar el nuevo artículo.
FRECUENCIA	1 a 2 veces por semana.	
IMPORTANCIA	Regular.	
URGENCIA	Regular.	
COMENTARIOS	Ninguno.	

Figura 22. Esquema CU-009 agregar artículo

CU-010	Actualizar datos y stock	
VERSIÓN	1.0	
DESCRIPCIÓN	El almacenero efectúa la actualización y modificación de los datos y el stock de un artículo.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	Se requiere actualizar los datos y stock del artículo.
	2	El sistema envía a otra ventana para iniciar la búsqueda del artículo.
	3	Después de elegir, podemos ver el formulario los datos en forma editable.
	4	El almacenero efectúa los cambios necesarios del artículo.
	5	Se comprueba que la información proporcionada sea correcta.
	6	Una vez concretado, el sistema genera y guarda los cambios.
FRECUENCIA	1 a 2 veces por semana.	
IMPORTANCIA	Regular.	
URGENCIA	Regular.	
COMENTARIOS	Ninguno.	

Figura 23. Esquema de CU-010 actualizar datos y stock

CU-011	Buscar artículo	
VERSIÓN	1.0	
DESCRIPCIÓN	El almacenero desea buscar datos sobre la existencia de un determinado artículo.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	Se pide datos del artículo, solicitado por el almacenero.
	2	Se visualiza el formulario para poder encontrarlo en la base de datos.
	3	El personal del almacén mientras digita, el sistema identifica y filtra una muestra de datos del artículo.
	4	El almacenero selecciona el artículo buscado y nos dirigimos al CU-010.
FRECUENCIA	1 a 2 veces por semana.	
IMPORTANCIA	Importante.	
URGENCIA	Regular.	
COMENTARIOS	Ninguno.	

Figura 24. Esquema de CU-011 buscar artículo

e. Gestión de informes

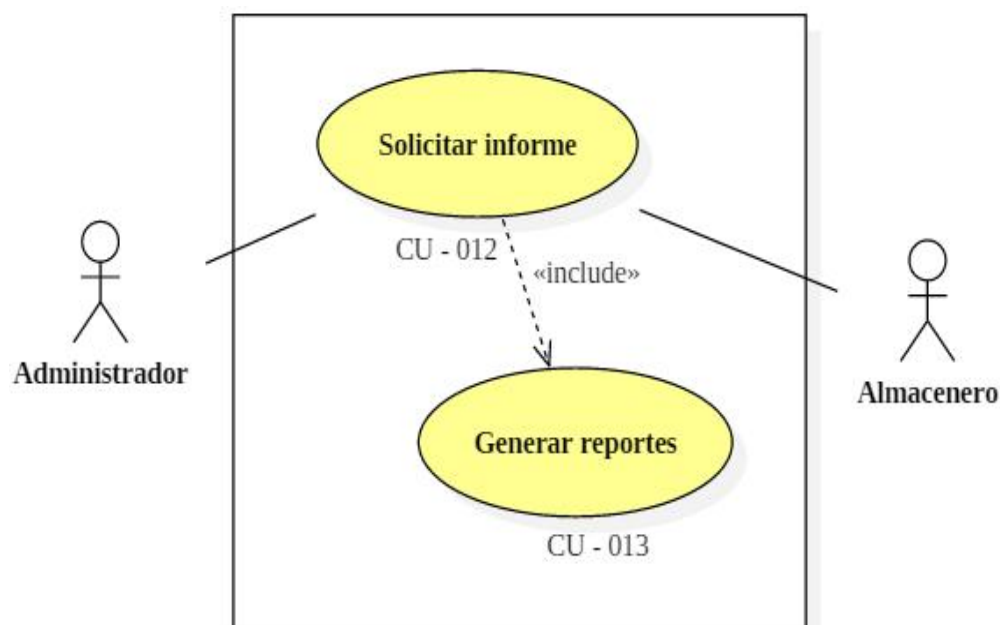


Figura 25. Gestión de informes – diagrama de CU

CU-012	Solicitar informe	
VERSIÓN	1.0	
DESCRIPCIÓN	El área de la administración solicita reportes de forma mensual al área del almacén.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	El administrador solicita un reporte de los movimientos del almacén, tanto salidas como los ingresos.
	2	Nos dirigimos al caso de uso CU-013.
	3	El almacenero verifica los documentos imprimidos.
	4	Entrega la información al administrador.
	5	Administrador recibe la información entregada.
FRECUENCIA	Mensualmente.	
IMPORTANCIA	Muy importante.	
URGENCIA	Inmediatamente.	

Figura 26. Esquema CU-012 solicitar informe

CU-013	Generar reportes	
VERSIÓN	1.0	
DESCRIPCIÓN	El encargado genera en el sistema los movimientos de entrada y salidas efectuadas hasta la actualidad.	
SECUENCIA	PASO	DESCRIPCIÓN
NORMAL	1	Almacenero procede consolidar la información en el sistema.
	2	Efectúa búsquedas por movimientos de salidas e ingresos de artículos.
	3	El sistema muestra y filtra la información, los reportes logran consolidarse.
	4	Se envía a imprimir los reportes solicitados.
FRECUENCIA	Mensualmente.	
IMPORTANCIA	Muy importante.	
URGENCIA	Inmediatamente.	
COMENTARIOS	Ninguno.	

Figura 27. Esquema CU-013 generar reportes

3.3. Representación de resultados

Después de realizar el análisis del planteamiento del caso práctico, podemos obtener los siguientes diagramas en las siguientes figuras.

3.3.1. Diagrama de actividad

A través de estos diagramas se detalla como fluye las actividades más resultantes en el área de almacenes.

3.3.1.1. Iniciar sesión CU-001.

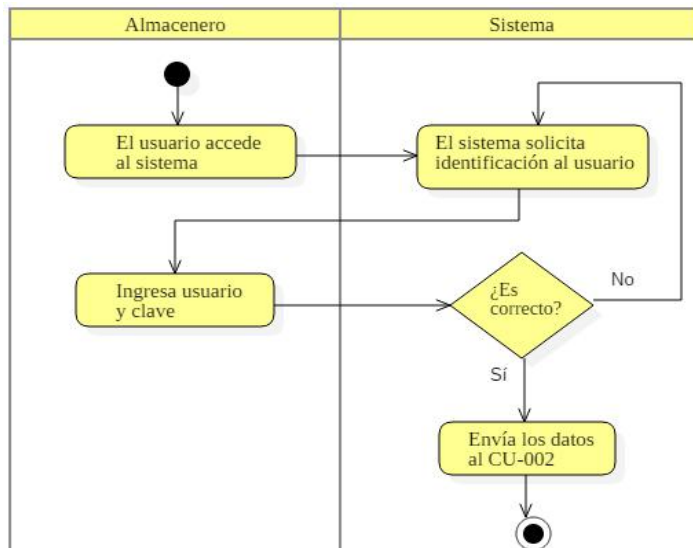


Figura 28. Diagrama de actividad para iniciar sesión

3.3.1.2. Validar datos CU-002.

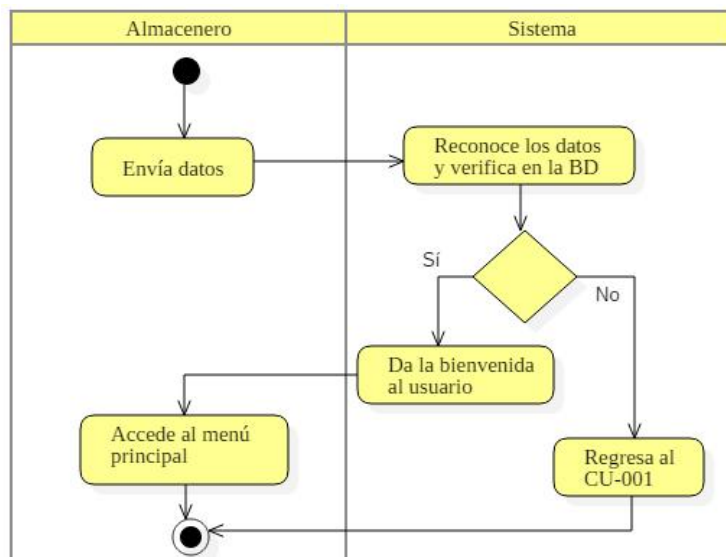


Figura 29. Diagrama de actividad para validar datos

3.3.1.3 Buscar artículo CU-003.

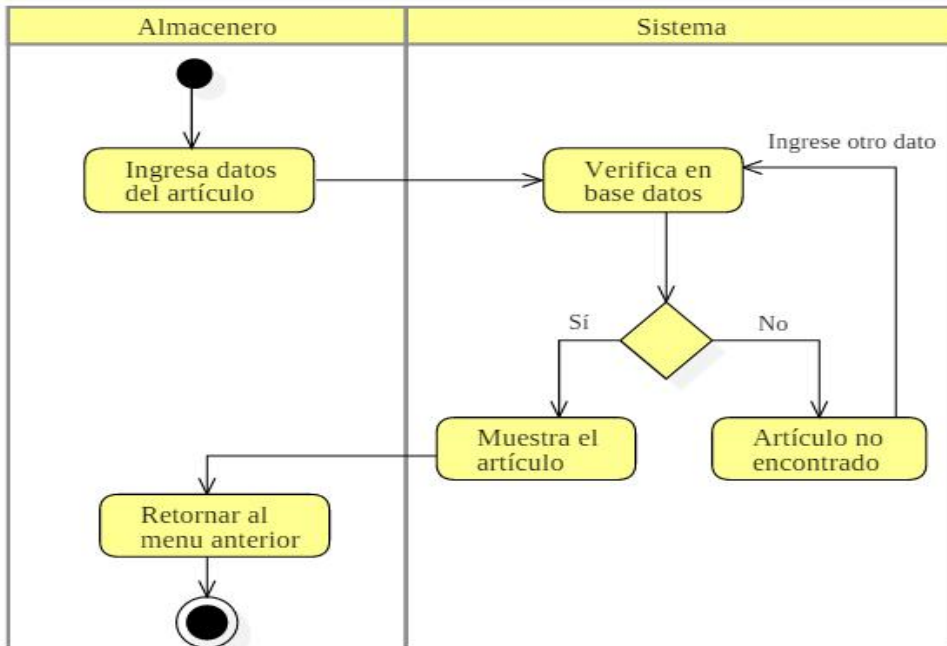


Figura 30. Diagrama de actividad para buscar artículo

3.3.1.4 Solicitar pedido CU-004.

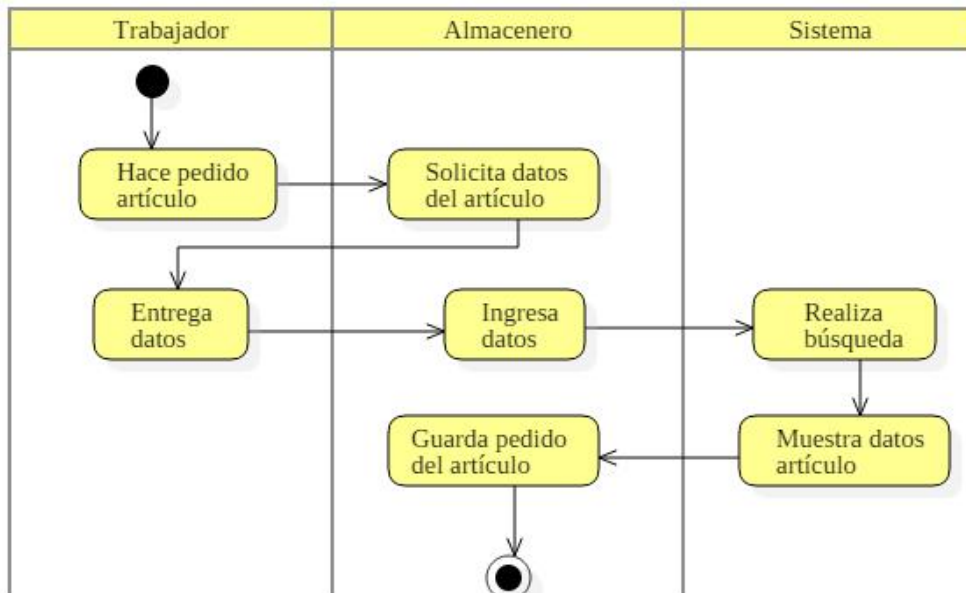


Figura 31. Diagrama de actividad para solicitar pedido

3.3.1.5. Entregar pedido CU-005.

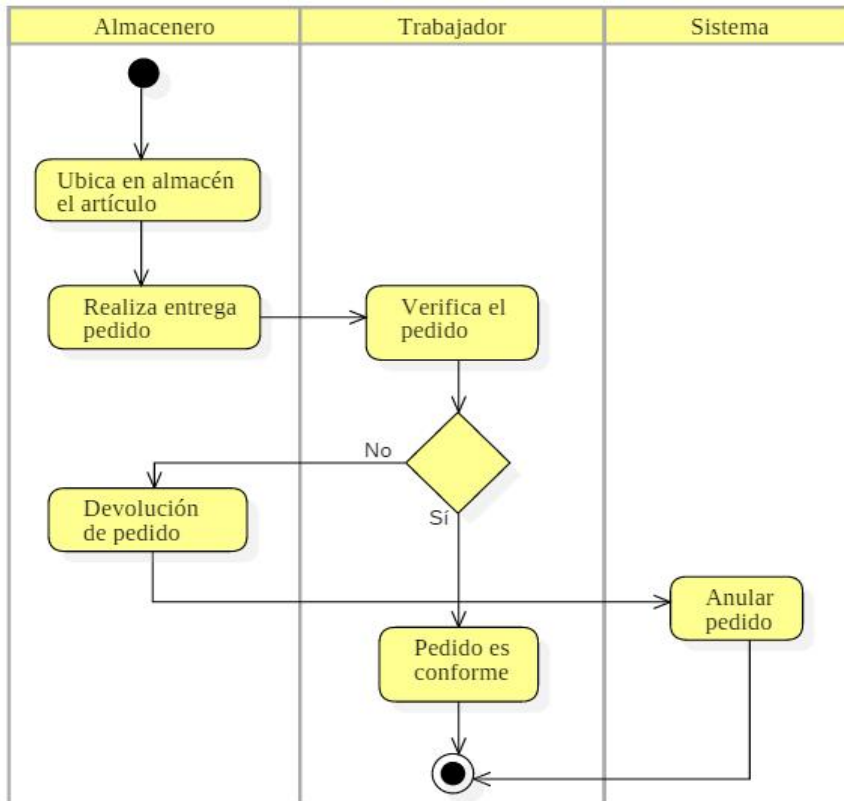


Figura 32. Diagrama de actividad para entregar pedido

3.3.1.6. Hacer pedido faltante CU-006.

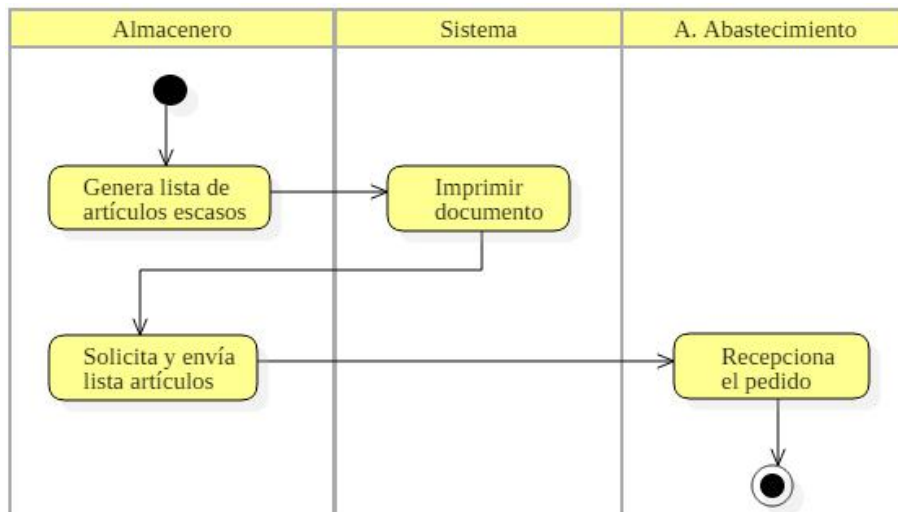


Figura 33. Diagrama de actividad para Hacer pedido faltante

3.3.1.7. Ingresar artículo faltante CU-007.

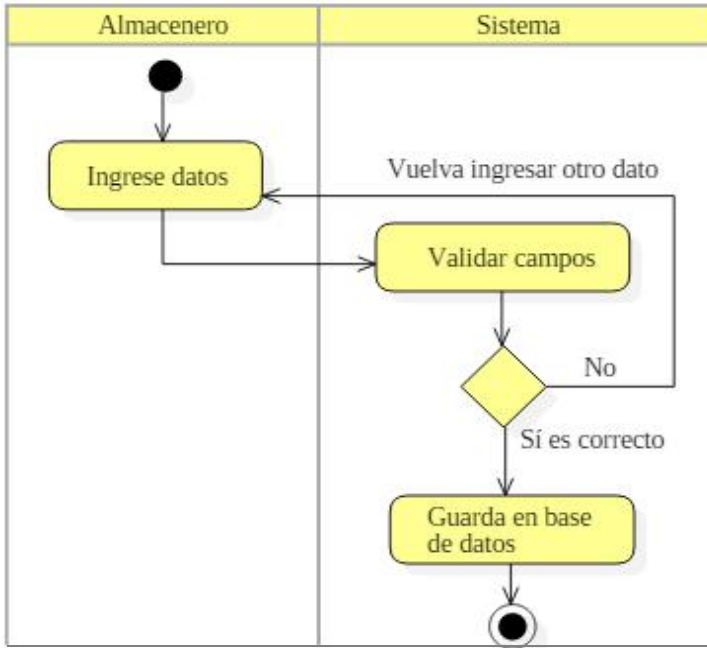


Figura 34. Diagrama de actividad para ingresar artículo faltante

3.3.1.8. Buscar artículo CU-008.

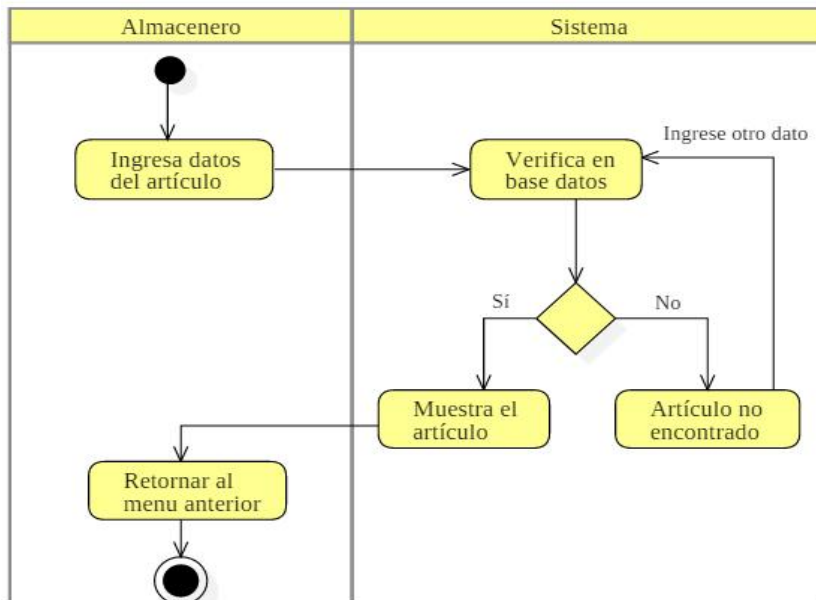


Figura 35. Diagrama de actividad para buscar artículo

3.3.1.9. Agregar artículo CU-009.

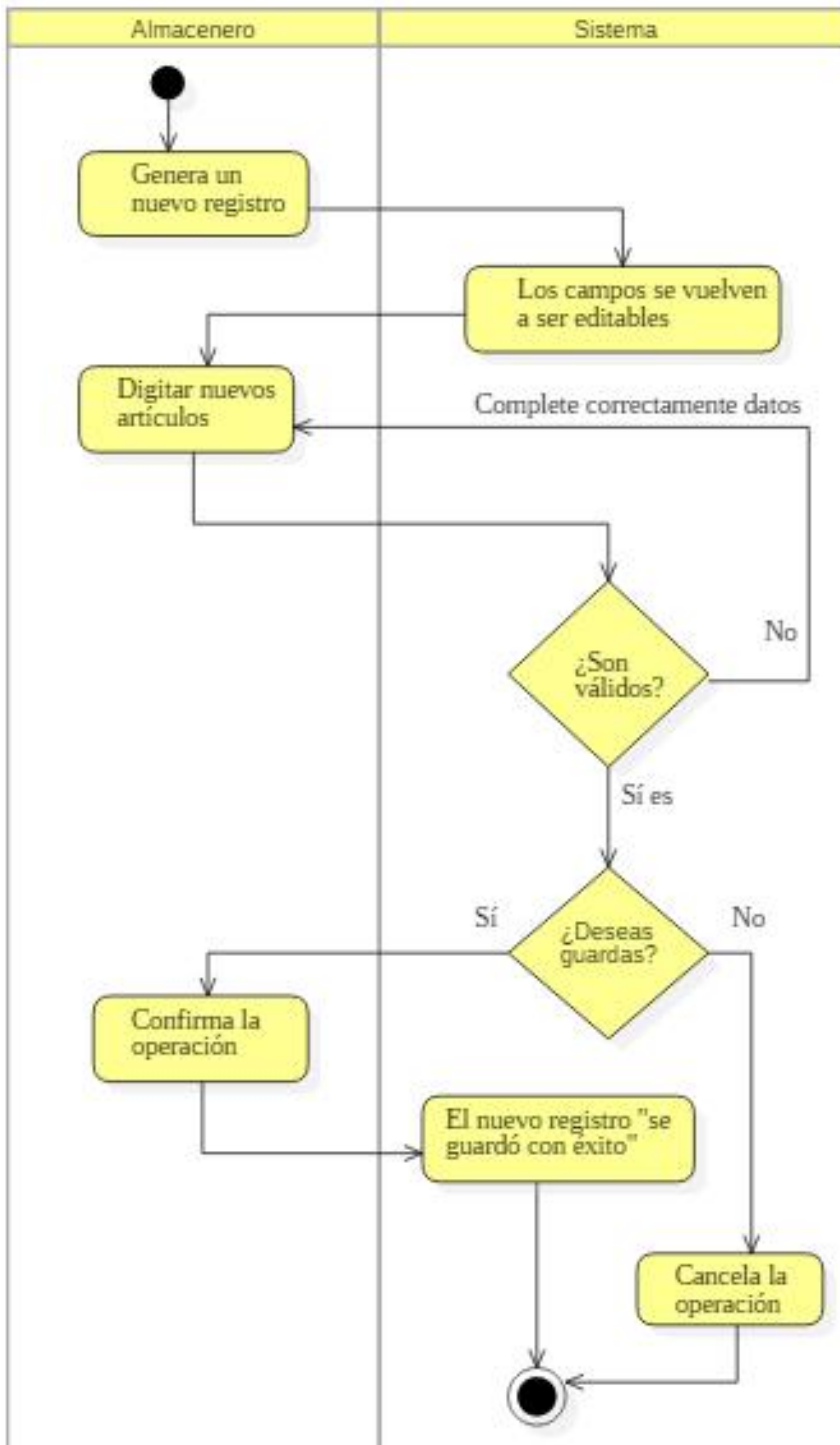


Figura 36. Diagrama de actividad para agregar artículo

3.3.1.10. Actualizar datos y stock CU-010.

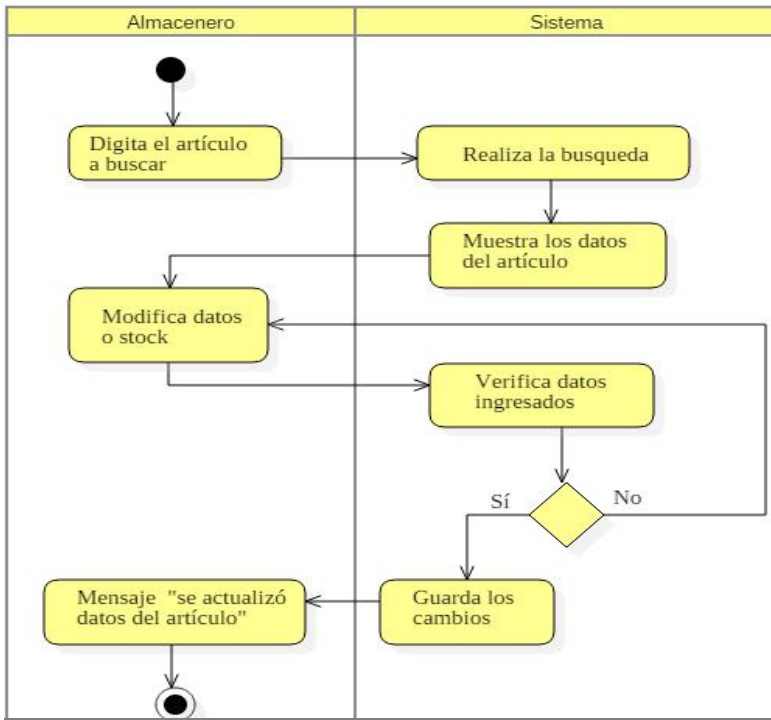


Figura 37. Diagrama de actividad para actualizar datos y stock

3.3.1.11. Buscar artículo CU-011.

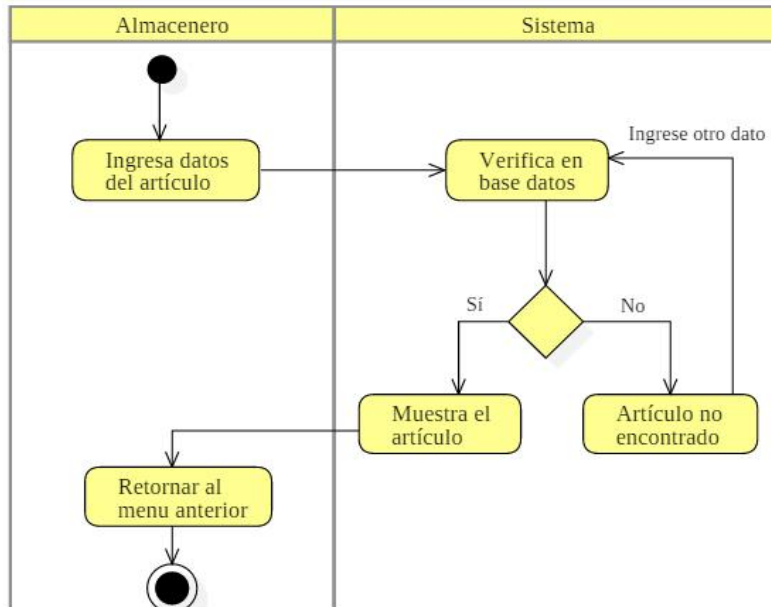


Figura 38. Diagrama de actividad para buscar artículo

3.3.1.12. Solicitar informe actual CU-012.

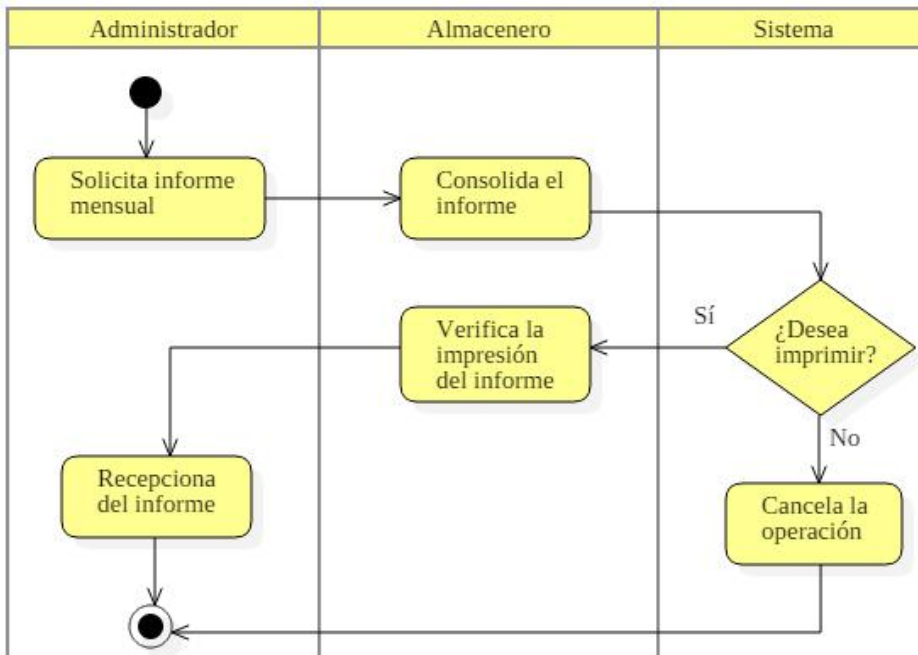


Figura 39. Diagrama de actividad para solicitar informe actual

3.3.1.13. Generar reportes CU-013.

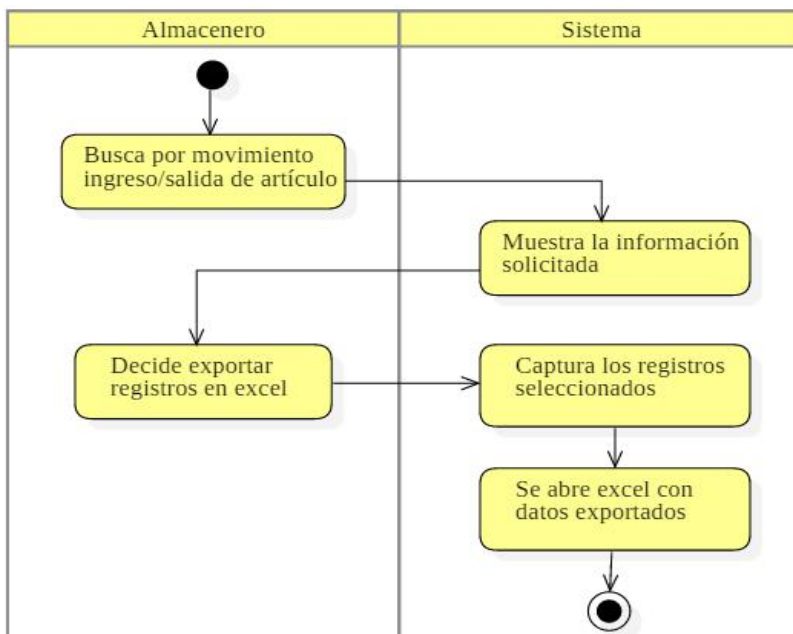


Figura 40. Diagrama de actividad para generar reportes

3.3.2. Diagrama de secuencia

En estos diagramas la interacción de un conjunto de objetos de las acciones de forma cronológica.

3.3.2.1. Iniciar sesión CU-001.



Figura 41. Diagrama de secuencia para iniciar sesión

3.3.2.2. Validar datos CU-002.

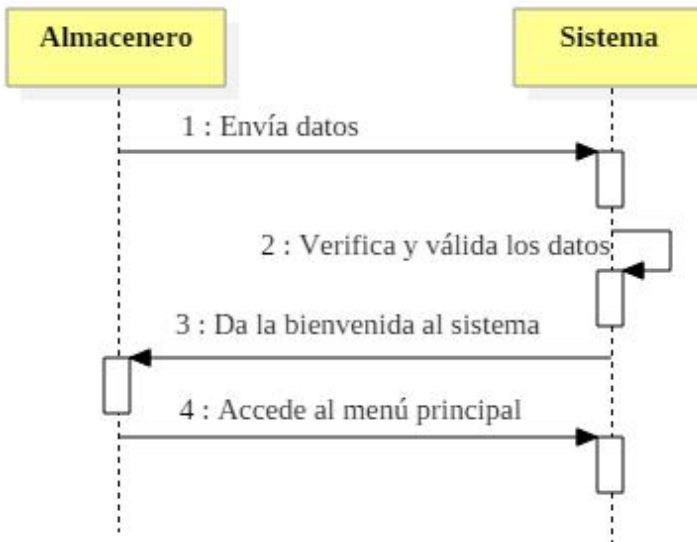


Figura 42. Diagrama de secuencia para validar datos

3.3.2.3. Buscar artículo CU-003.

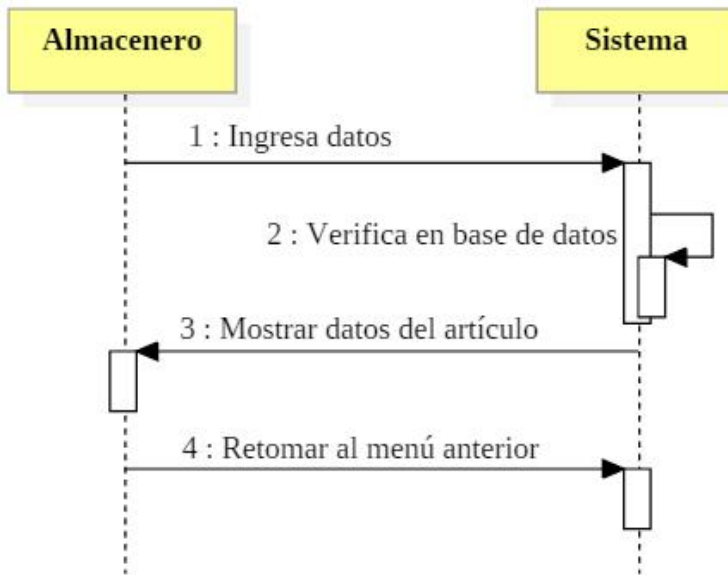


Figura 43. Diagrama de secuencia para buscar artículo

3.3.2.4. Solicitar pedido CU-004.

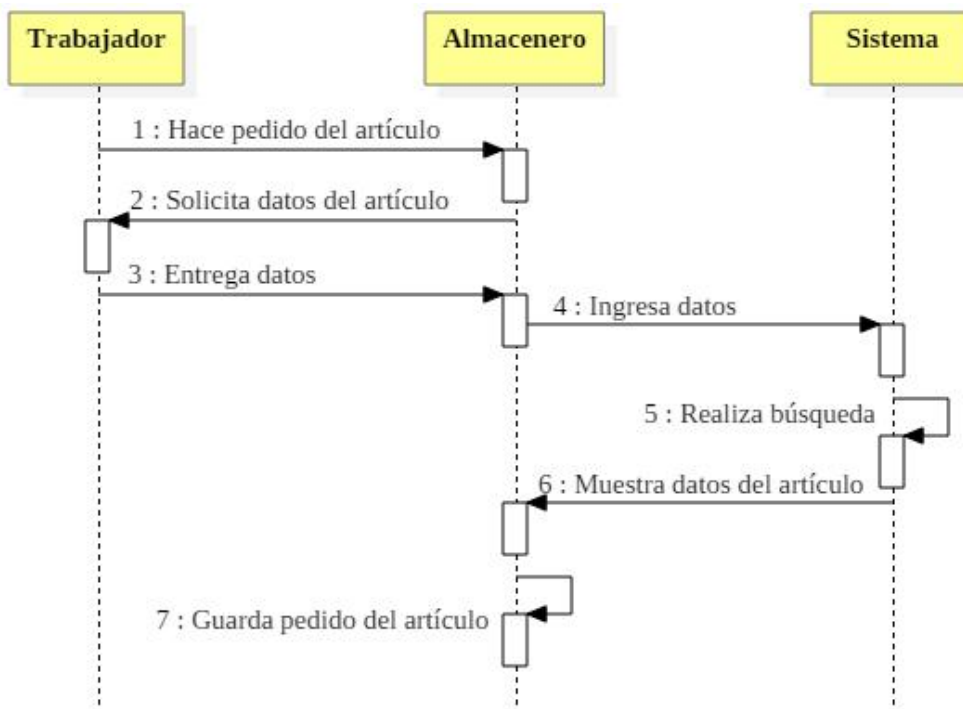


Figura 44. Diagrama de secuencia para solicitar pedido

3.3.2.5. Entregar pedido CU-005.

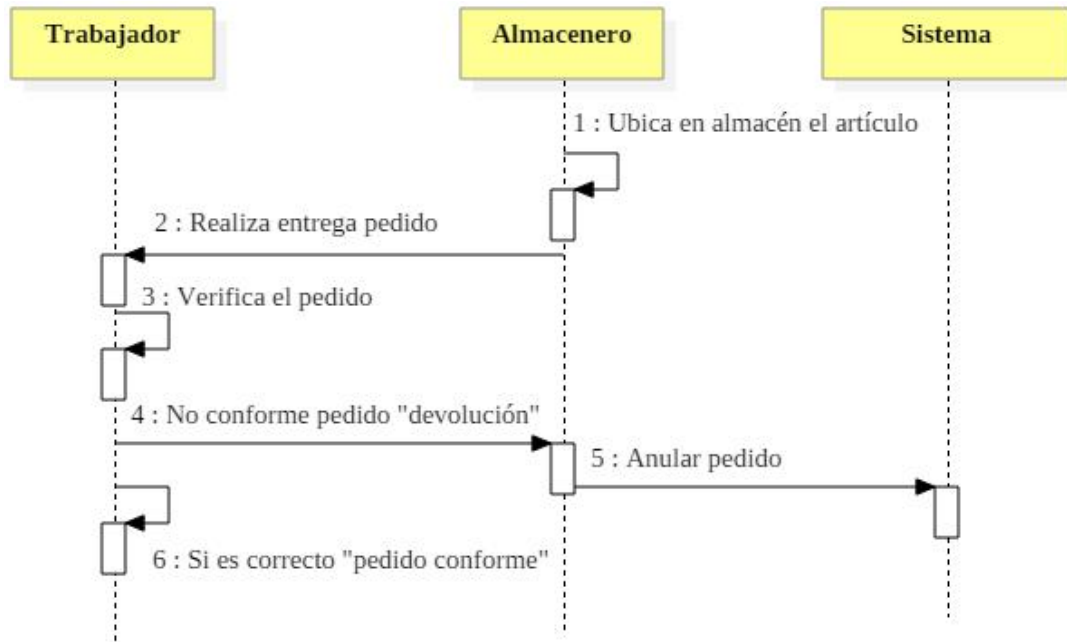


Figura 45. Diagrama de secuencia para entregar pedido

3.3.2.6. Hacer pedido faltante CU-006.

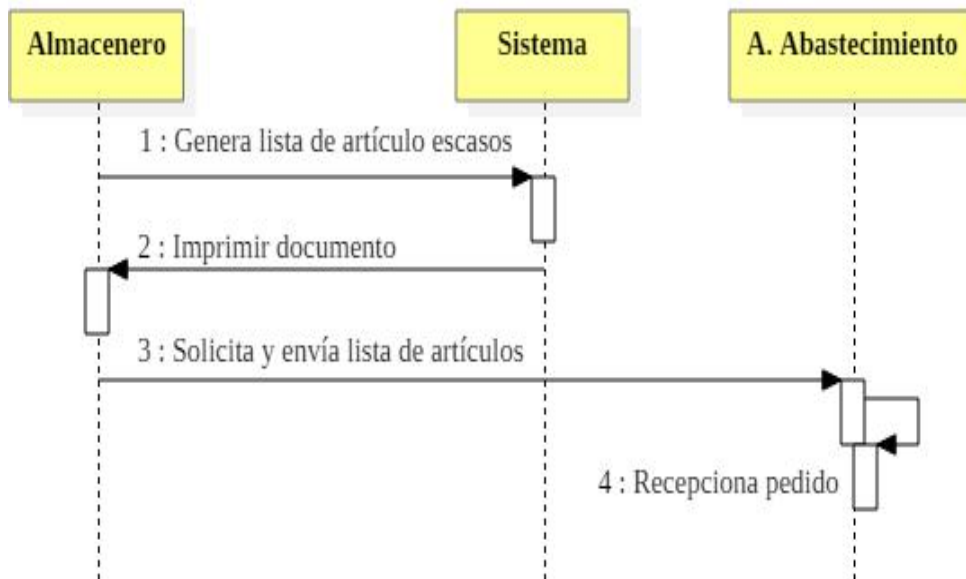


Figura 46. Diagrama de secuencia para hacer pedido faltante

3.3.2.7. Ingresar artículo faltante CU-007.

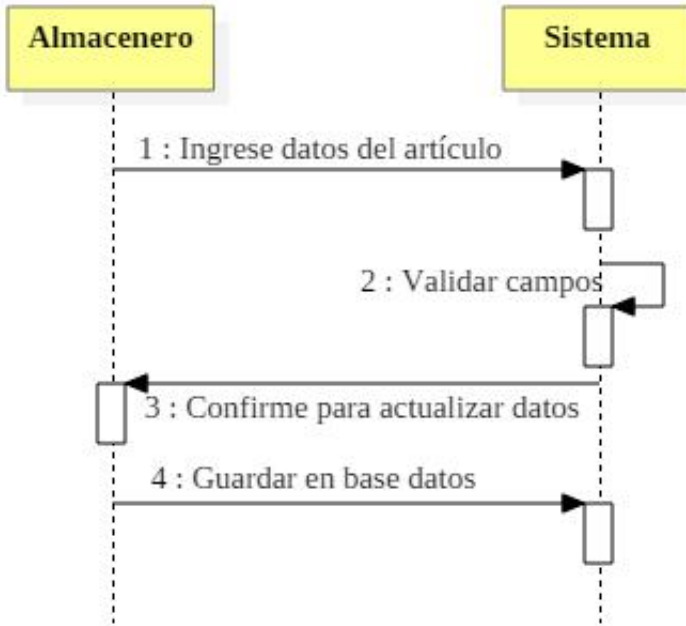


Figura 47. Diagrama de secuencia para ingresar artículo faltante

3.3.2.8. Buscar artículo CU-008.

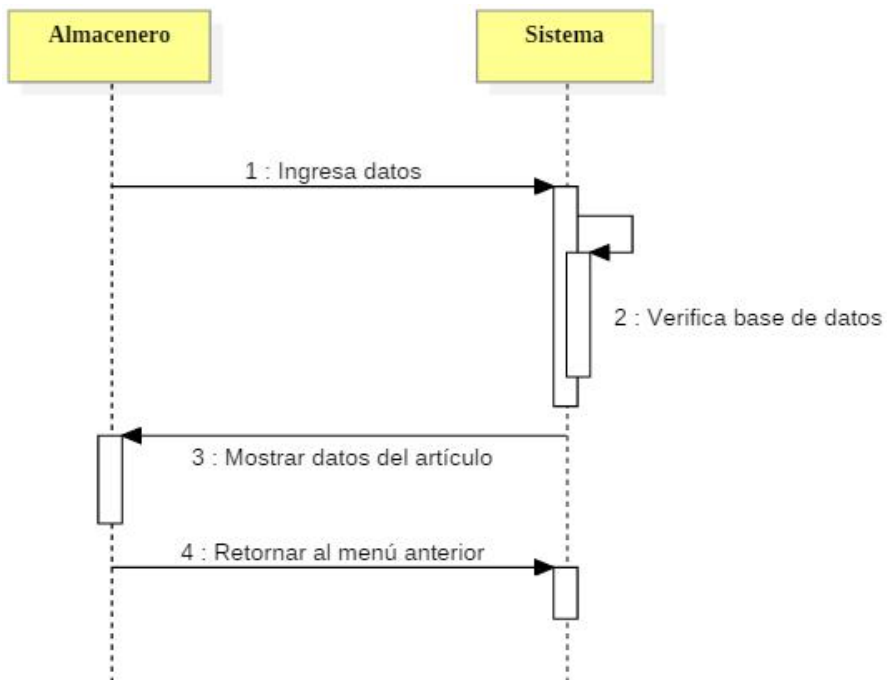


Figura 48. Diagrama de secuencia para buscar artículo

3.3.2.9. Agregar artículo CU-009.



Figura 49. Diagrama de secuencia para agregar artículo

3.3.2.10. Actualizar datos y stock CU-010.

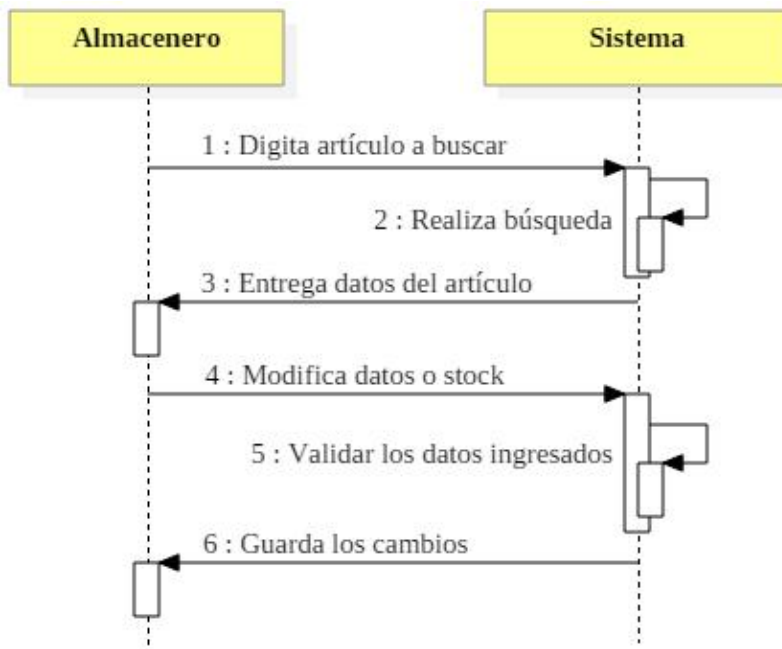


Figura 50. Diagrama de secuencia para actualizar datos y stock

3.3.2.11. Buscar artículo CU-011.

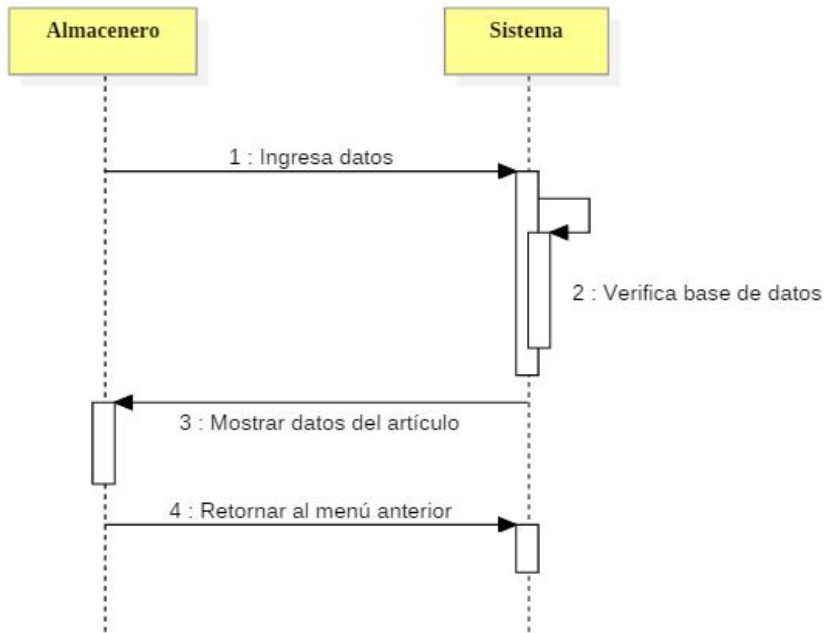


Figura 51. Diagrama de secuencia para buscar artículo

3.3.2.12. Solicitar informe actual CU-012.

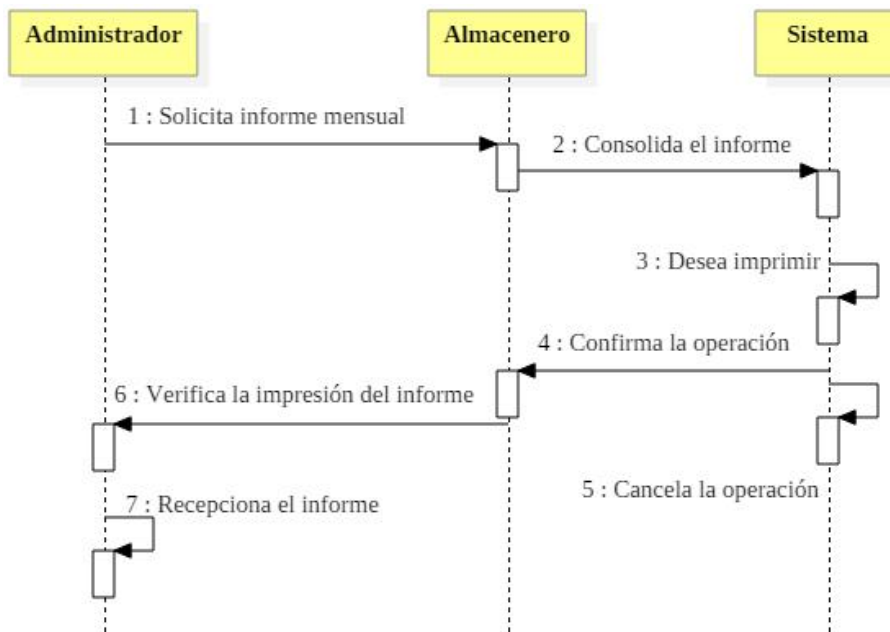


Figura 52. Diagrama de secuencia para solicitar informe actual

3.3.2.13. Generar reportes CU-013.

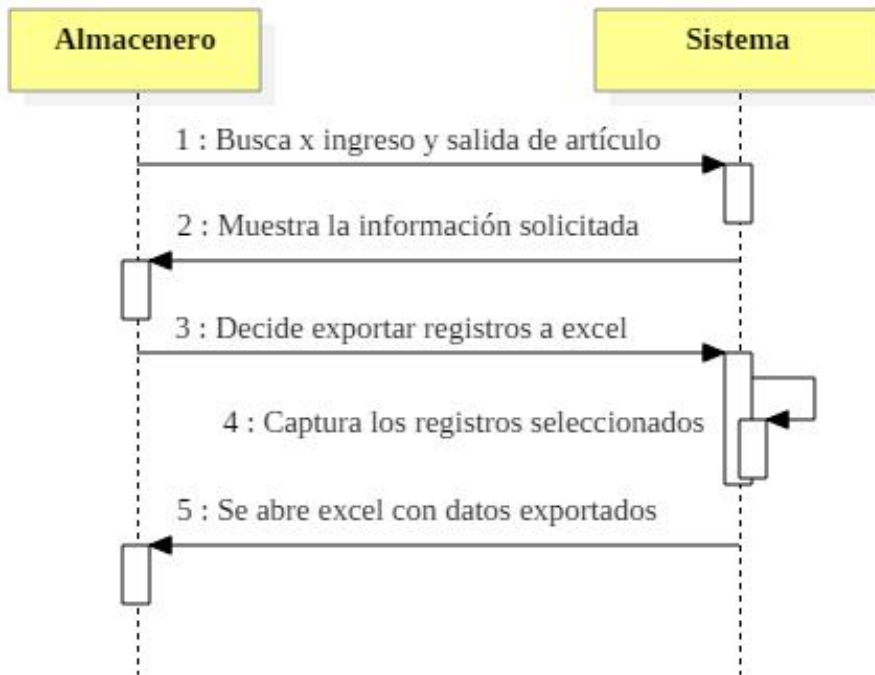


Figura 53. Diagrama de secuencia para generar reportes

3.3.3. Diagrama de colaboración

Por medio de estos diagramas podemos implementar la comunicación de mensajes entre los diferentes actores.

3.3.3.1. Iniciar sesión CU-001.



Figura 54. Diagrama de colaboración para iniciar sesión

3.3.3.2. Validar datos CU-002.

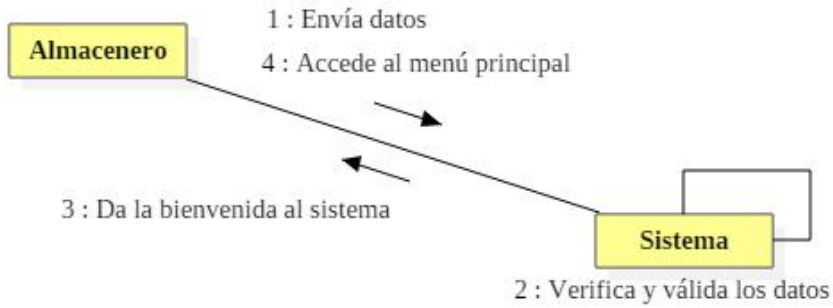


Figura 55. Diagrama de colaboración para validar datos

3.3.3.3. Buscar artículo CU-003.

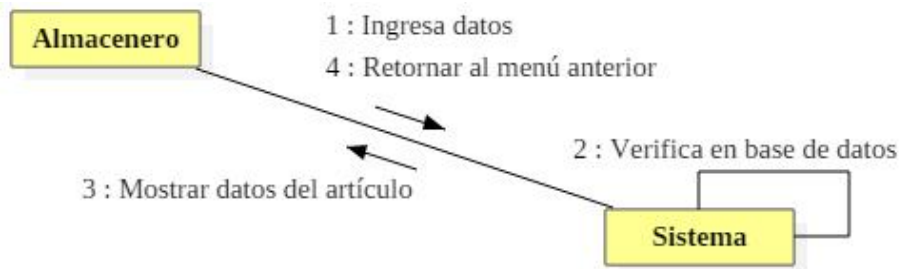


Figura 56. Diagrama de colaboración para buscar artículo

3.3.3.4. Solicitar pedido CU-004.

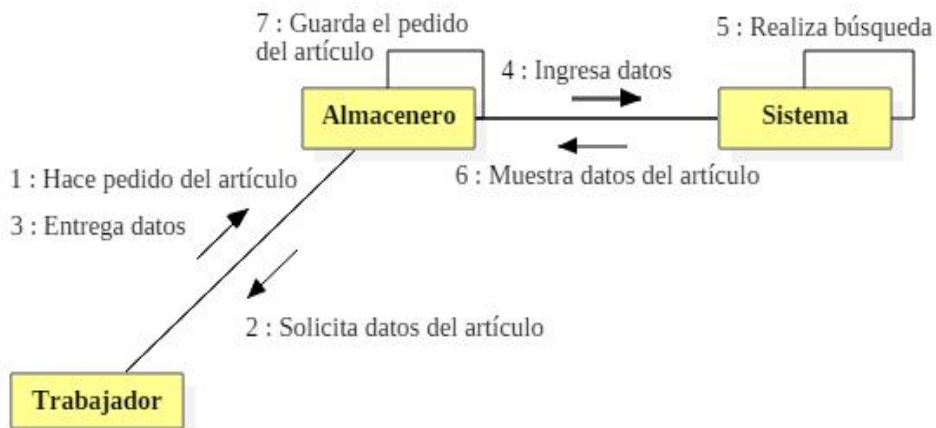


Figura 57. Diagrama de colaboración para solicitar pedido

3.3.3.5. Entregar Pedido CU-005.

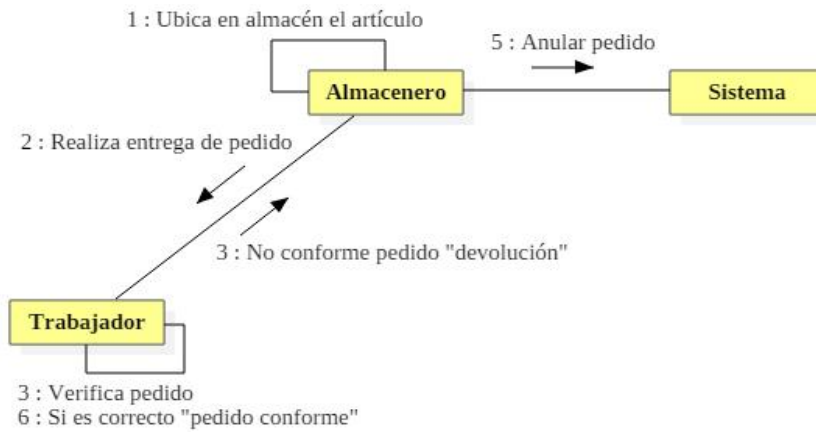


Figura 58. Diagrama de colaboración para entregar pedido

3.3.3.6. Hacer pedido faltante CU-006.

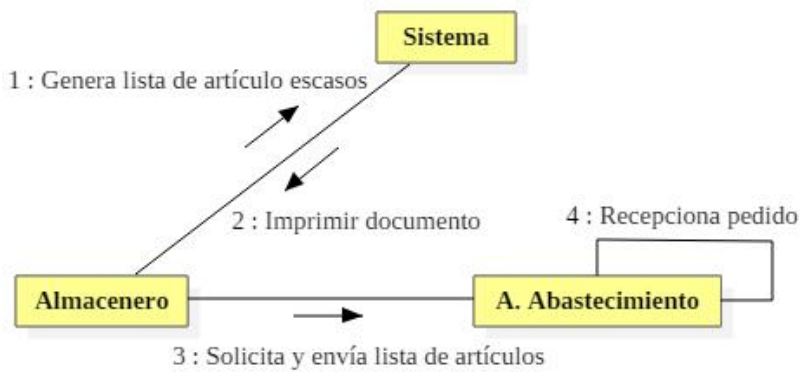


Figura 59. Diagrama de colaboración para hacer pedido faltante

3.3.3.7. Ingresar artículo faltante CU-007.

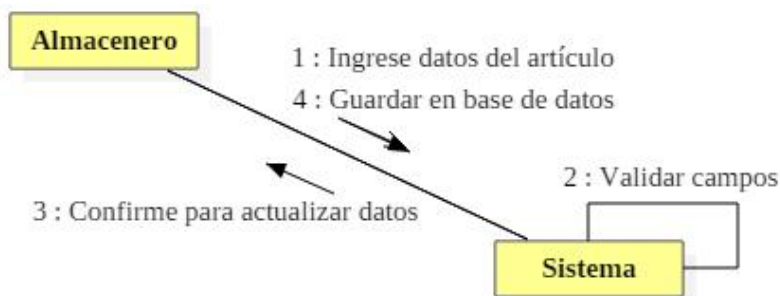


Figura 60. Diagrama de colaboración para ingresar artículo faltante

3.3.3.8. Buscar artículo CU-008.

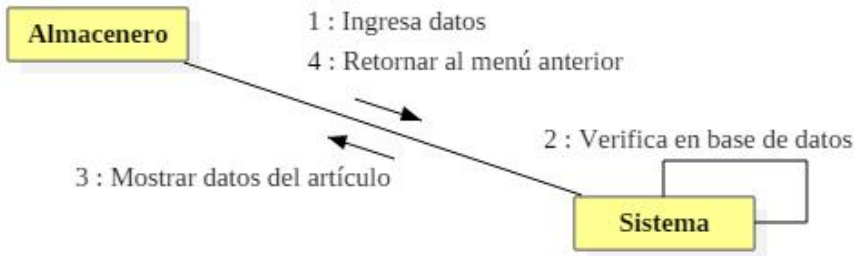


Figura 61. Diagrama de colaboración para buscar artículo

3.3.3.9. Agregar artículo CU-009.



Figura 62. Diagrama de colaboración para agregar artículo

3.3.3.10. Actualizar datos y stock CU-010.

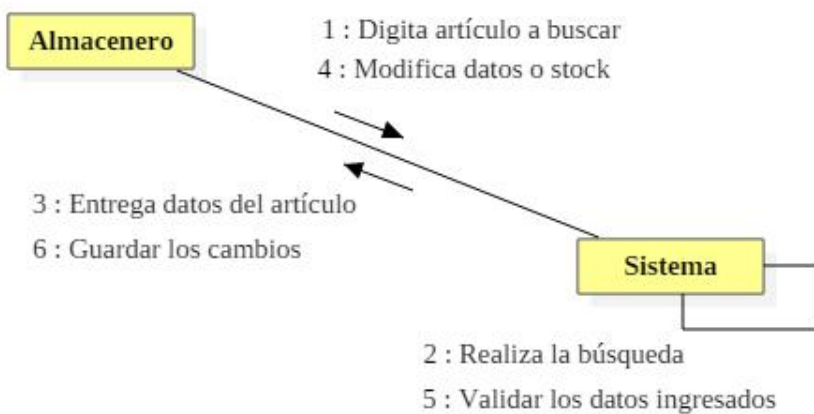


Figura 63. Diagrama de colaboración para actualizar datos y stock

3.3.3.11. Buscar artículo CU-011.

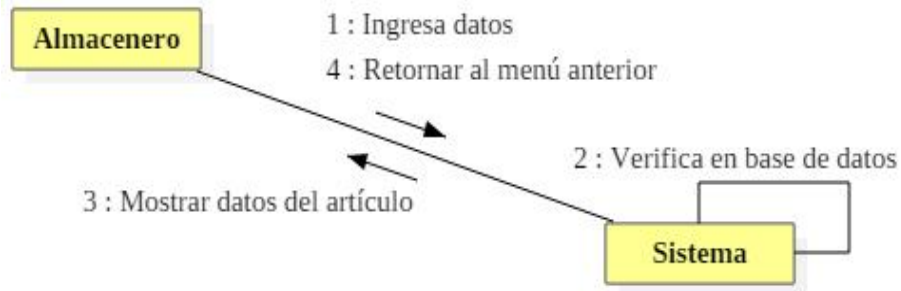


Figura 64. Diagrama de colaboración para buscar artículo

3.3.3.12. Solicitar informe actual CU-012.

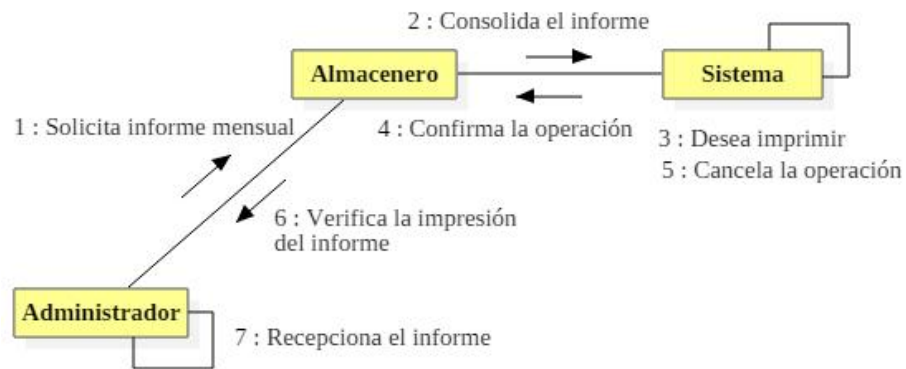


Figura 65. Diagrama de colaboración para solicitar informe actual

3.3.3.13. Generar reportes CU-013.



Figura 66. Diagrama de colaboración para generar reportes

3.3.4 Diagrama de estado

Estos diagramas también son importantes para ver la situación actual de los movimientos de cada actividad.

3.3.4.1. Pedido de artículo.



Figura 67. Diagrama de estado para pedido de artículo

3.3.4.2 Búsqueda de datos.

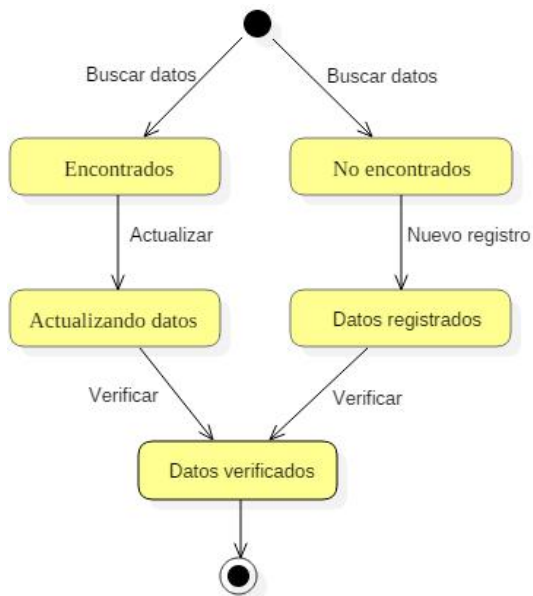


Figura 68. Diagrama de estado para búsqueda de datos

3.3.4.3. Registro de datos.

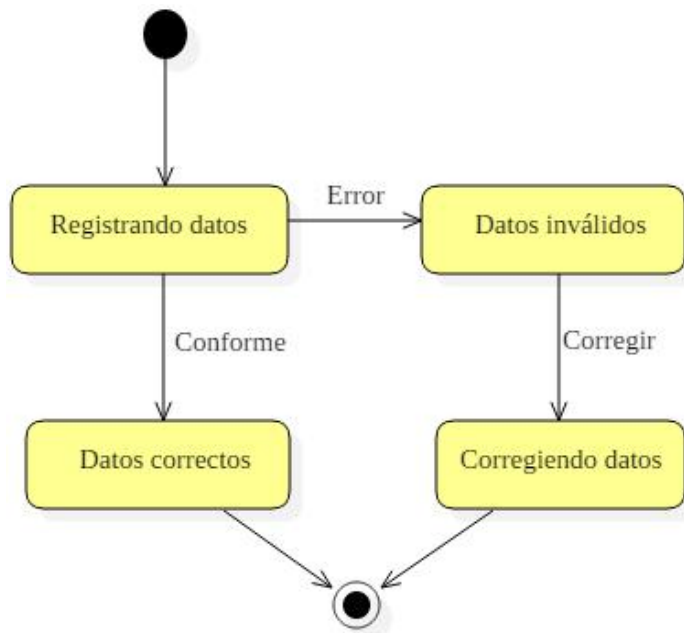


Figura 69. Diagrama de estado para registro de datos

3.3.4.4. Informes.



Figura 70. Diagrama de estado para informes

3.3.5. Diagrama de paquete

Este diagrama nos permite ver de forma general la interrelación de todos los actores, es una visión general y amplia el flujo de las diferentes acciones.

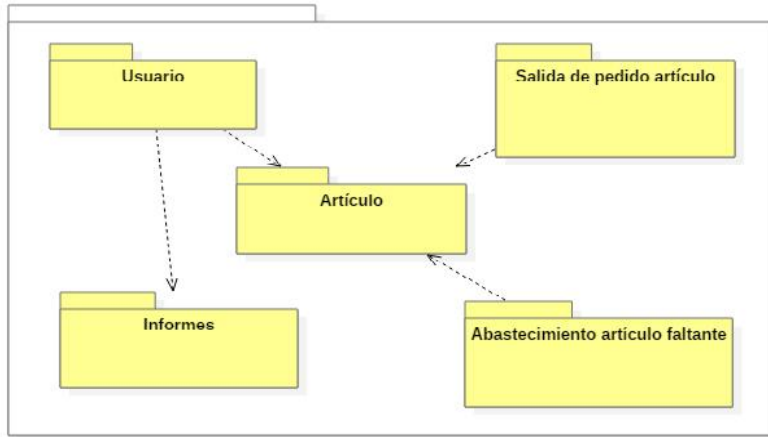


Figura 71. Diagrama de paquete para la gestión de almacén

3.3.6. Diagrama general de clases

Gráficamente podemos describir sus características de cada objeto, conocer a detalle cuáles son sus atributos y comportamientos.

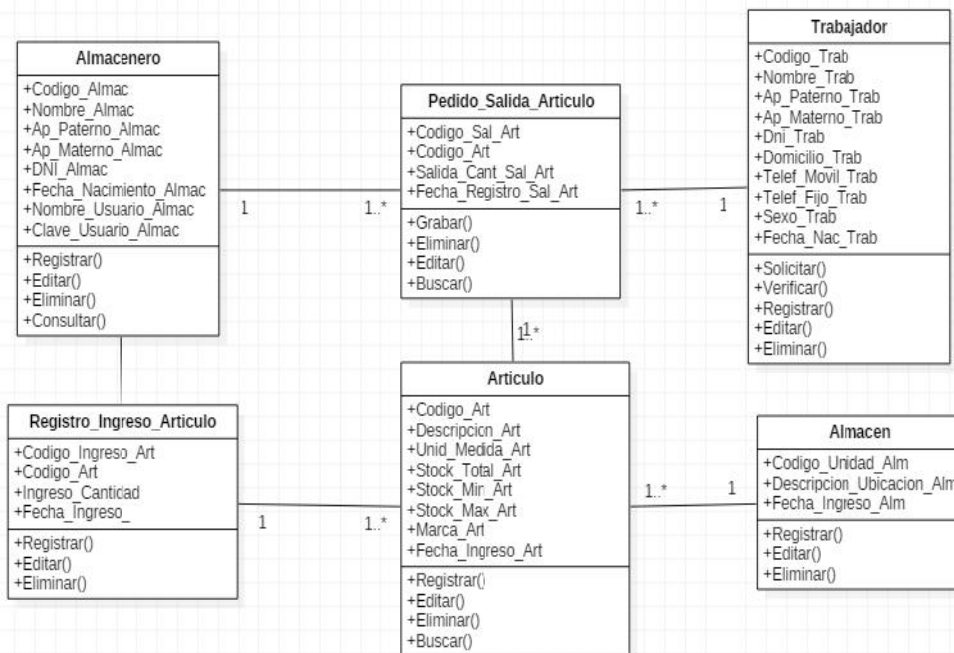


Figura 72. Diagrama general de clases para la gestión del almacén

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

Primera. Se logró la elaboración del análisis y diseño a través de metodología RUP en área de almacén de la empresa NCK INGENIEROS E.I.R.L.

Segunda. Se desarrolló los conceptos y fundamentos de la metodología RUP y UML.

Tercera. Se logró el análisis y diseño de las operaciones y movimientos que surgen en el área del almacén.

4.2 RECOMENDACIONES

Primera. Una de las principales consideraciones es asignar al personal adecuado y responsable para el manejo del almacén físico, optando las políticas de organización para que todo el funcionamiento del futuro sistema informático se logre a desarrollarse.

Segunda. Hacer seguimiento y la identificación de posibles cambios o nuevos problemas que se puedan presentar.

Tercera. Después de hacer el diseño y modelado, realizar su implementación o construcción de un sistema informático tomando como base el presente trabajo.

REFERENCIAS BIBLIOGRÁFICAS

- Booch, G., Jacobson, I. y Rumbaugh, J. (2000). *Proceso unificado del desarrollo del software*. Madrid, España: Editorial Addison Wesley.
- Booch, G., Jacobson, I. y Rumbaugh, J. (2000). *El lenguaje unificado de modelado*. Madrid, España: Editorial Addison Wesley.
- Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. México: Editorial Prentice Hall Latinoamérica.
- Larman, C. (2003). *UML y patrones (2ª ed.)*. Madrid, España: Prentice Hall.
- Kendall y Kenneth, J. (2005). *Análisis y diseño de sistemas (8ª ed.)*. México: Editorial Pearson educación.
- Kruchten, P. (2004). *The Rational unified process an introduction (3ª ed.)*. Boston, EE. UU.: Addison Wesley.
- Stevens, P. (2002). *Utilización de UML en ingeniería del software con objetos y componentes (2ª ed.)*. Madrid, España: Editorial Addison Wesley.
- Rumbaugh, J., Jacobson, I. y Booch, G. (2000). *El lenguaje unificado de modelado Manual de referencia*. Madrid, España: Editorial Addison Wesley.
- Fernández, C. (2000). *El proceso unificado rational para el desarrollo de software*. Universidad Tecnológica de la Mixteca, Oaxaca, México.
- Pressman, R. (2010). *Ingeniería de software. Un enfoque práctico (7ª ed.)*. México: Editorial McGraw-Hill Interamericana.

Vanzetti, J. (2006). *Un modelo del proceso de desarrollo de software guiado por la traceability* (Tesis de maestría). Universidad Nacional de La Plata, Argentina.

Hernandez, J., Galeana, S., Martínez, O. y Zepeda, D. (2013). *Sistema en línea para la evaluación docente en una institución educativa* (Tesis de pregrado). Universidad Nacional Autónoma de México, México.

Hernandez, A. (2004). *Aplicación del proceso unificado de desarrollo a proyecto de software* (Tesis de pregrado). Universidad Tecnológica de la Habana, La Habana, Cuba.

Recuperado de <https://researchgate.net/publication/312656269>.

Universidad José Carlos Mariátegui. (2018). *Manual de elaboración de tesis, Trabajo de Suficiencia Profesional y Artículo Científico* (versión 2). ISO 9001:2015.