



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3

**Sistema para la Dirección de Compras de la UCI. Módulos
Gestión de necesidad, Contratación, Seguimiento de la demanda
y Recepción de mercancía.**

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autores:

Patricia Hernández Rodríguez

José Pablo Bejarano Ruiz

Tutores:

MSc. Miguel Angel Sánchez Palmero

MSc. Olga Yarisbel Rojas Grass

Ing. Héctor García Llanes

La Habana, junio de 2019

“Año 61 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser los autores de este trabajo titulado: Sistema para la Dirección de Compras en la UCI. Módulos Gestión de necesidad, Contratación, Seguimiento de la demanda y Recepción de mercancía y se otorga a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente a los ____ días del mes de _____ del año _____.

Autores:

Patricia Hernández Rodríguez

José Pablo Bejarano Ruiz

Tutores:

MSc. Olga Y. Rojas Grass

MSc. Miguel Angel Sánchez Palmero

Ing. Héctor García Llanes

AGRADECIMIENTOS

Estos cinco años en la Universidad, me han dado experiencias inolvidables, ya han pasado cinco años de sacrificios y alegrías que hoy terminan cumpliendo un gran sueño, por eso quiero agradecerles:

A mi mamá por estar siempre en los momentos difíciles y siempre confiar en mí.

A mi papá por estar siempre que lo necesité, teníamos una meta en común que hoy estamos logrando.

A mis hermanitas jimaguas por ser esas personitas que alegran mis días, las quiero.

A mi novio la persona más especial en estos añitos que llevamos, por haber estado presente en todo momento y siempre ayudarme en todo, eres especial. Gracias por siempre estar a mi lado.

A mi tía, mi segunda mamá, por darme esos ánimos que siempre me hacían falta y estar orgullosa de mí. A papi por ser mi chofer durante estos años. A mi familia en general.

A mis amigas Maite, Laura, Selí, Danet y Haraid gracias por estar a mi lado en los buenos y malos momentos.

A mis tutores Migue y Olga por su apoyo y siempre estar disponibles en cualquier momento, son excelentes. Gracias a los dos por toda la ayuda, el tiempo dedicado y la paciencia fundamental.

A mi compañero de tesis por su apoyo.

A mis profesores que me guiaron en la carrera, y en especial a la profe Dariela por su preocupación y apoyo, gracias por guiarme al mundo profesional.

A los profesores y miembros del tribunal que ayudaron a la realización de esta tesis gracias por ayudarnos a ser unas mejores personas.

Patricia Hernández Rodríguez

AGRADECIMIENTOS

Quisiera agradecer a todas aquellas personas que me han apoyado a lo largo de esta experiencia universitaria, a mi familia en general, principalmente a mis padres por estar siempre apoyándome, a mi abuelita que me estuvo preguntando constantemente sobre mi proceso de tesis. A mi gordita que siempre se mantuvo pendiente a lo que necesitara. Especialmente quiero agradecer a Patry por sobrellevar junto a mí esta compleja travesía, a mis tutores Olga, Migue y Héctor por dedicarme su tiempo y apoyo. Hay una persona que siempre estuvo al tanto de todo desde mis comienzos en esta universidad, gracias Dariela.

José Pablo Bejarano Ruiz

DEDICATORIA

*A mis padres y hermanitas que son las
personas más importantes en mi vida.*

Patricia Hernández Rodríguez

A mi abuelo Juan.

José Pablo Bejarano Ruiz

RESUMEN

La Gestión de compras comprende un proceso complejo que se inicia a partir de la existencia de una necesidad de adquirir un bien o un producto en una entidad. La Dirección de Compras de la Universidad de las Ciencias Informáticas es la encargada de satisfacer las necesidades para la ejecución de los procesos claves. Para llevar a cabo este proceso se realizan un conjunto de actividades que presentan insuficiencias y conllevan a que existan demoras en la solución de las necesidades de las diferentes áreas. Con el propósito de agilizar la gestión de las compras, el presente trabajo tiene como objetivo desarrollar un Sistema para la Dirección de Compras. Para ello se realizó un estudio de los principales sistemas informáticos que efectúan este proceso, concluyendo que era necesario construir una solución propia. Se realizó un análisis de los procesos en dicha área, permitiendo conocer el estado actual del proceso de compras. Para la construcción del sistema, se utilizó la Metodología de desarrollo para la Actividad productiva de la Universidad, como marco de trabajo Odoos en su versión 12.0 el cual incluye la utilización del lenguaje XML (Lenguaje de Marcas Extensible), Python 3.6 y PostgreSQL 10.0, además se utilizó como entorno de desarrollo PyCharm 2018.2.3. El sistema desarrollado fue validado a través de pruebas que determinaron el cumplimiento de los requisitos definidos inicialmente. La solución informática contribuye a satisfacer de forma ágil la gestión de compras de la UCI, permitiendo dar solución y seguimiento a todas las necesidades registradas por las áreas de la universidad.

Palabras claves: compras, contrato, demanda, necesidad, producto.

ABSTRACT

Purchasing Management comprises a complex process that starts from the existence of a need to acquire a good or a product in an entity. The Purchasing Department of the University of Computer Science is in charge of satisfying the needs for the execution of the key processes. To carry out this process, a series of activities are carried out that present insufficiencies and lead to delays in the solution of the needs of the different areas. In order to streamline the management of purchases, this work aims to develop a System for Purchasing Management. To this end, a study was made of the main computer systems that carry out this process, concluding that it was necessary to build their own solution. An analysis of the processes in this area was carried out, allowing to know the current status of the purchasing process. For the construction of the system, the Development Methodology was used for the productive activity of the University, as an Odoo framework in its version 12.0 which includes the use of XML (Extensible Markup Language), Python 3.6 and PostgreSQL 10.0, in addition it was used as a development environment PyCharm 2018.2.3. The developed system was validated through tests that determined compliance with the initially defined requirements. The IT solution contributes to the agile management of the UCI purchases, allowing to solve and follow up all the needs registered by the areas of the university.

Keywords: purchases, contract, demand, need, product.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1. Introducción	5
1.2. Proceso de gestión de compras	5
1.2.1. Principales conceptos relacionados	6
1.3. Análisis de principales sistemas	9
1.3.1. Valoración de los sistemas analizados	13
1.4. Metodología de desarrollo de software	14
1.5. Notaciones y lenguajes	15
1.6. Herramientas y tecnologías	17
1.7. Validación	18
1.8. Conclusiones parciales	22
CAPÍTULO 2. PROPUESTA DE SOLUCIÓN	23
2.1. Introducción	23
2.2. Modelado del negocio	23
2.2.1. Descripción de procesos de negocio	23
2.2.2. Modelo conceptual	24
2.3. Requisitos del sistema	26
2.3.1. Técnicas utilizadas para la obtención de requisitos	26
2.3.2. Requisitos funcionales	26
2.3.3. Descripción de requisitos funcionales	28
2.3.4. Requisitos no funcionales	30
2.3.5. Validación de requisitos	32
2.4. Diseño de la propuesta de solución	32
2.4.1. Diseño Arquitectónico	32
2.4.2. Diagrama de clases con estereotipos web	33
2.4.3. Entidades persistentes en Odoo	34
2.4.4. Patrones de diseño	35
2.5. Validación del diseño	36
2.6. Conclusiones parciales	42
CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	43
3.1. Introducción	43
3.2. Implementación	43
3.2.1. Estándares de codificación	43
3.2.2. Diagrama de componentes	45
3.2.3. Interfaz de usuario funcional	46
3.3. Pruebas Internas	47

ÍNDICE

3.3.1. Pruebas de caja blanca.....	48
3.3.2. Prueba de caja negra.....	50
3.3.3. Pruebas de aceptación.....	54
3.4. Aplicación de la técnica de ladov	55
3.5. Beneficios del Sistema de compras desarrollado	57
3.6. Conclusiones parciales	58
CONCLUSIONES GENERALES	59
RECOMENDACIONES	60
REFERENCIAS BIBLIOGRÁFICAS	61

ÍNDICE DE TABLAS

Tabla 1 Comparación de sistemas estudiados	13
Tabla 2 Salidas del sistema.....	27
Tabla 3 Descripción del requisito Crear necesidad de balance de alimento	28
Tabla 4 Requisitos no funcionales.....	31
Tabla 5 Atributos de calidad evaluados por la métrica TOC	37
Tabla 6 Criterios de evaluación para la métrica TOC	37
Tabla 7 Atributos de calidad evaluados por la métrica RC.....	39
Tabla 8 Criterios de evaluación para la métrica RC.....	40
Tabla 9 Caso de prueba para el camino 1	49
Tabla 10 Caso de prueba para el camino 2	50
Tabla 11 Caso de prueba para el camino 3.....	50
Tabla 12 Diseño de Caso de Prueba de caja negra del requisito Crear producto.....	51
Tabla 13 Niveles de satisfacción	55
Tabla 14 Cuadro Lógico de ladov.....	56
Tabla 15 Resultados del cuestionario de ladov según la escala.....	57

ÍNDICE DE FIGURAS

Figura 1 Escenario tres de la metodología	15
Figura 2 Descripción del proceso Gestión de necesidad	24
Figura 3 Modelo conceptual	25
Figura 4 Arquitectura Modelo Vista Controlador	33
Figura 5 Diagrama de clases del diseño con estereotipos web	34
Figura 6 Resultado de la evaluación de la métrica para el atributo responsabilidad	38
Figura 7 Resultado de la evaluación de la métrica para el atributo complejidad	38
Figura 8 Resultado de la evaluación de la métrica para el atributo reutilización	39
Figura 9 Cantidad de dependencias entre las clases	40
Figura 10 Resultado de la evaluación de la métrica para el atributo acoplamiento	41
Figura 11 Resultado de la evaluación de la métrica para el atributo complejidad de mantenimiento ..	41
Figura 12 Resultado de la evaluación de la métrica para el atributo cantidad de pruebas	41
Figura 13 Resultado de la evaluación de la métrica para el atributo reutilización	42
Figura 14 Estándar de codificación Tabuladores o espacios	43
Figura 15 Estándar de codificación Líneas en blanco.....	44
Figura 16 Estándar de codificación Tamaño máximo de línea.....	44
Figura 17 Diagrama de componentes.....	46
Figura 18 Interfaz de usuario de la funcionalidad Listar productos	47
Figura 19 Interfaz de usuario de la funcionalidad Crear necesidad de balance de alimentos	47
Figura 20 Método registrar demanda en el balance de alimentos.....	48
Figura 21 Grafo resultante de aplicar la técnica Camino Básico.....	49
Figura 22 No Conformidades detectadas	54
Figura 23 Índice de satisfacción grupal	56
Figura 24 Rango de valores para el cálculo de satisfacción grupal	56

INTRODUCCIÓN

Las Tecnologías de la Información y la Comunicación (TIC) representan en este siglo el vector impulsor de profundos y estructurales cambios en los hombres y en las sociedades. Las empresas, hacen uso de estas tecnologías y en muchas ocasiones son las que propician su desarrollo. El rápido avance de las tecnologías ha modificado profundamente las formas de trabajar y vivir, pues la difusión y generalización de las TIC está asociada a innovaciones sociales, comerciales, empresariales y legales. Aún cuando el impacto en la productividad empresarial puede ser significativo, debido a las nuevas formas de gestionar y estructurar los negocios sobre la base de las TIC, no es menos cierto que plantea un desafío para los gestores del futuro. La productividad de las empresas ha crecido de la mano de la incorporación masiva de las TIC a sus procesos de negocios [1].

Una de las decisiones más complejas y arriesgadas para las empresas, es escoger adecuadamente los sistemas informáticos o aplicaciones capaces de gestionar los flujos de información y que soporten sus procesos. Entre ellos se pueden encontrar los sistemas ERP (por sus siglas en inglés, Planificación de Recursos Empresariales) y CRM (por sus siglas en inglés, Gestión de las Relaciones con Clientes), que sirven de apoyo a varios de los procesos empresariales, entre los que se encuentra el de Gestión de compras. En el mundo existen ERP que se utilizan en múltiples empresas, además en Cuba se han desarrollado sistemas de este tipo, los que de una forma u otra solucionaron los problemas de las empresas donde se implantaron. Uno de los ejemplos desarrollados en Cuba y en despliegue por la empresa XETID es el Sistema de Gestión Empresarial DISTRA, cuenta con varios módulos, pero existen insuficiencias en la gestión de los contratos [2].

La informatización de la sociedad cubana ha sido una de las estrategias prioritarias del Estado formando parte de ella varias entidades. En el 2002 se crea la Universidad de las Ciencias Informáticas (UCI) con un modelo de formación único, que relaciona la formación, con la investigación y la producción de software, respondiendo al proceso de informatización que se lleva a cabo en el país. En ella se encuentran adscritos los centros de desarrollo de productos y servicios, con un elevado nivel de compromiso, ética y profesionalidad. En la red de centros se encuentra el Centro de Informatización de Entidades (CEIGE), especializado en el desarrollo de soluciones, productos y servicios para la gestión empresarial.

Entre los procesos que se ejecutan en la universidad se encuentra la Gestión de compras. Este es un proceso de apoyo de la universidad, que permite el aprovisionamiento de alimentos e insumos necesarios para su abastecimiento. Este incluye la gestión de las necesidades de las áreas de la UCI, la contratación, seguimiento de la demanda y la recepción de mercancías. En este proceso existen

varias insuficiencias para llevar a cabo su ejecución, por lo que la Dirección de Compras (DC) y CEIGE deciden iniciar un trabajo conjunto para revisar el funcionamiento del proceso.

En el proceso de Gestión de compras se debe tener en cuenta las cuatro fases de un proceso administrativo (planificación, organización, ejecución y control). Para ello es de vital importancia seguir una serie de pautas [3]:

- Detectar las necesidades.
- Negociar con proveedores y contratar aquellos que, por calidad, tiempo de entrega y flexibilidad de condiciones se ajusten mejor a los requisitos.
- Recepcionar y revisar los pedidos solicitados.
- Realizar la compra y el seguimiento de cada pedido.
- Gestionar el pago de las facturas a los proveedores.

A partir de estas pautas se identificaron varios problemas que actualmente existen en la Dirección de Compras de la UCI:

- Las áreas de la universidad no brindan una demanda bien detallada de los productos que necesitan, existiendo retraso e incertidumbre en la determinación de las cantidades. Esto dificulta su correcta planificación, lo que trae consigo retraso y afectación en la cadena de aprovisionamiento. El mecanismo de la recepción de las solicitudes de las necesidades varía afectando la consistencia de la información recibida, no favoreciendo la toma de decisiones.
- Cuando una necesidad no se puede satisfacer debido a que los productos que se requieren no están en almacén, se revisa la cartera de proveedores para definir la solicitud de oferta. Durante estas actividades se debe tener en cuenta la vigencia de los contratos y la posibilidad de realizar nuevos contratos de ser necesario. El seguimiento que se realiza a los contratos es a través de la herramienta Excel, esta presenta limitaciones para realizar notificaciones de contratos vencidos o próximos a vencer, provocando demoras en el análisis y supervisión de la oferta.
- La información relacionada con el proceso no es oportuna, en ocasiones ocurren pérdidas de oportunidades de compra porque la manera en que se encuentra la información dificulta realizar un seguimiento adecuado de la ejecución de la compra.

Teniendo en cuenta la problemática anterior se plantea como **problema a resolver**: ¿cómo contribuir a la agilización de los procesos Gestión de necesidad, Contratación, Seguimiento de la demanda y Recepción de mercancía en la Dirección de Compras de la UCI?

El **objeto de estudio** se enmarca en el proceso de Gestión de compras.

Para solucionar el problema anteriormente planteado se define como **objetivo general** desarrollar los módulos Gestión de necesidad, Contratación, Seguimiento de la demanda y Recepción de mercancías en la Dirección de Compras de la UCI para la agilización de estos procesos.

Campo de acción: sistemas web para la gestión de compras.

Objetivos específicos:

OE1: Elaborar el marco teórico de la investigación relativo a los sistemas de gestión de compras mediante aplicaciones web.

OE2: Realizar el Modelado de negocio, Requisitos y Análisis y diseño para los módulos Gestión de necesidad, Contratación, Seguimiento de la demanda y Recepción de mercancías para la Dirección de Compras de la UCI.

OE3: Implementar los módulos Gestión de necesidad, Contratación, Seguimiento de la demanda y Recepción de mercancía para la Dirección de Compras de la UCI.

OE4: Validar la solución mediante la aplicación de técnicas, métricas y pruebas.

OE5: Evaluar la satisfacción de los usuarios a través de la técnica de ladov.

Idea a defender: si se desarrollan los módulos Gestión de necesidad, Contratación, Seguimiento de la demanda y Recepción de mercancía se contribuirá a la agilización de los procesos de la Dirección de Compras de la UCI.

Métodos científicos de investigación

Métodos teóricos

- **Histórico:** a través de este método se realizó un estudio del origen y desarrollo de los sistemas ERP. Además, se realizó un análisis para entender el proceso de Gestión de compras, enfocado en la ejecución de este en la Dirección de Compras de la UCI.
- **Modelación:** la modelación es el método mediante el cual se crean abstracciones con vistas a explicar la realidad. Su aplicación en este trabajo se evidencia en la elaboración de los diferentes diagramas, modelo conceptual y prototipos durante el desarrollo del sistema.

Métodos empíricos

- **Observación:** permite obtener conocimiento acerca del comportamiento del proceso de compra en la UCI y las actividades que se ejecutan. Se utilizó al apreciar sistemáticamente el proceso de compras para lograr un mejor entendimiento del funcionamiento del mismo.
- **Entrevista:** se realizaron entrevistas individuales al Director de Compras de la UCI para un mejor entendimiento del proceso a informatizar. Luego de aplicarla se obtuvo la descripción de la información referente al negocio y se definieron los requisitos del sistema.

Estructura del trabajo

Capítulo 1. Fundamentación teórica: se describen los principales conceptos relacionados con el proceso de compra. Se hace una valoración referente a los sistemas informáticos existentes para la gestión de compra. Se realiza la fundamentación de las herramientas y lenguajes a utilizar para dar solución a la problemática, así como la metodología que guiará el proceso de desarrollo de software.

Capítulo 2. Propuesta de solución: se realiza un estudio del negocio. Se definen los requisitos funcionales y no funcionales, así como su correspondiente descripción, se muestran los artefactos que propone la metodología de desarrollo de software seleccionada. Se realizó el diagrama de clases del diseño con estereotipos web y, por último, se valida el diseño a través de las métricas seleccionadas. Además, se elabora una propuesta de solución con los diagramas y otros artefactos que guían la posterior implementación.

Capítulo 3. Implementación y validación de la solución propuesta: en este capítulo se implementan las funcionalidades identificadas, se realiza la validación de los módulos desarrollados mediante las pruebas internas y las pruebas de aceptación. Además, se aplica la técnica de ladov para conocer en qué medida se sienten satisfechos los usuarios.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En el presente capítulo se presentan los principales conceptos abordados sobre el tema. Se realiza un estudio al proceso de Gestión de compras para establecer una base de conocimientos sólida que permita una mejor comprensión de los procesos de gestión de compras. Se fundamenta la selección de la metodología, herramientas y tecnologías que permiten llevar a cabo el proceso de desarrollo de software.

1.2. Proceso de gestión de compras

Cuando la empresa necesita comprar bienes o contratar servicios tiene que hacer un estudio del mercado, localizar fuentes de suministro, comprobar precios y servicios. La planificación de las compras es un proceso complejo, va más allá de la negociación con el proveedor y el trámite burocrático. Con una buena gestión de compras la empresa consigue ahorrar costos, satisfacer al cliente, en tiempo y cantidad, y obtener servicios directos, pues la gestión de compras es decisiva para que la empresa obtenga el éxito o el fracaso. En una empresa esta actividad la realiza el área de compras [4].

Entre las funciones del área de compras se encuentran [4]:

- Detección y descripción de la necesidad: además de la descripción material de la necesidad hace falta la visión, cantidad y para cuándo. En base a una previsión de ventas, se realiza el programa de fabricación y en función de sus plazos, se determina la necesidad para cada pieza o material concreto. De esta forma el área de compras puede asumir la responsabilidad de que la empresa disponga de lo necesario, en el momento preciso y al menor costo posible.
- Investigación y búsqueda de fuentes de aprovisionamiento: es fundamental la información que aporte el área sobre costos, calidades, tendencias y disponibilidad que ofrece el mercado. El área de compras debe reunir y analizar la información necesaria de todos los artículos y materiales que se necesiten. Lo primero en la búsqueda y selección de proveedores es la recogida de los datos para analizar cuáles pueden satisfacer las necesidades.
- Preparación de la compra: se debe acometer la eficaz prospección y examen de los posibles proveedores, para asegurarse que tienen las condiciones necesarias y suficientes para satisfacer plenamente las necesidades.
- Ejecución de la compra: una vez seleccionados los proveedores más adecuados y conocidas las condiciones generales que rigen en el mercado suministrador, se debe iniciar la petición de ofertas que deberán cumplir con las condiciones fundamentales. Con la firma de contratos por comprador y vendedor, se llegan a acuerdos de las condiciones definitivas para la adquisición de materiales o servicios de la negociación.

- Seguimiento y control: el seguimiento debe incluir el control administrativo y las visitas a los proveedores, para tener la seguridad del cumplimiento de los plazos o detectar a tiempo posibles retrasos.

Proceso de Gestión de compras en la UCI

El proceso de Gestión de compras en la UCI tiene como principal objetivo el aprovisionamiento de equipos, insumos y materiales, así como la adquisición de servicios necesarios para la realización de las actividades de la Universidad. Además, en la ejecución de las compras se debe cumplir con las cantidades y plazos establecidos, con los niveles de calidad necesarios y al menor precio que permita el mercado.

Este proceso se lleva a cabo en el momento en que surge una necesidad, la cual se analiza y se registra. Una vez registrada dicha necesidad en la base de datos se procede a revisar qué proveedor produce o brinda el producto o servicio que se ha demandado, si existe dicho proveedor se pasa a analizar el estado del contrato. Es entonces cuando se solicita la proforma de contrato y/o suplemento al proveedor y luego se revisa. Posteriormente se procede a la recepción, registro y entrega del contrato y/o suplemento. Una vez que se tiene definido cuál o cuáles proveedores pueden satisfacer lo demandado, se analizan las ofertas y se elige la propuesta de mayor conveniencia para la universidad. Posteriormente se selecciona el o los proveedores, el especialista encargado le asignará al comprador correspondiente el seguimiento de la demanda.

Una vez que el especialista encargado tiene la oferta, procede a confeccionar la solicitud de pago la cual posteriormente se revisa, se firma, se aprueba y se archiva una copia. Cuando está listo todo se pasa a solicitar el transporte de carga para comprar la mercancía y luego que sea revisada. Esta mercancía pasa por un proceso de comprobación, si se detecta alguna irregularidad se realiza la debida reclamación. Cuando el comprador o la persona autorizada firme el informe de recepción concluye la compra de la mercancía.

1.2.1. Principales conceptos relacionados

Comprador

Comprador según el Diccionario de la Real Academia Española (RAE), es el que compra. El comprador es el que cubre necesidades adquiriendo bienes o servicios a cambio de un precio justo. Los consumidores (y usuarios) son compradores de bienes y (contratadores de servicios) por definición y son una de las dos partes que tienen los mercados: la demanda por oposición a la oferta, compuesta por los vendedores [5].

Persona u organización que demanda bienes o servicios proporcionados por el productor o el proveedor de bienes o servicios. Es un agente económico con una serie de necesidades y deseos que cuenta con

una renta disponible con la que puede satisfacer esas necesidades y deseos a través de los mecanismos de mercado [6].

A partir de las definiciones anteriormente realizadas se concluye que un Comprador es una persona u organización que realiza el acto de compra de bienes o servicios. Es un agente económico que puede ser o no el consumidor del producto o servicio adquirido.

Producto

Para Kerin, Hartley y Rudelius, un producto es “un artículo, servicio o idea que consiste en un conjunto de atributos tangibles o intangibles que satisface a los consumidores y es recibido a cambio de dinero u otra unidad de valor” [7].

Stanton, Etzel y Walker, autores del libro “Fundamentos de Marketing”, definen el producto como “un conjunto de atributos tangibles e intangibles que abarcan empaque, color, precio, calidad y marca, más los servicios y la reputación del vendedor; el producto puede ser un bien, un servicio, un lugar, una persona o una idea” [8]. Según Jerome McCarthy y William Perrault, autores del libro “Marketing Planeación Estratégica de la Teoría a la Práctica”, el producto “es la oferta con que una compañía satisface una necesidad” [9].

A partir de lo planteado anteriormente se define que un producto es el resultado de un esfuerzo creador que tiene un conjunto de atributos tangibles e intangibles, los cuales son percibidos por sus compradores como capaces de satisfacer sus necesidades o deseos. Por tanto, un producto puede ser un bien, un servicio, una idea o un lugar y existe para propósitos de intercambio, la satisfacción de necesidades o deseos y para contribuir al logro de objetivos de una organización.

Servicio

Stanton, Etzel y Walker, definen los servicios “como actividades identificables e intangibles que son el objeto principal de una transacción ideada para brindar a los clientes satisfacción de deseos o necesidades” [10]. Para Richard L. Sandhusen, “los servicios son actividades, beneficios o satisfacciones que se ofrecen en renta o a la venta, y que son esencialmente intangibles y no dan como resultado la propiedad de algo” [11].

Según Lamb, Hair y McDaniel, “un servicio es el resultado de la aplicación de esfuerzos humanos o mecánicos a personas u objetos. Los servicios se refieren a un hecho, un desempeño o un esfuerzo que no es posible poseer físicamente” [12].

Teniendo en cuenta lo propuesto anteriormente se plantea la definición de servicio como las actividades que son resultado de esfuerzos humanos o mecánicos que producen un hecho, un desempeño o un esfuerzo. Dichas actividades implican generalmente la participación del cliente y no es posible poseer físicamente, ni transportarlos o almacenarlos, pero que pueden ser ofrecidos en renta o a la venta. Por

tanto, pueden ser el objeto principal de una transacción ideada para satisfacer las necesidades o deseos de los clientes.

Proveedor

Según Paulo Núñez en el contexto de las ciencias económicas y empresariales, el término Proveedor designa a toda la entidad que pone a disposición de otra entidad un determinado producto o servicio. De acuerdo con este concepto, el proveedor puede ser el productor de bienes y servicios o de sus distribuidores [13].

En la cadena de suministros, el proveedor es la parte fundamental para dar cumplimiento a la gestión de compras de cualquier compañía. El proveedor es el responsable de entregar a tiempo las órdenes de compra de acuerdo a los requerimientos exigidos por sus clientes [14].

A partir de lo planteado anteriormente, se define que un proveedor es una persona o empresa que abastece a otras empresas con existencias (productos), los cuales serán vendidos directamente o transformados para su posterior venta.

Precio

Según Philip Kotler y Gary Armstrong (2003), autores del libro “Fundamentos de Marketing”, “el precio es en el sentido más estricto la cantidad de dinero que se cobra por un producto o servicio. En términos más amplios, el precio es la suma de los valores que los consumidores dan a cambio de los beneficios de tener o usar el producto o servicio” [15].

Por su parte, la *American Marketing Association* (2006), define el precio como “la proporción formal que indica las cantidades de bienes de dinero o servicios necesarios para adquirir una cantidad dada de bienes o servicios” [15]. Patricio Bonta y Mario Farber, autores del libro “199 Preguntas Sobre Marketing y Publicidad”, definen el precio como la expresión de un valor. El valor de un producto depende de la imagen que percibe el consumidor; sin embargo, los consumidores perciben cualquier producto “bueno para la salud” como algo de valor superior [15].

Tomando en cuenta las anteriores definiciones, se plantea que precio es la cantidad de dinero que se dará a cambio de un beneficio, así mismo es un factor significativo para que el cliente este satisfecho, puesto que el cliente aprecia el bien o servicio de acuerdo al precio que fue adquirido, y en base a la percepción del precio es generalmente si se toma la decisión de compra.

Mercado

Stanton, Etzel y Walker, autores del libro “Fundamentos de Marketing”, definen el mercado (para propósitos de marketing) como “las personas u organizaciones con necesidades que satisfacer, dinero para gastar y voluntad de gastarlo” [16]. Para Patricio Bonta y Mario Farber, autores del libro “199 Preguntas Sobre Marketing y Publicidad”, el mercado es “donde confluyen la oferta y la demanda. En

un sentido menos amplio, el mercado es el conjunto de todos los compradores reales y potenciales de un producto” [17].

Según Philip Kotler, autor del libro “Dirección de Marketing”, un mercado era “una ubicación física donde se reunían compradores y vendedores para comprar y vender bienes. Los economistas describen el mercado como el grupo de compradores y vendedores que realizan transacciones sobre un producto o clase de productos (como el mercado de vivienda o el mercado de granos)” [16].

Teniendo en cuenta lo propuesto anteriormente se plantea que mercado es el conjunto de compradores reales y potenciales que tienen una determinada necesidad y/o deseo, dinero para satisfacerlo y voluntad para hacerlo, los cuales constituyen la demanda, y vendedores que ofrecen un determinado producto para satisfacer las necesidades y/o deseos de los compradores mediante procesos de intercambio, los cuales constituyen la oferta. Ambas, la oferta y la demanda son las principales fuerzas que mueven el mercado.

Contrato

Ignacio Galindo Garfías: “Es el acto jurídico típico del derecho privado y se caracteriza porque las declaraciones de voluntad de las partes que lo celebran son concurrentes, convienen en crear entre sí relaciones jurídicas; forman un acuerdo de voluntades o consentimiento, lo que constituye propiamente el elemento esencial del contrato. El contrato puede ser considerado como acto jurídico por una parte y como norma individualizada de conducta desde otro punto de vista [18].”

Azúa Reyes: “Contrato es el acuerdo de voluntades de dos o más personas para crear o transferir entre sí, derechos y obligaciones. De su naturaleza explica que es un acto jurídico bilateral o plurilateral, o si se prefiere, dos o más actos jurídicos emanados respectivamente de dos o más personas que concurren a la realización de un negocio en el que se encuentran interesadas [18].”

Rafael de Pina: “En su significación semántica, contrato es el pacto o convenio entre partes sobre materia o cosa determinada y a cuyo cumplimiento pueden ser exigidas. El contrato no es una norma jurídica en el mismo sentido en que lo son la ley y la costumbre, por ejemplo, sino un acto jurídico celebrado de acuerdo con determinadas normas jurídicas [18].”

Tomando en cuenta las anteriores definiciones, se plantea la siguiente definición de contrato: acuerdo, generalmente escrito, por el que dos o más partes se comprometen recíprocamente a respetar y cumplir una serie de condiciones.

1.3. Análisis de principales sistemas

Existen actualmente diferentes sistemas informáticos que sirven de apoyo a los procesos de gestión de compra. Para un mejor entendimiento de las funcionalidades a desarrollar en la solución, es

necesario, realizar un estudio de sistemas que poseen características similares los cuales se describen a continuación.

➤ **SAP Business One**

SAP Business One es un software de gestión empresarial, rápido y fácil de usar, hecho específicamente para pequeñas y medianas empresas. El sistema integra funciones empresariales, incluyendo, entre otros: finanzas, compras, logística, ventas, comercio electrónico y gestión de inventarios [19].

Dispone de un módulo de soporte para compras que permite estimar con mayor exactitud la demanda futura, con opciones de compras planificadas y más rentables. El módulo de compras de SAP permite gestionar las actividades relacionadas con los proveedores, como la emisión de pedidos de compra, la actualización de cantidades de existencias, el cálculo del precio de entrega de los artículos importados, la entrega de mercancías y la gestión de devoluciones y abonos. Además de contar con el análisis de mercado, ventas y clientes para planear inteligentemente las compras de materiales [19].

➤ **Assets NS**

Es un Sistema de Gestión Integral estándar y parametrizado que permite el control de los procesos de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos [20]. El módulo de Compras permite realizar todas las operaciones vinculadas al proceso de adquisición de mercancías para la venta o para insumo, pasando por un proceso de formación de costo a partir de las facturas de los diferentes proveedores, ya sean en torno al precio unitario del producto o teniendo en cuenta otros gastos incurridos en las gestiones de compras, manipulación, gestiones aduanales, transportación y seguros. Las opciones que brinda son las siguientes [21]:

- Emisión manual o generación automática de órdenes de compras al proveedor.
- Recepciones de mercancías.
- Recepciones de servicios.
- Recepciones de servicios asociadas a productos.
- Formación de costo con la posibilidad de estimar.

Actualmente en la universidad se utiliza este sistema para la gestión de varios procesos como: Inventario, Finanzas, Contabilidad, Activos Fijos y Recursos Humanos.

➤ **Plataforma de compras Fullstep**

Fullstep es una empresa líder en la gestión de compras y aprovisionamiento empresarial, además es un sistema de módulos que facilita el control sobre todo el ciclo del gasto. Una herramienta versátil, que integra todos los detalles y funciones que completan el proceso de compras en cada sector. En definitiva, tecnología para impulsar el rendimiento de Compras [22].

Es una herramienta desarrollada para mejorar el Área de Compras desde el punto de vista operativo y organizativo. Desde el punto de vista operativo, incrementa el rendimiento gracias a las características dinámica, versátil y escalable, y configurable. Desde el punto de vista organizativo, la plataforma de Compras aporta profesionalización, eficiencia y perfeccionamiento constante [22].

Fullstep permite consolidar toda la información en un único lugar, a la vez que facilita la colaboración entre los distintos departamentos de la empresa. Fullstep Portal es un canal de comunicación directo entre una empresa y sus proveedores [22].

Fullstep permite:

- Gestionar proveedores
- Solicitudes y autorización
- Negociaciones y adjudicación
- Gestión de contratos
- Emisión de pedidos y recepción
- Homologación y calidad de proveedores
- Facturación en línea
- Análisis y control del gasto

➤ ERP Odoo

OpenERP ahora llamado Odoo es una *suite* de negocios integral, libre, de código abierto, modular y basada en la aplicación. Dispone de módulos oficiales, módulos desarrollados por terceros y también permite realizar desarrollos propios para adaptarlo a las necesidades específicas de la empresa. El módulo de Compras se utiliza para gestionar la adquisición directa e indirecta de todos los bienes y servicios que pueda necesitar la empresa, con el objetivo de centralizar la capacidad de compra, respondiendo y dando soporte a las políticas y procesos de adquisición [23]. Mediante este módulo se puede: gestionar pedidos, líneas de pedido, productos, facturación, listas de precios, entregas, control y seguimiento, proveedores.

Características:

- Es un sistema utilizado para la gestión integrada de los recursos de una empresa.
- La arquitectura del sistema es cliente– servidor, lo que permite que todos los usuarios trabajen sobre el mismo repositorio de datos. Esto tiene la ventaja de que toda la información está disponible y sincronizada en todo momento además de que descarga la mayor parte del trabajo de procesamiento de datos de las máquinas cliente (donde trabajan efectivamente los usuarios).
- Dentro de la construcción misma del software se hace un flujo intensivo de flujos de trabajo que se pueden integrar con sus distintos módulos.

➤ **Microsoft Dynamics**

Microsoft Dynamics NAV surgió a mediados del año 2002. Esta herramienta cuenta con más de 2.000 módulos, garantizando que la solución sea adaptable a diferentes entornos empresariales. Según el distribuidor con el que se haya formalizado el vínculo para el despliegue del proyecto, se pueden utilizar servicios en la nube. Entre sus módulos principales se encuentra la gestión de la cadena de suministros [24].

- Este módulo permite personalizar los procesos de los ciclos de ventas, compras, recogida, empaquetado y envío para satisfacer necesidades específicas y mantener el ritmo de mercados competitivos con escasos márgenes
- Gestiona presupuestos
- Define la cantidad de autorizaciones para generar órdenes de compra
- Gestión de pedidos de compras
- Gestiona las solicitudes de compras

➤ **Openbravo**

Openbravo es una aplicación de gestión empresarial del tipo ERP destinada a empresas de pequeño y mediano tamaño. Se presenta en dos versiones: *Community*, de libre distribución y con acceso al código abierto; y *Profesional*, de código propietario con todas las funcionalidades activas. Es una aplicación con arquitectura cliente/servidor web implementada en Java. Se ejecuta sobre Apache y con soporte para base de datos PostgreSQL y Oracle. También es un completo y seguro sistema de gestión empresarial que permite integrar los procesos de una organización de acuerdo a su necesidad. Posee módulos a tener en cuenta como son el de gestión de compras, gestión de almacenes e inventarios y la generación de informes de inteligencia de negocio [25].

➤ **Compiere**

Es una sofisticada solución de negocios Open Source que se ha posicionado como una fuerte alternativa a los productos propietarios, es decir a aquellas aplicaciones que son desarrolladas y comercializadas por una organización comercial. Compiere maneja los siguientes procesos de negocio como son la Requisición a Pago y Administración de la Cadena de Suministro. Compiere proporciona una completa solución de ERP de punta a punta que automatiza los procesos desde la contabilidad a la compra, la realización de las órdenes, la producción y el almacenamiento [25].

1.3.1. Valoración de los sistemas analizados

Los sistemas descritos anteriormente presentan un grupo de funcionalidades y características relacionadas a la gestión de compras. El análisis de las características que presentan permiten obtener una visión sobre la propuesta de solución. La Tabla 1 muestra el comportamiento de los indicadores seleccionados a partir de las necesidades del cliente y teniendo en cuenta las políticas de soberanía e independencia tecnológica.

Tabla 1 Comparación de sistemas estudiados

Sistemas	Multiplataforma	Software libre	Soporte	Gestión de proveedores	Gestión de pedidos y solicitudes de compras	Gestión de contratos	Facturación
Sap Business One	Sí	No	Sí	Sí	Sí	Sí	Sí
Assets NS	Sí	No	Sí	Sí	No	Sí	Sí
ERP Odoo	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Fullstep	Sí	No	Sí	Sí	Sí	Sí	Sí
Microsoft Dynamics	Sí	No	Sí	Sí	Sí	Sí	Sí
Openbravo	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Compiere	Sí	Sí	Sí	Sí	Sí	Sí	Sí

Después de realizar un estudio de los sistemas mencionados anteriormente, se puede concluir que, Sap Business One, Fullstep y Microsoft Dynamics son sistemas que poseen grandes ventajas y funcionalidades, pero que no se pueden tener en cuenta por ser software privativo. Además, Assets NS a pesar de ser también software privativo, no gestiona pedidos ni solicitudes de compras. Los sistemas Openbravo y Compiere son ERP de software libre, ambos presentan todas las ventajas de un ERP de código abierto. Openbravo a pesar de ser de software libre, es necesario pagar una cuota que depende de su uso en la nube. También al analizar el alcance sobresale que Openbravo es recomendado para su uso en medianas y pequeñas empresas, mientras que el mercado de Odoo alcanza a grandes empresas. Por otra parte, Compiere tiene el código fuente y las diferentes aplicaciones que pueden ser extendidos dentro de la aplicación mediante la adicción de componentes modulares. Sin embargo, la documentación y el soporte solo están disponibles mediante pago. A nivel de módulos y de posibilidades ofrecidas, Compiere es similar a Openbravo. Compiere ofrece mejores soluciones en los módulos compra y almacenamiento, pero como resultado del estudio entre Compiere

y Odo, Odo resultó claramente superior debido a las facilidades de trabajo e integración de sus componentes. Por lo que se decide utilizar como marco de trabajo para el desarrollo de la solución.

Una de las grandes ventajas que presenta Odo es el uso del marco de trabajo OpenObject, que presenta un proceso denominado desarrollo rápido de aplicaciones, proporcionando un desarrollo acelerado. Otra de las posibilidades que brinda es la integración con herramientas de negocio, utiliza un flujo de trabajo flexible y dinámico, permitiendo agregar funciones y módulos. En CEIGE se utiliza como marco de trabajo para el desarrollo de aplicaciones empresariales, por lo que es un elemento más a tener en cuenta para su utilización. Luego de analizar el módulo de Compras de Odo se llegó a la conclusión de que este no resolvía las solicitudes realizadas por el cliente por lo que se decide realizar un nuevo sistema.

Una vez definida la propuesta se describe la metodología que guiará el proceso de desarrollo, así como los lenguajes y herramientas que permitieron realizar los productos de trabajo.

1.4. Metodología de desarrollo de software

Para desarrollar un proyecto de software se debe establecer un enfoque disciplinado y sistemático. Es necesario hacer uso de una metodología que guíe el ciclo de vida de un software para que el mismo tenga éxito. Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de crear un determinado producto.

La UCI propone para el desarrollo de sus proyectos de software la metodología AUP-UCI. El Proceso Unificado Ágil de *Scott Ambler* o *Agile Unified Process* (AUP) en inglés es una versión simplificada del *Proceso Unificado de Rational* (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. [26].

De las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición) el ciclo de vida de los proyectos de la UCI mantiene la fase de Inicio, unifica las restantes tres fases en una sola, llamada Ejecución y se agrega la fase de Cierre. Además, se define para los proyectos de la UCI, siete disciplinas las cuales son nombradas como: Modelado de Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas Internas, Pruebas de Liberación y Pruebas de Aceptación. En la presente investigación se desarrollaron las siguientes: Modelado de Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas Internas y Pruebas de Aceptación.

Escenarios de la metodología

La metodología posee cuatro escenarios para modelar el sistema en los proyectos. En el presente trabajo se utiliza el escenario tres, este escenario es aplicado a proyectos que hayan evaluado el

negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad.



Figura 1 Escenario tres de la metodología

1.5. Notaciones y lenguajes

Las notaciones y lenguajes en los procesos llevan ya varias décadas usándose en numerosos campos de la industria. Su objetivo principal siempre ha sido la búsqueda de costos y tiempos óptimos. Sin embargo, el uso de ellos es relativamente reciente dentro del ámbito de la Ingeniería de Software [27]. A continuación, se muestran los lenguajes y notaciones empleados:

- **BPMN 2.0**

BPMN (*Business Process Model and Notation*, por sus siglas en inglés) es una notación gráfica que describe la lógica de los pasos de un proceso de negocio. El objetivo principal de BPMN es proporcionar una notación estándar fácilmente comprensible por todos los interesados del negocio. Estos incluyen los analistas de negocios que crean y perfeccionan los procesos, los desarrolladores técnicos responsables de implementarlos y los gerentes de negocios que los monitorean y administran. BPMN es un estándar para el modelado de procesos de negocio, que proporciona una notación gráfica para especificar procesos de negocio en un Business Process Diagram (BPD). BPMN está diseñado para cubrir muchos tipos de modelado y permite la creación de segmentos de proceso, así como procesos de negocio de extremo a extremo, en diferentes niveles de fidelidad [28]. Se utilizó para modelar los procesos de una manera unificada y estandarizada brindando un entendimiento común entre el cliente y el equipo de desarrollo.

- **UML 2.4**

UML (*Unified Modeling Language*, por sus siglas en inglés), es un estándar aprobado por la norma ISO (*International Organization for Standardization*), que, en la ingeniería del software, se utiliza como diagramado para representar el comportamiento y las funcionalidades de un programa. En contexto, estos diagramas son un medio de comunicación entre los profesionales involucrados en los diferentes procesos de análisis, diseño y desarrollo; permitiendo comunicar y clarificar requisitos, que se solucionarán con la creación de determinado software. En términos generales, los diagramas UML tienen las siguientes características [29]:

- Muestran una idea general del software a desarrollar.
- Muestran posibles comportamientos que podría tener el programa al ejecutar tareas específicas
- Dan una perspectiva sobre todos los aspectos que quiere cubrir la aplicación.
- Permiten mejorar o corregir posibles inconformidades o puntos imprecisos en el levantamiento de los requisitos.
- Permiten visualizar y realizar los ajustes necesarios en la etapa de desarrollo.

Se utilizó el lenguaje de modelado UML pues permite que el modelo y el código estén actualizados, manteniendo la visión en el diseño de la estructura de un proyecto.

- **XML 1.0**

XML (por sus siglas en inglés, *eXtensible Markup Language*), describe una clase de objetos de datos llamados documentos XML y describe parcialmente el comportamiento de los programas informáticos que los procesan. XML es una aplicación perfil o forma restringida del Lenguaje de Marcado Estándar Generalizado (SGML) [ISO 8879]. Por construcción, los documentos XML son documentos conformes a SGML y ha sido diseñado para facilitar la implementación y para la interoperabilidad con SGML y HTML [30].

Este lenguaje se utiliza para la generación de las vistas y reportes, pues es el lenguaje de marcas que utiliza Odoo.

- **Python 3.6**

Python es un lenguaje de programación que cuenta con facilidades para la programación orientada a objetos, imperativa y funcional, por lo que se considera un lenguaje multi-paradigmas. Es un lenguaje de alto nivel que contiene implícitas algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, que permiten realizar algunas tareas complejas en pocas líneas de código y de manera legible. [31]. El marco de trabajo Odoo utiliza el lenguaje Python, el cual será utilizado en el desarrollo de la solución.

- **PostgreSQL 10.0**

Es un sistema de gestión de bases de datos relacional orientado a objetos, el cual incluye características como herencia, restricciones, tipos de datos, reglas e integridad transaccional. Tiene soporte total para transacciones, disparadores, vistas, procedimientos almacenados y almacenamiento de objetos de gran tamaño. Se destaca en ejecutar consultas complejas, consultas sobre vistas, subconsultas y joins. Permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Utiliza el modelo cliente servidor y es un manejador de base de datos de código abierto liberado bajo la licencia BSD8 (*Berkeley Software Distribution*). PostgreSQL está diseñado para

administrar grandes volúmenes de datos [21]. Por las ventajas que posee PostgreSQL es utilizado por Odoos y se tuvo en cuenta en este trabajo para el desarrollo de la solución.

1.6. Herramientas y tecnologías

En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que pueden servir de base para la organización y desarrollo de software. Típicamente, pueden incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto [32].

- **Visual Paradigm 8.0**

Visual Paradigm es una herramienta de Ingeniería de Software Asistida por Computación (CASE). Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Se caracteriza por tener disponibilidad en múltiples plataformas y se enfoca al negocio generando un software de mayor calidad. Esta herramienta permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. También realiza la distribución automática de diagramas, así como la reorganización de las figuras y conectores de los diagramas UML [33]. Se utilizó esta herramienta pues permite realizar todos los tipos de diagramas, además de que permite modelar el negocio orientado a procesos usando la notación BPMN.

- **Pencil Project**

Herramienta de código abierto que puede ser instalada como una extensión (*addon*) en el navegador Firefox. Es una solución muy flexible que ofrece una librería formada por más de 50 elementos gráficos, que admiten una edición posterior y la incorporación de nuevos elementos gráficos externos. Provee la exportación a *html*, *png*, *openoffice*, *word*, *pdf* y *pencil*; pero sólo importa su propio formato [34]. Se utilizó Pencil Project como herramienta para realizar los prototipos porque ofrece una visualización más detallada y cercana a las interfaces de usuario.

- **Odoos 12.0**

Odoos es un sistema de ERP integrado de código abierto, producido por OpenERP S.A. Actualmente el nombre de OpenERP ha sido cambiado por el de Odoos. Entre las principales características de Odoos se encuentra su escalabilidad, modularidad y flexibilidad. Posee una comunidad activa de desarrolladores, con presencia en varios países. Odoos internamente usa un modelo de flujos de trabajo (*workflow*), con arquitectura en tres capas. Odoos es una plataforma de código abierto altamente

integrada que mantiene la integridad de la empresa con módulos personalizables. Además, permite integrar sus módulos de manera ágil y flexible [23].

- **PyCharm 2018.2.3**

PyCharm es un IDE (entorno de desarrollo integrado) multiplataforma utilizado en el ámbito de la programación. Es uno de los entornos de desarrollo más completos para Python. Es parte de la colección de herramientas de programación ofrecidas por *JetBrains*, que cuenta con entornos para construir código en distintos idiomas como PHP y Ruby. PyCharm tiene un editor inteligente, que permite completar código con algunos atajos de teclado. Una de las características notables de PyCharm es la posibilidad que tiene de refactorizar el código, que, en términos generales, significa modificar el código sin comprometer la ejecución del mismo. El editor de código inteligente de PyCharm brinda soporte de primera clase para Python [29]. Se utilizó el PyCharm como entorno de desarrollo por ser uno de los entornos más completos para el desarrollo en Python.

- **PgAdmin 1.22.2**

Es una aplicación gráfica de diseño y manejo de base de datos PostgreSQL. PgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, Mac OSX y Solaris [35].

- **Marco de trabajo OpenObject 1.0**

Es un marco de trabajo de código abierto, inteligente, profesional y rápido en el desarrollo de aplicaciones en Python. Está basado en la arquitectura modelo-vista-controlador, además de poseer Inteligencia de Negocios, Mapeador Relacional de Objetos(ORM), casos de pruebas, motores de flujos de trabajo, grabador de módulos, envases de módulos, entre otros. OpenObject ofrece, en un solo paquete el componente básico para la construcción de una aplicación de negocios: multilinguaje, servicios web, campos traducibles, ingeniería de reportes, PostgreSQL, Python como lenguaje de programación y licencia GNU AGPL v3 [36]. Se utilizó por ser el ORM que utiliza OdoO para el manejo de los datos.

Luego de definir y estudiar la metodología, los lenguajes y herramientas se procede a describir cómo se validaron los productos de trabajo realizados en cada disciplina del desarrollo del software.

1.7. Validación

Validación es el nombre que se da a los procesos de comprobación y análisis que aseguran que el software que se desarrolla está acorde a su especificación y cumple las necesidades de los clientes

[37]. En este epígrafe se explica cómo la solución fue validada en cada una de las disciplinas durante su desarrollo.

1.7.1. Modelado de negocio

Según el escenario número tres de la metodología a desarrollar en esta disciplina se generan los siguientes artefactos:

- Modelo conceptual
- Descripción de procesos de negocio

Estos productos de trabajo se validan con el cliente a partir de revisiones que se realizan a las descripciones de los procesos de negocio. Se debe constatar que los procesos modelados se corresponden con los procesos que se ejecutan en la Dirección de Compras de la UCI.

1.7.2. Requisitos

La validación de los requisitos trata de mostrar que éstos realmente definen el sistema que el cliente desea. [38]. Existen técnicas para la validación de los requisitos, a continuación se describen las más utilizadas [39]:

- **Revisiones Técnico Formales:** los requisitos son analizados sistemáticamente por un equipo de revisores, incluyendo el analista principal del proyecto y su equipo de desarrolladores.
- **Construcción de prototipos:** se muestra un modelo ejecutable del sistema a los usuarios finales y a los clientes, los cuales pueden experimentar con este modelo para ver si cumple sus necesidades reales.
- **Generación de casos de pruebas:** los requisitos deben poder probarse. Las pruebas para éstos se conciben como parte del proceso de validación y a menudo revelan los problemas en los requisitos.

Para la validación de las descripciones de requisitos por procesos se utilizaron las técnicas de construcción de prototipos y la generación de los diseños de casos de prueba.

1.7.3. Análisis y diseño

Lorenz y Kidd concentran las métricas basadas en clases en cuatro categorías: tamaño, herencia, valores internos y valores externos. Las métricas orientadas a tamaños para una clase se centran en cálculos de atributos y de operaciones para una clase individual, y promedian los valores para el sistema en su totalidad. Las métricas basadas en herencia se centran en la forma en que se reutilizan las operaciones a lo largo y ancho de la jerarquía de clases. Las métricas para valores internos de clase examinan la cohesión y asuntos relacionados con el código, y las métricas orientadas a valores externos examinan el acoplamiento y la reutilización. Estas métricas están enfocadas a las características internas del diseño orientado a objeto y de esta manera, contribuyen a asegurar la

mantenibilidad de los productos de software [40]. Para la validación del diseño se utilizó la categoría *tamaño* basado en las métricas Tamaño Operacional de Clase (TOC) y las Relaciones entre Clases (RC).

1.7.4. Implementación y pruebas

En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema. En la disciplina de pruebas se realiza una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funcione tal como está establecido, y verificando que se cumplan los requisitos [26].

Pruebas unitarias (caja blanca)

La prueba de caja blanca del software se basa en el examen cercano de los detalles de procedimiento. Las rutas lógicas a través del software y las colaboraciones entre componentes se ponen a prueba al revisar conjuntos específicos de condiciones y/o bucles. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que: 1) garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, 2) revisen todas las decisiones lógicas en sus lados verdadero y falso, 3) ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y 4) revisen estructuras de datos internas para garantizar su validez [41].

Técnica de Camino básico

El método de ruta básica permite al diseñador de casos de prueba derivar una medida de complejidad lógica de un diseño de procedimiento y usar esta medida como guía para definir un conjunto básico de rutas de ejecución. Los casos de prueba derivados para revisar el conjunto básico tienen garantía para ejecutar todo enunciado en el programa, al menos una vez durante la prueba [41].

Antes de considerar el método de ruta básica, debe introducirse una notación simple para la representación del flujo de control, llamado gráfico de flujo (o gráfico de programa). El gráfico de flujo muestra el flujo de control lógico que usa la notación. Luego se determina la complejidad ciclomática $V(G)$ del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo, es decir, es cualquier camino dentro del código que introduce por lo menos un nuevo conjunto de sentencias de proceso o una nueva condición [41].

La complejidad se calcula en una de tres formas:

1. $V(G) = R$, donde R es el número de regiones que favorece a estimar el valor de la complejidad ciclomática.
2. $V(G) = E - N + 2$, donde E es el número de aristas, y N el número de nodos de la gráfica de flujo.
3. $V(G) = P + 1$, donde P es el número de nodos predicados incluidos en el grafo.

Pruebas funcionales (caja negra)

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca [41].

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes: 1) funciones incorrectas o faltantes, 2) errores de interfaz, 3) errores en las estructuras de datos o en el acceso a bases de datos externas, 4) errores de comportamiento o rendimiento y 5) errores de inicialización y terminación [41].

Técnica de Partición de equivalencia

La partición de equivalencia es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Un caso de prueba ideal descubre de primera mano una clase de errores (por ejemplo, procesamiento incorrecto de todos los datos carácter) que de otro modo podrían requerir la ejecución de muchos casos de prueba antes de observar el error general [41].

El diseño de casos de prueba para la partición de equivalencia se basa en una evaluación de las clases de equivalencia para una condición de entrada. Las clases de equivalencia pueden definirse de acuerdo con los siguientes lineamientos:

1. Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos inválidas.
2. Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y dos inválidas.
3. Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una inválida.
4. Si una condición de entrada es booleana, se define una clase válida y una inválida.

Al aplicar los lineamientos para la derivación de clases de equivalencia, pueden desarrollarse y ejecutarse los casos de prueba para cada ítem de datos del dominio de entrada. Los casos de prueba se seleccionan de modo que se revise a la vez el número más grande de atributos de una clase de equivalencia [41].

1.8. Conclusiones parciales

A partir de la bibliografía consultada sobre sistemas que apoyan el proceso de gestión de compras, se determinó el uso de ERP Odoó como propuesta de solución en la investigación. Esta plataforma permite el desarrollo rápido de los módulos basados en OpenObject, utiliza un flujo de trabajo flexible y dinámico para el trabajo con funciones, módulos y su integración a los ya existentes.

Se determinó el uso de la Metodología de desarrollo para la Actividad productiva de la UCI en la propuesta de solución, debido a que en cada una de las disciplinas utilizadas de la fase de Ejecución se definen las actividades y los productos de trabajo necesarios para lograr la comunicación en el equipo de desarrollo y la calidad del producto.

La utilización de Odoó como plataforma de desarrollo implica el uso de lenguajes, tecnologías y herramientas de código abierto, aprovechando las bondades que ofrecen estas herramientas para el desarrollo de software y su distribución.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

2.1. Introducción

En este capítulo se realiza el modelado de negocio, requisitos, análisis y diseño de los módulos Gestión de necesidad, Contratación, Seguimiento de la demanda y Recepción de mercancía para la Dirección de Compras de la UCI. Para ello se obtienen las descripciones de procesos de negocio y el modelo conceptual. Se identifican y describen los requisitos funcionales y no funcionales. Se realizó el diagrama de clases del diseño con estereotipos web y, por último, se valida el diseño a través de las métricas seleccionadas.

2.2. Modelado del negocio

Para modelar el negocio se utilizó el escenario tres de la metodología de desarrollo para la actividad productiva de la universidad, con el objetivo de comprender los procesos involucrados en la gestión de compras. El Modelado del negocio es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito [26].

2.2.1. Descripción de procesos de negocio

La gestión de compras en la UCI abarca diferentes subprocesos que permiten llevar a cabo toda la cadena de aprovisionamiento para satisfacer las necesidades de las áreas de la universidad. A continuación, se describen los procesos relacionados con la investigación:

- **Gestión de necesidad**

Este proceso inicia cuando surge una necesidad presentada por un área de la universidad. Una vez que la DC tiene las informaciones correspondientes a dicha necesidad, se procede a su análisis, revisando primeramente si existe en el almacén. El personal designado de la DC determina qué productos y/o servicios no están disponibles para proceder a registrar esta necesidad en el registro de demanda. Seguidamente se asigna la necesidad según su categoría o tipo al comprador correspondiente según sus funciones con su información correspondiente.

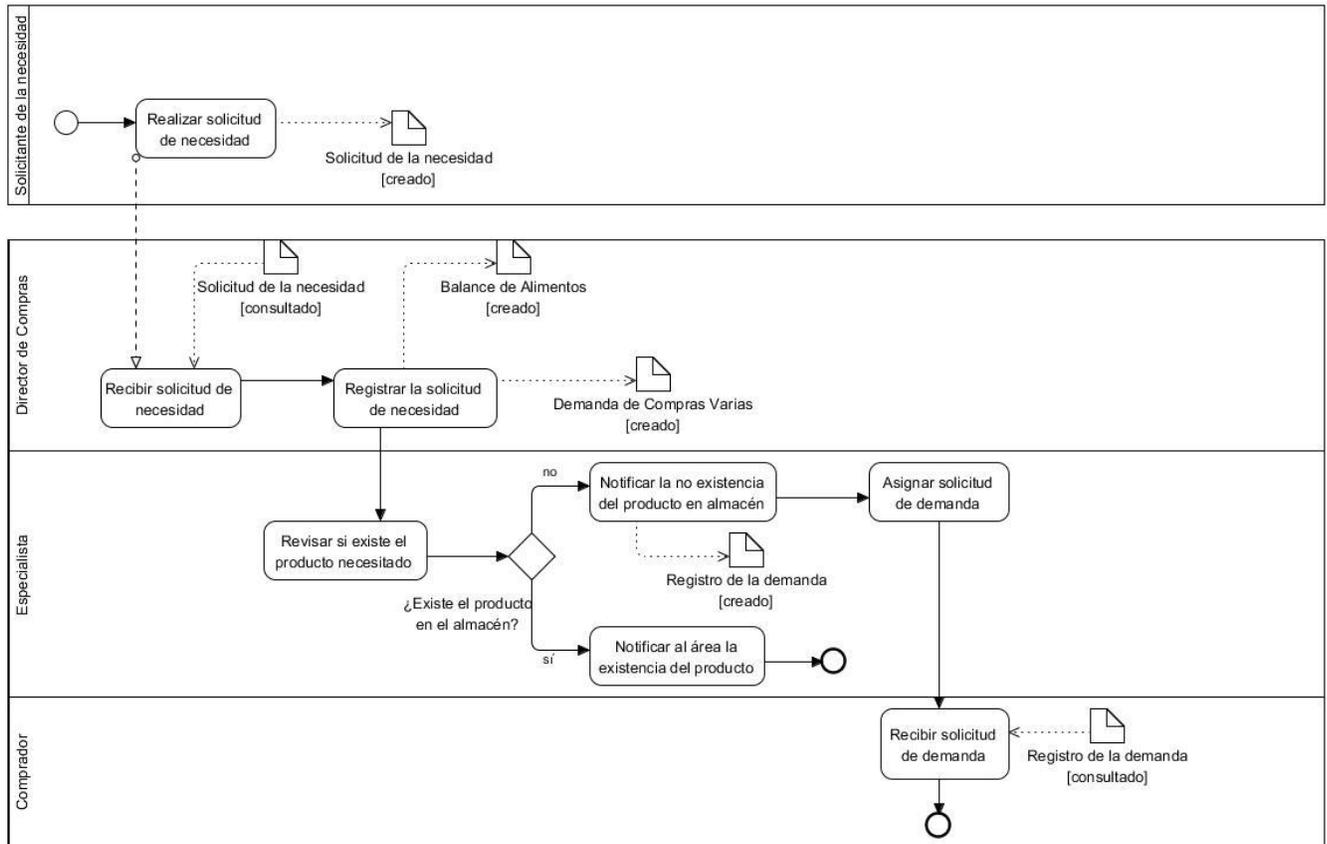


Figura 2 Descripción del proceso Gestión de necesidad

El modelado del negocio comprende además los subprocesos Gestión de Contratación, Seguimiento de la demanda y Recepción de mercancía. Las descripciones de estos subprocesos pueden ser consultadas en el Anexo 2.

Para una mayor comprensión de los términos tratados en las descripciones de procesos de negocio, se realizó el modelo conceptual. En este producto de trabajo se detallan cada uno de los conceptos utilizados y sus relaciones.

2.2.2. Modelo conceptual

El modelo conceptual está constituido por un conjunto de conceptos y las relaciones que se establecen entre ellos. Es una representación visual, mediante un determinado lenguaje, de los conceptos u objetos del mundo real de un dominio de interés. Explica los conceptos más significativos en un dominio del problema, mostrando los conceptos, las asociaciones entre conceptos y los atributos de los conceptos [3].

La realización del modelo conceptual del proceso de Compras en la Universidad de las Ciencias Informáticas permitió obtener una representación gráfica de los conceptos que se manejan en dicho

proceso, así como las relaciones que existen entre ellos. La realización del modelo de dicho proceso para el sistema que se desea desarrollar permitió identificar las clases conceptuales del negocio, así como los atributos y las relaciones que existen entre dichas clases. A continuación, se muestra dicho modelo conceptual:

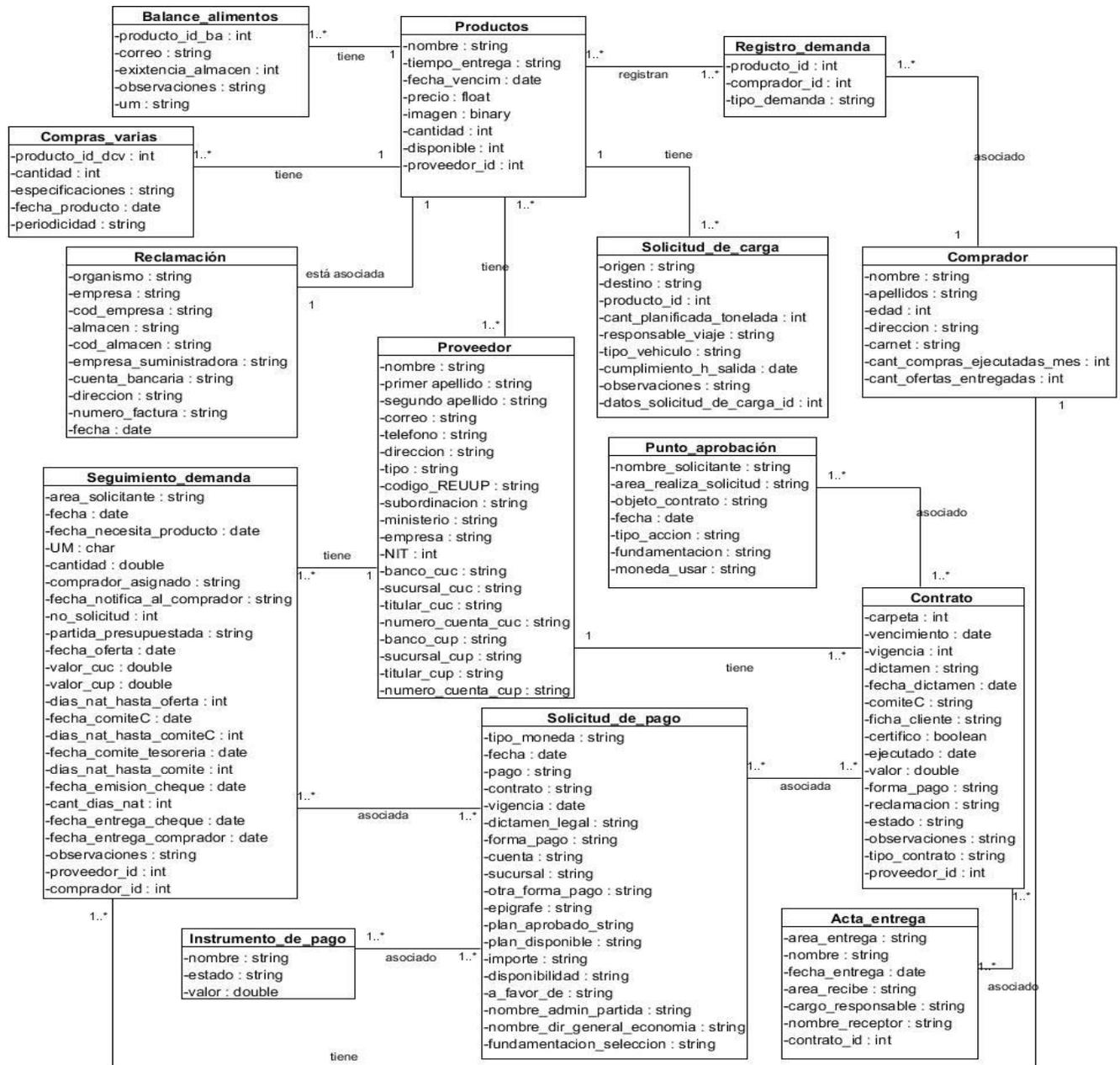


Figura 3 Modelo conceptual

2.3. Requisitos del sistema

Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requisitos reflejan las necesidades de los clientes de un sistema que ayudan a resolver algún problema [38]. Una vez realizadas las descripciones de procesos y el modelo conceptual es posible identificar y describir los requisitos funcionales del sistema.

2.3.1. Técnicas utilizadas para la obtención de requisitos

En la etapa de obtención de requisitos los ingenieros de software trabajan con los clientes y los usuarios finales del sistema para determinar el dominio de la aplicación, analizando los servicios que debe proporcionar el sistema [38].

Se aplicaron las siguientes técnicas para la obtención de requisitos:

- **Tormenta de ideas:** es una herramienta de trabajo grupal que facilita el surgimiento de nuevas ideas sobre un tema o problema determinado. Se utilizó la tormenta de ideas para la captura de requisitos logrando un mejor entendimiento de las necesidades del cliente.
- **Entrevista:** es de gran utilidad para obtener información cualitativa como opiniones, descripciones de actividades. Se aplicó realizando preguntas relacionadas con el proceso al Director de Compras de la UCI, para así obtener la información deseada.

2.3.2. Requisitos funcionales

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer, estos dependen del tipo de software que se desarrolle, de los posibles usuarios y del enfoque general tomado por la organización al redactar requerimientos [38].

Luego de aplicar las técnicas de tormenta de ideas y entrevista para la obtención de los requisitos, se obtuvieron 128 requisitos funcionales, resumiéndose en 21 agrupaciones de requisitos (los requisitos correspondientes a cada agrupación se encuentran en el anexo x)

- Gestionar necesidades del balance de alimentos
- Gestionar nomenclador de área para balance de alimentos
- Gestionar nomenclador de unidad de medida para balance de alimentos
- Gestionar necesidades de compras varias
- Gestionar nomenclador de área para compras varias
- Gestionar nomenclador de unidad de medida para compras varias
- Gestionar demanda
- Gestionar proveedor

- Gestionar nomenclador de tipo de proveedor
- Gestionar comprador
- Gestionar producto
- Gestionar nomenclador de clasificación de producto
- Gestionar seguimiento de la demanda
- Gestionar solicitud de pago
- Gestionar nomenclador tipo de instrumento de pago
- Gestionar contrato
- Gestionar nomenclador de tipo de contrato
- Gestionar punto de aprobación
- Gestionar acta de entrega
- Gestionar solicitud de transporte de carga
- Gestionar reclamación

Salidas del sistema

Una salida o reporte es aquella que se entrega al usuario, donde los datos requieren un procesamiento extenso antes que se conviertan en salida adecuada del sistema. Los reportes como salidas del sistema son útiles, comprenden varios objetivos que un analista de sistemas debe tener en cuenta para el diseño de la salida [41].

Tabla 2 Salidas del sistema

1.	Obtener necesidades aprobadas y no aprobadas
2.	Obtener necesidades por áreas
3.	Demandas asignadas a un comprador en un período
4.	Necesidades por período
5.	Listado de proveedores
6.	Listado de proveedores por producto
7.	Contratos por proveedores
8.	Perfil de proveedores
9.	Listado de contratos vencidos
10.	Informe de reclamación
11.	Acta de entrega
12.	Punto de aprobación
13.	Solicitud de pago

2.3.3. Descripción de requisitos funcionales

La descripción de los requisitos funcionales de los módulos Gestión de necesidad, Contratación, Seguimiento de la demanda y Recepción de mercancía, permitió obtener una versión completa del comportamiento del sistema a desarrollar, en dichas descripciones se reflejan los flujos por los que transita el requisito, la información que muestra, así como las restricciones que posee. A continuación, se muestra un ejemplo de la descripción del requisito Crear proveedor, perteneciente a la agrupación de requisitos Gestionar necesidad de balance de alimento:

Tabla 3 Descripción del requisito Crear necesidad de balance de alimento

Precondiciones	El usuario se ha autenticado en el sistema y tiene permiso para realizar esta acción.
Flujo de eventos	
Flujo básico Crear necesidad de balance de alimento	
1.	Se introducen los datos de la necesidad: Nombre* Correo* Área* Cargo* Descripción Desglose de Producto*: UM* Cantidad* Fecha* Observaciones
2.	El sistema valida (ver validación 1) los datos introducidos.
3.	Si los datos son correctos el sistema los registra.
4.	El sistema confirma el registro de los datos.
5.	Concluye el requisito.
Pos-condiciones	
	Se registró en el sistema una nueva necesidad
Flujos alternativos	
Flujo alternativo 3.a Información incorrecta	
	El sistema señala los datos incorrectos y permite corregirlos
1	El usuario corrige los datos
3	Volver al paso 2 del flujo básico

Pos-condiciones		
	N/A	
Flujo alternativo 3.b Información incompleta		
1	El sistema señala los datos vacíos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 2 del flujo básico	
Pos-condiciones		
1	N/A	
Flujo alternativo *.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se registran los datos.	
Validaciones		
1	Validar que se carguen de forma automática el nombre del solicitante de la necesidad que está logueado en el sistema.	
2	Validar que se carguen de forma automática el correo del solicitante de la necesidad que está logueado en el sistema.	
3	Validar que se carguen de forma automática el cargo del solicitante de la necesidad que está logueado en el sistema.	
4	Validar que se seleccione el campo obligatorio área a partir del listado que se carga automático.	
5	Validar que se inserte el campo descripción por el usuario, verificando que sea campo de texto y que al menos el primer caracter se escriba con mayúscula.	
6	Validar que se seleccione el campo obligatorio producto a partir del listado que se carga automático.	
7	Validar que se seleccione el campo obligatorio UM a partir del listado que se carga automático, asociado al tipo de producto que puede ser de compras varias o de alimentos.	
8	Validar que se inserte el campo obligatorio cantidad por el usuario, permitiendo solo caracteres numéricos.	
9	Validar que se seleccione o inserte el campo obligatorio fecha, verificando el formato date.	
10	Validar que se inserte el campo observaciones por el usuario, verificando que sea campo de texto y que al menos el primer caracter se escriba con mayúscula.	
Conceptos	Producto	Visibles en la interfaz:

		Nombre* Correo* Área* Cargo* Descripción Desglose de Producto*: UM* Cantidad* Fecha* Observaciones
Requisitos especiales	N/A	
Asuntos pendientes	N/A	
Prototipo elemental de interfaz gráfica de usuario		

2.3.4. Requisitos no funcionales

Los requisitos no funcionales especifican las propiedades o cualidades que debe tener la solución a desarrollar, son aquellos que no se refieren directamente a las funciones específicas que proporciona

el sistema, sino a las propiedades emergentes de éste. Además, representan las características que hacen al producto atractivo, usable, rápido o confiable [38]. A continuación, se listan algunos de ellos:

Tabla 4 Requisitos no funcionales

No	Requisitos no funcionales
Seguridad	
RNF1	El sistema manejará la seguridad de acceso y administración de usuarios mediante el otorgamiento de privilegios y roles, así como la asignación de perfiles.
RNF2	Se concederá acceso al sistema a partir de un nombre de usuario y una contraseña.
RNF3	El sistema concederá acceso a cada usuario autenticado solo a las funciones que le estén permitidas, de acuerdo a la configuración del sistema.
Usabilidad	
RNF4	El idioma de todas las interfaces de la aplicación será en español.
RNF5	Los errores cometidos por el usuario les serán notificados.
RNF6	En cada formulario del sistema los campos obligatorios serán señalados en azul.
RNF7	Todos los mensajes de error del sistema deberán incluir una descripción textual del error.
Confiabilidad	
RNF8	El sistema debe prever en cada formulario, la entrada de datos incorrectos.
RNF9	El sistema deberá detectar fallos internos y notificar al usuario de la ocurrencia de estos.
Software	
RNF10	Como marco de trabajo se utiliza la plataforma de desarrollo Odoo 12.0.
RNF11	Se empleará como Gestor de Base de Datos, PostgreSQL 10.0.
RNF12	El sistema se desarrollará con tecnología Python 3.6.
RNF13	Sistemas operativos compatibles: Windows XP o superior, Linux.
Hardware	
RNF14	El sistema para su instalación en las máquinas clientes requiere: Procesador: 1.40 GHZ. RAM: 512 MB (recomendado 1024 MB).
RNF15	El sistema para su instalación en el servidor de aplicación requiere: Procesador: 3.00 GHZ. RAM: 1GB (recomendado 4 GB). Disco duro: 160 GB (recomendado 500 GB)

2.3.5. Validación de requisitos

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto [42].

Se utilizaron las siguientes técnicas para la validación de los requisitos:

- **Construcción de prototipos:** una vez identificados y descritos cada uno de los requisitos se diseñaron prototipos de interfaces de usuario utilizando como herramienta de modelado Pencil. Para realizar un prototipo de interfaz es necesario poder contar con una descripción lo más clara y precisa posible, las mismas son un reflejo del futuro sistema. Mediante los prototipos de interfaz de usuario se obtuvo una visión inicial del sistema a implementar demostrando cómo se van a disponer posteriormente los conceptos que intervienen en el mismo. Estos prototipos fueron revisados, llegando a un entendimiento de la información especificada que permitió la aprobación de cada uno de los requisitos por parte del cliente.
- **Generación de casos de pruebas:** se generaron 128 casos de pruebas con el objetivo de probar y verificar el cumplimiento de cada uno de los requisitos descritos, los mismos serán tratados en el epígrafe 3.3.2 del próximo capítulo.

Concluida la validación de los requisitos finaliza esta disciplina, permitiendo el inicio del Análisis y diseño con mayor definición acerca de las funcionalidades que debe implementar el sistema. Con los requisitos funcionales aprobados se facilita el diseño de la arquitectura y su implementación.

2.4. Diseño de la propuesta de solución

Durante el desarrollo de esta disciplina se modela el sistema, teniendo en cuenta la arquitectura, para que soporte tanto los requisitos funcionales como los no funcionales. Propiciando que dichos elementos sirvan de base a la etapa de implementación.

2.4.1. Diseño Arquitectónico

El diseño arquitectónico representa la estructura de los datos y los componentes del programa que se requieren para construir un sistema basado en computadora. Constituye el estilo arquitectónico que tendrá el sistema, la estructura y las propiedades de los componentes que ese sistema comprende, y las interrelaciones que tienen lugar entre todos los componentes arquitectónicos del sistema [41].

Odoó sigue una arquitectura Modelo-Vista-Controlador (MVC), donde la capa del modelo define la estructura de los datos, la vista describe la interfaz con el usuario y el controlador soporta la lógica de negocio de la aplicación [43]:

Modelo: objetos Python cuyos datos son almacenados en una base de datos PostgreSQL. Los modelos se encuentran dentro de la carpeta models, ejemplos de ellos son persona, necesidad y producto. El mapeo de la base de datos es gestionado automáticamente por Odoor, y el mecanismo responsable por esto es el modelo objeto relacional (ORM- object relational model).

Vista: las vistas en Odoor manejan la presentación visual de los datos representados por el modelo a través de archivos XML. Se encuentran almacenadas dentro de la carpeta views, como proveedor_view.xml, producto_view.xml.

Controlador: el Controlador está compuesto por las clases controladoras, se pone de manifiesto a través del modelo controller.py, el cuál es el encargado de hacer peticiones al modelo cuando se hace alguna solicitud de la información por parte del cliente.

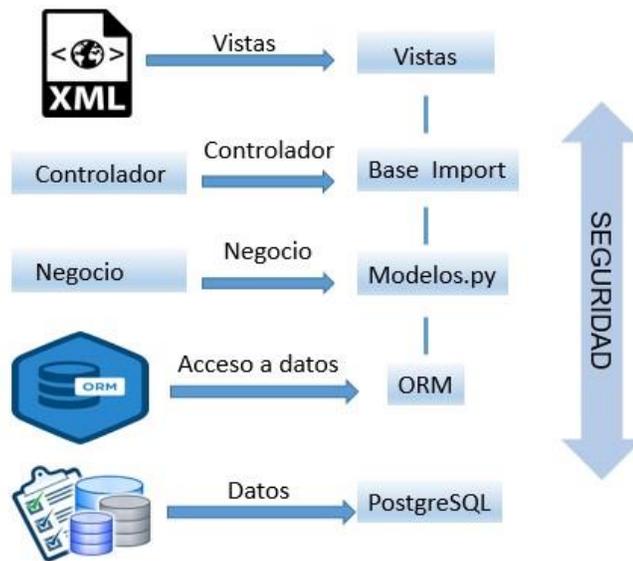


Figura 4 Arquitectura Modelo Vista Controlador

2.4.2. Diagrama de clases con estereotipos web

Las clases del diseño sirven para describir la estructura del sistema mostrando sus clases, atributos y las relaciones entre ellos. A continuación, se presenta el modelo del diseño del requisito “Gestionar proveedor”.

En este caso se utilizan los estereotipos web: las páginas clientes, las servidoras y los formularios. En el siguiente diagrama de clases del diseño con estereotipos web, la Página Cliente Principal le hace una petición a la Página del Servidor (Gestionar Proveedor), esta es la encargada de construir las Páginas Clientes (Crear Proveedor, Editar Proveedor, Suprimir Proveedor, Listar Proveedor y Buscar Proveedor), con la colección de elementos de entrada Formularios, para luego mostrarlos a través de la vista correspondiente a la función solicitada, dígame: crear/editar/suprimir/listar/buscar proveedor.

Para dar respuesta a las peticiones realizadas por la página cliente, el controlador tiene la responsabilidad de realizar las operaciones, sobre la información de las tablas de la base de datos accediendo a las clases del modelo y enviar los resultados. (Ver Figura 5).

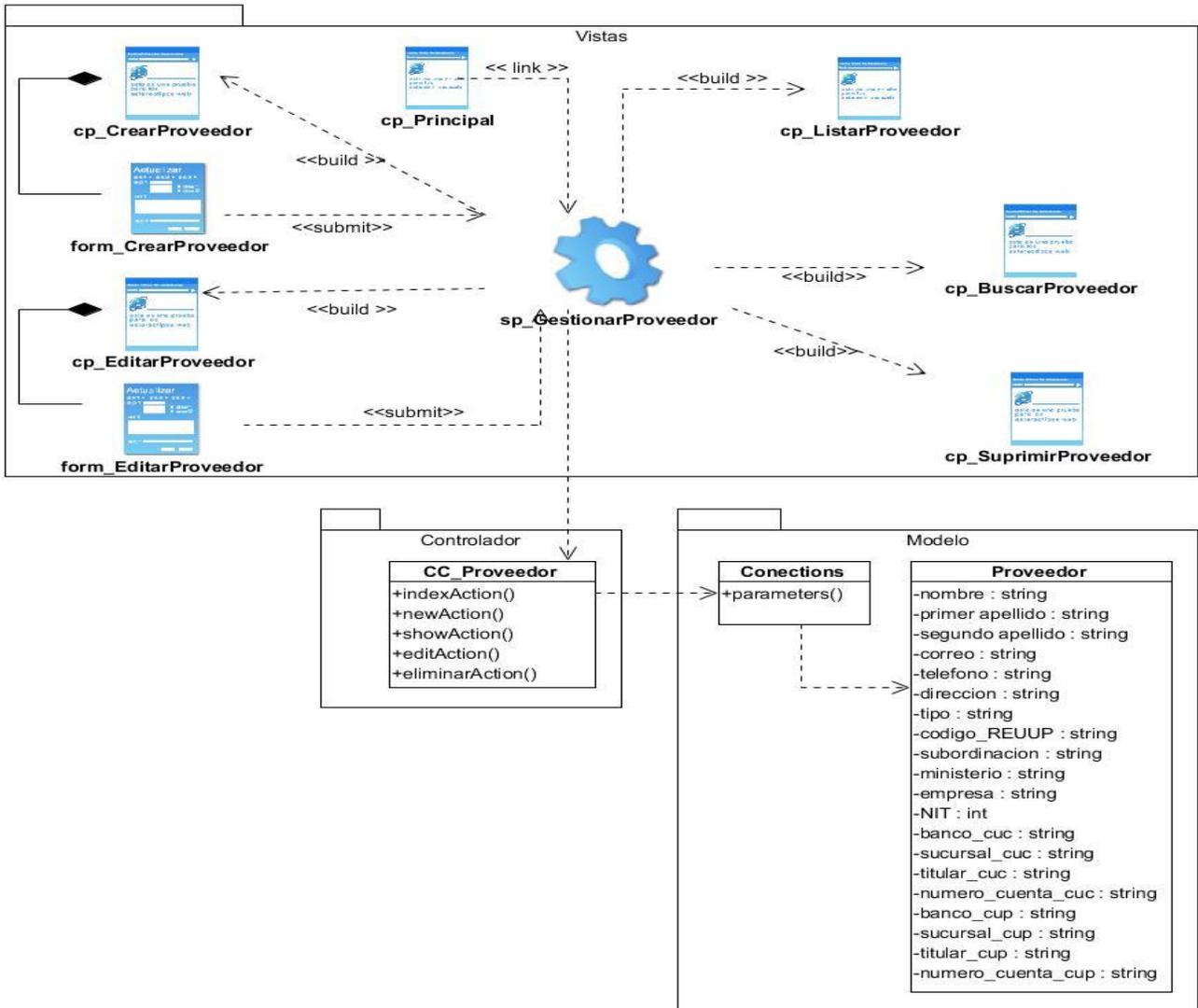


Figura 5 Diagrama de clases del diseño con estereotipos web

2.4.3. Entidades persistentes en Odo

Cuando se desarrolla un módulo en Odo, éste se encuentra relacionado con cualquier objeto que se haya creado dentro del marco de trabajo; en este caso, todo objeto creado dentro de él representa una entidad, o sea, una tabla de base de datos. Odo trabaja con un motor de ORM (*Object Relational Mapping*) que se encarga de manipular los modelos dentro de la base de datos programando en el código [44].

El ORM se encarga de crear las entidades y las relaciones en la base de datos sin necesidad de utilizar sentencias SQL, ni tampoco generar MER (Modelo de Entidad-Relación) en el motor de base de datos. El IDE de PyCharm permite desarrollar los módulos requeridos, creando nuevas clases y usando las heredadas de la plataforma de Odoon [44].

2.4.4. Patrones de diseño

Los patrones son principios y estilos que guían en la creación de un software, que se codifican con un formato estructurado que describe un problema y su solución, acumulados por desarrolladores orientados a objetos con experiencia y otros desarrolladores de software [45].

Patrones GRASP

GRASP (por sus siglas en inglés, Patrones Generales de Software para Asignar Responsabilidades). Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. Se utilizaron los siguientes patrones GRASP, con el objetivo de propiciar la robustez y flexibilidad, permitiendo asignar responsabilidades a las clases [45]:

- **Bajo acoplamiento:** asignar una responsabilidad de manera que el acoplamiento permanezca bajo. Las clases deberán estar relacionadas lo menos posible, de forma tal que, en caso de producirse una modificación en una de ellas, se tenga la mínima repercusión posible. Se observa en la mayor parte del diseño pues las clases fueron creadas teniendo en cuenta que las mismas presentan la menor cantidad de relaciones de dependencia posible.
- **Alta cohesión:** asignar una responsabilidad de manera que la cohesión siempre permanezca alta, o sea, cada clase se encargará de realizar solamente las funciones que estén en correspondencia con la responsabilidad que esta posee. Propone que la información que almacena una clase debe de ser coherente y debe estar, en la medida de lo posible, relacionada con la clase.
- **Experto:** propone asignar una responsabilidad al experto en información, la clase que contiene la información necesaria para realizar la responsabilidad; es un principio de guía básico que se utiliza continuamente en el diseño de objetos. Fue usado en la mayoría de las clases ya que a cada una le fue asignada la responsabilidad en dependencia de la información que esta posee.
- **Creador:** asignar a una clase la responsabilidad de crear una instancia de otra clase cumpliéndose determinados casos. Dicho patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. Este patrón se puede apreciar principalmente en las clases controladoras las cuales necesitan crear objetos de las clases del modelo para acceder a las funcionalidades que estas poseen.

Patrones GOF

Los patrones GOF (por sus siglas en inglés, Gang Of Four) describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación entre clases y la formación de estructuras de mayor complejidad. Nos permiten crear grupos de objetos para ayudarnos a realizar tareas complejas. Existen tres tipos de patrones GOF: de creación, estructurales y de comportamiento [39].

- **Decorador:** su principal objetivo es añadir responsabilidades a objetos concretos de manera dinámica y transparente sin afectar a otros objetos. Este patrón brinda más flexibilidad que la herencia estática y evita que las clases más altas en la jerarquía estén demasiado cargadas de funcionalidad y sean complejas. Se evidencia a través del método *def registrarDemandaBA*, que implementa todos los métodos de esta clase y además añade la responsabilidad de dar valor al campo validado.
- **Observador:** define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. El patrón Observador puede ser utilizado cuando hay objetos que dependen de otro, se puede apreciar en la relación que existe entre Registro de demanda y Comprador.

2.5. Validación del diseño

Las métricas de diseño permiten medir de forma cuantitativa la calidad de los atributos internos del software. Se centran en cuantificar tanto la complejidad, como la funcionalidad y eficiencia inmersa en el desarrollo de software. Inclina sus objetivos a mejorar la comprensión de la calidad del producto, a estimar la efectividad del proceso y mejorar la calidad del trabajo [46].

Las métricas empleadas están diseñadas para evaluar los siguientes atributos de calidad [46]:

- **Responsabilidad:** consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- **Complejidad de implementación:** consiste en el grado de dificultad que tiene implementado un diseño de clases determinado.
- **Reutilización:** consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.
- **Acoplamiento:** consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, está muy ligada a la característica de Reutilización.

- **Complejidad del mantenimiento:** consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costos y la planificación del proyecto.
- **Cantidad de pruebas:** consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (componente, módulo, clase, conjunto de clases, etc.) diseñado.

Para la validación del diseño se aplicaron las métricas Tamaño Operacional de Clases (TOC) y Relaciones entre Clases (RC) debido al conjunto de atributos de calidad de diseño que ambos miden.

Métrica de Tamaño Operacional de Clases (TOC): está dado por el número de métodos asignados a una clase. Mediante el cual se calcula el nivel de Responsabilidad de los métodos, la Complejidad de implementación de los mismos y su Reutilización, a fin de inspeccionar la efectividad del diseño, existiendo una relación directa con los dos primeros e inversa con el último antes mencionado [47]:

Tabla 5 Atributos de calidad evaluados por la métrica TOC

Atributo de calidad	Modo en que lo afecta
Responsabilidad	Aumento del TOC provoca aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Aumento del TOC provoca aumento de la complejidad de implementación de la clase.
Reutilización	Aumento del TOC provoca disminución del grado de reutilización de la clase.

Los criterios y categorías definidos para la evaluación de los atributos de calidad anteriores se presentan en la siguiente tabla:

Tabla 6 Criterios de evaluación para la métrica TOC

Atributos	Categoría	Criterio
Responsabilidad	Baja	\leq Promedio
	Media	Entre Promedio y $2 \cdot$ Promedio
	Alta	$> 2 \cdot$ Promedio
Complejidad e implementación	Baja	\leq Promedio
	Media	Entre Promedio y $2 \cdot$ Promedio
	Alta	$> 2 \cdot$ Promedio

Reutilización	Baja	$> 2 * \text{Promedio}$
	Media	Entre Promedio y $2 * \text{Promedio}$
	Alta	$\leq \text{Promedio}$

Resultados obtenidos al aplicar la métrica TOC:

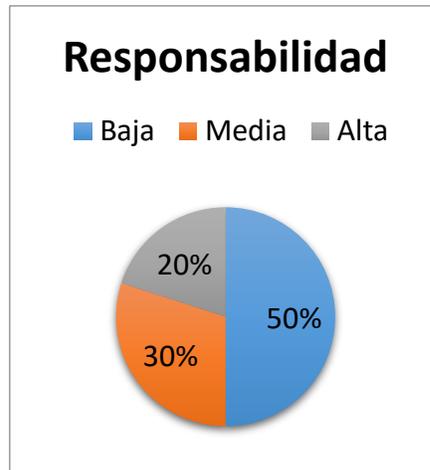


Figura 6 Resultado de la evaluación de la métrica para el atributo responsabilidad

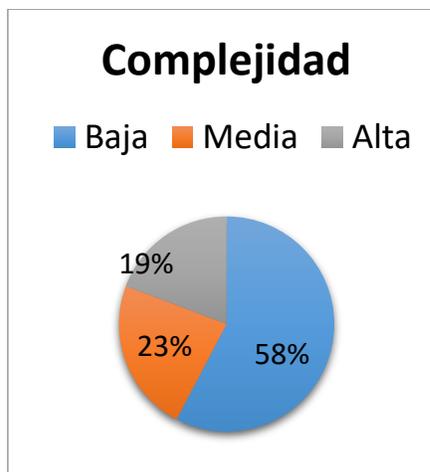


Figura 7 Resultado de la evaluación de la métrica para el atributo complejidad



Figura 8 Resultado de la evaluación de la métrica para el atributo reutilización

Luego de aplicarse la métrica de diseño TOC se obtuvieron resultados que permiten evaluar el diseño propuesto de calidad aceptable. Los atributos de calidad se encuentran en un nivel satisfactorio en las clases, teniendo en cuenta que el 50% de las clases tienen una baja responsabilidad y 58% de complejidad, y a su vez un alto nivel de reutilización (73 %) entre ellas.

Métrica de Relaciones entre Clases (RC): está dado por el número de relaciones de uso de una clase con otra, o sea el número de dependencias que una clase tiene con otra. Mediante la cual se calcula el Acoplamiento, la Complejidad de mantenimiento, la Reutilización y la Cantidad de pruebas a fin de inspeccionar la efectividad del diseño, existiendo una relación directa con los tres primeros e inversa con el último antes mencionado [47]:

Tabla 7 Atributos de calidad evaluados por la métrica RC

Atributos de calidad	Modo en que los afecta
Acoplamiento	Aumento del RC provoca aumento del acoplamiento de la clase.
Complejidad de mantenimiento	Aumento del RC provoca aumento de la complejidad del mantenimiento de la clase.
Reutilización	Aumento del RC provoca disminución en el grado de reutilización de la clase
Cantidad de pruebas	Aumento del RC provoca aumento de la cantidad de pruebas de unidad necesarias para probar una clase.

Los criterios y categorías definidos para la evaluación de los atributos de calidad anteriores se

presentan en la siguiente tabla:

Tabla 8 Criterios de evaluación para la métrica RC

Atributos	Categoría	Criterio
Acoplamiento	Ninguna	0
	Bajo	1
	Medio	2
	Alto	>2
Complejidad de mantenimiento	Baja	< =Promedio
	Media	Entre Promedio y 2* Promedio
	Alta	> 2* Promedio
Reutilización	Baja	> 2*Promedio
	Media	Entre Promedio y 2* Promedio
	Alta	< =Promedio
Cantidad de pruebas	Baja	< =Promedio
	Media	Entre Promedio y 2* Promedio
	Alta	> 2* Promedio

Resultados obtenidos al aplicar la métrica RC

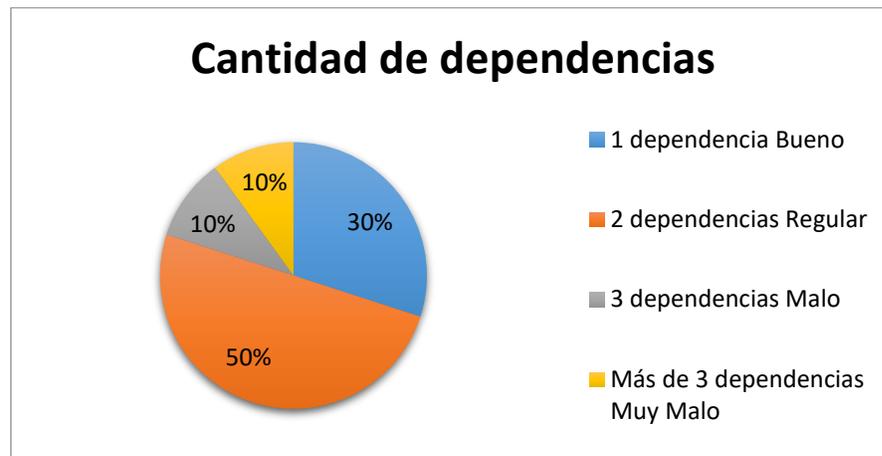


Figura 9 Cantidad de dependencias entre las clases

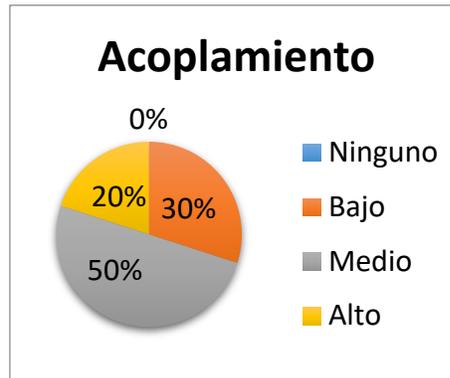


Figura 10 Resultado de la evaluación de la métrica para el atributo acoplamiento



Figura 11 Resultado de la evaluación de la métrica para el atributo complejidad de mantenimiento

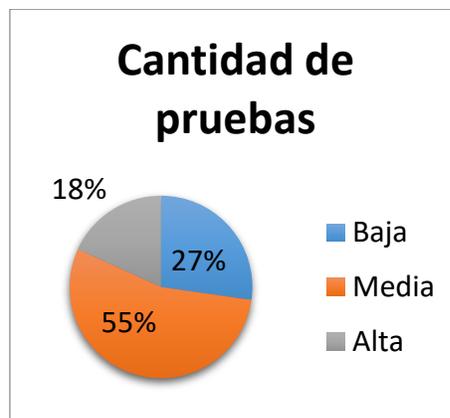


Figura 12 Resultado de la evaluación de la métrica para el atributo cantidad de pruebas



Figura 13 Resultado de la evaluación de la métrica para el atributo reutilización

Luego de aplicarse la métrica de diseño RC se obtuvieron resultados que permiten evaluar el diseño propuesto de calidad aceptable teniendo en cuenta que más del 50% de las clases empleadas en el sistema poseen 2 dependencias con otras clases lo que conlleva a evaluaciones positivas de los atributos de calidad involucrados (Acoplamiento, la Complejidad de mantenimiento, la Reutilización y la Cantidad de pruebas).

2.6. Conclusiones parciales

Se generaron los productos de trabajo del modelado de negocio, requisitos, análisis y diseño, propuestos en cada disciplina de la metodología de desarrollo utilizada, que permitieron en gran medida realizar el diseño de la propuesta de solución. La aplicación de diferentes técnicas para la obtención de requisitos, arrojaron como resultado la identificación de 128 RF y 15 RNF, los cuales fueron descritos y prototipados obteniendo como resultado la validación de estos por parte del cliente. Además, se realizó el diseño, validándolo mediante la métrica TOC donde se evidencia una alta reutilización de las clases, y la métrica RC en la que se obtuvo un valor medio de reutilización y acoplamiento.

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1. Introducción

En el presente capítulo se realizan las disciplinas de implementación, pruebas internas y pruebas de aceptación teniendo presente los requisitos definidos con el cliente. En la implementación, se define el estándar de codificación y se representa mediante el diagrama de componentes la organización y relación de cada uno de los componentes que formarán parte del sistema. Además, se realizan las pruebas internas utilizando las técnicas de caja blanca y caja negra y se realizan con el cliente las pruebas de aceptación. Se aplica la técnica de ladov para la medición de la satisfacción de los usuarios con el sistema y se describen los beneficios que aporta la solución al proceso de Gestión de Compras de la Universidad.

3.2. Implementación

La implementación se realiza a partir de los resultados obtenidos en disciplinas anteriores, el Análisis y Diseño brinda elementos de arquitectura que permiten formalizar la construcción del sistema [26]. El objetivo de esta disciplina es la obtención del código fuente, estas instrucciones son escritas en un lenguaje de programación que consiste en un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones, traduciéndose finalmente en el sistema que se le entregará al cliente. Antes de iniciar la implementación es fundamental definir el estándar de codificación, que detalla la nomenclatura a utilizar por los desarrolladores.

3.2.1. Estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del sistema, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Definir dichos estándares a utilizar en la investigación permite facilitar el estudio y comprensión del código haciendo que sea más fácil a la hora de realizar cambios [42].

A continuación, se muestran ejemplos de los estándares de codificación definidos para la implementación de la aplicación:

Tabuladores o espacios

- Para la indentación se utiliza sólo espacios.

```
class BalanceAlimento(models.Model):
    ..._name = 'necesidad.balance_alimento'

    ...balance_alimento_id = fields.One2many('necesidad.datos_balance_alimento', 'datos_balance_alimento_id')
    ...correo_demandante = fields.CharField('Correo', required=True)
```

Figura 14 Estándar de codificación Tabuladores o espacios

Líneas en blanco

- Se separan las funciones con una línea en blanco y las definiciones de clases con dos líneas en blanco.

```
class BalanceAlimento(models.Model):
    ... _name = 'necesidad.balance_alimento'
    ... balance_alimento_id = fields.One2many('necesidad.datos_balance_alimento', 'datos_balance_alimento_id')
    ... correo_demandante = fields.Char('Correo', required=True)
    ... @api.one
    ... def sendEmail(self):...
    ... @api.model
    ... def registrarDemanda(self):...

class DatosBalanceAlimento(models.Model):
    ... name = 'necesidad.datos balance alimento'
```

Figura 15 Estándar de codificación Líneas en blanco

Tamaño máximo de línea

- Se limitan todas las líneas a un máximo de 120 caracteres.

```
... message = 'Los productos ' + str(prodfalt) + ' no se encuentran en el almacén y ' + str(proddisp) + \
... ' si en encuentran disponible en el almacén'
```

Figura 16 Estándar de codificación Tamaño máximo de línea

Espacios en blanco en expresiones y sentencias

Se utiliza espacio en blanco:

- Inmediatamente después de entrar en un paréntesis o antes de salir de un paréntesis, corchete o llave.
- Inmediatamente antes de una coma, punto y coma, o dos puntos.
- Inmediatamente antes de abrir un paréntesis para una lista de argumentos de una llamada a una función.

```
smtplib.login(from_address, password)
```

- Inmediatamente antes de abrir un paréntesis usado como índice o para particionar.

```
mime_message["From"] = from_address
```

Nombres de Clases

- Para los nombres de clases se utiliza UpperCamelCase: es un estilo de escritura que se aplica a frases o palabras compuestas, cuando la primera letra de la cada una de las palabras es mayúscula.

```
class BalanceAlimento(models.Model):
```

Nombres de métodos

- Se utiliza para el nombre de los métodos lowerCamelCase: variante de CamelCase, usado cuando la primera letra es minúscula.

def sendEmail(self):

Nombres de paquetes y módulos

- Los módulos deben tener nombres cortos formados en su totalidad por letras minúsculas, se puede utilizar también guiones bajos en el nombre del módulo. Ejemplo: persona.py.

Nombre de los archivos

- Vistas: nombre_modelo_view

Ejemplo de nombre de archivos: balance_alimento_view.xml.

3.2.2. Diagrama de componentes

Una vez terminado el diseño de la arquitectura tiene lugar el diseño a nivel de componentes. En esta etapa se ha establecido la estructura general de los datos y del programa del software. El objetivo es traducir el modelo del diseño a software operativo. Pero el nivel de abstracción del modelo de diseño existente es relativamente alto y el del programa operativo es bajo [41].

El diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos ficheros de código fuente, binarios o ejecutables. Los elementos de modelado que lo conforman son los componentes y paquetes que muestran la estructura del sistema en términos de implementación a un alto nivel [41].

La siguiente figura muestra el diagrama de componentes del sistema, la organización y dependencia existente entre los componentes Vista, Controlador, ORM y Modelo. El paquete Modelo representa los componentes que garantizan la información persistente en el sistema y se relaciona con el componente ORM; Odoos utiliza OpenObject como ORM para gestionar automáticamente el mapeo de la base de datos. El paquete Vistas está compuesto por 23 componentes al igual que el Modelo, los de este último representan los modelos existentes en la arquitectura y el paquete Controlador representa la clase controladora del sistema.

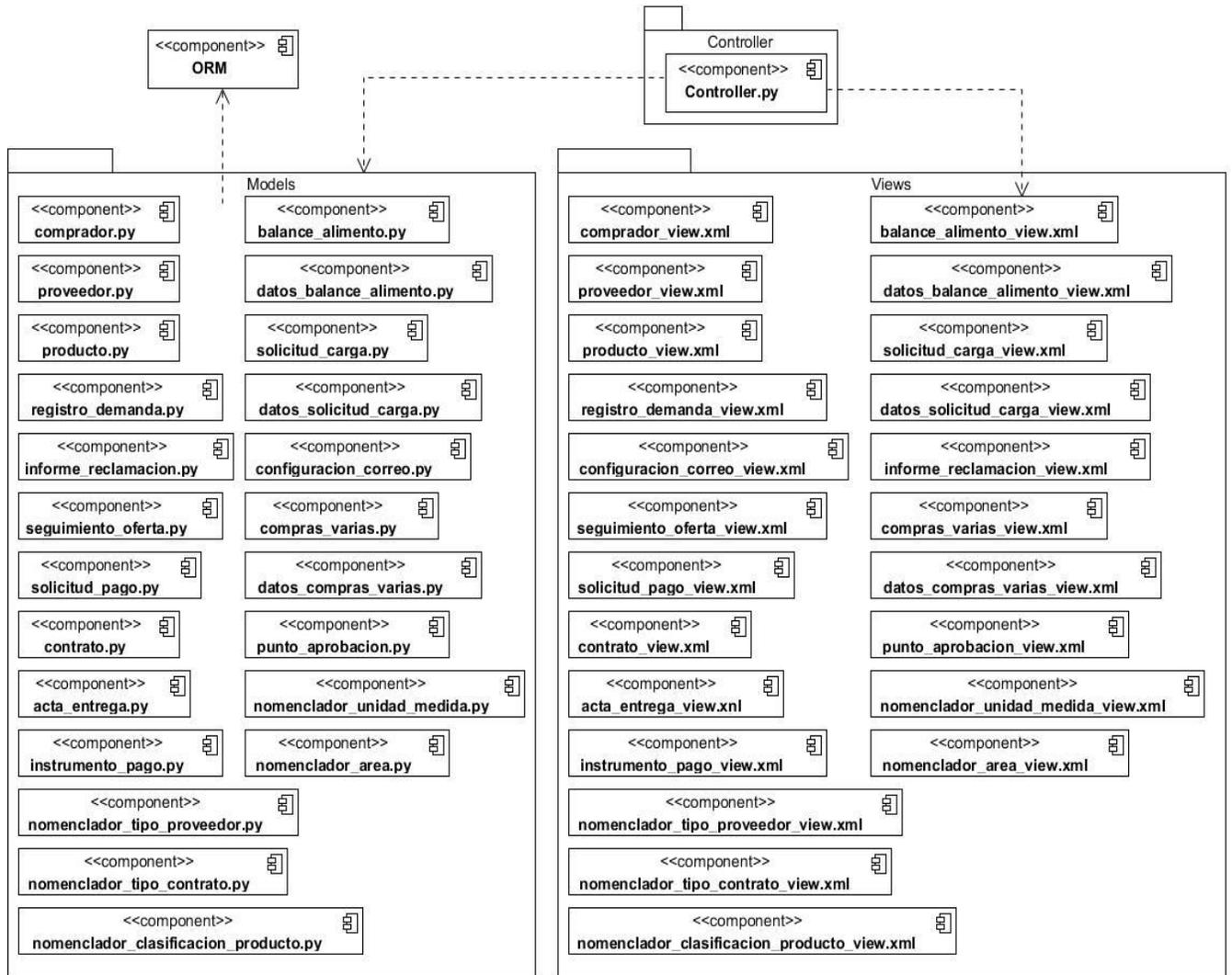


Figura 17 Diagrama de componentes

3.2.3. Interfaz de usuario funcional

La solución propuesta permite, de manera sencilla, gestionar el proceso de compras en la universidad. A continuación, se muestra una interfaz de usuario de la funcionalidad Listar producto (Figura 16) y otra del requisito funcional Listar proveedor (Figura 17) del sistema desarrollado donde se pueden apreciar los elementos antes abordados:

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

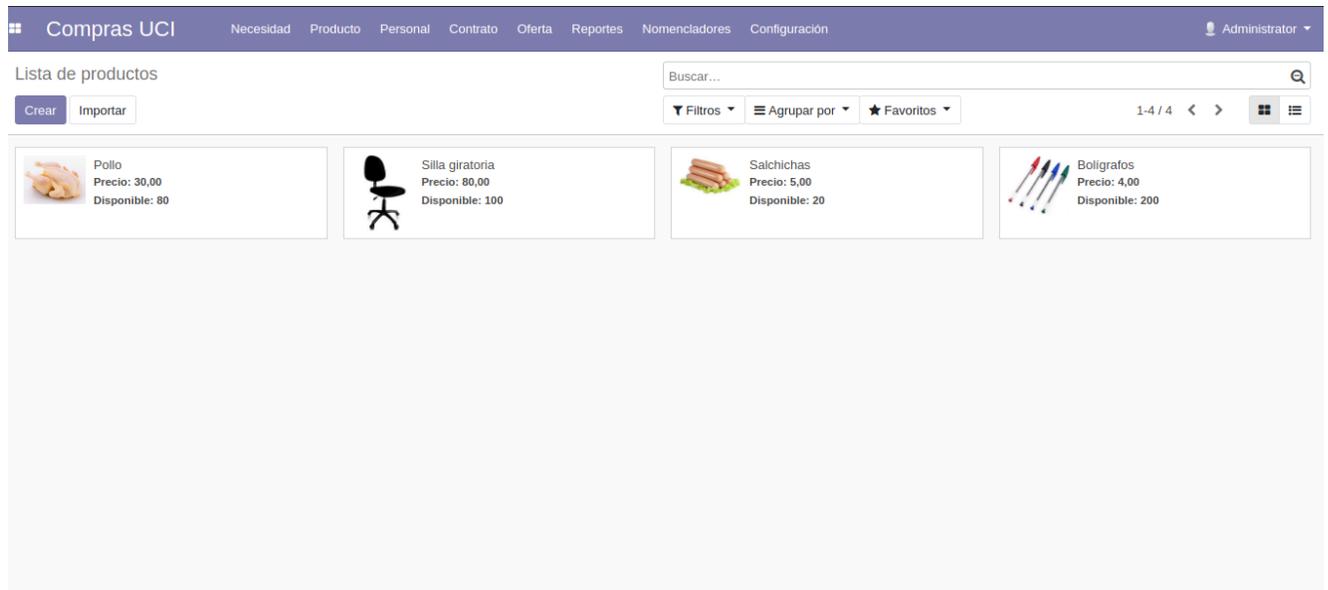


Figura 18 Interfaz de usuario de la funcionalidad Listar productos

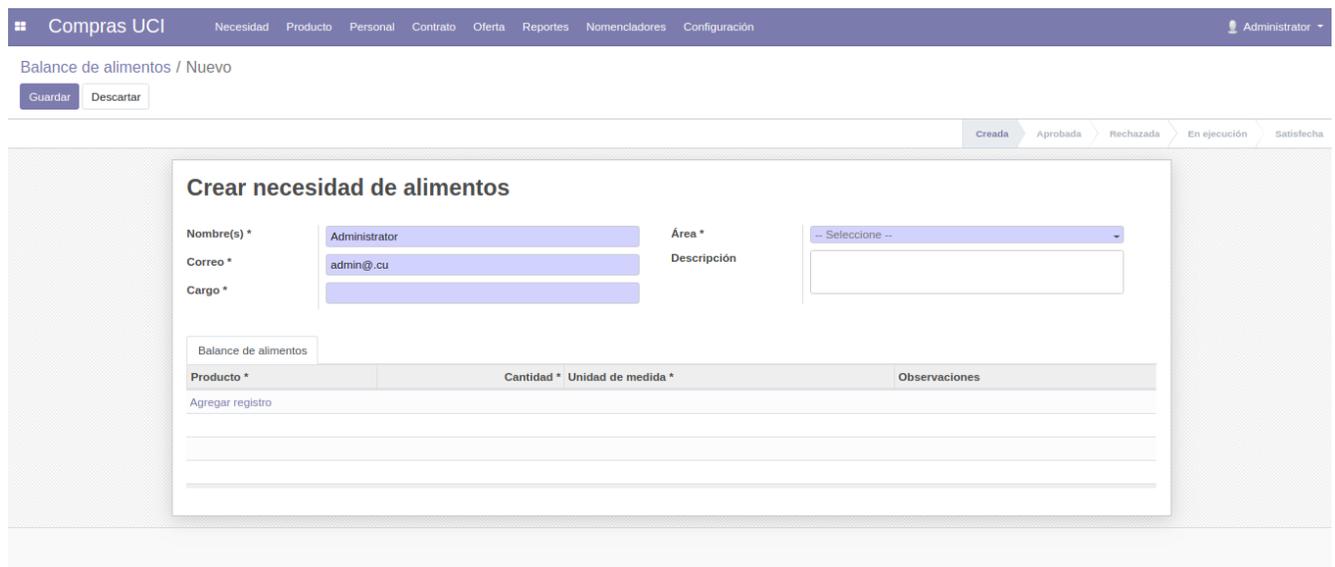


Figura 19 Interfaz de usuario de la funcionalidad Crear necesidad de balance de alimentos

Al concluir la disciplina de Implementación, fue necesario realizar pruebas al software para comprobar que las funcionalidades desarrolladas cumplen con los objetivos para el que fueron implementadas. Las pruebas se realizaron en la disciplina de Pruebas internas, a través de diferentes métodos y técnicas.

3.3. Pruebas Internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas [26].

3.3.1. Pruebas de caja blanca

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba.

Técnica del Camino Básico

La prueba del camino básico es una técnica de prueba de caja blanca propuesta por Tom McCabe. El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa [41].

La complejidad ciclomática es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba del camino básico, el valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y define un límite superior para el número de pruebas que se deben realizar asegurando la ejecución de cada sentencia al menos una vez [41].

La siguiente figura muestra el código de la funcionalidad *registrarDemandaBA(self)*:

```
@api.multi
def registrarDemandaBA(self):
1  self.ensure_one()
2  if self.hay_faltantes_ba:
3      prodfalt = self.balance_alimento_id.filtered(
         lambda balimento: balimento.producto_id_ba.disponible < balimento.cantidad
         ).ids
4      for i in prodfalt:
5          self.env['necesidad_registro_demanda'].create(dict(
             tipo_demanda='Balance de Alimentos',
             producto_id_rd=[[6, False, [i]]]
         ))
6  self.estado_ba = 'en ejecucion'
7  return True
```

Figura 20 Método registrar demanda en el balance de alimentos

Para obtener los casos de prueba a partir de la técnica seleccionada se debe construir el grafo de flujo correspondiente al código de la función como se muestra en la figura:

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

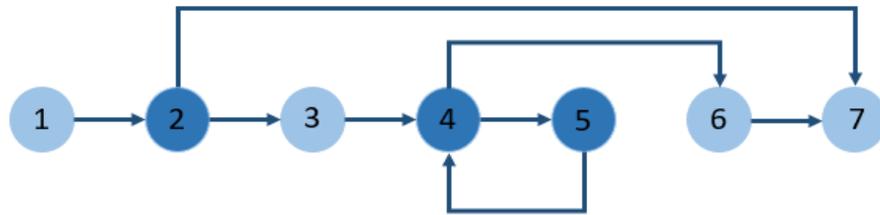


Figura 21 Grafo resultante de aplicar la técnica Camino Básico

Luego se determina la complejidad ciclomática $V(G)$ del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo. Puede ser calculada de tres formas:

Cálculo de la Complejidad Ciclométrica

1. $V(G) = A - N + 2$, siendo A la cantidad de aristas o arcos del grafo y N la cantidad de nodos del grafo.
2. $V(G) = P + 1$, siendo P los nodos predicados, es decir los que tienen más de una arista de salida.
3. $V(G) = R$, siendo R el número de regiones cerradas del grafo.

Al realizar los cálculos correspondientes se obtiene por cualquiera de las variantes el siguiente resultado:

- | | | |
|-----------------------|-------------------|---------------|
| 1. $V(G) = A - N + 2$ | 2. $V(G) = P + 1$ | 3. $V(G) = R$ |
| $V(G) = 8 - 7 + 2$ | $V(G) = 2 + 1$ | $V(G) = 3$ |
| $V(G) = 3$ | $V(G) = 3$ | |

El cálculo arrojó que $V(G) = 3$, definiendo como posibles caminos básicos:

Camino básico 1: 1, 2, 3, 4, 5, 4, 6, 7

Camino básico 2: 1, 2, 7

Camino básico 3: 1, 2, 3, 4, 6, 7

Luego se definen los casos de prueba para cada uno de los caminos básicos obtenidos. A continuación, se muestra dicho resultado:

Tabla 9 Caso de prueba para el camino 1

Descripción	Analizar los productos faltantes en la instancia del modelo actual para ser registrados en la demanda.
Condición de ejecución	Que el balance analizado tenga productos faltantes en el almacén.
Entrada	Balance de alimentos.
Resultado	Se registra una demanda con los productos del balance de alimentos que tienen faltantes en el almacén

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 10 Caso de prueba para el camino 2

Descripción	Analizar los productos faltantes en la instancia del modelo actual para ser registrados en la demanda.
Condición de ejecución	Que el balance analizado tenga productos faltantes en el almacén.
Entrada	Balance de alimentos
Resultado	Ningún resultado

Tabla 11 Caso de prueba para el camino 3

Descripción	Analizar los productos faltantes en la instancia del modelo actual para ser registrados en la demanda.
Condición de ejecución	Que el balance analizado tenga productos faltantes en el almacén.
Entrada	Balance de alimentos.
Resultado	Se registra una demanda con los productos del balance de alimentos que tienen faltantes en el almacén

3.3.2. Prueba de caja negra

Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un sistema.

Método de Partición de Equivalencias

La partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirán la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar [41].

El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Mediante conceptos introducidos en la sección anterior, si un conjunto de objetos puede unirse por medio de relaciones simétricas, transitivas y reflexivas, entonces existe una clase de equivalencia. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada [41].

A continuación, se muestra el flujo central del caso de prueba de la agrupación de requisitos Crear producto:

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 12 Diseño de Caso de Prueba de caja negra del requisito Crear producto.

Escenario	Descripción	Nombre	Precio	Disponible	Tiempo entrega	Fecha vencimiento	Respuesta del sistema	Flujo central
EC 1.1: Crear producto introduciendo datos válidos.	El sistema debe permitir crear un producto en el sistema.	V Arroz	V 100	V 5	V 13/01/19	V 25/09/22	El sistema registra un nuevo producto.	1. Se selecciona la opción de producto. 2. Se presiona el botón "Crear". 3. Se introducen los datos necesarios para crear un producto. 4. Se selecciona el botón "Guardar".
EC 1.2: Crear producto introduciendo datos inválidos.	El sistema debe permitir crear un producto en el sistema.	I Arr1oz	V 100	V 5	V 13/01/19	V 25/09/22	El sistema muestra un mensaje : "El campo Nombre está incorrecto".	1. Se selecciona la opción de producto. 2. Se presiona el botón "Crear". 3. Se introducen los datos necesarios para crear un producto. 4. Se selecciona el botón "Guardar".
		I arroz	I 70g	V 5	V 13/01/19	V 25/09/22	El sistema muestra un mensaje : "El campo Nombre está incorrecto" y "El campo Precio está incorrecto".	

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

		I arroz	V 100	I Cinco	V 13/01/1 9	V 25/09/22	El sistema muestra un mensaje : "El campo Nombre está incorrecto" y "El campo Disponible está incorrecto".	
		I arroz	V 100	V 5	I mañana	V 25/09/22	El sistema muestra un mensaje : "El campo Nombre está incorrecto" y "El campo Tiempo de entrega está incorrecto".	
		I arroz	V 100	V 5	V 13/01/1 9	I 32/09/22	El sistema muestra un mensaje : "El campo Nombre está incorrecto" y "El campo Fecha de Vencimiento está incorrecto".	

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

EC 1.3: Crear producto dejando campos vacíos.	Se dejan campos vacíos al crear un producto.	I (vacío)	V 100	V 5	V 13/01/19	V 25/09/22	El sistema muestra los errores en una alerta indicando que contiene campos vacíos.	<ol style="list-style-type: none"> 1. Se selecciona la opción de producto. 2. Se presiona el botón "Crear". 3. Se introducen los datos necesarios para crear un producto. 4. Se selecciona el botón "Guardar".
		V Arroz	I (vacío)	V 5	V 13/01/19	V 25/09/22		
		V Arroz	V 100	I (vacío)	V 13/01/19	V 25/09/22		
		V Arroz	V 100	V 5	I (vacío)	V 25/09/22		
		V Arroz	V 100	V 5	V 13/01/19	I (vacío)		
EC 1.4: Cancelar.	Se cancela la operación de crear un producto.	V(N/A)	V(N/A)	V(N/A)	V(N/A)	V(N/A)	El sistema no realiza ninguna acción y cierra la ventana.	<ol style="list-style-type: none"> 1. Se selecciona la opción de producto. 2. Se presiona el botón "Crear". 3. Se introducen los datos necesarios para crear un producto. 4. Se selecciona el botón "Descartar".

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Con la aplicación de la prueba de caja negra y su método Partición de Equivalencias, se obtuvo, luego de aplicar los Diseños de Casos de Pruebas resultados satisfactorios, demostrando el correcto funcionamiento de todas las funcionalidades implementadas. Para revisar el resto de los casos de pruebas de caja negra aplicados a la solución consultar los Diseño de Casos de Pruebas del presente trabajo.

Se aplicaron las pruebas internas, definida como disciplina de la metodología AUP-UCI, que guía la presente investigación, realizadas por especialistas del Centro CEIGE, a partir de los diseños de casos de prueba de los requisitos funcionales del sistema. El método de prueba aplicado demostró resultados satisfactorios desde el punto de vista funcional. Las no conformidades encontradas fueron analizadas y corregidas en el tiempo establecido, logrando un correcto comportamiento ante diferentes situaciones (entradas válidas y no válidas). A continuación, se muestra la figura 20, con el número de no conformidades detectadas en cada iteración de prueba realizada:

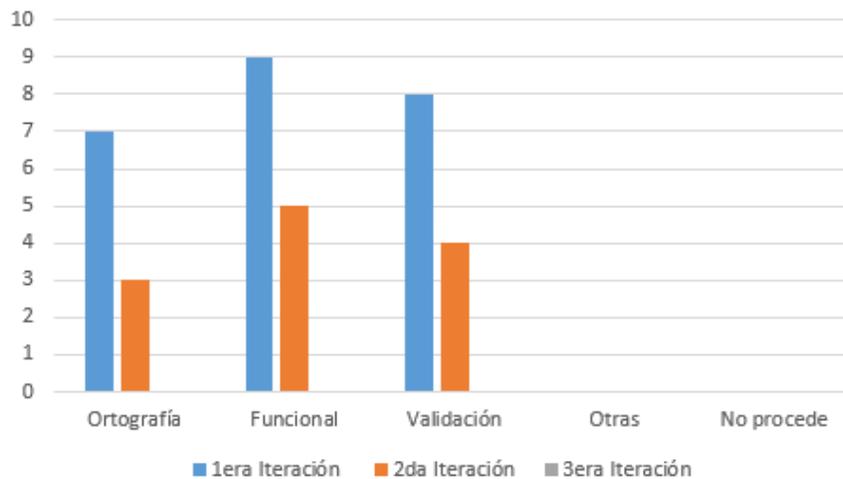


Figura 22 No Conformidades detectadas

En la primera iteración fueron detectadas un total de 24 No conformidades (NC): 7 de ortografía, 9 funcionales y 8 de validación. En la segunda iteración fueron detectadas 12 NC: 3 de ortografía, 5 funcionales y 4 de validación. En la tercera iteración no fueron detectados errores de ningún tipo, para un total de 0 NC, quedando de esta forma validado el sistema.

3.3.3. Pruebas de aceptación

Las pruebas de aceptación se realizan para que el cliente certifique que el sistema cumple con sus especificaciones. La planificación detallada de estas pruebas debe haberse realizado en etapas tempranas del desarrollo del proyecto, con el objetivo de utilizar los resultados como indicador de su validez. Si se ejecutan las pruebas documentadas a satisfacción del cliente, el producto se considera correcto y, por lo tanto, adecuado para su utilización [23].

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Luego de las pruebas internas realizadas al Sistema de Compras se ejecutaron las pruebas de aceptación con el cliente. La aceptación del producto arrojó como resultado un total de cuatro solicitudes de cambios en tres iteraciones, desglosadas en tres en la primera iteración, una en la segunda iteración y cero en la tercera iteración, las cuales se resolvieron en tiempo. Estas solicitudes fueron solucionadas para el correcto funcionamiento del sistema. Una vez comprobada la correcta implementación de los requisitos, se emitió el Acta de aceptación con el cliente, la cual puede ser consultada en el Anexo 4.

3.4. Aplicación de la técnica de ladov

Para la validación de la investigación se utilizó la Técnica de ladov, conformada por cinco preguntas: tres cerradas y 2 abiertas, para evaluar la satisfacción. Las preguntas cerradas se relacionan a través de lo que se denomina el “Cuadro Lógico de ladov”, por otra parte, las preguntas abiertas permiten profundizar en la naturaleza de las causas que originan los diferentes niveles de satisfacción. La Técnica de ladov constituye una vía indirecta para el estudio de la satisfacción, pues los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce [49]. Para la realización de esta técnica se seleccionaron especialistas y compradores de la Dirección de compras de la UCI, a los cuales se les aplicó el cuestionario para determinar el grado de satisfacción con la propuesta de solución.

El número resultante de la interrelación de las tres preguntas indica la posición de cada sujeto en la escala de satisfacción. A continuación, se muestra la escala:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma:

Tabla 13 Niveles de satisfacción

+1	Máximo de satisfacción
0.5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0.5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

La satisfacción grupal se calcula por la siguiente fórmula:

$$ISG = \frac{A (+1) + B (+0.5) + C (0) + D (-0.5) + E (-1)}{N}$$

Figura 23 Índice de satisfacción grupal

En esta fórmula A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 ó 6; 4; 5 y donde N representa el número total de sujetos del grupo.

El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción.

Estos valores representados gráficamente en un eje, se aprecian de la forma siguiente:

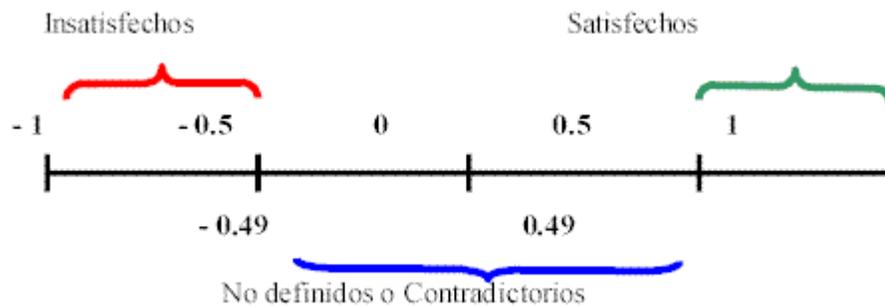


Figura 24 Rango de valores para el cálculo de satisfacción grupal

Cuadro Lógico de ladov:

Tabla 14 Cuadro Lógico de ladov

6. ¿Le satisface este sistema para su utilización en la Dirección de Compras?	1. ¿Considera factible el desarrollo de la gestión de compras sin contar con una herramienta que centralice la información relacionada con el proceso y esté disponible desde cualquier área de la universidad?								
	No			No sé			Sí		
	4. ¿Utilizaría usted el Sistema de Compras desarrollado para la gestión de Gestión de necesidad, Contratación, Seguimiento de la demanda y Recepción de mercancía?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Resultados obtenidos al aplicar el cuestionario:

Tabla 15 Resultados del cuestionario de ladov según la escala

Total de cuestionados	8	Escala
Máximo de satisfacción	5	A
Más satisfecho que insatisfecho	2	B
No definida	1	C
Más insatisfecho que satisfecho	0	D
Clara insatisfacción	0	E
Contradictoria	0	C

Resultados obtenidos:

$$ISG = \frac{5 (+1) + 2 (+0.5) + 1 (0) + 0 (-0.5) + 0 (-1)}{8}$$

$$ISG = 0.75$$

La aplicación de la técnica de ladov constituye un instrumento de gran valor para el estudio de la satisfacción de los encuestados sobre el Sistema de Gestión de compras. La valoración de los usuarios sobre el sistema confirmó un alto Índice de Satisfacción Grupal (ISG = 0.75), lo cual indica que, según el rango de valores de estos, es un elevado valor de satisfacción de los encuestados respecto a dicho sistema y cualitativamente en los criterios emitidos donde evidencian su satisfacción por la contribución del sistema.

3.5. Beneficios del Sistema de compras desarrollado

El desarrollo del Sistema de compras permite la gestión de los subprocesos Gestión de necesidad, Contratación, Seguimiento de la demanda y Recepción de mercancía. En él se gestionan los datos necesarios sobre la necesidad, los cuales se obtienen en tiempo real y son ingresados en el sistema por cada una de las áreas de la universidad. Estas necesidades son verificadas por el Director de Compras y convertidas en demanda cuando el producto que se solicita no se satisface con la existencia del mismo en el almacén. Cuando esto ocurre la demanda es asignada al comprador y se ejecutan los demás subprocesos mencionados inicialmente. Entre los principales beneficios que aporta el desarrollo del sistema se encuentran:

CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

- Es un sistema Web que permite la disponibilidad desde cualquier parte de la universidad. Lo que contribuye a mejorar la comunicación a tiempo de las necesidades.
- Estandariza los canales de la DC para recibir las solicitudes de las necesidades por área. Todas las áreas utilizan una misma interfaz para las solicitudes, anteriormente se utilizaban diferentes vías como el correo, teléfono y Excel.
- Disminuye los errores introducidos en las solicitudes de las necesidades al contar con elementos de selección y carga automática de campos desde el perfil del solicitante.
- Facilita el chequeo de consistencia de la información a través de métodos que validan diferentes campos como la generación de la demanda, vigencia de los contratos, entre otros.
- Provee un conjunto de reportes que muestran la información y permiten la toma de decisiones de forma operativa de la DC.

3.6. Conclusiones parciales

Se implementaron los requisitos funcionales en correspondencia con las descripciones realizadas obteniéndose una solución acorde a las especificaciones del cliente. La utilización de los estándares de codificación permitió organizar el código para un mejor entendimiento de todas las clases y métodos desarrollados. Se ejecutaron las pruebas internas y pruebas de aceptación, para el control de la calidad del sistema, arrojando resultados satisfactorios que permitieron la corrección en tiempo y forma de las no conformidades detectadas.

CONCLUSIONES GENERALES

- A partir del estudio de los sistemas relacionados se comprobó que existen soluciones que desde la perspectiva de negocio satisfacen las necesidades del cliente, pero no están en correspondencia con las políticas de soberanía e independencia tecnológica. Sin embargo, los sistemas de software libre analizados presentan limitaciones para su uso, aunque se comprobó la factibilidad de la plataforma Odoó para el desarrollo de un nuevo sistema.
- Las disciplinas Modelado de negocio, Requisitos y Análisis y diseño permitieron el desarrollo de productos de trabajo para el análisis de la solución y diseño de la arquitectura. Además, proporcionaron los cimientos para realizar la disciplina de implementación.
- La utilización de estándares de codificación permitió obtener un código estructurado, logrando un lenguaje común y comprensible. La implementación del sistema permitió cumplir con las funcionalidades identificadas, utilizando para ello lenguajes, notaciones, herramientas y tecnologías distribuidas bajo licencias de software libre en correspondencia con las políticas de la Universidad y del país.
- Se realizaron las pruebas de software al sistema, los artefactos generados por la metodología de desarrollo utilizada fueron revisados por el Director de Compras de la UCI, obteniendo finalmente el Acta de Aceptación del cliente, elementos que demuestran el cumplimiento de los objetivos propuestos.
- La aplicación de la técnica de ladov arrojó que los usuarios se sienten satisfechos con el Sistema de compras desarrollado para la universidad.

RECOMENDACIONES

Para dar continuidad a la presente investigación se recomienda:

- Continuar el estudio con el objetivo de añadir nuevas funcionalidades al sistema, de acuerdo a las necesidades del cliente.

REFERENCIAS BIBLIOGRÁFICAS

1. Volpentesta, J.R., *El impacto de las TIC sobre las estructuras organizacionales y el trabajo del hombre en las empresas*. 2016.
2. Placeres, L.R. and H.R. Companioni, *SUBSISTEMA DE GESTIÓN COMERCIAL PARA EL SISTEMA DE GESTIÓN EMPRESARIAL DISTRA*. Universidad&Ciencia, 2018. 7(1): p. 103-114.
3. Martha Dunia Delgado Dapena, U.T.d.L.H., *MODELO CONCEPTUAL DE UN PROYECTO DE SOFTWARE UTILIZANDO EL RAZONAMIENTO BASADO EN CASOS*. octubre, 2012.
4. Serrano, M.J.E., *Gestión de compras*. 2014: Ediciones Paraninfo, SA.
5. Vielba Fernández, J., *Estrategias de marketing en las pequeñas y medianas empresas*. 2017.
6. Calle Alférez, M.M., et al., *Análisis de la Influencia de la Responsabilidad Social Empresarial y el Comportamiento de Compra de los Consumidores de Servicios Hoteleros de 3 Estrellas de Tacna, Año 2015*. 2016.
7. Kerin Roger, H.S.y.R.W., «*Marketing*», *Novena Edición*. 2009.
8. Stanton William, E.M.y.W.B., «*Fundamentos de Marketing*», *Decimocuarta Edición*. 2007.
9. McCarthy y Perrault, *Marketing Planeación Estratégica de la Teoría a la Práctica*, 11a edición.
10. Pérez Ríos, C.K., *La calidad del servicio al cliente y su influencia en los resultados económicos y financieros de la empresa Restaurante Campestre SAC, Chiclayo periodo enero a setiembre 2011 y 2012*. 2014.
11. EGAS ORDÓÑEZ, A.C., *ANÁLISIS DE LA CALIDAD DEL SERVICIO DE RESTAURACIÓN QUE OFRECEN LOS PYMES, PARA LA ELABORACIÓN DE TALLERES SOBRE SERVICIO AL CLIENTE EN EL ÁREA DE ALIMENTOS Y BEBIDAS EN EL CANTÓN "EL CARMEN", PROVINCIA DE MANABÍ*. 2017, UNIVERSIDAD DE GUAYAQUIL.
12. Jaramillo Morales, E.G. and S.P. Cruz Pineda, *Creación y aplicación de un modelo de evaluación de la calidad del servicio orientado a 5 instituciones privadas según correspondan a la actividad económica "Asistencia social" en la ciudad de guayaquil*. 2013.
13. Paulo Nunez, E.e.I.U.d.N.d.L., 2016.
14. Zuluaga, W. and L. Prieto, *Elementos fundamentales del desarrollo de proveedores como modelo de gestión en procesos de compras y aprovisionamiento*. 2017.
15. Alpízar Domínguez, A.J. and L. Hernández Jaimes, *ESTUDIO DE MERCADO PARA DETERMINAR EL NIVEL DE SATISFACCIÓN DEL CLIENTE EN ESTELARIS MUEBLERÍAS, 2015*.
16. Trejos, S. and L. Chaves, *Necesidades de los consumidores para sentirse únicos:(Caso aspirantes a Posgrado en proceso de admisión)*. 2017.
17. Farber, P.B.M., *199 Preguntas sobre marketing*. 2002: Editorial Norma.
18. Murillo Pulgar, M.G., *La hipoteca y su incidencia como garantía del contrato de préstamo de mutuo en las sentencias dictadas por la Unidad Judicial Civil y Mercantil de la Ciudad de Riobamba periodo 2015*. 2016, Riobamba: Universidad Nacional de Chimborazo, 2016.
19. García Guanilo, Y.A. and J.M. Choquehuanca Nuñez, *La implementación del Sap Business One y su influencia en la gestión de la empresa Letreros Corporativos SA*. 2016.
20. Rodríguez, N.D. and C.J.N.P. Guerrero, *Metodología para evaluar el impacto de la maquinaria agrícola sobre los recursos naturales del medio ambiente*. Ciencias Holguín, 2010. 13(2).
21. *PostgreSQL*. 2016; Available from: www.postgresql.org.
22. *Plataforma de compras | Software de gestión de compras Fullstep*. Available from: <https://www.fullstep.com/plataforma-de-compras/>.
23. Tuya, J., I.R. Román, and J.J.D. Cosín, *Técnicas cuantitativas para la gestión en la ingeniería del software*. 2007: NetBiblo.
24. Riera Romero, E., *Análisis de planificadores de recursos empresariales para una PYME*. 2016.

25. GALEANO, R.D.M., *ESTUDIO COMPARATIVO DE TRES ERP CON DISTRIBUCIÓN FLOSS. DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMA Y COMPUTACIÓN*. QUITO, ENERO DE 2017.
26. Sánchez, T.R., *Metodología de desarrollo para la Actividad productiva de la UCI*. 2015.
27. Pérez, J.D., *Notaciones y lenguajes de procesos. Una visión global*. Departamento de sistemas y lenguajes informáticos, Universidad De Sevilla, 2007.
28. González Navarro, D., *Expresividad de BPMN 2.0 desde la perspectiva del problema de secuenciamiento*. 2018.
29. Ortiz, L., A. Aristizabal, and F. Caraballo, *UML diagramming software: Definition of accessibility criteria required for the construction of diagrams by visually impaired users Software de diagramado de UML: Definición de criterios de accesibilidad necesarios para la construcción de diagramas por usuarios con limitación visual*. Actas de Ingeniería, 2016. **2**: p. 344-351.
30. Bray, T., et al., *Extensible Markup Language (XML)*. World Wide Web Journal, 2014. **2**(4): p. 27-66.
31. Challenger-Pérez, I., Y. Díaz-Ricardo, and R.A. Becerra-García, *El lenguaje de programación Python*. Ciencias Holguín, 2014. **20**(2).
32. *Mejoras de Procesos de Software*. Available from: <http://mejoras.prod.uci.cu/>.
33. Orihuela, A.V. *Visual Paradigm*. 2016; Available from: <https://es.scribd.com/document/166415572/Visual-Paradigm>.
34. Pérez-Montoro, M. and L. Codina, *Software de prototipado para la arquitectura de la información: funcionalidad y evaluación*. El profesional de la información, 2010. **19**(4).
35. *PgAdmin. Postgresql Tools*. 2016; Available from: www.pgadmin.org.
36. *ODOO*. 2016; Available from: <http://www.odoo.com>.
37. López, J.M.D.y.P. and C.y.T. Real, *Verificación y Validación. Ingeniería de software. Ingeniería de Programación*.
38. SOMMERVILLE, I., *Ingeniería del Software, Séptima Edición*. Pearson Educación, Madrid.
39. SOMMERVILLE, I., *Ingeniería del Software. Séptima edición.*: p. 712.
40. Rodríguez, C.R.R., *Herramienta para aplicar métricas al diagrama de clases del diseño orientado a objetos*. Avanzada Científica, 2012. **15**(3): p. 39-52.
41. Pressman, R.S., *Ingeniería del Software. Un enfoque práctico. Séptima edición*. 2010.
42. Pressman, R.S., *Ingeniería del Software Un enfoque práctico, quinta edición*. 589.
43. Reis, D., *Odoos development essentials*. 2015: Packt Publishing Ltd.
44. Sánchez, B.A.N. and J.P.O. Pérez, *MEDDPOP: Una aplicación para Pymes del sector salud basada en ODOO*. INNOVA Research Journal, 2018. **3**(8.1): p. 212-231.
45. Larman, C., *UML y Patrones. Una introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado. Segunda Edición*. 2003.
46. Vázquez Escudero, P.J., M.N. Moreno García, and F.J. García-Peñalvo, *Métricas orientadas a objetos*. 2001.
47. Vega, M.R.F. and F.I. Delgado, *Desarrollo de una herramienta para la evaluación de la mantenibilidad mediante métricas en el Laboratorio de Pruebas del CEIGE*. 2013.
48. Cadavid, A.N., J.D.F. Martínez, and J.M. Vélez, *Revisión de metodologías ágiles para el desarrollo de software*. Prospectiva, 2013. **11**(2): p. 30-39.
49. Dr. Alejandro López Rodríguez*, D.V.G.M., *La técnica de ladov. Una aplicación para el estudio de la satisfacción de los alumnos por las clases de educación física* Abril de 2002.

Anexos

Anexo 1 Entrevista para la obtención de los requisitos

Preguntas realizadas con el objetivo de obtener los requisitos funcionales del negocio.

1. ¿Cómo se realiza el proceso de Compras en la Universidad?
2. ¿Cuántos procesos intervienen en dicho proceso?
3. ¿Cuáles son los procesos más importantes de Compras?
4. ¿Qué áreas intervienen en este proceso?
5. ¿Cuáles documentos de salida se generan?

Anexo 2 Descripción de subprocesos

- **Contratación**

Una vez registrada dicha necesidad se procede a revisar qué proveedor produce el producto o brinda el servicio que se requiere, si existe dicho proveedor se pasa a analizar el estado del contrato. Es entonces cuando se solicita la proforma de contrato y/o suplemento al proveedor y luego se revisa. Luego se procede a la recepción, registro y entrega del contrato y/o suplemento.

- **Seguimiento de la demanda**

Una vez seleccionado el proveedor, el especialista encargado le asignará al comprador correspondiente el seguimiento de la demanda. Cuando el especialista encargado ya lo tiene, procede a confeccionar la solicitud de pago la cual posteriormente se revisa, se firma, se aprueba y se archiva una copia. Posteriormente pasan a la confección del instrumento de pago.

- **Recepción de la mercancía**

Una vez terminado con lo anterior se solicita el transporte de carga para realizar la compra de la mercancía. En el momento de cargar la mercancía, se debe efectuar su revisión. Esta mercancía pasa por un proceso para comprobar todo lo que se había solicitado y sino pasar a la reclamación. Una vez que el comprador o la persona autorizada firme el informe de recepción concluye la compra de la mercancía.

Anexo 3 Listado de requisitos funcionales

No	Requisitos Funcionales
	Gestionar necesidades del balance de alimentos
RF1	Crear necesidad de balance de alimentos
RF2	Buscar necesidad de balance de alimentos
RF3	Editar necesidad de balance de alimentos
RF4	Suprimir necesidad de balance de alimentos
RF5	Listar necesidad de balance de alimentos
RF6	Mostrar necesidad de balance de alimentos
	Gestionar nomenclador de área para balance de alimentos

RF7	Crear nomenclador de área para balance de alimentos
RF8	Editar nomenclador de área para balance de alimentos
RF9	Mostrar nomenclador de área para balance de alimentos
RF10	Listar nomenclador de área para balance de alimentos
RF11	Buscar nomenclador de área para balance de alimentos
RF12	Suprimir nomenclador de área para balance de alimentos
	Gestionar nomenclador de unidad de medida para balance de alimentos
RF13	Crear nomenclador de unidad de medida para balance de alimentos
RF14	Editar nomenclador de unidad de medida para balance de alimentos
RF15	Mostrar nomenclador de unidad de medida para balance de alimentos
RF16	Listar nomenclador de unidad de medida para balance de alimentos
RF17	Buscar nomenclador de unidad de medida para balance de alimentos
RF18	Suprimir nomenclador de unidad de medida para balance de alimentos
	Gestionar necesidades de compras varias
RF19	Crear necesidad de compras varias
RF20	Buscar necesidad de compras varias
RF21	Editar necesidad de compras varias
RF22	Suprimir necesidad de compras varias
RF23	Listar necesidad de compras varias
RF24	Mostrar necesidad de compras varias
	Gestionar nomenclador de área para compras varias
RF25	Crear nomenclador de área para compras varias
RF26	Editar nomenclador de área para compras varias
RF27	Mostrar nomenclador de área para compras varias
RF28	Listar nomenclador de área para compras varias
RF29	Buscar nomenclador de área para compras varias
RF30	Suprimir nomenclador de área para compras varias
	Gestionar nomenclador de unidad de medida para compras varias
RF31	Crear nomenclador de unidad de medida para compras varias
RF32	Editar nomenclador de unidad de medida para compras varias
RF33	Mostrar nomenclador de unidad de medida para compras varias
RF34	Listar nomenclador de unidad de medida para compras varias
RF35	Buscar nomenclador de unidad de medida para compras varias
RF36	Suprimir nomenclador de unidad de medida para compras varias
	Gestionar demanda
RF37	Crear demanda
RF38	Listar demanda
RF39	Mostrar demanda
RF40	Buscar demanda
RF41	Editar demanda
RF42	Suprimir demanda
	Gestionar proveedor
RF43	Crear proveedor
RF44	Editar proveedor
RF45	Listar proveedor
RF46	Buscar proveedor
RF47	Suprimir proveedor
RF48	Mostrar proveedor

	Gestionar nomenclador de tipo de proveedor
RF49	Crear nomenclador de tipo de proveedor
RF50	Editar nomenclador de tipo de proveedor
RF51	Mostrar nomenclador de tipo de proveedor
RF52	Listar nomenclador de tipo de proveedor
RF53	Buscar nomenclador de tipo de proveedor
RF54	Suprimir nomenclador de tipo de proveedor
	Gestionar comprador
RF55	Crear comprador
RF56	Editar comprador
RF57	Mostrar comprador
RF58	Listar comprador
RF59	Buscar comprador
RF60	Suprimir comprador
	Gestionar producto
RF61	Crear producto
RF62	Listar producto
RF63	Buscar producto
RF64	Suprimir producto
RF65	Mostrar producto
RF66	Editar producto
	Gestionar nomenclador de clasificación de producto
RF67	Crear nomenclador de clasificación de producto
RF68	Editar nomenclador de clasificación de producto
RF69	Mostrar nomenclador de clasificación de producto
RF70	Listar nomenclador de clasificación de producto
RF71	Buscar nomenclador de clasificación de producto
RF72	Suprimir nomenclador de clasificación de producto
	Gestionar seguimiento de la demanda
RF73	Crear solicitud de oferta
RF74	Editar solicitud de oferta
RF75	Listar solicitud de oferta
RF76	Buscar solicitud de oferta
RF77	Suprimir solicitud de oferta
RF78	Mostrar solicitud de oferta
	Gestionar solicitud de pago
RF79	Crear solicitud de pago
RF80	Editar solicitud de pago
RF81	Listar solicitud de pago
RF82	Buscar solicitud de pago
RF83	Suprimir solicitud de pago
RF84	Mostrar solicitud de pago
	Gestionar nomenclador tipo de instrumento de pago
RF85	Crear nomenclador tipo de instrumento de pago
RF86	Editar nomenclador tipo de instrumento de pago
RF87	Mostrar nomenclador tipo de instrumento de pago
RF88	Listar nomenclador tipo de instrumento de pago
RF89	Buscar nomenclador tipo de instrumento de pago

RF90	Suprimir nomenclador tipo de instrumento de pago
	Gestionar contrato
RF91	Crear contrato
RF92	Editar contrato
RF93	Listar contrato
RF94	Buscar contrato
RF95	Suprimir contrato
RF96	Mostrar contrato
RF97	Importar documento de contrato
	Gestionar nomenclador de tipo de contrato
RF98	Crear nomenclador de tipo de contrato
RF99	Editar nomenclador de tipo de contrato
RF100	Mostrar nomenclador de tipo de contrato
RF101	Listar nomenclador de tipo de contrato
RF102	Buscar nomenclador de tipo de contrato
RF103	Suprimir nomenclador de tipo de contrato
	Gestionar punto de aprobación
RF104	Crear punto de aprobación
RF105	Editar punto de aprobación
RF106	Mostrar punto de aprobación
RF107	Listar punto de aprobación
RF108	Buscar punto de aprobación
RF109	Suprimir punto de aprobación
	Gestionar acta de entrega
RF110	Crear acta de entrega
RF111	Editar acta de entrega
RF112	Mostrar acta de entrega
RF113	Listar acta de entrega
RF114	Buscar acta de entrega
RF115	Suprimir acta de entrega
	Gestionar solicitud de transporte de carga
RF116	Crear solicitud de transporte de carga
RF117	Editar solicitud de transporte de carga
RF118	Listar solicitud de transporte de carga
RF119	Buscar solicitud de transporte de carga
RF120	Suprimir solicitud de transporte de carga
RF121	Mostrar solicitud de transporte de carga
RF122	Imprimir solicitud de transporte de carga
	Gestionar reclamación
RF123	Crear reclamación
RF124	Listar reclamación
RF125	Buscar reclamación
RF126	Suprimir reclamación
RF127	Mostrar reclamación
RF128	Editar reclamación

Anexo 4 Acta de aceptación



Acta de aceptación

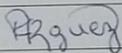
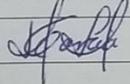
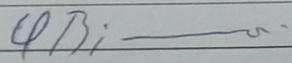
Fecha: 18/06/2019

ACTA DE ACEPTACIÓN

En cumplimiento del Convenio de colaboración entre la **Dirección de Compras** y en función de la ejecución: **Centro de Informatización de Entidades**, se hace entrega de los productos que se relacionan a continuación:

- *Modelo de procesos*
- *Descripción de Requisitos de Software*
- *Diseños de Casos de Prueba*
- *Código fuente*
- *Revisión y aprobación de los Requisitos por el cliente.*
- *Realización de pruebas exploratorias al Sistema de Compras de la Universidad de las Ciencias Informáticas.*

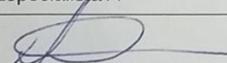
La **Parte Cliente**, luego de haber revisado los productos de trabajo determina que acepta los productos de trabajo desarrollados.

Entrega	Recibe
Nombre y apellidos: Patricia Hernández Rodríguez	Nombre y apellidos: Héctor García Llanes
Cargo: -	Cargo: Director de Compras
Firma: 	Firma: 
Entrega	
Nombre y apellidos: José Pablo Bejarano Ruiz	
Cargo: -	
Firma: 	

Representante Parte Suministradora (CEIGE)

Nombre y Apellidos: Miguel Angel Sánchez Palmero

Cargo: Especialista A

Firma: 

Nombre y Apellidos: Olga Yarisbel Rojas Grass

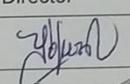
Cargo: Jefa de Departamento de Aplicaciones de Gestión Empresarial

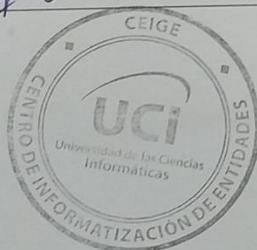
Firma: 

Observador independiente

Nombre y Apellidos: Yoansy López Reyes

Cargo: Director

Firma: 



Anexo 5 Cuestionario de la técnica ladov

Evaluación de la técnica de ladov		
1.	¿Considera factible el desarrollo de la gestión de compras sin contar con una herramienta que centralice la información relacionada con el proceso y esté disponible desde cualquier área de la universidad?	Sí ___ No sé___ No___
2.	¿Considera usted que el sistema propuesto facilita la estandarización de la información para la gestión de necesidades?	Sí ___ No___
3.	¿Considera usted que el sistema desarrollado realiza un seguimiento adecuado de la ejecución de la compra?	Sí ___ No___
4.	¿Utilizaría usted el Sistema de Compras desarrollado para la gestión de Gestión de necesidades, Contratación, Seguimiento de la demanda y Recepción de mercancía?	Sí ___ No sé___ No___
5.	¿Considera usted que la solución propuesta permite agilizar los procesos Gestión de necesidades, Contratación, Seguimiento de la demanda y Recepción de mercancía?	Sí ___ No___
6.	¿Le satisface este sistema para su utilización en la Dirección de Compras?	<input type="checkbox"/> Me gusta mucho <input type="checkbox"/> No me gusta tanto <input type="checkbox"/> Me da lo mismo <input type="checkbox"/> Me disgusta más de lo que me gusta <input type="checkbox"/> No me gusta nada <input type="checkbox"/> No sé qué decir

7. ¿Qué elementos considera positivo del Sistema de Compras desarrollado?

8. ¿Qué elementos considera negativo del Sistema de Compras desarrollado?

9. ¿Qué importancia le concede al sistema desarrollado para la gestión de las compras?