



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**  
**Centro de Informatización de Entidades**  
**FACULTAD 3**

**Componente para la gestión de información del Comité  
de Contratación en el Sistema de Importación de la  
empresa TECNOTEX**

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

**Autor:**

**Gabriel Reyes Garay**

**Tutora:**

**MSc. Olga Yarisbel Rojas Grass**  
**Ing. Dagoberto Félix Pérez Montesinos**

**La Habana, 2020**  
**“Año 62 de la Revolución”**

## DECLARACIÓN DE AUTORÍA

Declaro ser el autor de este trabajo titulado: Componente para la Gestión de Información del Comité de Contratación en el Sistema de Importación de la Empresa TECNOTEX 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\_\_\_\_\_.

Autor:

---

Gabriel Reyes Garay

Tutores:

---

MSc. Olga Yarisbel Rojas Grass

---

Ing. Dagoberto Félix Pérez Montesinos

## **AGRADECIMIENTOS**

*A mis profesores durante todo mi tiempo de vida estudiantil por guiarme como una segunda familia.*

*A mis tutores Olga Yarisbel Rojas Grass y Dagoberto Félix Pérez Montesino por su dedicación y esfuerzo en mi último reto antes de convertirme en un profesional.*

*A mi ejemplar familia por todo el apoyo incondicional que me han ofrecido a lo largo de todos mis años de vida. A mis abuelos por su amor y sabiduría. A mi padre Osvaldo Reyes Acosta, mi primer amigo, por ser un gran ejemplo a seguir como hombre y como futuro ingeniero; de ti he adquirido mi forma de ser y pensar; pueden existir muchas personas en mi vida, pero eso nunca podrá cambiar. A mi madre Hilda Elisa Garay Pérez por ser una fiel educadora de principios y ética, por ofrecerme ese amor incondicional que es imposible describir, por tratarme como un bebé a pesar de mis años; frente a tus ojos siempre seré aquel niño pequeño que en tus brazos creció, eso jamás se podrá cambiar, pero no importa, no quiero que cambie, eso me recuerda que jamás estamos solos en esta vida y que no importa quiénes somos, siempre comenzamos de algo que jamás debemos olvidar. A mi hermano Osvaldo Reyes Garay, por ser aquella persona que a mi lado crece; la sabiduría adquirida de ti no se logra expresar, somos aquel árbol que bajo las sombras de un bosque creció. Siempre estaremos, uno al lado del otro, para recordar quiénes somos y de dónde provenimos.*

*A mi gran amigo Erne, mi hermano. Aunque embarazosas palabras sean, gracias por apoyarme en cada momento de mi vida, por ofrecerme tus grandiosos consejos en los momentos más cruciales. A pesar de no poder seguir compartiendo todos los días a las 10 de la noche nuestras interesantes charlas sobre el mundo, te agradezco enormemente por siempre estar ahí. A mi querida compañera Adriana, una ejemplar amiga de mis recientes años, gracias por ayudarme a crecer como hombre. En general, a mis amigos y compañeros de aula, los nuevos conocidos y los que dejé atrás, por todo el apoyo brindado en los momentos malos y buenos.*

*A todos ustedes,*

**MUCHAS GRACIAS**

## DEDICATORIA

*Dedicado a mis padres,*

*Oswaldo Reyes Acosta e Hilda Elisa Garay Pérez*

*y a mi hermano,*

*Oswaldo Reyes Garay*

## RESUMEN

Los sistemas de gestión empresarial ofrecen mejoras en la utilización de la información que soportan el desarrollo y el crecimiento económico de cualquier organización. Estos sistemas permiten gestionar y controlar cada actividad de las empresas para lograr una mayor competitividad en el mercado. El proceso de contratación es una de las actividades en las que estos sistemas pueden ser de gran apoyo. Este proceso, en la empresa TECNOTEX, permite controlar todos aquellos contratos relacionado a los pedidos que realizan los clientes y los proveedores relacionados con su mercado. Actualmente, esta empresa no cuenta con un sistema informatizado, que gestione de forma eficiente esta actividad. Por tal motivo, en el presente trabajo se desarrolló un componente para el Comité de Contratación en la empresa TECNOTEX que permite disminuir el tiempo en la obtención de la información utilizada en la gestión de contratos. Para la construcción del componente, se utilizó la Metodología de desarrollo para la Actividad productiva de la Universidad, como marco de trabajo Odoon en su versión 13.0, el cual incluye la utilización del lenguaje XML (Lenguaje de Marcas Extensible) en su versión 1.0, Python en su versión 3.7 y PostgreSQL en su versión 12.1. Además, se utilizó como entorno de desarrollo PyCharm en su versión 2019.2.4. El diseño e implementación del componente fue validado a través de técnicas que determinaron el cumplimiento de los requisitos definidos en una fase previa a la formalización del presente trabajo.

**Palabras Clave:** acta del comité, agenda del comité, comité de contratación, nota informativa, sistema de gestión.

## **ABSTRACT**

Business management systems offer improvements in the use of information that support the development and economic growth of any organization. These tools allow managing and controlling each area of the companies and obtain greater competitiveness in the market. One of the areas in which the use of this technology most influences is related to the hiring processes. This process, in the TECNOTEX Company, allows to control all those orders and suppliers related to its market. At this moment, there is no computer system efficiently managing this activity. For this reason, in the present work a component was developed for the Hiring Committee in the company TECNOTEX. The developed component reduces the time for planning and obtaining the information used in contract management. For the construction of the component, the Development Methodology for the Productive Activity of the University was used, as an Odoo framework in its version 13.0, which includes the use of the XML language (Extensible Markup Language) in its version 1.0, Python in its version 3.7 and PostgreSQL in its version 12.1. In addition, PyCharm was used as the development environment in its version 2019.2.4. The design and implementation of the component were validated through techniques that determined compliance with the requirements defined in a phase prior to the formalization of this work.

**Key Words:** briefing note, committee agenda, hiring committee, information note, management system.

## ÍNDICE

<b>Introducción</b> .....	1
<b>Capítulo 1: Fundamentación Teórica</b> .....	6
1.1 <b>Introducción</b> .....	6
1.2 <b>Principales conceptos relacionados</b> .....	6
1.3 <b>Sistemas homólogos</b> .....	7
1.4 <b>Metodología de desarrollo de software</b> .....	11
1.5 <b>Lenguajes</b> .....	13
1.6 <b>Herramientas y tecnologías</b> .....	14
1.7 <b>Conclusiones parciales</b> .....	15
<b>Capítulo 2: Propuesta e Implementación de la Solución</b> .....	16
2.1 <b>Introducción</b> .....	16
2.3 <b>Requisitos Funcionales</b> .....	16
2.3.3 <b>Descripción de requisitos funcionales</b> .....	17
2.3.4 <b>Requisitos no Funcionales</b> .....	21
2.4 <b>Análisis y diseño</b> .....	23
2.4.1 <b>Diseño arquitectónico</b> .....	23
2.4.3 <b>Patrones de diseño</b> .....	24
2.4.2 <b>Diagrama de clases del diseño</b> .....	26
2.4.4 <b>Modelo de datos</b> .....	27
2.4.5 <b>Validación del diseño</b> .....	28
2.5 <b>Implementación</b> .....	33
2.5.1 <b>Diagrama de Componentes</b> .....	33
2.5.2 <b>Estándares de codificación</b> .....	34
2.6 <b>Conclusiones parciales</b> .....	37
<b>Capítulo 3: Validación de la propuesta de solución</b> .....	38

<b>3.1 Introducción.....</b>	<b>38</b>
<b>3.2 Pruebas internas .....</b>	<b>38</b>
<b>3.2.1.1 Técnica del Camino Básico.....</b>	<b>38</b>
<b>3.2.2 Prueba de Caja Negra .....</b>	<b>41</b>
<b>3.2.2.1 Técnica de Partición de Equivalencia .....</b>	<b>41</b>
<b>3.3 Aplicación de la Técnica de ladov .....</b>	<b>44</b>
<b>3.5 Conclusiones parciales .....</b>	<b>47</b>
<b>CONCLUSIONES GENERALES .....</b>	<b>48</b>
<b>RECOMENDACIONES .....</b>	<b>49</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>50</b>
<b>ANEXOS.....</b>	<b>52</b>

## INDICE DE FIGURAS

Figura 1. Escenario No 1 .....	12
Figura 2. Escenario No 2 .....	12
Figura 3. Escenario No 3 .....	12
Figura 4. Escenario No 4 .....	12
Figura 5. Modelo Vista Controlador.....	24
Figura 6. Ejemplo de Patrón de comportamiento, Observador .....	26
Figura 7. Ejemplo de Patrón Creacional, Instancia Única.....	26
Figura 8. Diagrama de clase de diseño .....	26
Figura 9. Modelo de datos de la solución .....	28
Figura 10. Resultado del atributo Responsabilidad .....	30
Figura 11. Resultado del atributo Complejidad.....	30
Figura 12. Resultado del atributo Reutilización .....	30
Figura 13. Resultado del Acoplamiento.....	32
Figura 14. Resultado del Complejidad de Mantenimiento .....	32
Figura 15. Resultado del atributo Cantidad de pruebas.....	32
Figura 16. Resultado del atributo Reutilización .....	32
Figura 17. Diagrama de componente .....	33
Figura 18. Módulo Nota Informativa .....	34
Figura 19. Prototipo funcional: crear Agenda de Comité .....	36
Figura 20. Prototipo funcional: crear Acta de Comité .....	36
Figura 21. Gráfico de Código .....	39
Figura 22. Grafo resultante del método.....	39
Figura 23. Resultado del ISG .....	45
Figura 24. Intervalo de satisfacción.....	45
Figura 25. Resultado del ISG .....	46

## ÍNDICE DE TABLAS

Tabla 1. Tabla comparativa de los sistemas citados .....	10
Tabla 2. Listado de requisitos funcionales.....	16
Tabla 3. Descripción del Requisito Funcional Crear Agenda de Comité.....	17
Tabla 4. Requisitos no Funcionales .....	22
Tabla 5. Métrica de tamaño operacional TOC.....	29
Tabla 6. Criterio de evaluación para la métrica TOC.....	29
Tabla 7. Atributos de calidad evaluados por la métrica RC .....	31
Tabla 8. Criterios de evaluación para la métrica RC .....	31
Tabla 9. Camino básico 1: 1-2-5 .....	40
Tabla 10. Camino básico 2: 1-2-3-4-5.....	40
Tabla 11. Camino básico 3: 1-2-3-2-5.....	40
Tabla 12. Camino básico 4: 1-5 .....	40
Tabla 13. Casos de Prueba del RF: Crear agenda de comité .....	42
Tabla 14. Niveles de satisfacción.....	44
Tabla 15. Cuadro lógico de ladov .....	45
Tabla 16. Resultado de la encuesta.....	46

### Introducción

El desarrollo científico-tecnológico permite acceder, organizar y comunicar información relevante con mayor eficiencia. Por esta razón, una de las metas del sector empresarial está destinada a la adopción de nuevas tecnologías relacionadas con la información. Esto se debe a que generan un conjunto de conocimientos claves para la evolución tecnológica. [1]

En la actualidad, la implementación de herramientas y técnicas para la administración de las empresas, han generado grandes cantidades de datos que necesitan ser procesados. Esto ha conllevado a la aparición sistemas informáticos que permiten analizar y procesar dichos datos, han transformado la manera de trabajar y gestionar los recursos de las mismas. Por lo que, un buen uso de estos sistemas permite mejorar la administración empresarial y lograr mayor competitividad en el mercado. [2]

En Cuba, el proceso de adquisición de sistemas informáticos comienza a partir de la necesidad de la informatización de la sociedad, siendo la Universidad de las Ciencias Informáticas (UCI) su principal promotor. En ella se encuentran adscritos los centros de desarrollo de productos y servicios, con un elevado nivel de compromiso, ética y profesionalidad. Actualmente unos de los centros de desarrollo que permite satisfacer las necesidades y problemas de la sociedad con la introducción de productos informáticos es el Centro de Informatización de Entidades (CEIGE). Este tiene como misión desarrollar productos y servicios informáticos para la gestión de entidades, contribuyendo a la formación de estudiantes y profesionales. CEIGE desarrolla varios proyectos de gestión empresarial, uno de ellos está dedicado al desarrollo del Sistema de Importación y Suministro para la empresa TECNOTEX.

TECNOTEX es una empresa cubana que se dedica a la importación de productos para la construcción. Las importaciones son la introducción legítima de bienes y servicios del extranjero adquiridos por un país para distribuirlos en el interior de éste. Actualmente, esta actividad es regulada por la resolución No.50 de 2014 del Ministerio del Comercio Exterior con el objetivo de lograr un adecuado ordenamiento de la actividad comercial exterior. Existen múltiples motivos para realizar una importación de productos: innovación, mejora de la calidad, reducción de costos, nuevas posibilidades de negocio, pero siempre resulta imprescindible disponer de un plan de importación. En Cuba varias empresas se dedican a la importación de mercancías entre las que se pueden citar: CUBAZUCAR, TRANSIMPORT, BK-IMPORT EXPORT y TECNOTEX. Esta última, dedicada a la ejecución directa y concreta de las

operaciones de exportación e importación de artículos y productos técnicos especializados que se determinen por el organismo y autoridad competente. Para la realización de las actividades de exportación e importación, TECNOTEX tiene distribuidas sus acciones en tres procesos fundamentales, Presentación y Revisión de pedidos, Contratación y Suministro de mercancía.

Durante la ejecución del proceso de Contratación actúa un Comité de Contratación. Este órgano es el encargado de evaluar y aprobar, previa a su formalización, las operaciones de importación y exportación, garantizando que las mismas satisfagan los requerimientos establecidos en materia comercial, financiera, técnica y legal. [3]

En TECNOTEX, se realiza una reunión por el Comité de Contratación una vez cada tres semanas. En estas reuniones se analizan los pedidos, las notas informativas del momento y se discuten los estados de las que fueron revisadas en otro comité. Para ello, el personal implicado en dicho proceso utiliza una Agenda del Comité, donde quedan plasmadas informaciones referentes a los pedidos y notas informativas a analizar, así como indicaciones y acuerdos tomados. Cada Agenda del Comité guarda información histórica relacionada con las reuniones pasadas, a partir de los pedidos y las notas informativas en la que su estado aún está pendiente a cierre, o no proceden debido a algún problema relacionado con el proceso de Importación y Suministro de mercancía.

Durante la identificación de los pedidos a analizar, se verifica si los productos o servicios a contratar superan un monto máximo definido por TECNOTEX. Una vez identificados estos pedidos, se analizan los proveedores relacionados con ellos. Éstos, son aquellas entidades que ofertan una gama de productos o servicios a un valor específico. El análisis de ellos se realiza a través de un estudio del mercado en el que se decide si comprar una parte o todo de sus ofertas.

Las notas informativas se realizan con el objetivo de formalizar el proceso de aceptación de estos proveedores. Estas son de tres tipos, de inclusión permanente, de inclusión excepcional o de inclusión a la salida del mercado de TECNOTEX. La diferencia principal radica en la selección de las ofertas de los proveedores. Esto se debe, a que existen proveedores que ofertan sus productos a mejor precio o son de mayor ventaja para la empresa. La inclusión de forma permanente se les da a aquellos proveedores a los que TECNOTEX decide aceptar la compra de la gama completa de sus ofertas. La inclusión excepcional se realiza, por decisión de la empresa, cuando una parte del contenido de una oferta de un proveedor es más favorable, en comparación con otros proveedores. Sin embargo, no ocurre con el resto de la

oferta. La inclusión a la salida del mercado de TECNOTEX, se ofrece cuando la empresa decide que los productos de un proveedor son de gran beneficio para ella, y decide incluirlo en su propio mercado de ofertas.

Una vez culminada la reunión del Comité de Contratación, se elabora un Acta del Comité por cada uno de los pedidos y notas informativas. Para ello, se asigna una persona encargada de escribir, en un Excel complejo, todas las decisiones tomadas, los miembros implicados, así como las indicaciones dadas por los especialistas del mercado de TECNOTEX. Además, dicha persona es la encargada de incluir los datos en el documento digital de las Actas del Comité a imprimir. Sin embargo, a pesar de ser un proceso organizado, necesariamente requiere de mucho tiempo y atenta contra la eficiencia del proceso global de la empresa.

En la actualidad, los procesos antes descritos presentan dificultades en los sistemas informáticos que contribuyen a su adecuada gestión, lo que trae consigo que existan:

- Pérdida de oportunidades para la empresa, producto a que no se dispone con facilidad la información necesaria para analizar las ofertas que ofrecen los proveedores.
- La información obtenida para la generación de las Actas del Comité está propensa a errores. Esto se debe a que la persona encargada de registrar cada una estas actas, obtiene los datos a partir de un Excel complejo.
- Pérdida de documentos, debido a que el almacenamiento de las Actas y Agendas del Comité se realiza en formato duro. Esto trae consigo que estén propensos al deterioro por el tiempo, así como de la ineficiencia a la hora de archivarlas. Además, debido al dinamismo de la información utilizada en ambos, resulta complejo el empleo de un Excel para su almacenaje.
- Dificultad en la obtención de información histórica de las agendas antes definidas, debido a la inexistencia de una base de datos para almacenar dichos datos.

Por lo que, a partir de la problemática antes planteada se define como:

**Problema a resolver:** ¿Cómo disminuir el tiempo en la obtención de la información que utiliza el Comité de Contratación en la empresa TECNOTEX para la gestión de los contratos?

**El objeto de estudio:** la gestión de los contratos en sistemas empresariales.

**Campo de acción:** los sistemas de gestión empresarial.

**Objetivo general:** desarrollar un componente que permita disminuir el tiempo en la obtención de la información, que utiliza el Comité de Contratación en la empresa TECNOTEX para la gestión de los contratos.

### Objetivos específicos:

- Construir el marco teórico de la investigación para sustentar los conceptos, la propuesta de desarrollo de las funcionalidades, las herramientas y tecnologías a utilizar.
- Realizar el análisis y diseño de la solución a implementar, teniendo en cuenta las necesidades del cliente.
- Implementar el componente para la gestión de información del Comité de Contratación.
- Validar la propuesta de solución, con el empleo de pruebas de software, técnicas y métodos.
- Validar los resultados de la investigación con el empleo de la técnica de ladov.

**Idea a defender:** si se desarrolla un componente para el Sistema de Importación de la Empresa TECNOTEX, permitirá disminuir el tiempo en la obtención de la información que utiliza el Comité de Contratación en la empresa TECNOTEX para la gestión de los contratos.

### Métodos científicos de investigación

#### Métodos teóricos:

- **Histórico-Lógico:** se emplea con el objetivo de analizar el proceso de gestión de la información que realiza el Comité de Contratación del grupo empresarial TECNOTEX, así como para comprender los sistemas que permiten mejorar la gestión empresarial.
- **Modelación:** se utiliza para crear modelos y diagramas que permitan un mejor entendimiento de los procesos del sistema, destacando: el modelo conceptual, los diagramas de clases del diseño, el modelo entidad-relación y el diagrama de componentes.
- **Hipotético-deductivo:** se utiliza para elaborar la idea a defender y trazar estrategias para cumplirla, validando finalmente a través de técnicas y métodos.

#### Métodos empíricos:

- **Observación:** se utiliza para adquirir experiencias reales sobre la gestión empresarial que realiza el comité de contratación del grupo empresarial TECNOTEX.
- **Entrevista:** se realiza a los especialistas que laboran en los procesos relacionados directamente con el proyecto. Tiene el objetivo de precisar el funcionamiento del negocio y de los requisitos relacionados con el Comité de Contratación.

### **Estructura de los capítulos**

La presente investigación consta de tres capítulos. A continuación, se describen los principales elementos de cada uno de los capítulos.

**Capítulo 1. Fundamentación teórica:** en este capítulo se describe la fundamentación teórica que sirve como base para la investigación, referente a los conceptos relacionados con el proceso administrativo que realiza el comité de contratación del grupo empresarial TECNOTEX. Se realiza un estudio a sistemas empresariales para la propuesta de solución teniendo como criterios de búsquedas las necesidades definidas por el cliente. Además, se describen las tecnologías, lenguajes y herramientas utilizadas para el modelado e implementación, así como la metodología que guiará el proceso de desarrollo de software.

**Capítulo 2. Propuesta e implementación de la solución:** en este capítulo se realiza el análisis y diseño de la propuesta de solución. Se describe la arquitectura base, los patrones de diseño, el uso y los beneficios de los mismos. También se incluye la elaboración de los diagramas de clases del diseño, el modelo entidad relación, así como la validación de análisis y diseño de éstos. Por último, se describe la implementación, mostrando así un fragmento del código y el uso adecuado de estándares para el mismo.

**Capítulo 3. Validación de la propuesta de solución:** en este capítulo se describen y utilizan las pruebas para la validación de la disciplina de implementación, teniendo en cuenta la metodología seleccionada. Además, se realiza la validación de la investigación.

### Capítulo 1: Fundamentación Teórica

#### 1.1 Introducción

En el presente capítulo se abordan los principales conceptos relacionados con la gestión de los contratos en empresas importadoras de artículos y productos. Se realiza un estudio del estado del arte de distintos sistemas informáticos que apoyan el proceso de contratación en las empresas. Entre ellos, se analizan sistemas ERP y otros sistemas a la medida que permiten crear y modificar documentos técnicos legales relacionados con los contratos. Además, se definen las herramientas, metodología y lenguajes, que se emplean en la propuesta del sistema.

#### 1.2 Principales conceptos relacionados

##### Contratos

Según la Real Academia Española de idioma, un contrato es un pacto o convenio, oral o escrito, entre partes que se obligan sobre materia o cosa determinada, y a cuyo cumplimiento puede ser con obligación o por autoridad. Es un documento que recoge las condiciones de este pacto o convenio. [4]

Las definiciones de contrato usualmente pueden ser establecidas en dos sentidos, tanto en términos de acuerdos o como promesas. El contrato es un acuerdo, el cual es legalmente forzoso o legalmente reconocido como creador de un deber. Otra definición usada es la que considera al contrato como una promesa o conjunto de promesas, que en caso de incumplimiento el derecho otorga un remedio o el reconocimiento por parte del derecho, en cierto sentido, como un deber. [5]

El tratadista argentino Jorge Mossete Iturraspe, en alusión a la definición legal argentina del contrato, define el contrato como: el acuerdo de varias personas sobre una declaración de voluntad común destinada a reglar sus derechos, es una de las especies dentro del género acto o negocio jurídico.” [5]

Tomando como referencia lo antes citado, un contrato es: un acuerdo tanto oral como escrito sobre condiciones definidas entre varias personas dentro de las cuales están legalmente forzado o reconocido a darle cumplimiento.

### **Sistemas de gestión empresarial**

El surgimiento de sistemas para la gestión empresarial está comprendido como parte de la evolución de la administración de materiales de las empresas, y del avance de la tecnología en los últimos cien años. La integración de diferentes áreas de las empresas como ingeniería, ventas, fabricación o compras bajo un mismo sistema de información, permitió mejorar el enfoque de la gestión empresarial incorporando otras áreas en un único sistema integral. Debido a esto, comienzan a aparecer sistemas que abordan la planificación de recursos humanos o financieros junto con la planificación de necesidades de materiales y recursos de producción. [6]

En el contexto actual de la gestión empresarial coexisten dos tendencias complementarias: la primera es la implementación de sistemas de gestión a partir de los correspondientes estándares nacionales e internacionales, que facilitan de forma separada el desempeño de cada una de las funciones técnicas asociadas a las diferentes partes interesadas, y la segunda es el diseño de esquemas de excelencia empresarial que garantiza la satisfacción de los grupos de interés implicados. [7]

Un ejemplo de ellos son los sistemas ERP (Enterprise Resource Planning), softwares soportados por módulos que interactúan entre sí. Estos son un factor crucial para el acceso instantáneo a la información, ya que facilita el flujo de información dentro de la organización y con los proveedores, clientes y otros intervinientes de la cadena de abastecimiento. En esos módulos se incluyen funcionalidades de marketing y ventas, distribución, gestión de producción, control de inventarios, gestión de la calidad, recursos humanos, gestión financiera, contabilidad y gestión de información, entre otros. [8]

### **1.3 Sistemas homólogos**

El análisis de los sistemas de gestión empresarial se realiza a partir de las necesidades que posee la empresa en el proceso de Contratación. Este tiene dentro de sus actividades, el control de los pedidos realizados por los clientes, así como el análisis de las ofertas que envían los proveedores. Para la realización de estas actividades, el Comité de Contratación establece una reunión una vez cada tres semanas, donde se analizan los pedidos de los clientes, donde se analizan los pedidos de los clientes, en el que su costo supere un monto máximo establecido por TECNOTEX o su estado aún está pendiente a cierre por algún problema con el proceso de Importación y Suministros. Para que este proceso ocurra de forma consecuente se crea una Agenda del Comité, en la que se guardan y analizan informaciones referentes a los

pedidos y las notas informativas de un plazo de tiempo definido, así como las almacenadas en agendas anteriores. Como culminación de esta actividad se elabora un Acta del Comité por cada pedido y nota informativa analizadas. Además, se registran las decisiones tomadas, los miembros implicados, así como las indicaciones dadas por los especialistas del mercado de TECNOTEX.

Como propuesta de solución, CEIGE definió que se utilizará el software Odoo, un marco de trabajo modular y escalable bajo la licencia de código libre. Dispone de funcionalidades oficiales y módulos desarrollados por terceros. Este funciona a través de una única plataforma en la que se encuentran integrados todos los módulos. Odoo permite llevar toda la gestión de operaciones, control financiero, productivo y de recursos humanos de las empresas, ya que es sumamente flexible para adaptarse a los procesos comerciales de éstas. Además, ofrece la creación y utilización de módulos externos completamente personalizables y la creación digital de documentos técnicos legales. [9]

Sin embargo, éste no posee un módulo dedicado al proceso de contratación que permita abarcar la totalidad de los requisitos definidos por el cliente, tales como:

- Crear documentos técnicos legales como Actas de Comité, Agendas de Comité y notas informativas con mayor facilidad.
- Crear un Acta del Comité por cada una de las notas informativas y pedidos analizados en las Agendas de Comité.
- Obtener datos históricos de notas y pedidos anteriormente definidos, para así poder analizarlos en las Agenda de Comité.

A partir de lo antes descrito, surge la necesidad de crear un módulo externo, que permita darle cumplimiento al objetivo de la investigación, y de buscar sistemas que permitan realizar la gestión de contratos con el fin de adquirir mayor conocimiento sobre la temática. Para ello se analizaron cuatro sistemas informáticos que permiten gestionar parte del proceso de Contratación. A continuación, se describen dichos sistemas:

**OrbitTrade:** Es un software que se adapta a distintos escenarios para proporcionar un control y un seguimiento exhaustivo sobre el proceso de exportación/importación. Permite gestionar la generación de pedidos, realizar proyecciones, además de ofrecer información detallada sobre el estado de los pedidos y de los estados con relación a las entregas, remisiones de planta a

puertos, mercancía en puertos, embarques, liquidación de exportaciones, costos y estadísticas de ventas. Dentro de sus principales funcionalidades se encuentran: [10]

- ❖ Diseñar contratos con clientes y/o proveedores y anexos al mismo.
- ❖ Catálogos de productos por cliente/proveedor.
- ❖ Gestión documental en cualquier fase del proceso, como por ejemplo contratos, cartas de crédito y transporte.
- ❖ Definición de flujos de procesos con sus validaciones correspondientes.
- ❖ Identificación y control de la responsabilidad de cada inspección.
- ❖ Sistema de decisión de compra que permite analizar la propuesta más adecuada de los proveedores para cada cliente.

**Doxis4:** Es un software que permite, independientemente de la fuente y el formato, registrar, gestionar, procesar, regular, almacenar (con garantía de integridad) y emitir de nuevo en cualquier momento toda la información dentro del ámbito de una empresa y sus sedes. Doxis4 dispone de funciones e información sin redundancias, salvando así las restricciones de los sistemas de información verticales basados en documentos. La arquitectura abierta API facilita la incorporación de Doxis4 en cualquier aplicación empresarial (ERP, CRM, HR, entre otros). El paquete de Doxis4 iECM integra de forma modular captura, archivado electrónico, gestión de documentos, expedientes digitales, colaboración, así como gestión de tareas y procesos empresariales. [11]

**Softland:** Es un software que brinda la posibilidad de realizar el proceso administrativo de contratos de una forma fácil y eficiente. Está especialmente diseñado para empresas que requieren automatizar sus procesos contractuales, tales como contratos de servicios, contratos de arrendamiento de condominios y oficinas, maquinarias, entre otros. El módulo permite administrar diferentes características de los contratos: [12]

- ❖ Tipo de Contrato
- ❖ Estados del Contrato
- ❖ Histórico de documentos relacionados con el contrato
- ❖ Copia de contratos y recalcular de impuestos
- ❖ Razones de suspensión
- ❖ Periodicidades de facturación
- ❖ Procesos de aumento de precios
- ❖ Generación de facturas en lotes e individuales

## Capítulo 1: Fundamentación Teórica

**AB pro:** Es un sistema que permite incrementar el valor de las empresas, optimizar su desarrollo y destacar sus ventajas competitivas. Dentro de los módulos que ofrece se encuentra el desarrollo de la logística y control documental para las empresas, en el cual muestra la disponibilidad de los productos y define los tiempos para lograr el alcance de objetivos en las entregas de pedidos a los clientes. Establece los parámetros para mejorar el tiempo de entrega de pedidos, con el fin de garantizar la satisfacción del mercado. Algunas de las funcionalidades que ofrece son: [13]

- ❖ Un módulo para el flujo completo de pedidos y órdenes de compra, controlando el proceso desde el origen, hasta la entrega final.
- ❖ Genera y adjunta automáticamente el formato de Pedidos y Cotizaciones de clientes, así como Órdenes de Compra de proveedores en formato PDF. Opcionalmente, también lo puede enviar por correo electrónico.
- ❖ Establece la planeación y el flujo de los pedidos de los clientes, así como las órdenes de compra que la empresa genera.
- ❖ Permite la gestión de remisiones de productos, así como la búsqueda de remisiones y facturas desde pedidos.
- ❖ Gestiona y controla la administración de contratos comerciales.

Teniendo en cuenta las características de los sistemas antes planteados, se muestra la siguiente tabla comparativa. Esta es creada a partir de las necesidades del cliente y las políticas de soberanía e independencia tecnológica.

Tabla 1. Tabla comparativa de los sistemas citados

SISTEMAS	MULTIPLATAFORMA	SOFTWARE LIBRE	SOPORTE	GESTIONAR CONTRATOS	GESTIONAR ACTAS	GESTIONAR AGENDA
<b>ORBITTRADE</b>	Sí	Sí	Sí	Sí	Sí	No
<b>DOXSI4</b>	Sí	Sí	Sí	Sí	Sí	No
<b>SOFTLAND</b>	Sí	No	Sí	Sí	Sí	No
<b>ODOO ERP</b>	Sí	Sí	Sí	Sí	Sí	Sí
<b>AB PRO</b>	Sí	No	Sí	Sí	Sí	Sí

Después de la realización del estudio de los sistemas mencionados anteriormente, se puede concluir que:

- ❖ SOFTLAND y AB pro permitieron ver como es el funcionamiento de sistemas relacionados con el proceso de contratación a través del enfoque de trabajo orientado a la nube. El estudio de éstos, se realizó con el fin de desarrollar la propuesta de solución

lo más adaptable posible, disminuyendo la dependencia entre los datos y creando la menor cantidad de relaciones posibles.

- ❖ El sistema DOXSI4 permitió ver como se realiza un flujo completo del proceso de contratación y la utilización de los documentos técnicos legales con alta flexibilidad a la hora de su creación.
- ❖ OrbitTrade al ser un software dedicado a la gestión de importaciones y exportaciones, permite ver cómo funciona el control de los pedidos a través de interfaces amigables. Como generar diferentes estadísticas por ofertas, países, proveedores, así como el análisis de resultados de la gestión de toda la operación relacionada con la actividad de importación y exportación.

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

### 1.4 Metodología de desarrollo de software

Las metodologías de desarrollo de software son indispensables en la creación o actualización de un proyecto de software. Son una parte fundamental en su ciclo de vida y constituyen el estudio y determinación del método más adecuado para el desarrollo. [14]

En la Universidad de las Ciencias Informáticas, este proceso está enmarcado por la utilización de la metodología AUP-UCI, que es una variante del proceso Unificado Ágil (AUP, por sus siglas en inglés) de Scott Ambler y a su vez es una versión simplificada del Proceso Unificado de Rational (RUP). AUP describe de manera simple la forma de desarrollar aplicaciones de software de negocio, con el uso de técnicas ágiles y conceptos, que aún se mantienen válidos en RUP. El objetivo de hacer una variación de la metodología AUP es el de crear un entorno de trabajo, que se adapte al ciclo de vida definido para la actividad productiva de la UCI. [15]

AUP-UCI propone tres fases (Inicio, Ejecución y Cierre) para el desarrollo de software y 11 roles por el cual debe guiarse el equipo de desarrollo para lograr una mayor organización. Propone también 7 disciplinas (Modelado de Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas Internas, Pruebas de Liberación y Pruebas de Aceptación) al igual que AUP; pero a un nivel más atómico. Una vez definidos los requisitos en el modelado de negocio, AUP-UCI define 4 escenarios para la modelación de éstos, publicados por Sánchez, T.R. [15]

### Escenario No 1:

Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.



Figura 1. Escenario No 1

### Escenario No 2:

Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.



Figura 2. Escenario No 2

### Escenario No 3:

Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.



Figura 3. Escenario No 3

### Escenario No 4:

Proyectos que no modelen negocio solo pueden modelar el sistema con HU.



Figura 4. Escenario No 4

En el presente trabajo, se presenta la solución del sistema a partir de la fase de ejecución, generando los productos de trabajo de las disciplinas de análisis y diseño, implementación, pruebas internas y pruebas de aceptación. El escenario utilizado es el número tres, el cual aplica a los 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. Además, este escenario es el seleccionado para la propuesta de solución por la complejidad y cantidad de requisitos que responden a los procesos de negocios identificados durante la etapa de Levantamiento de Información.

Una vez identificada la metodología a utilizar, se prosigue a describir los lenguajes y herramientas utilizadas en el proceso de desarrollo del software.

### 1.5 Lenguajes

Un lenguaje es un conjunto de símbolos y reglas estandarizados que se utilizan para modelar y desarrollar parte de un diseño de software orientado a objetos. Comúnmente son utilizados en combinación con una metodología de desarrollo de software para llegar de una especificación inicial a la implementación. [16]

**Lenguaje unificado de modelación (UML) 2.4:** es un lenguaje destinado al modelado de sistemas y procesos. Está basado en la orientación a objetos, lo cual condujo a la creación de lenguajes de programación como Java, C++, C#, o Smalltalk. El objetivo de UML es la unificación de los métodos de modelado de objetos, por medio de la identificación y definición de la semántica de los conceptos fundamentales y la elección de una representación gráfica con una sintaxis simple, expresiva e intuitiva. En la actualidad es promovido por el OMG (Object Management Group) y se ha convertido en un lenguaje de modelación muy extendido, sobre todo por su riqueza semántica, que lo abstrae de numerosos aspectos técnicos. [17] El UML es utilizado para el modelado de los diagramas, que se realizan durante el análisis, diseño e implementación de la propuesta de solución.

**Lenguaje de Marcado Extensible(XML) 1.0:** por definición, es un lenguaje de marcado textual, lo que significa que, a diferencia del caso de RDBMS, los elementos de datos están ordenados por naturaleza. La información de esquema proporcionada externamente, que puede o no estar presente, ayuda a evitar coacciones excesivas y costosas entre tipos de datos. Además, para hacer frente a la estructura de árbol de los documentos XML y la complejidad resultante, las relaciones jerárquicas entre datos, se utilizan las expresiones de ruta regulares. Las referencias se pueden usar para modelar las relaciones que exceden las limitaciones de las estructuras de árbol y requieren lógicas de mapeo adicionales. [18] Se utiliza para la implementación de las interfaces que utilizará el usuario.

**PYTHON 3.7:** es un lenguaje de programación interpretado. La forma en que se desarrolla en el lenguaje y su enfoque orientado a objetos, tiene como objetivo ayudar a los programadores a escribir códigos claros y lógicos para proyectos de pequeña y gran escala. Cuenta con estructuras de datos eficientes y de alto nivel. La sintaxis de Python y su enfoque dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y el desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C). Python también puede

usarse como un lenguaje de extensiones para aplicaciones personalizables. [19] Es empleado como lenguaje para la programación de las clases modelos y controladoras.

### 1.6 Herramientas y tecnologías

Las tecnologías y herramientas a utilizar en el desarrollo de la propuesta de solución fueron definidas a partir de la selección de los lenguajes descritos en el epígrafe anterior. Estas son programas, aplicaciones o simplemente instrucciones usadas para efectuar tareas de modo más sencillo.

**PostgreSQL12.1:** es un gestor de bases de datos relacional, orientado a objetos. Su licencia y desarrollo es de código abierto y es mantenida por una comunidad de desarrolladores, colaboradores y organizaciones comerciales de forma libre y desinteresadamente. Esta comunidad es denominada PDGD (PostgreSQL Global Development Group, por sus siglas en inglés). Es reconocido actualmente como uno de los sistemas más potentes del mercado. Presenta fácil accesibilidad, es multiplataforma y está disponible para su uso en casi todos los sistemas operativos utilizados en la actualidad, sin disminuir su rendimiento.[20] Se usa como gestor de la base de datos utilizado por el marco de trabajo de Odoos para almacenar los datos de los proyectos creados.

**Visual Paradigm 8.0:** es una de las herramientas CASE utilizadas para el modelado de software. Esta herramienta cuenta con los medios necesarios para extender sus funcionalidades, pues da soporte a las extensiones de aplicación. Provee de forma libre una interfaz de programación, que permite a los desarrolladores implementar y reutilizar clases e interfaces, así como, desarrollar funciones agregadas, que son útiles para el desarrollo de software. [16] Se utiliza para la realización del modelado de las clases y sus relaciones a través de UML.

**Pycharm 2019.2.4:** es un IDE o entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje de programación Python. Proporciona análisis de código, depuración gráfica, integración con VCS / DVCS y soporte para el desarrollo web con Django, entre otras bondades. Entre las características fundamentales que posee el PyCharm se encuentran el autocompletado, resaltador de sintaxis, herramientas de análisis y refactorización. Posee un depurador avanzado, además de la integración con lenguajes de plantillas como Mako, Jinja2, Django. Soporta entornos virtuales e intérpretes de Python 2.x, 3.x, PyPy, Iron Python y Jython.

[21] Se utiliza por las ventajas que ofrece como entorno de trabajo para el desarrollo de sistemas basados en el lenguaje de programación Python.

**Odoo 13:** es un software para la gestión empresarial conocido en sus primeras versiones como OpenERP. La marca cambió su enfoque, de una solución ERP a un conjunto de aplicaciones comerciales. Actualmente ofrece más de trescientos sesenta aplicaciones, gracias a los múltiples desarrollos realizados por Odoo S.A. y su comunidad. Su flexibilidad para conectar al cliente con el proveedor, su facilidad para la creación de módulos externos y la posibilidad de personalizarlos son unas de las mayores ventajas que ofrece. Odoo se encuentra bajo la licencia de código abierto (open source) y está disponible en tres ediciones: Community, Enterprise y Online. [22]

**OpenObject 1.0:** Es un marco de trabajo que permite relacionar un lenguaje de programación de base de datos con un nuevo lenguaje de programación. Dentro de sus características se encuentra su enfoque ORM (mapeo relacional de objetos), que tiene el objetivo de eliminar el desajuste de impedancia entre estos lenguajes. [23] Se emplea en el presente trabajo por ser el ORM que utiliza Odoo para el manejo de los datos.

**pgAdmin III 1.22.2:** Es un sistema integral de diseño y gestión de bases de datos PostgreSQL para sistemas Unix y Windows. Está disponible gratuitamente bajo los términos de “La licencia PostgreSQL” y se puede redistribuir, siempre que se cumplan los términos de la licencia. El proyecto está gestionado por el equipo de desarrollo de pgAdmin. La conexión a PostgreSQL se realiza con el empleo de la biblioteca libpq nativa. [24]

### 1.7 Conclusiones parciales

El análisis de los sistemas de gestión empresarial permitió obtener un mayor conocimiento para la definición de la propuesta de solución, que consistió en el desarrollo de un componente para el proceso de Contratación de la empresa TECNOTEX. Se eligió Odoo 13 como marco de trabajo para la implementación de la solución. Además, se seleccionaron PostgreSQL12.1, Visual Paradigm 8.0, Pycharm 2019.2.4 y OpenObject 1.0, como herramientas que cumplen con la política de soberanía tecnológica, que define el país para el desarrollo de sistemas. Se utilizó la metodología AUP-UCI como guía del proceso de desarrollo del ciclo de vida del software.

### Capítulo 2: Propuesta e Implementación de la Solución

#### 2.1 Introducción

En el presente capítulo se describen y validan los artefactos de trabajo propuestos por AUP-UCI para el desarrollo de la solución de las disciplinas de análisis y diseño e implementación en la fase de ejecución. En esta etapa, se detallan los principales artefactos de dichas disciplinas, destacando los diagramas de clases, el modelo entidad-relación y el diagrama de componente. Igualmente, se especifica la utilización de los patrones de arquitectura y de diseño para la concepción de la solución, así como el estándar de codificación para la implementación del sistema.

En una fase previa a la formalización del presente trabajo, se identificó y validó con el cliente los requisitos a implementar. Fueron definidos los artefactos utilizados en la disciplina de modelado de negocio y requisitos, destacando el modelo conceptual, la descripción de los requisitos funcionales y no funcionales. A continuación, se muestra los requisitos funcionales y no funcionales descritos para la propuesta de solución.

#### 2.3 Requisitos Funcionales

Los requisitos funcionales son servicios que debe proporcionar el sistema, manera en que debe reaccionar a entradas y cómo se debe comportar en situaciones particulares. Describen lo que el sistema debe hacer. [25] A continuación, se listan los requisitos funcionales asociados a la propuesta de solución.

Tabla 2. Listado de requisitos funcionales

	<b>Requisitos Funcionales</b>
	<b>Gestionar nota informativa</b>
RF1	Crear nota informativa
RF2	Editar nota informativa
RF3	Listar nota informativa
RF4	Mostrar nota informativa
RF5	Buscar nota informativa
	<b>Gestionar agenda de comité</b>
RF6	Crear agenda de comité
RF7	Editar agenda de comité
FR8	Buscar agenda de comité
FR9	Listar agenda de comité
FR10	Mostrar agenda de comité

## Capítulo 2: Propuesta e Implementación de la solución

	<b>Gestionar acta de comité</b>
RF11	Crear acta de comité
RF12	Editar acta de comité
RF13	Listar acta de comité
RF14	Mostrar acta de comité
RF15	Buscar acta de comité

### 2.3.3 Descripción de requisitos funcionales

La descripción de requisitos especifica con mayor detalle la acción que realiza el mismo en el sistema. Contiene los pasos o las actividades que se deberán realizar para darle cumplimiento al requisito, así como las restricciones a las que está sujeto y las posibles respuestas del sistema. Es una herramienta ampliamente utilizada en el análisis y diseño de un sistema informático.[26] A continuación, se muestra la descripción del requisito funcional: Crear Agenda de Comité.

Tabla 3. Descripción del Requisito Funcional Crear Agenda de Comité

<b>Precondiciones</b>	<p>El usuario se ha autenticado en el sistema y tiene permisos para realizar esta acción.</p> <p>Se ha creado en el sistema al menos una Acta de comité</p>
<b>Flujo de eventos</b>	
<b>Flujo básico Crear agenda del comité</b>	
	Se selecciona la opción Crear agenda del comité
	<p>Se introduce y/o selecciona los siguientes datos:</p> <ul style="list-style-type: none"> <li>- Fecha</li> <li>- Hora</li> <li>- Lugar</li> <li>- Orden del Día (que va estar permanente, pero opción de cambio)</li> <li>- Indicaciones</li> <li>- Acuerdos</li> </ul> <p>Tiene dos pestañas para elegir porque dentro de una agenda puede haber pedidos o nota informativa</p> <p>Cuando es para pedido se introduce los siguientes datos (una vez que se cree un pedido se va crear automáticamente un acta):</p> <ul style="list-style-type: none"> <li>- Unidad comercial (ir al requisito Listar unidades comerciales de la agrupación Gestionar</li> </ul>

## Capítulo 2: Propuesta e Implementación de la solución

unidad comercial)

- Producto (ir al requisito gestionar Acta de Comité)
- Pedido
- Comprador (ir al requisito gestionar Acta de Comité)
- Clientes (ir al requisito Listar clientes de la agrupación Gestionar cliente)
- Fecha de presentación del pedido (ir al requisito gestionar Acta de Comité)
- Salida al mercado (ir al requisito gestionar Acta de Comité)
- Ofertas recibidas (ir al requisito gestionar Acta de Comité)
- Señalamientos (ir al requisito gestionar Acta de Comité)
- Capacidad de Resolver Reclamación
- Tiempo de Entrega (ver Evaluación del proveedor)
- Cantidad Faltante (ver Evaluación del proveedor)
- Cantidad contratos
- Importe Contratado(Total)
- Capital Social (ir gestionar proveedor)

Cuando es para nota informativa se introduce los siguientes datos (una vez que se cree una Nota Informativa se va crear automáticamente un acta):

muestra una lista de notas informativas que tendrá (Unidad comercial, No. pedido, Producto, Estado) que se podrá encontrar en el requisito Gestionar Nota informativa.

Nota: la agenda va hacer creada según las actas

Se selecciona la opción Guardar.

El sistema valida (Ver validación 1, 2, 3, 4, 5, 6, 7) los datos introducidos.

Si los datos son correctos el sistema los registra.

El sistema confirma el registro de los datos.

Concluye el requisito.

### Pos-condiciones

Se registró en el sistema una nueva agenda

### Flujos alternativos

#### Flujo alternativo 7.a Cuando se introducen datos incorrectos.

El sistema señala los datos incorrectos y permite corregirlos.

El usuario corrige los datos.

Volver al paso 4 del flujo básico.

### Pos-condiciones

N/A

#### Flujo alternativo 7.b Cuando se dejan datos vacíos y son obligatorios.

El sistema señala los datos obligatorios y permite corregirlos.

El usuario corrige los datos.

## Capítulo 2: Propuesta e Implementación de la solución

Volver al paso 4 del flujo básico.		
<b>Pos-condiciones</b>		
N/A		
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
Concluye el requisito.		
<b>Pos-condiciones</b>		
N/A		
<b>Validaciones</b>		
Se valida que la Unidad comercial muestre una lista de las unidades comerciales registrada en el sistema.		
Se valida que el Especialista de compra venta y Especialista de precios y mercado muestre una lista de los empleados con esos cargos.		
Se valida que el Pedido permita entrar solo números.		
Se valida que la Cliente muestre una lista de los clientes registrado en el sistema.		
Se valida que se va crear una agenda si su estado está en pendiente a comité y el tipo de comité es comité del GAE o comité de TECNOTEX.		
Se valida que la Fecha de presentación de pedido tenga el formato DD/MM/AAAA.		
Se validan los datos según lo establecido en el CEIGE_TECNOTEX_Modelo Conceptual.		
<b>Conceptos</b> <b>Se deben especificar los atributos obligatorios con un asterisco (*) en el grupo visibles en la interfaz.</b>	Orden del día	Visibles en la interfaz: - Fecha - Hora - Lugar - Orden del Día (que va estar permanente, pero opción de cambio) - Indicaciones - Acuerdos Tiene dos pestañas para elegir porque dentro de una agenda puede haber pedidos o nota informativa Cuando es para pedido se introduce los siguientes datos (una vez que se cree un pedido se va crear automáticamente un acta): - Unidad comercial (ir al requisito Listar unidades comerciales de la agrupación Gestionar unidad comercial) - Producto (ir al requisito gestionar Acta de Comité) - Pedido - Comprador (ir al requisito gestionar Acta de Comité)

## Capítulo 2: Propuesta e Implementación de la solución

	<ul style="list-style-type: none"> <li>- Clientes (ir al requisito Listar clientes de la agrupación Gestionar cliente)</li> <li>- Fecha de presentación del pedido (ir al requisito gestionar Acta de Comité)</li> <li>- Salida al mercado (ir al requisito gestionar Acta de Comité)</li> <li>- Ofertas recibidas (ir al requisito gestionar Acta de Comité)</li> <li>- Señalamientos (ir al requisito gestionar Acta de Comité)</li> <li>- Capacidad de Resolver Reclamación</li> <li>- Tiempo de Entrega (ver Evaluación del proveedor)</li> <li>- Cantidad Faltante (ver Evaluación del proveedor)</li> <li>- Cantidad contratos</li> <li>- Importe Contratado(Total)</li> </ul> <p>Cuando es para nota informativa se introduce los siguientes datos una vez que se cree una Nota Informativa se va crear automáticamente un acta):</p> <p>muestra una lista de notas informativas que tendrá (Unidad comercial, No. pedido, Producto, Estado), esto se podrá encontrar en el requisito Gestionar Nota informativa.</p> <p>Utilizados internamente:</p> <ul style="list-style-type: none"> <li>- Fecha*</li> <li>- Hora*</li> <li>- Lugar*</li> <li>- Unidad comercial *</li> <li>- Propuesta</li> <li>- Pedido*</li> <li>- Especialista de compra y venta*</li> <li>- Clientes</li> <li>- Fecha de presentación del pedido</li> <li>- Salida al mercado</li> <li>- Propuesta de cierre</li> <li>- Especialista de precios y mercado</li> </ul>
<b>Requisitos especiales</b>	Son los requisitos no funcionales específicos para el requisito. Por ejemplo, estándares de intercambio de información.
<b>Asuntos pendientes</b>	Posibles mejoras al requisito.
<b>Prototipo elemental de interfaz gráfica de usuario</b>	

## Capítulo 2: Propuesta e Implementación de la solución

Agenda / Nuevo

Guardar

Fecha:

Hora:

Lugar:

Indicaciones

**Pedidos**

<input type="checkbox"/>	UM
<input type="checkbox"/>	
<input type="checkbox"/>	

Agregar

Agenda del comite de contratación / Nuevo

Guardar

UC  Fecha de presentación pedido

Producto  Salida al mercado

Pedido  Ofertas recibidas

Comprador  Propuesta cierre

Cliente  Capital Social

Señalamientos

Logisticos  Capacidad de Resolver Reclamaciones

Juridico  Tiempo de entrega

Mercado  Cantidad Faltantes

Cantidad contratos

Importe Contratado

Agenda / Nuevo

Guardar

Fecha:

Hora:

Lugar:

Indicaciones

Creado > Confirmado

Orden del día

Acuerdos pendientes

**NI** **Pedidos**

<input type="checkbox"/>	UC	No. NI	Producto	Cierre	Estado
<input type="checkbox"/>					
<input type="checkbox"/>					

### 2.3.4 Requisitos no Funcionales

Los requisitos no funcionales describen aspectos del sistema que son visibles por el usuario que no incluyen una relación directa con el comportamiento funcional del sistema; incluyen restricciones como el tiempo de respuesta, la precisión, recursos consumidos y seguridad. [25] A continuación, se listan los requisitos no funcionales asociados a la propuesta de solución.

## Capítulo 2: Propuesta e Implementación de la solución

Tabla 4. Requisitos no Funcionales

No.	Requisitos no funcionales
<b>Disponibilidad</b>	
RnF 1	Capacidad de almacenamiento: La base de datos debe permitir almacenar datos pasivos con un máximo de hasta 5 años.
RnF 2	Sistema centralizado: El sistema debe ser centralizado con la capacidad de brindarle a los polos el acceso a los datos desde sus regiones.
RnF 3	Tiempo de actividad: El sistema debe mantenerse en ejecución 24x7x365 con una disponibilidad total de 0.99.
<b>Usabilidad</b>	
RnF 4	El componente de la interfaz de usuario debe ejecutarse en un navegador de Internet.
<b>Portabilidad</b>	
RnF 5	La base de datos debe poder ser divisible, cada polo debe poder tener su base de datos.
<b>Interoperabilidad</b>	
RnF 6	Sincronización con servidores de correos (Outlook, Zimbra, etc.)
RnF 7	El sistema debe permitir la exportación o importación de ficheros en formato Excel o XML.
<b>Seguridad</b>	
RnF 8	El sistema debe registrar todas las acciones realizadas por los usuarios en el mismo, así como definir un control de acceso por roles.
RnF 9	Todos los usuarios del sistema operarán sobre este una vez se hayan autenticado, siendo de la misma forma para la comunicación.
<b>Escalabilidad</b>	
RnF 10	La aplicación debe poder manejar una carga máxima de 300 usuarios simultáneos.
<b>Software</b>	
RnF 11	Navegador Mozilla Firefox versión 71 o superior
RnF 12	Como marco de trabajo se utiliza la plataforma de desarrollo Odoo 13.0
RnF 13	Se empleará como Gestor de Base de Datos , PostgreSQL 12.1

## Capítulo 2: Propuesta e Implementación de la solución

RnF 14	El sistema se desarrollará con tecnología Python 3.6
RnF 15	Sistema Operativo Linux utilizando Docker
Hardware	
RnF 16	El sistema para su instalación en las máquinas clientes requiere: Procesador 2.0 GHZ RAM : 2 GB de Memoria Tarjeta de red: 1
RnF 17	El sistema para su instalación en el servidor de aplicaciones requiere: Procesador: 8 GHZ RAM: 16 GB Disco duro 512 GB Tarjeta de red: 1

Una vez culminada la disciplina de Requisitos, se inicia el Análisis y diseño de la solución.

### 2.4 Análisis y diseño

El análisis permite definir en detalle el ámbito del software. Se crean modelos de los requisitos de datos, flujo de información, flujo de control y del comportamiento operativo. Además, la etapa de diseño permite describir como el sistema va a satisfacer los requisitos. El resultado obtenido de la etapa de diseño facilitará la implementación posterior de la solución, pues proporciona la estructura básica y cómo los diferentes componentes actúan y se relacionan entre sí. [27]

#### 2.4.1 Diseño arquitectónico

Para la definición de la arquitectura a utilizar en el presente trabajo, se utilizó el patrón Modelo-Vista-Controlador (MVC), que surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma, que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, de manera que garantiza así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo.[28]

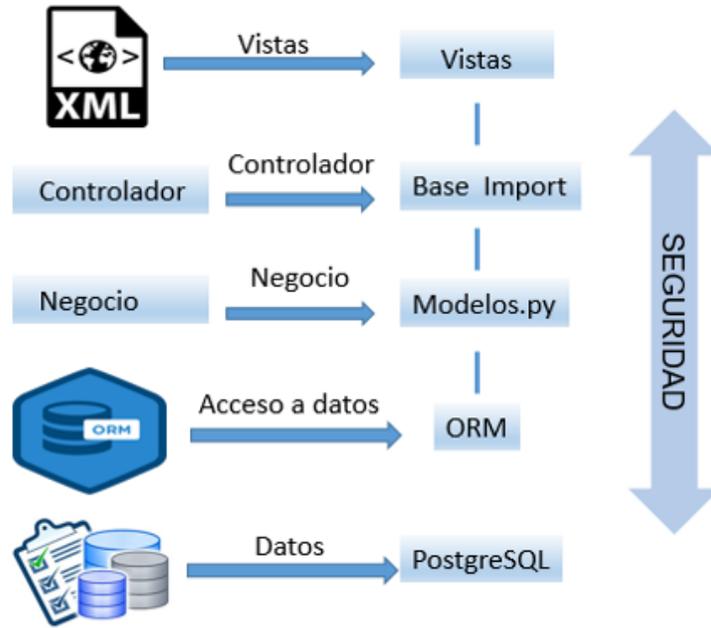


Figura 5. Modelo Vista Controlador

### 2.4.3 Patrones de diseño

El concepto de patrones existe desde hace tiempo y tiene profundas raíces en el trabajo, que el arquitecto Christopher Alexander desarrolló a finales de los 60 y comienzo de los 70. En los años 90 este trabajo comenzó a adaptarse a la ingeniería del software. [29]

Un patrón de diseño nombra, motiva y explica de forma sistemática a una arquitectura, que afronta un problema de diseño recurrente en los sistemas orientados a objetos. Describe el problema, la solución y cuándo aplicarla, además de sus consecuencias. [30]

#### Patrones generales de software para asignación de responsabilidades (GRASP)

La calidad de diseño de la interacción de los objetos y la asignación de responsabilidades presentan gran variación. Las decisiones poco acertadas dan origen a sistemas y componentes frágiles y difíciles de mantener, entender, reutilizar o extender. Una implementación hábil se funda en los principios cardinales, que rigen un buen diseño orientado a objetos. En los patrones GRASP se codifican algunos de ellos, que se aplican al preparar los diagramas de interacción, cuando se asignan las responsabilidades o durante ambas actividades. [31]

A continuación, se citan cinco de estos patrones:

- ❖ **Experto:** un modelo de clases puede definir docenas y hasta cientos de clases de software. Una aplicación puede contener uno o varios modelos de clase, por consiguiente, requiere del cumplimiento de cientos o miles de responsabilidades.

## Capítulo 2: Propuesta e Implementación de la solución

Durante el diseño orientado a objetos, cuando se definen las interacciones entre los objetos, se toman decisiones sobre la asignación de responsabilidades a las clases. Sí se hacen en forma adecuada, los sistemas tienden a ser más fáciles de entender, mantener y ampliar, y se nos presenta la oportunidad de reutilizar los componentes en futuras aplicaciones. [31]

- ❖ **Creador:** la creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. En consecuencia, conviene contar con un principio general para asignar las responsabilidades concernientes a ellas. El diseño, bien asignado, puede soportar un bajo acoplamiento y una mayor claridad, el encapsulamiento y reutilización. [31]
- ❖ **Bajo Acoplamiento:** el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras. Si estas presentaran alto acoplamiento, tendrían los siguientes problemas: los cambios de las clases afines ocasionan cambios locales, son más difíciles de entender cuando están aisladas y de reutilizar, debido a que se requiere la presencia de las otras clases de las que depende. [31]
- ❖ **Alta Cohesión:** en la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que no realicen un trabajo enorme. Una clase con baja cohesión tiende a realizar un trabajo excesivo y trae consigo que: sean más difíciles de comprender, reutilizar y conservar, además de ser afectadas constantemente por nuevos cambios. [31]
- ❖ **Controlador:** es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. [31]

### Patrones GOF (Gang of FOUR)

Los patrones GOF pertenecen al campo del diseño orientado a objetos. Están conformados por veintitrés patrones que se clasifican, según su propósito, en creacionales, estructurales y de comportamiento: [32]

- Creacionales: definen la forma en la que un objeto puede ser creado teniendo en cuenta la reutilización y mutabilidad. Estos describen la mejor manera de manejar instancias.
- Estructurales: describen cómo los objetos y las clases se pueden combinar para formar estructuras.

## Capítulo 2: Propuesta e Implementación de la solución

- Comportamiento: describen la forma de cómo organizar, administrar y combinar conductas y responsabilidades de objetos, centrándose en la comunicación entre ellos.

En el presente trabajo se evidencia la utilización de los patrones GoF:

- Observador: define una dependencia de uno a muchos de forma que, cuando el objeto cambie su estado, los otros objetos sean notificados y se actualicen automáticamente. Separa los objetos de presentación (vistas) de los objetos de datos, de forma que se pueden tener varias vistas sincronizadas de los mismos datos.

```
comite_acta_ni_id = fields.One2many('tecnote.comite_acta_ni', 'comite_acta_ni_id', string='NI')
```

Figura 6. Ejemplo de Patrón de comportamiento, Observador

- Instancia Única: garantiza que una clase solo tenga una instancia y proporciona un punto de acceso global a la misma. En Odoo se define cada clase por un atributo “\_name” a través del cual es el que se tiene acceso a dicha clase

```
acta_buscar_ni = self.env['tecnote.comite_acta'].search([('pedido','=', False)], order='estado desc')
```

Figura 7. Ejemplo de Patrón Creacional, Instancia Única

- Adaptador: convierte la interfaz de una clase en otra distinta, que es la que se esperan los clientes. Esto se evidencia en las Actas del Comité en las que a partir de una misma clase modelo se realiza diferentes interfaces visuales para adaptar la información que necesita el cliente en el momento de crearlas.

### 2.4.2 Diagrama de clases del diseño.

A continuación, se presenta el modelo del diseño del requisito “Gestionar Nota Informativa”.

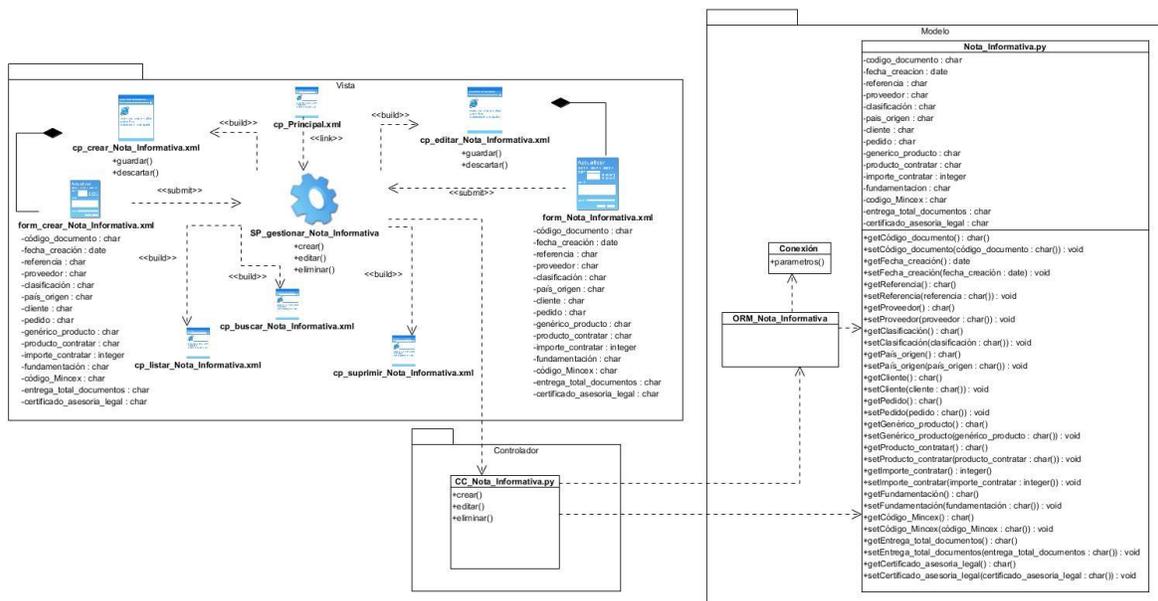


Figura 8. Diagrama de clase de diseño

## Capítulo 2: Propuesta e Implementación de la solución

Las clases del diseño sirven para describir la estructura del sistema mostrando sus clases, atributos y las relaciones entre ellos. En este caso se utilizan los estereotipos web: las páginas clientes, las servidoras y los formularios. La Página Cliente Principal le hace una petición a la Página del Servidor (gestionar Nota Informativa), esta es la encargada de construir las Páginas Clientes (crear Nota Informativa, editar Nota Informativa, suprimir Nota Informativa, listar Nota Informativa y Buscar Nota Informativa, imprimir Nota Informativa), 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 nota informativa.

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 en la base de datos accediendo a las clases del modelo y enviando los resultados.

En el diagrama, ver figura 8, se evidenciaron los patrones GRASP:

- ❖ **El patrón experto:** se evidencia en la clase modelo Nota\_Informativa producto a que posee únicamente los parámetros necesarios para su funcionamiento.
- ❖ **El patrón bajo acoplamiento:** se evidencia en la relación entre la clase modelo Nota\_Informativa y la clase controladora cc\_Nota\_Informativa
- ❖ **El patrón alto cohesión:** se aprecia en la clase Nota\_Informativa
- ❖ **El patrón controlador:** se evidencia en las clases cc\_Nota\_Informativa y Conexión producto a que contienen métodos que permiten darles respuesta a las peticiones del cliente y funcionan ajenas a este.

### 2.4.4 Modelo de datos

El modelo de datos se basa en una percepción de un mundo real que consiste en un conjunto de objetos básicos llamados entidades y de relaciones entre estos objetos. Se desarrolló para facilitar el diseño de la base de datos permitiendo especificar un esquema empresarial. Este esquema representa la estructura lógica general de la base de datos. [33]

En el presente trabajo se realizó el modelo de entidad-relación, ver Figura 9, para el diseño de las tablas que son utilizadas para almacenar la información del sistema propuesto para la solución. Estas se encuentran normalizadas, por tanto, cumplen con las normas establecidas para el diseño de la base de datos. Además, la base de datos se encuentra en tercera forma normal, pues sus atributos dependen directamente de la clave primaria, o sea, no se relacionan a través de otros atributos, eliminando de esta manera la dependencia transitiva.

## Capítulo 2: Propuesta e Implementación de la solución

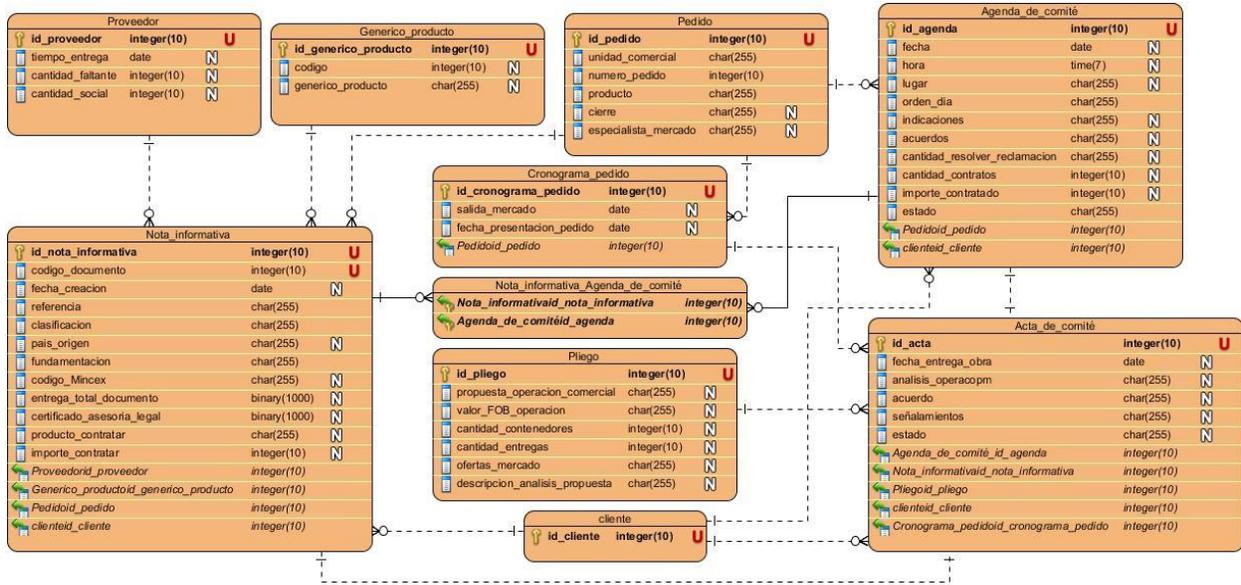


Figura 9. Modelo de datos de la solución

### 2.4.5 Validación del diseño

La validación es un conjunto de procesos de comprobación y análisis que aseguran que el desarrollo del software esté acorde a su especificación y cumpla los requisitos definidos por los clientes. [34]

En el presente trabajo se realizó la validación del diseño del sistema propuesto para la solución, con el empleo de las métricas Tamaño Operacional de Clase (TOC) y Relaciones entre Clases (RC), ya que éstas miden factores como: [35]

- ❖ **Encapsulamiento:** es la proporción entre la suma de los grados de invisibilidad de los métodos en todas las clases y el número total de métodos definidos en el sistema. El grado de invisibilidad de un método es el porcentaje sobre el número total de clases desde las cuales un método no es visible. Es decir, MHF es la proporción entre los métodos definidos como protegidos o privados y el número total de métodos.
- ❖ **Complejidad de implementación:** consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
- ❖ **Cohesión:** número de grupos de métodos locales que no acceden a atributos comunes. Indica la calidad de la abstracción hecha en la clase. Usa el concepto de grado de similitud de métodos. Si no hay atributos comunes, el grado de similitud es cero. Una baja cohesión incrementa la complejidad y por tanto la facilidad de cometer errores durante el proceso de desarrollo.

## Capítulo 2: Propuesta e Implementación de la solución

- ❖ **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.

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.

**Tamaño Operacional de Clase (TOC):** las métricas orientadas a tamaño para una clase orientada a objetos (OO), está dado por el número de métodos asignados a una clase (tanto métodos heredados como métodos internos de dicha clase). Para esta métrica es importante que la reutilización sea inversamente proporcional a la complejidad de implementación y la responsabilidad. A mayor reutilización habrá menor complejidad y responsabilidad. [36]

Tabla 5. Métrica de tamaño operacional TOC

Atributo de calidad	Modo en que lo afecta
<b>Responsabilidad</b>	Aumento del TOC provoca aumento de la responsabilidad asignada a la clase.
<b>Complejidad de implementación</b>	Aumento del TOC provoca aumento de la complejidad de implementación de la clase.
<b>Reutilización</b>	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. Criterio de evaluación para la métrica TOC

Atributos	Categoría	Criterio
<b>Responsabilidad</b>	Baja	$\leq$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
<b>Complejidad e implementación</b>	Baja	$\leq$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
<b>Reutilización</b>	Baja	$> 2 \times$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$\leq$ Promedio

Resultados obtenidos al aplicar la métrica TOC:

### Responsabilidad

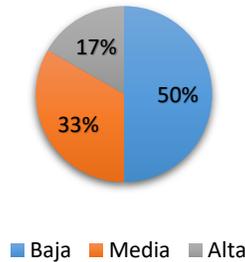


Figura 10. Resultado del atributo Responsabilidad

### Complejidad

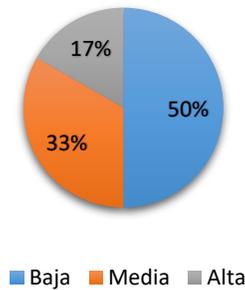


Figura 11. Resultado del atributo Complejidad

### Reutilización

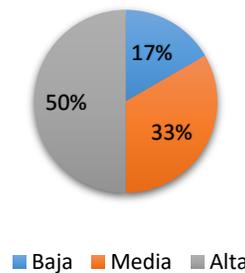


Figura 12. Resultado del atributo Reutilización

Luego de aplicarse la métrica de diseño TOC se obtuvo, de **calidad aceptable**, la validación del diseño de la propuesta de solución. Los atributos de calidad se encuentran en un nivel aceptable en todas las clases, teniendo en cuenta que el 50% de las clases tienen una baja responsabilidad y complejidad, pero a su vez solo el 50% de las clases poseen un nivel alto de reutilización entre ellas.

## Capítulo 2: Propuesta e Implementación de la solución

**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. [36]

Tabla 7. Atributos de calidad evaluados por la métrica RC

Atributo de calidad	Modo en que lo afecta
<b>Acoplamiento</b>	Aumento del RC provoca aumento del acoplamiento de la clase.
<b>Complejidad de mantenimiento</b>	Aumento del RC provoca aumento de la complejidad del mantenimiento de la clase.
<b>Reutilización</b>	Aumento del RC provoca disminución en el grado de reutilización de la clase.
<b>Cantidad de pruebas</b>	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
<b>Acoplamiento</b>	Ninguna	0
	Bajo	1
	Medio	2
	Alto	>2
<b>Complejidad de mantenimiento</b>	Baja	$\leq$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
<b>Reutilización</b>	Baja	$> 2 \times$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$\leq$ Promedio
<b>Cantidad de pruebas</b>	Baja	$< =$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio

Resultados obtenidos al aplicar la métrica RC:

### Acoplamiento

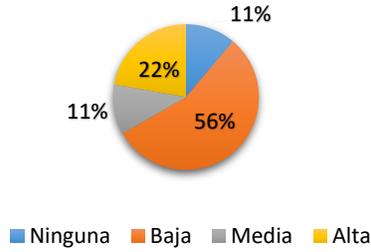


Figura 13. Resultado del Acoplamiento

### Complejidad de Mantenimiento

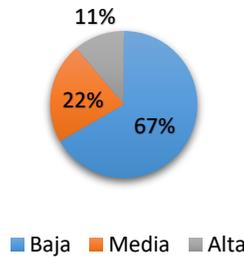


Figura 14. Resultado del Complejidad de Mantenimiento

### Reutilización

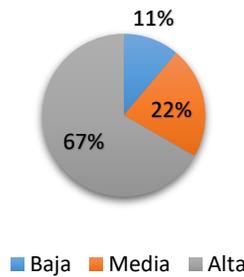


Figura 15. Resultado del atributo Cantidad de pruebas

### Cantidad de Pruebas

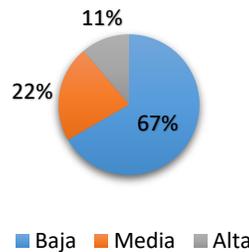


Figura 16. Resultado del atributo Reutilización

## Capítulo 2: Propuesta e Implementación de la solución

Luego de aplicarse la métrica de diseño RC, los resultados obtenidos permiten evaluar el diseño propuesto de “**calidad aceptable**”, teniendo en cuenta que, más del 67% de las clases empleadas en el sistema poseen de 1 a 2 dependencias con otras clases, lo que conlleva a evaluaciones positivas de los atributos de calidad involucrados (acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización).

Posterior a la validación del diseño se inicia, según la metodología, la disciplina de implementación.

### 2.5 Implementación

Entre las disciplinas que conforman el proceso de desarrollo de software se encuentra la de implementación. Esta 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. [37]

#### 2.5.1 Diagrama de Componentes

El diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. [27]

En el presente trabajo se identificaron los componentes, ver figura 17: Nota Informativa, Agenda de Comité y Acta de Comité. Además, se definió la relación existente entre ellos y con otros componentes necesario para su funcionamiento.

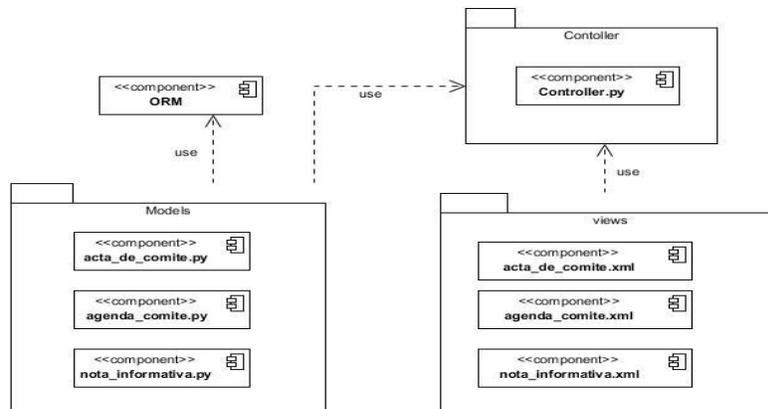


Figura 17. Diagrama de componente

## Capítulo 2: Propuesta e Implementación de la solución

El diagrama de componentes está compuesto por el paquete *Views* que agrupa los componentes que son utilizados en las interfaces y con los cuales el usuario puede interactuar directamente. El paquete *Models* que agrupa los componentes relacionados con las clases de acceso a datos, las cuales son utilizadas por el *Controller* para dar respuesta a la vista. El componente *Controller* relacionado con la lógica del negocio, el cual obtiene las peticiones del usuario y da respuesta a la misma mediante su interacción con el modelo.

### 2.5.2 Estándares de codificación

El estándar de código es un conjunto de convenciones establecidas para la codificación. Estos varían dependiendo del lenguaje de programación elegido y las características del proyecto en cuestión. [26]

A continuación, se presenta algunos de los estándares de codificación utilizados en el presente trabajo:

#### Los Módulos

Cada módulo en Odoe se guía por una estructura de archivos y carpetas, las cuáles tiene una determinada responsabilidad en el funcionamiento del módulo. Esta estructura de carpetas se basa en el patrón arquitectónico Modelo-Vista-Controlador y garantiza la alta cohesión y el bajo acoplamiento, favoreciendo la reutilización. Los archivos principales de un módulo son el “\_init\_.py” donde se importan los archivos que serán utilizados por el módulo y el archivo “\_manifest\_.py” que contiene información útil del módulo como: nombre, resumen del módulo, dependencias, categoría, versión, vistas que utiliza el módulo, etc. [38]

Si el nombre de un módulo es compuesto, se utilizará el carácter ( ) para separar cada palabra del nombre del módulo, de la forma: palabra1\_palabra2.

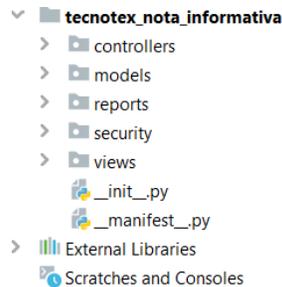


Figura 18. Módulo Nota Informativa

#### Estructura organizativa de los directorios de un módulo.

El nombre de las carpetas se escribe en minúscula. La estructura de carpetas necesarias es la siguiente:

## Capítulo 2: Propuesta e Implementación de la solución

- ❖ report: esta carpeta contiene los archivos para generar reportes. La misma contiene las carpetas models, wizard, templates en esta carpeta se creará la estructura de cada uno de los reportes y el archivo papel\_format.xml el formato en que saldrá el reporte.
- ❖ wizard: contiene los wizard o asistentes. Cada uno de los .xml que se creen pueden ser divididos por subcarpetas que identifiquen el módulo o requisito a desarrollar.
- ❖ views: esta carpeta contiene los archivos de vista en formato .xml. Estos archivos serán divididos por módulos o agrupaciones de requisitos en carpetas diferentes. Cada una de estas carpetas contendrá el archivo menuAction.xml donde se añaden los menú y submenú del módulo, el viewType.xml tendrá las vistas y por último el workflow en caso que sea necesario.
- ❖ data: contiene los datos a introducir en una base de datos una vez instalado el módulo, cada uno de los .xml que se creen pueden ser divididos por subcarpetas que identifiquen el módulo o requisito a desarrollar.
- ❖ models: contiene los modelos donde será desarrollada la lógica de negocio. Cada uno de los .py que se creen pueden ser divididos por subcarpetas que identifiquen el módulo o requisito a desarrollar.
- ❖ static: en esta carpeta se definen todos los ficheros estáticos del módulo. Los css, img, js, less y xml. Esto son archivos globales que se usan en el sistema.
- ❖ security: contiene los ficheros de seguridad. En el fichero .xml de esta carpeta se pueden definir los roles y grupos que tendrá el sistema, así como los permisos, los cuales serán configurados en él .csv, este mismo mecanismo puede ser realizado a nivel de sistema.

### 2.5.3 Interfaz de usuario funcional

A continuación, se muestran interfaces del componente que responden a los requisitos funcionales de crear agenda de comité y acta de comité.

## Capítulo 2: Propuesta e Implementación de la solución

The screenshot shows the 'Comite Contratación' system interface. The top navigation bar includes 'Comite Contratación', 'Agenda', and 'Acta'. The current page is 'Agenda / Nuevo'. There are two buttons: 'Guardar' and 'Descartar'. The main form area contains several sections:

- Fecha:** 22/08/2020
- Hora:** 22/08/2020 19:14:42
- Lugar:** (empty field)
- Orden del día:** (empty field)
- Indicaciones:** (empty field)
- Acuerdos Pendientes:** (empty field)
- Pedido:** NI
- Table:** A table with columns: UC, Pedido, Producto, Propuesta cierre, Estado. Below the table is a link 'Agregar línea' and three empty rows.

Figura 19. Prototipo funcional: crear Agenda de Comité

The screenshot shows the 'Comite Contratación' system interface. The top navigation bar includes 'Comite Contratación', 'Agenda', and 'Acta'. The current page is 'Acta / Nuevo'. There are two buttons: 'Guardar' and 'Descartar'. The main form area contains several sections:

- Tipo:** NI
- Table:** A table with columns: Nombre y Apellidos, Cargo, Tipo. Below the table is a link 'Agregar línea' and three empty rows.
- Datos de la propuesta:** A section with several input fields:
  - Referencia:** (empty field)
  - Especialista mercado:** (empty field)
  - Nombre proveedor:** (empty field)
  - Clasificación:** (empty field)
  - Estado:** Aprobado
  - Pais origen:** (empty field)
  - Fundamentación de la inclusión del proveedor:** (empty field)

Figura 20. Prototipo funcional: crear Acta de Comité

### 2.6 Conclusiones parciales

En este capítulo se definieron 15 requisitos funcionales y 17 no funcionales. Se empleó la arquitectura modelo-vista-controlador a partir de la cual, se diseñaron los diagramas de clases del diseño y el modelo de datos. Se aplicaron las métricas TOC y RC para la validación del diseño, arrojando resultados de calidad aceptable para cada uno de los parámetros que se midieron. Se diseñó el diagrama de componente para establecer las relaciones existentes entre estos. El empleo de los estándares de codificación permitió un mejor entendimiento de las clases y los métodos desarrollados.

## Capítulo 3: Validación de la propuesta de solución

### Capítulo 3: Validación de la propuesta de solución

#### 3.1 Introducción

En el presente capítulo se realizan las disciplinas de pruebas internas y pruebas de aceptación teniendo en cuenta los requisitos definidos por el cliente. Las pruebas internas se realizan utilizando las técnicas de caja blanca y caja negra y las pruebas de aceptación se realizan con el cliente. 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 Contratación.

#### 3.2 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. [15]

##### 3.2.1 Prueba de Caja Blanca

Las pruebas de caja blanca son una filosofía de diseño de caso de prueba, que usan la estructura de control del diseño a nivel de componente para derivar los casos de pruebas. Mediante estos métodos, se pueden obtener casos de prueba que garanticen que se ejercite, por lo menos una vez, todos los caminos independientes de cada módulo; se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas; se ejecuten todos los bucles en sus límites y con sus limitaciones operacionales; y se ejerciten las estructuras internas de datos para asegurar su validez. [27]

La técnica a aplicar es la de camino básico, la cual permite al diseñador de caso de prueba obtener una medida de complejidad lógica de un diseño de procedimiento, o usar esta 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. [27]

##### 3.2.1.1 Técnica del Camino Básico

Para obtener el conjunto de caminos independientes se construye el grafo de flujo asociado a una función y se calcula su complejidad. Para este caso, se tomó como ejemplo el método `change_value()`, encargado de verificar que el campo pedido solamente admita números enteros.

## Capítulo 3: Validación de la propuesta de solución

Primeramente, se enumeran las sentencias de código.

```
@api.onchange('pedido')
def change_value(self):
    1-if (self.pedido):
        2-for aux in str(self.pedido):
            3-if (aux not in "[0-9]{10}$"):
                4-raise UserError('pedido: error %s' % self.pedido)
```

Figura 21. Gráfico de Código

Posteriormente se diseña el grafo:

Los elementos del grafo son:

- ❖ **Nodos:** son los círculos en el grafo, representan una o más secuencias del procedimiento.
- ❖ **Aristas:** son las saetas y unen los Nodos representando el flujo de control del procedimiento.
- ❖ **Regiones:** son las áreas delimitadas por las aristas y nodos.

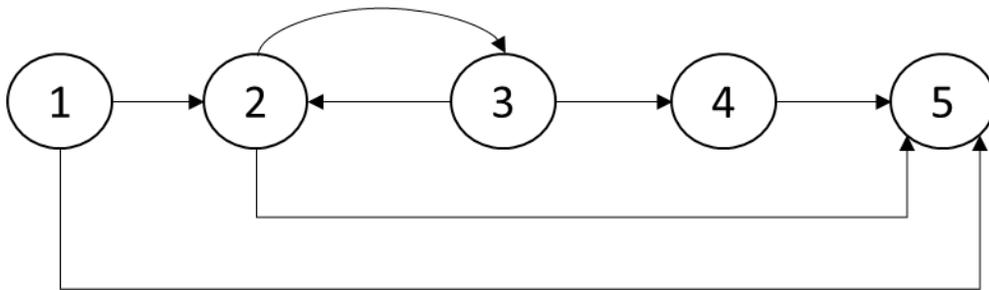


Figura 22. Grafo resultante del método

Por último, se calcula la complejidad ciclomática:

La complejidad ciclomática es una medición de software con que se define la cantidad de caminos independientes del conjunto básico de un programa, brindando el número máximo de pruebas que se debe realizar para hacer que se ejecute cada sentencia al menos una vez. [27]

La misma se calcula de la siguiente manera:

- ❖ **1.  $V(G) = (A - N) + 2$**

Siendo "A" la cantidad total de aristas y "N" la cantidad de nodos.

$$V(G) = (7 - 5) + 2$$

$$V(G) = 4.$$

- ❖ **2.  $V(G) = P + 1$**

Siendo "P" la cantidad de nodos predicados (son los nodos de los cuales parten dos o más aristas).

### Capítulo 3: Validación de la propuesta de solución

$$V(G) = 3 + 1$$

$$V(G) = 4.$$

#### ❖ 3. $V(G) = R$

Siendo "R" la cantidad total de regiones, se incluye el área exterior del grafo, contando como una región más.

$$V(G) = 4.$$

Para este caso se obtienen 3 posibles caminos independientes y cantidad de pruebas que se deben realizar para comprobar que las sentencias se ejecuten al menos una vez.

- ❖ Camino 1: 1, 2, 5.
- ❖ Camino 2: 1, 2, 3, 4, 5.
- ❖ Camino 3: 1, 2, 3, 2, 5.
- ❖ Camino 4: 1, 5.

Luego se elaboran los casos de prueba, quedando de la siguiente manera.

Tabla 9. Camino básico 1: 1-2-5

<b>Descripción</b>	Sucede cuando el campo válido: valores numéricos.
<b>Condición de ejecución</b>	Ocurre cuando se pretende comprobar la validez del campo.
<b>Entrada</b>	Que los datos sean correctos
<b>Resultado</b>	Se registra el pedido.

Tabla 10. Camino básico 2: 1-2-3-4-5

<b>Descripción</b>	Sucede cuando el campo es inválido: valores no numéricos.
<b>Condición de ejecución</b>	Ocurre cuando se pretende comprobar la validez del campo.
<b>Entrada</b>	Que los datos sean incorrectos
<b>Resultado</b>	No registra el pedido.

Tabla 11. Camino básico 3: 1-2-3-2-5

<b>Descripción</b>	Sucede cuando el campo válido: valores numéricos.
<b>Condición de ejecución</b>	Ocurre cuando se pretende comprobar la validez del campo.
<b>Entrada</b>	Que los datos sean correctos
<b>Resultado</b>	Se registra el pedido.

Tabla 12. Camino básico 4: 1-5

<b>Descripción</b>	Sucede cuando el campo es vacío.
<b>Condición de ejecución</b>	Ocurre cuando se pretende comprobar la validez del campo.
<b>Entrada</b>	Que no existan datos.
<b>Resultado</b>	No registra el pedido.

## **Capítulo 3: Validación de la propuesta de solución**

Luego se ejecutan los casos de prueba para comparar los resultados obtenidos con los esperados, una vez comprobado que estos coinciden, se puede asegurar que todas las sentencias del método se han ejecutado al menos una vez.

### **3.2.2 Prueba de Caja Negra**

Las pruebas de caja negra o pruebas funcionales se realizan sobre la interfaz del software, comprobando las entradas y salidas de datos. Estas pruebas se realizan para comprobar que cada función es operativa, utilizando el artefacto diseño de caso de prueba, que tienen como objetivo introducir juegos de datos que ayuden a la ejecución de los casos y facilite que el sistema se ejecute en todas sus variantes. [27] La técnica utilizada para desarrollar las pruebas de Caja Negra fue partición equivalente.

#### **3.2.2.1 Técnica de Partición de Equivalencia**

Un método 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. [27]

### Capítulo 3: Pruebas y validación de la solución

#### Casos de Pruebas

Para los casos de pruebas se definieron un conjunto de estados válidos y no válidos para las condiciones de entrada del sistema.

Tabla 13. Casos de Prueba del RF: Crear agenda de comité

Escenario	Descripción	Fecha	Hora	Lugar *	Orden del día	Indicaciones	Acuerdos Pendientes	Respuesta del sistema	Flujo central
EC 1.1: Crear Agenda de Comité introduciendo datos válidos	El sistema debe permitir crear una Agenda de Comité	V 25/06/2020	V 25/06/2020 15:03	V Prueba	V Prueba	V Prueba	V Prueba	El sistema registra una nueva Agenda de Comité	1. Se selecciona la opción de Agenda de Comité. 2. Se presiona el botón "Crear". 3. Se introducen los datos necesarios para crear una Agenda de Comité. 4. Se selecciona el botón "Guardar"
EC 1.2: Crear Agenda de Comité introduciendo datos inválidos	El sistema debe no permitir crear una Agenda de Comité	I 25/aa/2020	V 25/06/2020 15:03	V Prueba	V Prueba	V Prueba	V Prueba	El sistema corrige la fecha automáticamente	1. Se selecciona la opción de Agenda de Comité. 2. Se presiona el botón "Crear". 3. Se introducen los datos necesarios para crear una Agenda de Comité. 4. Se selecciona el botón "Guardar".
		V 25/06/2020	I d5/06/2020 1a:03	V Prueba	V Prueba	V Prueba	V Prueba	El sistema corrige la fecha y hora automáticamente	
		V 25/06/2020	V 25/06/2020 15:03	I Pru3ba	V Prueba	V Prueba	V Prueba	El sistema registra una nueva Agenda de Comité	
		V 25/06/2020	V 25/06/2020 15:03	V Prueba	I Prueb@	V Prueba	V Prueba		
		V 25/06/2020	V 25/06/2020 15:03	V Prueba	V Prueba	I Pru""eba	V Prueba		
		V 25/06/2020	V 25/06/2020 15:03	V Prueba	V Prueba	V Prueba	I 1Prueba		
EC 1.3: Crear	Se dejan campos vacíos al	I vacío	V 25/06/2020 15:03	V Prueba	V Prueba	V Prueba	V Prueba	El sistema registra una nueva Agenda de Comité	1. Se selecciona la opción de Agenda de Comité. 2. Se presiona el botón "Crear". 3

### Capítulo 3: Validación de la propuesta de solución

Agenda de Comité dejando campos vacíos	de crear una Agenda de Comité	V 25/06/2020	I vacío	V Prueba	V Prueba	V Prueba	V Prueba	El sistema registra una nueva Agenda de Comité	Se introducen los datos necesarios para crear una Agenda de Comité. 4.Se selecciona el botón "Guardar"
		V 25/06/2020	V 25/06/2020 15:03	I vacío	V Prueba	V Prueba	V Prueba	El sistema muestra el campo obligatorio por llenar	
		V 25/06/2020	V 25/06/2020 15:03	V Prueba	I vacío	V Prueba	V Prueba	El sistema registra una nueva Agenda de Comité	
		V 25/06/2020	V 25/06/2020 15:03	V Prueba	V Prueba	I vacío	V Prueba		
		V 25/06/2020	V 25/06/2020 15:03	V Prueba	V Prueba	V Prueba	I vacío		
EC 1.4: Cancelar	Se cancela la operación de crear Agenda de Comité	V (N/A)	V (N/A)	V (N/A)	V (N/A)	V (N/A)	V (N/A)	El sistema no realiza ninguna acción y cierra la pestaña	1.Se selecciona la opción de Agenda de Comité. 2.Se presiona el botón "Crear". 3. Se introducen los datos necesarios para crear una Agenda de Comité. 4.Se selecciona el botón "Descartar"

### Capítulo 3: Pruebas y validación de la solución

Como parte de la ejecución de la prueba de caja negra, se identificó en la primera iteración 4 No Conformidades (NC) como faltas de ortografías y funciones incorrectas. Para la segunda iteración se realizaron correcciones a las 4 no conformidades pendientes de la iteración anterior y se comprobó que existe aún problemas con la admisión de caracteres raros en los campos no relacionados con las fechas. Por último, se corrigió el problema pendiente y se prosiguió al desarrollo de una tercera iteración arrojando resultados satisfactorios.

#### 3.3 Aplicación de la Técnica de ladov

Para validar esta investigación se utilizó la Técnica de ladov. Esta constituye una vía indirecta para el estudio de la satisfacción, puesto que los criterios se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario aplicado y en cuya relación el sujeto se desconoce. Estas tres preguntas se relacionan a través de lo que se denomina como el “**Cuadro Lógico de ladov**”, ver tabla 14. [39]

Para la realización de esta técnica se seleccionaron especialistas del centro CEIGE pertenecientes al proyecto, a los cuales se les aplicó el cuestionario, ver anexo 2, para determinar el grado de satisfacción del componente para la gestión de la información del Comité de Contratación en el sistema de importación de la empresa TECNOTEX desarrollado. [39]

El número resultante de la interrelación de las tres preguntas cerradas 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 14. Niveles de satisfacción

Valores	Clasificación	Escala
+1	Clara satisfacción	A
0,5	Más satisfecho que insatisfecho	B
0	No definido y contradictorio	C
-0,5	Más insatisfecho que satisfecho	D
-1	Clara insatisfacción	E

### Capítulo 3: Validación de la propuesta de solución

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. Resultado del ISG

En esta fórmula A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 o 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:

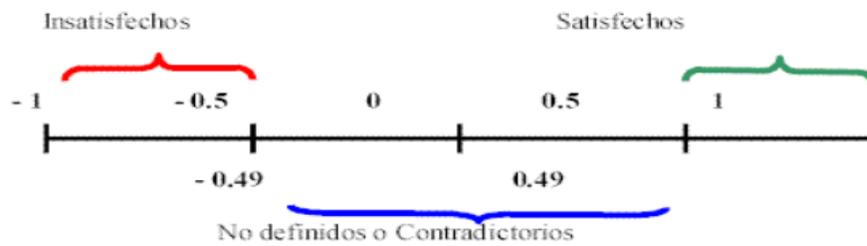


Figura 24. Intervalo de satisfacción

Cuadro lógico de ladov de la investigación:

Tabla 15. Cuadro lógico de ladov

5. ¿Le satisface la concepción de este componente para disminuir el tiempo en la obtención de la información utilizada para la gestión de contratos en el Sistema de Importación y Suministro de la empresa TECNOTEX?	2. ¿Considera usted que es aconsejable no contar con un componente que permita gestionar las notas informativas, las actas y agenda de comité en el Sistema de Importación y Suministro de la empresa TECNOTEX?								
	No			No sé			Sí		
	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
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

### Capítulo 3: Validación de la propuesta de solución

Resultado obtenido al aplicar la encuesta

Tabla 16. Resultado de la encuesta

Total de cuestionados	9	Escala
Clara satisfacción	6	A
Más satisfecho que insatisfecho	1	B
No definida	1	C
Más insatisfecho que satisfecho	0	D
Clara insatisfacción	0	E
Contradictoria	1	C

Al aplicar la fórmula para el cálculo del ISG se obtuvo el siguiente resultado:

$$ISG = \frac{6(1) + 1(0.5) + 2(0) + 0(0) + 0(0)}{9} = 0,72$$

Figura 25. Resultado del ISG

Luego de aplicar la técnica de ladov a través de una encuesta sobre el Componente para la gestión de la información del Comité de Contratación en el sistema de importación de la empresa TECNOTEX creado, donde se tuvo la participación de 9 especialistas del centro CEIGE, directamente relacionados con el proyecto, se obtuvo de los encuestados (especialistas) una valoración positiva con un alto **Índice de Satisfacción Grupal** como se observa en la Figura 25. El valor se encuentra comprendido entre 0,5 y 1, por lo que, se corrobora según la técnica de ladov, que el componente desarrollado se clasifica como satisfactoria.

#### 3.4 Beneficios del componente desarrollado

El Componente para la gestión de la información del Comité de Contratación en el sistema de importación de la empresa TECNOTEX, corresponde al módulo Contratación, que interviene en la gestión de los subprocesos Presentación y Revisión de Pedidos, Contratación y Suministro de mercancía. En él se gestiona principalmente los datos asociados con las Acta y Agendas del Comité, así como las notas informativas relacionadas con ambos. Entre los principales beneficios que aporta el desarrollo del componente están:

- Es un sistema web, lo que permitirá el acceso desde cualquier punto dentro de la empresa.
- Permite la elaboración automática de las actas utilizadas en las reuniones del Comité a partir de las informaciones extraídas por cada nota informativa y pedido analizado.
- Disminuye el tiempo en la obtención de los documentos necesarios para realizar cada reunión de Comité.

### **Capítulo 3: Validación de la propuesta de solución**

- Permite analizar la información archivadas en las Actas y Agendas del Comité de reuniones anteriores.
- Muestra en las Agendas del Comité cada pedido y nota informativa en que su estado aún está pendiente a cierre.

#### **3.5 Conclusiones parciales**

En el presente capítulo se ejecutaron las pruebas internas, tales como, Caja Blanca y Caja Negra para el control de la calidad del sistema a realizar. El resultado de estas pruebas funcionales y no funcionales, permitió la corrección en tiempo y forma de las no conformidades detectadas. Se efectuó, además, la validación de la investigación mediante la técnica de ladov con la participación de especialistas del CEIGE. El grado de satisfacción evaluado del componente fue de satisfactorio, lo que permite concluir que se cumplieron los objetivos planteados en la investigación.

### CONCLUSIONES GENERALES

Con el propósito de darle cumplimiento al objetivo general y a la problemática planteada en la presente investigación, se han llevado a cabo satisfactoriamente los objetivos específicos trazados:

- ❖ El estudio y análisis de los sistemas de gestión empresarial asociadas a la gestión de contratos evidenció la no existencia de una solución que permita darle respuesta al problema existente. Por este motivo, se desarrolló un componente para la gestión de información del Comité de Contratación, a través de la utilización de Odoó como marco de trabajo y la metodología AUP-UCI, garantizando una mayor organización en el desarrollo del trabajo.
- ❖ La verificación del diseño realizado, mediante la aplicación de las métricas TOC y RC, evidenció una baja responsabilidad, cantidad de pruebas y complejidad de implementación, la cual favoreció la reutilización corroborándose así una correcta asignación de responsabilidades.
- ❖ En la disciplina Implementación se utilizaron estándares de codificación que permitieron 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.
- ❖ Las pruebas unitarias y de funcionalidad permitieron validar el correcto funcionamiento de las interfaces desarrolladas y el cumplimiento de los objetivos de la investigación.
- ❖ La validación de la investigación mediante el uso de la técnica IADOV arrojó que la propuesta de solución contribuye disminuir el tiempo en la obtención de la información utilizada en la gestión de contratos en la empresa TECNOTEX.

## **RECOMENDACIONES**

Para dar continuidad a la siguiente investigación se recomienda:

- Integrar el componente desarrollado al Sistema de Importación y Suministro.
- Para una nueva versión del componente, tener en cuenta las referencias de los campos relacionales definidos en las descripciones de los requisitos.
- Continuar con el estudio, desarrollo y mantenimiento del componente implementado.

## REFERENCIAS BIBLIOGRÁFICAS

1. Blanca Hernández Ortega, J.J.M., José Martín de Hoyos, Influencia de las TIC en la gestión de la información empresarial Dialnet, 2007(2232713): p. 15.
2. Pita, G.E.C.J.D.d.I.C., Las TICs en las empresas: evolución de la tecnología y cambio estructural en las organizaciones. 2018. 4(1): p. 499-510.
3. Comité de Contratación. [cited 2019; Available from: <http://www.cubalegalinfo.com/mercantil/importacion-exportacion/comite-de-contratacion1>.
4. Española, R.A., Real academia española. 1983: Espasa Calpe.
5. FRANCO, V.A.S., ENFOQUE ACTUAL DE LA TEORIA GENERAL DEL CONTRATO.
6. Delgado, J. and F.J.E.i. Marín, Evolución en los sistemas de gestión empresarial. Del MRP al ERP. 2000. 331(1): p. 51-58.
7. Cabrera, H.R., et al., La integración de Sistemas de Gestión Empresariales, conceptos, enfoques y tendencias. 2015. 46(3): p. 3-8.
8. Burgos, R.N., Software ERP: análisis y consultoría de software empresarial. 2016: IT Campus Academy.
9. Ganesh, A., et al. OpenERP/Odoo-an open source concept to ERP Solution. in 2016 IEEE 6th International Conference on Advanced Computing (IACC). 2016. IEEE.
10. OrbitTrade: Gestión de Importación y Exportación. [cited 2020 29/01/2020]; Available from: <https://www.orbit.es/soluciones-y-servicios/soluciones-de-negocio/crm-bpm/orbittrade/>.
11. Gestión de contratos digitalizada con el software Doxis4 2020 [cited 2020 30/01/2020]; Available from: <https://www.sergroup.com/es/productos-soluciones/expedientes-electronicos/gestion-de-contratos-software.html>.
12. Softland. [cited 2020 29/01/2020]; Available from: <https://www.softland.cl/>.
13. ABpro. [cited 2020 29/01/2020]; Available from: <https://www.abprosystems.com/Soluciones/logistica.html>.
14. Navarro, M.E., et al. Selección de metodologías ágiles e integración de arquitecturas de software en el desarrollo de sistemas de información. in XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires). 2017.
15. Sánchez, T.R.J.H.s., Metodología de desarrollo para la Actividad productiva de la UCI. 2014.
16. Peña, D.M., Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso. 2016, Universidad de las Ciencias Informáticas.
17. Debrauwer, L. and F. Van der Heyde, UML 2.5: iniciación, ejemplos y ejercicios corregidos. 2016: Ediciones ENI.
18. Schmidt, A.R., et al., The XML benchmark project. 2001.
19. Van Rossum, G. Python Programming Language. in USENIX annual technical conference. 2007.
20. Smith, G., PostgreSQL 9.0: High Performance. 2010: Packt Publishing Ltd.
21. Brains, J., PyCharm.2018.
22. Almugadam, S.H., et al. Developing tool for Odoo platform. in 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE). 2017. IEEE.

23. Wells, D.L., J.A. Blakeley, and C.W.J.C. Thompson, Architecture of an open object-oriented database management system.1992. 25(10): p. 74-82.
24. pgAdmin. [cited 2020 28/01/2020]; Available from: <https://www.pgadmin.org/>.
25. SOMMERVILLE, I., Ingeniería del Software. septima edición ed. 2017.
26. Dr. Edgar Danilo Domínguez, M.C.V.B.D., M.C. Arturo del Ángel Ramírez, M.C. José Antonio Moreno Barrios., REGLAS DE CALIDAD PARA LA CODIFICACIÓN ESTANDARIZADA EN LENGUAJE C: UNA PROPUESTA PARA LA ENSEÑANZA A NIVEL SUPERIOR 2016.
27. Pressman, R.S., Software engineering: a practitioner's approach. 2005: Palgrave macmillan.
28. González, Y.D. and Y.F.J.R.T. Romero, Patrón Modelo-Vista-Controlador. 2012. 11(1): p. 47-57.
29. Díaz, M.P., S. Montero, and I.J.P.P.H. Aedo, Ingeniería de la web y patrones de diseño.2005.
30. Gamma, E., Design patterns: elements of reusable object-oriented software. 1995: Pearson Education India.
31. Larman, C., UML y Patrones. 2003: Pearson Educación ^ eMadrid Madrid.
32. Álvarez, R.L., et al., Facultad 3. 2018, Universidad de las Ciencias Informáticas.
33. López Gaona, A.J.I.P.e.C.e.I.d.I.C., Fac. Ciencias, UNAM Disponible en Internet:, El modelo Entidad-Relación. 2011.
34. Puello, O.J.I.e.I.e.I., Modelo de verificación y validación basado en CMMI. 2013. 1(1).
35. Vázquez Escudero, P.J., M.N. Moreno García, and F.J. García-Peñalvo, Métricas orientadas a objetos. 2001.
36. Patricia Hernandez Rodriguez, J.p.B.R., Sistema para la dirección de compra de la UC. 2019.
37. T.R, S., Metodología de desarrollo para la ctividad productiva de la UCI. 2015.
38. Pressman, R., Ingeniería del software. Un enfoque práctico. 2017. 7ta edición: p. 589.
39. Fernández de Castro Fabre, A. and A.J.R.C.T.A. López Padrón, Validación mediante criterio de usuarios del sistema de indicadores para prever, diseñar y medir el impacto en los proyectos de investigación del sector agropecuario. 2014. 23(3): p. 77-82.

## ANEXOS

### Anexo 1. Entrevista para la comprensión del negocio a los analistas del proyecto.

- ¿Cómo se realiza el proceso contratación en la empresa?
- ¿Cuántos procesos intervienen en dicho proceso?
- ¿Cómo realmente una agenda de comité y qué es?
- ¿Cómo realmente un acta de comité y qué es?
- ¿Cómo realmente una nota informativa y qué es?
- ¿Qué áreas intervienen en este proceso?
- ¿Qué es y cómo intervienen las características técnicas y las políticas de estandarización en la generación de las actas de comité?

### Anexo 2. Encuesta de la técnica de ladov

Encuesta: Evaluación de la Técnica de ladov		
1	¿Considera usted que el componente propuesto disminuye el tiempo en la obtención de la información utilizada en la gestión de contratos en la empresa TECNOTEX?	Sí ___ No___
2	¿Considera usted que es aconsejable no contar con un componente que permita gestionar las notas informativas, las Actas y Agenda de Comité en el Sistema de Importación y Suministro de la empresa TECNOTEX?	Sí ___ No sé ___ No___
3	¿Utilizaría el Componente genérico de productos desarrollado para disminuir el tiempo en la obtención de la información utilizada en la gestión de contratos en la empresa TECNOTEX?	Sí ___ No sé___ No___
4	¿Considera usted que la solución propuesta permite disminuir el tiempo en la obtención de la información utilizada en la gestión de contratos en la empresa TECNOTEX?	Sí ___ No sé___ No___
5	¿Le satisface la concepción de este componente para disminuir el tiempo en la obtención de la información utilizada para la gestión de contratos en el Sistema de Importación y Suministro de la empresa TECNOTEX?	___ Me gusta mucho ___ No me gusta tanto ___ Me da lo mismo ___ Me disgusta más de lo que me gusta ___ No me gusta nada ___ No sé qué decir

6. ¿Qué elementos considera positivo del para la gestión de la información del Comité de Contratación en el sistema de importación de la empresa TECNOTEX desarrollado?

7. ¿Qué elementos considera negativo del Componente para la gestión de la información del Comité de Contratación en el sistema de importación de la empresa TECNOTEX desarrollado?

8. ¿Qué influencia tendría el componente para la gestión de la información del Comité de Contratación en el sistema de importación de la empresa TECNOTEX?

9. ¿Qué elementos le añadiría al componente para la gestión de la información del Comité de Contratación en el sistema de importación de la empresa TECNOTEX desarrollado?