

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3



**Componente para la gestión del plan de importación
del Sistema de importación de la empresa Tecnotex**

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

Autor: Patricia Rivalta Machado

Tutores: Ing. Liannet Baez Fernández

Ing. Rachel Pérez Cruz

La Habana, 2020

“Año 62 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaro ser única autora del trabajo **Componente para la gestión del plan de importación del sistema de importación de la empresa Tecnotex** y autorizo a la Universidad de las Ciencias Informáticas hacer el uso que estimen pertinente con el mismo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2020.

Patricia Rivalta Machado

Ing. Liannet Baez Fernández

Ing. Rachel Pérez Cruz

Las importaciones son uno de los principales conceptos económicos que sustentan las operaciones básicas comerciales de un país, razón por la cual es importante contar con la planificación del presupuesto que se destina a dicha actividad. Teconotex, empresa cubana que se dedica a la importación y exportación de artículos y servicios presenta entre sus procesos el plan de importación, a través del cual se realiza la planificación del presupuesto dedicado a importaciones. Actualmente las actividades que se llevan a cabo para realizar dicho proceso presentan insuficiencias que inciden en su calidad. Con el fin de aumentar la calidad del plan de importación, el presente trabajo tiene como objetivo desarrollar un componente para la gestión del plan de importación. Se realizó un estudio de los sistemas existentes concluyendo que era necesario realizar una solución propia. Se analizó el negocio para comprender los procesos que se realizaban en la empresa y sus estados actuales. Para el desarrollo de la solución se utilizó: la metodología AUP-UCI, Odoó v13.0 como marco de trabajo, Python 3.6 como lenguaje de desarrollo, para las vistas, Lenguaje Extendido de Modelado (XML), PostgreSQL 12.01 como gestor de base de datos y Pycharm como entorno de desarrollo integrado (IDE). El sistema desarrollado fue validado a través de pruebas que determinaron el cumplimiento de los requisitos definidos. La solución contribuye a una mejora de la calidad del plan de importación posibilitando el acceso desde todas las áreas a la información del plan de importación y el cálculo automático de los porcentajes de ejecución del mismo.

ABSTRACT

Imports are one of the main economic concepts that sustain the basic commercial operations of a country, which is why it is important to have the budget planning for this activity. Tecnotex, a Cuban company dedicated to the import and export of articles and services, presents among its processes the import plan, through which the planning of the budget dedicated to imports is carried out, currently the activities that are carried out to carry out said process present insufficiency that affect quality. In order to increase the quality of the import plan, the present work purposes to develop a component for the management of the import plan. A study of the existing systems was carried out, concluding that it was necessary to carry out an own solution. The business was analyzed to understand the processes that were carried out in the company and its current states. For the development of the solution, the following was used: the AUP-UCI methodology, Odoo v13.0 as the framework, Python 3.6 as the development language, XML for the views, PostgreSQL 12.01 as the database manager and Pycharm as IDE. The developed system was validated through tests that determined compliance with the defined requirements. The solution contributes to an improvement in the quality of the import plan, allowing access from all areas to the IP information and the automatic calculation of the percentage of execution.

Índice

Introducción	1
1 Capítulo 1: Fundamentación Teórica de la Investigación	5
1.1 Conceptos asociados al dominio del problema	5
1.2 Análisis de principales sistemas de gestión para empresas importadoras.	5
1.3 Metodología de desarrollo de software	9
1.4 Notaciones y Lenguajes.....	10
1.5 Herramientas y tecnologías para el desarrollo del sistema	12
1.6 Conclusiones parciales	13
2 Capítulo 2: Propuesta de solución.	14
2.1 Modelado de negocio.....	14
2.2 Requisitos del sistema	16
2.3 Diseño de la propuesta de solución	23
2.4 Validación del Análisis y Diseño.....	27
2.5 Conclusiones parciales	32
3 Capítulo 3: Implementación y validación de la solución propuesta	33
3.1 Implementación	33
3.2 Interfaz de usuario funcional	35
3.3 Pruebas de Software	36
3.4 Aplicación de la técnica de ladov	43
3.5 Beneficios del sistema desarrollado.....	46
3.6 Conclusiones parciales	47
CONCLUSIONES GENERALES	48
RECOMENDACIONES	49
REFERENCIAS BIBLIOGRÁFICAS	50
ANEXOS	60

Índice de Tablas

Tabla 1: Comparación entre los sistemas analizados.....	7
Tabla 2: Descripción del requisito Crear Plan de importación	18
Tabla 3: Requisitos no funcionales	21
Tabla 4: Atributos de calidad evaluados por la métrica TOC.....	27
Tabla 5: Criterios de evaluación para la métrica TOC	28
Tabla 6: Atributos de calidad evaluados por la métrica RC	29
Tabla 7: Criterios de evaluación para la métrica RC	29
Tabla 8: Caso de prueba para el camino 1.....	39
Tabla 9: Caso de prueba para el camino 2.....	39
Tabla 10: Caso de prueba para el camino 3.....	39
Tabla 11: Cuadro lógico de ladov	43
Tabla 12: Niveles de satisfacción de la técnica de ladov.....	44

Índice de Figuras

Figura 1: Descripción del proceso de negocio Presentar Solicitud de PI.....	15
Figura 2: Modelo Conceptual	16
Figura 3: Arquitectura MVC.....	24
Figura 4: Diagrama de clases del diseño con estereotipos web para la DRP Gestionar_Pi:	25
Figura 5: Código de la clase Aprobación donde se evidencia el uso del patrón Observador	27
Figura 6: Resultado de la evaluación métrica para el atributo responsabilidad	28
Figura 7: Resultado de la evaluación métrica para el atributo complejidad	28
Figura 8: Resultado de la evaluación métrica para el atributo reutilización.....	29
Figura 9: Cantidad de dependencias entre las clases	30
Figura 10: Resultado de la evaluación de la métrica para el atributo acoplamiento.....	30
Figura 11: Resultado de la evaluación de la métrica para el atributo complejidad de mantenimiento... ..	31
Figura 12: Resultado de la evaluación de la métrica para el atributo reutilización	31
Figura 13: Resultado de la evaluación de la métrica para el atributo cantidad de pruebas.....	31
Figura 14: Estándar de codificación Identación y Nombre de clases.....	34
Figura 15: Estándar de codificación Nombre de métodos	34
Figura 16: Diagrama de componentes	35
Figura 17: Interfaz de usuario funcional del requisito Listar Plan de Importación	36
Figura 18: Interfaz de usuario funcional del requisito Crear Aprobación.....	36
Figura 19: Código de la funcionalidad def_onchange_enmienda	38
Figura 20: Grafo de flujo	38
Figura 21: Diseño del caso de prueba. Crear Aprobación	41
Figura 22: Diseño del caso de prueba. Crear Aprobación	42
Figura 23: Rango de valores para la satisfacción grupal	45

Introducción

El comercio exterior en Cuba ejerce un fuerte y determinante impacto en la economía del país, su política está encaminada a elevar la eficiencia en la gestión de las empresas vinculadas al comercio exterior con el objetivo de incrementar y consolidar los ingresos por concepto de exportaciones, la sustitución de importaciones y el desarrollo local. Para ello tiene en cuenta la diversidad de los destinos de los rubros exportables, con preferencia en los de mayor valor agregado por su contenido o soluciones tecnológicas, priorizando así a los principales socios del país y garantizando que todos los bienes y servicios destinados a los mercados internacionales respondan a los más altos estándares de calidad. Otro de sus objetivos está dirigido a aumentar la eficiencia en la gestión importadora del país, haciendo énfasis en la disponibilidad oportuna de las importaciones, su racionalidad y el uso eficaz del poder de compra(1).

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. Un plan no es más que un conjunto de actividades sincronizadas armónicamente para alcanzar determinada meta u objetivo. Para cada una de las actividades deben identificarse los distintos recursos que se requieren en la ejecución de las misma. Estos recursos pueden ser humanos, materiales, tecnológicos y financieros (2). Las importaciones por su parte son la introducción legítima de bienes y servicios del extranjero adquiridos por un país para distribuirlos en el interior de este. (3). Entre las necesidades que estas satisfacen pueden destacarse la adquisición de tecnología moderna, la recuperación y empleo máximo de las capacidades industriales y técnicas, la restauración de la infraestructura de apoyo a los procesos productivos y la obtención de financiamientos, entre otros. Se puede definir entonces un plan de importación, como el documento en el cual se indica el proceso o procedimiento a seguir para la importación de un producto, teniendo en cuenta un conjunto de recursos. (4)

Cuba actualmente mantiene relaciones comerciales con más de 160 países, con los que se realizan operaciones de exportación e importación en las cuales resulta necesario tener presente el marco regulatorio vigente que facilite el cumplimiento de lo establecido en las citadas operaciones. Es por ello que el Ministerio del Comercio Exterior emitió la resolución No.50 en el año 2014, donde regula las actividades de importación y exportación de mercancías, para que las empresas afines a estas actividades en nuestro país se rijan por ella(5). Entre las empresas que se dedican a la importación de mercancías en Cuba 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.

INTRODUCCIÓN

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. Entre las diferentes áreas de apoyo a las actividades fundamentales que realiza la empresa se encuentra el área de Economía. En ella son recepcionados a inicio de año (los tres primeros meses) los planes de importación de los clientes de la empresa. Posteriormente en el área se realiza el plan de importación (PI) de Tecnotex para el año en cuestión, donde se tiene en cuenta los planes de importación presentados por sus clientes además de sus propias importaciones. En la elaboración de los planes de importación se deben verificar las aprobaciones (presupuesto aprobado por concepto de importación) de las entidades clientes, además de considerar posibles modificaciones o extraplanes (aumento del presupuesto) posterior a la confección del PI para el año vigente.

A pesar de que los diferentes procesos que se ejecutan en la empresa se encuentran informatizados, no se logra obtener una información íntegra de las importaciones realizadas debido a que cada proceso cuenta con un sistema informático independiente. Por otro lado, la elaboración del PI de manera manual, incide en la calidad de su gestión, ocasionando un conjunto de insuficiencias que repercuten de manera negativa en los resultados final del proceso de importación en Tecnotex. Entre las principales dificultades identificadas se encuentran:

- Demora en la elaboración del PI de Tecnotex: los clientes en ocasiones no presentan sus PI en el período establecido provocando demoras en la gestión del PI de la empresa.
- Inconsistencia en la información manejada: debido a que la empresa no cuenta con un único sistema para gestionar el proceso de importación de mercancía y la información es duplicada en cada sistema que sea necesario, las modificaciones realizadas a los PI de los clientes, corren el riesgo de no actualizarse en todos los sistemas que se estén empleando.
- Presencia de errores financieros graves para los clientes: al no contar con un sistema que integre todos los procesos, la relación que existe entre el presupuesto planificado con las aprobaciones que son afectadas por cada pedido que presentan los clientes, se realizan de forma manual, lo que puede dar al traste con errores humanos que se traduzcan en errores financieros para el negocio.
- Pérdida de información: la imposibilidad de no poder gestionar la trazabilidad de los PI con respecto a sus relaciones y posibles afectaciones, impide poder mostrar en tiempo real informaciones importantes para la toma de decisiones como, los porcentajes de ejecución de los PI y comparaciones entre los valores iniciales planificados y lo real ejecutado.
- Duplicidad de información: las diferentes áreas de la empresa que realicen las actividades de importación y necesiten contar con el PI utilizan una copia de dicho documento atentando esto contra la consistencia de datos.

INTRODUCCIÓN

- Errores asociados a la gestión financiera, al realizarse esta de forma manual en ocasiones se presentan errores de cálculo en los porcentajes de ejecución del presupuesto del PI lo que ocasiona errores al afectar las aprobaciones financieras de los clientes respecto a las formas de financiamiento presentadas en sus PI atentando esto contra el cumplimiento de dichos PI.

Para automatizar dichos procesos, la empresa de Tecnotex solicitó los servicios de la Universidad de las Ciencias Informáticas, institución dedicada al proceso de información que se lleva a cabo en el país. Dicha entidad cuenta con varios centros productivos entre los que se encuentra el Centro de Informatización de Entidades (CEIGE). El mismo posee varios proyectos dedicados al desarrollo de productos que están destinados a la gestión de entidades. Actualmente, CEIGE se encuentra desarrollando un nuevo sistema informático para la empresa Tecnotex, garantizando en este la automatización de los procesos vinculados a esta investigación.

A partir de la problemática antes planteada se define como **problema a resolver**: ¿Cómo aumentar la calidad en la gestión de los planes de importación en la empresa Tecnotex?

El problema definido se enmarca en el siguiente **objeto de estudio**: los sistemas de gestión empresarial, delimitando como **campo de acción**: los sistemas de gestión empresarial en la gestión de planes de importación.

Se define como **objetivo general**: Desarrollar un componente para la gestión de planes de importación que contribuya al aumento de la calidad de los planes.

A su vez para darle cumplimiento al objetivo general se abordarán los siguientes **objetivos específicos**:

1. Construir el marco teórico de la investigación para sustentar los conceptos, la propuesta de desarrollo de las funcionalidades, las herramientas y tecnologías a utilizar.
2. Realizar el análisis y diseño de la solución a implementar teniendo en cuenta las necesidades del cliente.
3. Implementar el componente para la gestión de planes de importación.
4. Validar la propuesta de solución a partir de los métodos definidos en la investigación.

Para desarrollar los objetivos planteados se llevarán a cabo las siguientes **tareas investigativas**:

1. Estudio del estado del arte y referentes teóricos de la investigación.
2. Definición y estudio de metodologías, tecnologías y herramientas a emplear para el desarrollo del componente.
3. Identificación de patrones a utilizar en el diseño e implementación del componente
4. Elaboración del análisis y diseño de la solución.
5. Identificación, especificación y validación de los requisitos funcionales, no funcionales de la solución.
6. Implementar los requisitos identificados para solucionar las necesidades del cliente.
7. Validar la solución mediante técnicas de pruebas para garantizar la calidad del resultado final.

Se plantea como **ideas a defender** que, si se desarrolla un componente para la gestión de planes de importación, se contribuye positivamente en el aumento de la calidad en la gestión de los planes.

Métodos de investigación científica:

Métodos teóricos:

Análisis histórico-lógico: Este método se empleó para la realización de un estudio sobre el comportamiento y funcionamiento de los sistemas existentes de gestión empresarial que llevan a cabo la gestión del plan de importación.

Análítico-sintético: Este método permitió analizar el proceso de gestión del plan de importación, logrando sintetizar la información adquirida durante el análisis para el entendimiento del funcionamiento de este proceso.

Modelación: Se utiliza en el diseño del sistema mediante el esbozo de los diferentes diagramas definidos en la metodología seleccionada, permitiendo así un mejor entendimiento de las funcionalidades de la aplicación.

Métodos empíricos

Entrevista: Se utilizó para adquirir información relacionado con la realización del proceso de gestión de los planes de importación en la empresa.

Observación: Se utilizó para obtener una información más precisa de cómo se realiza el proceso de gestión de planes de importación de la empresa Tecnotex.

Estructura del trabajo

Capítulo 1. Fundamentación teórica de la investigación se describen los principales conceptos relacionados con el proceso de gestión de planes de importación. Se hace un estudio y valoración de las soluciones existentes en el ámbito de las importaciones y el comercio exterior. Se realiza la fundamentación de metodología que guiará el proceso y las herramientas y lenguajes adecuados para la elaboración del sistema.

Capítulo 2: Propuesta de solución se definen los requisitos funcionales y no funcionales, y su descripción. Se documentan los principales artefactos que propone la metodología de desarrollo de software seleccionada a partir de la solución que se diseñó para la gestión del plan de importación. Se realizó el diagrama de clases del diseño con estereotipos web y, por último, se valida el diseño a través de las métricas seleccionadas.

El **Capítulo 3: Implementación y validación de la solución propuesta** en este capítulo se implementan las funcionalidades identificadas, se realiza la validación del componente desarrollado mediante las pruebas internas y pruebas de aceptación. Se valida la investigación y se exponen los beneficios del nuevo componente.

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

1 Capítulo 1: Fundamentación Teórica de la Investigación

Introducción

En este capítulo se abordan los conceptos necesarios para la comprensión del problema. Se realiza un estudio a sistemas de importación y con ello una valoración de los mismos. Además, se explica la metodología de desarrollo de software a utilizar, las tecnologías y herramientas seleccionadas para el diseño e implementación del componente.

1.1 Conceptos asociados al dominio del problema

Para una mejor comprensión de lo planteado en la situación problemática es necesario identificar y fundamentar los principales conceptos que abarcan tanto el objeto como el campo de acción de la presente investigación.

Importación: Transporte legítimo de bienes y servicios nacionales exportados por un país, pretendido para el uso o consumo interno de otro país (3).

Plan de importación: Un plan de importación se define como el documento en el cual se indica el proceso o procedimiento a seguir para la importación de un producto. Para la empresa Tecnotex un plan de importación se define como el documento mediante el cual se presenta la planificación del presupuesto para los pedidos a realizar por la empresa cliente para el año siguiente (4).

Aprobación: Es el instrumento financiero que respalda el presupuesto para los pedidos a realizar por la empresa cliente.

Extraplan: Son las modificaciones que se le han realizado al plan de importación presentado inicialmente por el cliente.

1.2 Análisis de principales sistemas de gestión para empresas importadoras.

En la actualidad existen varios sistemas de importación, los cuales favorecen la gestión de los procesos involucrados en esta actividad. En este acápite se describen y analizan distintos sistemas de importación, en particular los relacionados a la gestión del proceso de importación, en específico la gestión de los planes de importación o gestión del presupuesto asociado a las importaciones, así como el control de la ejecución de dicho financiamiento

Mistral Import

Mistral Import es un sistema informático orientado a la gestión comercial y logística de una central de compras; diseñada para pequeñas y medianas empresas. Este sistema está destinado a automatizar el proceso de compras(6). Es utilizada en Cuba por sus bondades para:

- Control de contratos de proveedores y clientes.
- Condiciones de pago, cronograma de entregas, e incidencias.
- Gestión de las solicitudes de pago al departamento económico.

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

- Generación por el sistema del texto del contrato.
- Control del cumplimiento del proveedor en tiempo y con la calidad requerida de las entregas pactadas.
- Obtención del cronograma depagos pendientes de las facturas recibidas.

OrbitTrade

Es una herramienta que se adapta a diferentes escenarios para proporcionar un control y un seguimiento sobre el proceso de exportación/importación. Contiene elementos comunes en todo proceso de importación y exportación(7).

Entre las principales funcionalidades que posee se pueden citar:

- Generación de pedidos y control de su estado.
- Planificación de entregas y recepción.
- Gestión de enmiendas (estado de las mismas, tiempos de resolución, etc.).
- Contratos marco con clientes/proveedores y anexos a contratos.
- Catálogos de producto por cliente/proveedor.
- Control de costes financieros por transacción.
- Gestión de facturas y proformas.
- Gestión documental en cualquier fase del proceso: contratos, cartas de crédito, transporte.

Odoo

Odoo es un sistema de planificación de recursos (ERP) integrado, de código abierto y multiplataforma, es uno de los principales sistemas de gestión empresarial de software libre. Odoo es sencillo, ligero e integra los módulos de administración basada en la relación con los clientes(CRM), facturación, contabilidad y gestión de proyectos. Al ser de código abierto, tiene posibilidades de ser implantado en una nube empresarial, y con una base de datos centralizada(8).

Odoo permite realizar la gestión de presupuestos utilizando la contabilidad general y la analítica apoyándose de los módulos Contabilidad y Facturación contando con las siguientes funcionalidades:

- Crear línea de presupuesto (cada línea contiene una cuenta analítica, una posición presupuestaria, fecha de inicio y fecha de finalización, cantidad planificada)
- Confirmar presupuesto.
- Aprobar presupuesto.
- Consultar estado del presupuesto (importe planificado, importe real, importe teórico).

SixDegreesPlanning:

Es un software de presupuesto creado por la empresa del mismo nombre. Permite informatizar todo el ciclo de planificación financiera a partir de la fijación de objetivos y presupuestos para su previsión y consolidación. Presenta la capacidad de generar informes, análisis predictivo, análisis de los datos

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

estadísticos, crear múltiples versiones, monitorear el desempeño y alinear recursos de acuerdo con las necesidades del mercado. Sus principales funcionalidades permiten: (9)

- Planificación y análisis: crear y analizar planes, presupuestos y pronósticos.
- Análisis predictivo: Modelar dimensiones y métricas para medir el progreso hacia los objetivos y vincularlos con acciones y pronósticos.
- Modelado: Desarrollar e implementar modelos de planificación y análisis más complejos.

SISTRADE MIS/ERP: Es un sistema para la elaboración de presupuesto que permite controlar los elementos que suministra el cliente y calcula los costes relacionados con el trabajo. El sistema ERP abarca los procesos de: solicitud de presupuesto, cálculo de costos, definición de márgenes, simulación de otras cantidades, envío del presupuesto y la aprobación del presupuesto (10).

CMP/Presupuesto y Planificación: Es un sistema que se adapta a las necesidades presupuestarias ya sea por supuesto tradicional, flexible o por actividades, los datos que almacena se encuentran integrados y centralizados. Entre sus principales funcionalidades se pueden destacar: la gestión del plan financiero que permite la creación y ajustes de los planes financieros de la empresa, la generación de reportes mensuales, la simulación de varios escenarios para cada plan financiero, la planificación de proyectos y su integración con planes financieros(11).

1.2.1 Valoración de los sistemas analizados

Los sistemas descritos anteriormente presentan características y funcionalidades relacionadas con la gestión de los planes de importación (también vistos como presupuestos o planificación de presupuestos) y las aprobaciones financieras. En la siguiente tabla se muestra una comparación entre las características que se consideran de relevancia para la propuesta de solución teniendo en cuenta las políticas de soberanía e independencia tecnológica, estos indicadores son:

- Software Libre.
- Multiplataforma.
- Soporte.
- Planificación de presupuestos.
- Gestión de presupuestos asociados a la importación.
- Manejo y control de financiamiento.

Tabla 1: Comparación entre los sistemas analizados

Sistema/ Características	Software Libre	Multi- plataforma	Soporte	Planificación de presupuesto	Planificación de presupuesto asociados a la importación	Manejo y control del financiamiento
-----------------------------	-------------------	----------------------	---------	------------------------------------	--	---

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

Mistral Import	No	Si	Si	No	Si	No
OrbitTrade	No	Si	Si	No	No	Si
SixDegreesPlanning	No	Si	Si	Si	No	No
SISTRADE MIS/ERP	No	Si	Si	Si	No	Si
CPM/Presupuesto y Planificación	No	Si	Si	Si	No	Si
Odo	Si	Si	Si	Si	No	Si

Haciendo un análisis de la tabla se puede apreciar que de los 6 sistemas estudiados solo 1 es software libre mientras que todos son multiplataformas y poseen soporte. La mayoría de los sistemas permiten la planificación del presupuesto y el manejo y control del financiamiento, sin embargo, solo uno de los sistemas permite la planificación de presupuestos asociados a la importación. Por otra parte, el 50 % solo cumple con una de las características enunciadas: Mistral Import solo presenta características afines a la planificación de presupuestos asociados a la importación, OrbitTrade al control y manejo del financiamiento y SixDegreePlanning a la planificación de presupuestos de forma general. SISTRADE MIS/ERP, CPM/Presupuesto y Planificación y Odo permiten la planificación de presupuestos y el manejo y control del financiamiento: SISTRADE los aborda como solicitud de presupuesto, definición de márgenes y aprobación del presupuesto, CPM/Presupuesto y Planificación lo contiene como gestión de planes financieros y Odo a través de la creación de la línea presupuestaria, la aprobación del presupuesto y los estados de los mismos.

A pesar de dichos sistemas contar con la mayoría de las características planteadas no presentan todas las necesarias para resolver el problema en cuestión, por lo que sería necesario adaptar dichas funcionalidades al contexto empresarial de Tecnotex, lo que no sería posible utilizando los sistemas SISTRADE o CPM/Presupuesto y Planificación debido a que estos no son software libre. Odo, por otra parte, es software libre bajo licencia pública general de Affero (AGPL), este posee un sistema modular y configurable que brinda facilidades de trabajo e integración de sus componentes, especialmente para herramientas de negocio, Odo además, puede ser utilizado como marco de

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

trabajo, lo que permitiría apoyarse en las características ya incluidas en dicho sistema para la creación de un nuevo componente para la gestión del plan de importación en la empresa Tecnotex.

Teniendo en cuenta las facilidades de Odo o como marco de trabajo se decide utilizarlo para el desarrollo del componente. La aprobación de su utilización se definió por el equipo de proyecto, la dirección del centro CEIGE y el equipo de consejo técnico de la universidad.

1.3 Metodología de desarrollo de software

Para comenzar el desarrollo del software es necesario la utilización de metodologías de desarrollo de software, estas proponen la utilización de procedimientos, guías y técnicas que establezcan el modo de construcción del software. Existen diversas metodologías de desarrollo de software, estas están orientadas en función de los nuevos principios de desarrollo del software y pueden ser ajustables de acuerdo a las características del proyecto atendiendo a esto se clasifican en ágiles y robustas, Agile Unified Process (AUP), SCRUM y Extreme Programming (XP) son consideradas metodologías ágiles, mientras que Rational Unified Process (RUP) y Microsoft Solution Framework (MSF) son metodologías robustas. (12)(13).

La Universidad de las Ciencias Informáticas (UCI) para sus procesos productivos cuenta con una metodología para el desarrollo de la actividad productiva denominada AUP-UCI. Esta metodología se basa en una variación de la metodología AUP en unión con el modelo Capability Maturity Model Integration for Development (CMMI-DEV)v1.3, con el objetivo de lograr una adaptación a la UCI y contribuir así al proceso de mejoras de la Universidad.

La metodología AUP-UCI propone tres fases para el desarrollo: Inicio, Ejecución y Cierre, además cuenta con 7 disciplinas, estas son:

- Modelado de negocio: destinada a comprender los procesos de negocio de una organización.
- Requisitos: comprende la administración y gestión de los requisitos funcionales y no funcionales del producto.
- Análisis y diseño: se modela el sistema y su forma para que soporte todos los requisitos incluyendo los no funcionales.
- Implementación: a partir de los resultados del Análisis y Diseño se construye el sistema.
- Prueba: se verifica el resultado de la implementación, se divide en tres disciplinas, pruebas internas, de liberación y de aceptación

A partir del Modelado de negocio la metodología propone tres variantes a utilizar en el modelado de los proyectos: Casos de Uso del Negocio (CUN), Descripción de Proceso de Negocio (DPN) y Modelo Conceptual (MC). Además, existen tres formas de encapsular los requisitos: Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP), teniendo esto en

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

cuentasurgen cuatro escenarios para modelar el sistema en los proyectos, quedando de la siguiente forma:(14)

Escenario No 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

$CUN + MC = CUS$

Escenario No 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

$MC = CUS$

Escenario No 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

$DPN + MC = DRP$

Escenario No 4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

HU.

La selección de la metodología estuvo condicionada en gran parte a que la propuesta de solución tributa a un software de la universidad desarrollado por un proyecto de uno de sus centros productivos y por tanto debe cumplir las políticas productivas establecidas por el centro como lo es la utilización de la metodología AUP-UCI. De esta forma se logra una convergencia entre todos los procesos que se lleven a cabo en cuanto al desarrollo del componente.

Una vez analizados todos los escenarios que propone la metodología, se seleccionó el escenario número tres puesto que es el escenario que se aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado se obtengan procesos muy complejos. Los procesos del negocio son complejos y se desea representar una gran cantidad de detalles y relaciones entre los procesos. Además, el negocio está documentado y el cliente tiene dominio sobre el negocio.

1.4 Notaciones y Lenguajes

Debido a la complejidad que presentan los procesos de software se hizo necesario los procesos de modelados de software y junto a estos la utilización de lenguajes de modelado y notaciones que permiten representar los elementos que componen al software de forma que los procesos sean entendidos con mayor facilidad, formalizando algunos aspectos que permiten representar de manera precisa y completa aspectos necesarios para llevar a cabo el modelado de procesos de software (15) (16).

A continuación, se describen las notaciones y lenguajes a utilizar en la propuesta a solución.

UML 2.4:

El Lenguaje Unificado de Modelado (UML) es el resultado de un esfuerzo por estandarizar el modelado de software Orientado a Objetos. UML fue adoptado en 1997 por Object Management Group(OMG) como una de sus especificaciones, y desde entonces se ha convertido en un estándar para visualizar, especificar y documentar los modelos que se crean durante la aplicación de un proceso de software.

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

Este permite una forma de modelar conceptualmente procesos de negocio y funciones de sistema, además de elementos concretos como son esquemas de base de datos, componentes de software reutilizables y escribir clases en un lenguaje determinado. Es interactivo, orientado a objetos, es altamente usado por la comunidad internacional y se ha convertido en un estándar para el desarrollo de sistemas informáticos(17)(18).

Se seleccionó UML por las múltiples potencialidades que brinda para el modelado de los diagramas de clases y de componentes permitiendo reflejar todos los elementos del sistema en dicho modelado con la mayor cantidad de detalles posibles.

XML 1.0

XML es un metalenguaje extensible de etiquetas que fue desarrollado por el World Wide Web Consortium (W3C). Este lenguaje representa una clase de objetos de datos llamados documentos XML que describen parcialmente el comportamiento de los programas informáticos que los procesan(19)(20)(21).

Debido a que Odoo define a XML como lenguaje de marcas a utilizar se selecciona XML como lenguaje para la generación de vistas y reportes.

Python 3.6

Python es un lenguaje de programación de alto nivel y propósito general, su filosofía de diseño enfatiza en la legibilidad del código. La sintaxis de Python permite a los programadores expresar conceptos en menos líneas de código de lo que sería posible en lenguajes como C++ o Java. Debido en parte a este enfoque, se ha convertido en un lenguaje popular para la informática científica y la ciencia de datos, con un amplio ecosistema de bibliotecas. Por un lado, hay un gran número de bibliotecas de Python, con las cuales se puede integrar prácticamente cualquier sistema de terceros, por otro lado, las bibliotecas de muchos otros idiomas de programación también se pueden usar en Python(22)(23)(24).

Python posee una gran comunidad alrededor y existe abundante documentación de este lenguaje lo que facilita el aprendizaje centrándose más en el problema a resolver que en el propio lenguaje. Por sus numerosas ventajas este lenguaje es utilizado por diferentes marcos de trabajos como Django, Flask, Engine y Odoo.

PostgreSQL 12.1

PostgreSQL es un sistema de base de datos relacional de objetos de código abierto. Este lenguaje usa y amplía el lenguaje de consulta estructurada (SQL) combinado con características que almacenan y escalan las más complicadas cargas de datos. PostgreSQL cuenta con una distinguida reputación por su arquitectura, confiabilidad, integridad de datos y la dedicación de la comunidad de código abierto detrás del software(25).

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

PostgreSQL es un sistema robusto y estable con gran escalabilidad y un buen sistema de seguridad, así como buena capacidad de almacenamiento. Se utilizó para el desarrollo de la propuesta de solución al estar definida por el marco de trabajo Odoo.

1.5 Herramientas y tecnologías para el desarrollo del sistema

En el presente epígrafe se describen las herramientas y tecnologías que se emplearán en el desarrollo e implementación del sistema conformando así la propuesta de solución de esta investigación.

Odoo 13.0

Odoo es una plataforma de código abierto para aplicaciones de negocio. Sobre esta se encuentra una suite de aplicaciones estrechamente integradas, que cubren todas las áreas de negocio desde ERP y Ventas hasta Contabilidad y Suministros(26).

Odoo cuenta con un núcleo sencillo, pero a su vez estable y robusto. Este núcleo funciona como motor de un sistema modular y configurable a gusto del cliente optando por lo que desea instalar según sus requerimientos o necesidades. Sus flujos de trabajo flexibles junto a la base de datos relacional, arquitectura Modelo Vista Controlador (MVC), interfaz gráfica dinámica y sencilla para el usuario, hacen un sistema sólido y estable. Utiliza como gestor de base de datos PostgreSQL y para la arquitectura MVC utiliza una vista mostrada por XML y un controlador hecho con el lenguaje de programación Python. El sistema funciona con un modelo cliente/servidor haciendo que todos los usuarios accedan a la misma información de manera consistente(27).

Visual Paradigm 8.0

Visual Paradigm es una herramienta de ingeniería de software asistida por computadora(CASE) multiplataforma que propicia un conjunto de ayudas para el desarrollo de programas informáticos dando soporte al modelado visual con UML. Es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación(28). Se empleó como herramienta para el modelado tanto del negocio como del sistema.

PyCharm 2018.2.3

PyCharm es un IDE o entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje de programación Python. PyCharm proporciona terminación de código inteligente, inspecciones de código, resaltado de errores sobre la marcha y soluciones rápidas, junto con refactorizaciones de código automatizadas. También, ofrece un excelente soporte específico de framework para frameworks de desarrollo web modernos como Django, Flask, Google App y Engine(29)(30).

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

PgAdmin 1.22.2

PgAdmin es una aplicación gráfica de diseño y manejo de base de datos PostgreSQL que brinda múltiples modelos de implementación, herramientas y mantenimientos de rutinas que permiten escribir simples consultas SQL hasta desarrollar bases de datos complejas. Su interfaz gráfica soporta todas las características de PostgreSQL y hacer simple la administración. Posee una gran variedad de características favorecedoras entre estas destacan que: es multiplataforma, está diseñado para múltiples versiones de PostgreSQL y posee amplia documentación. (31)

1.6 Conclusiones parciales

En este capítulo se abordaron una serie de elementos que fundamenta al marco teórico de la investigación logrando un mayor entendimiento del mismo. Como metodología de desarrollo se seleccionó la Metodología AUP-UCI, en su escenario No.3 ya que en este se definen todos los artefactos, actividades y productos de trabajo necesarios para llevar a cabo la implementación. A partir del marco de trabajo Odoos se describen las herramientas y tecnologías a utilizar para darle solución al problema planteado, las cuales son definidas en la estructura y arquitectura de Odoos.

2 Capítulo 2: Propuesta de solución.

Introducción

En este capítulo se realiza el modelado de negocio, la definición de requisitos y el análisis y diseño del componente para la gestión del plan de importación en la empresa Tecnotex. A partir de las descripciones de procesos de negocios y el modelo conceptual se identifican y se describen los requisitos funcionales y no funcionales. Se realiza el diagrama de clases del diseño con estereotipos web y se valida el diseño de la propuesta de solución.

2.1 Modelado de negocio

El modelado de negocio, ofrece una vista abstracta y simplificada de la realidad compleja en la que se expresan conceptos sobre el funcionamiento del negocio de una entidad u organización empresarial. Es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito(32).

Para modelar los procesos relacionados con la gestión del plan de importación en la empresa Tecnotex se utilizó el escenario número tres de la metodología AUP-UCI contribuyendo a lograr una mayor comprensión del negocio a informatizar y los procesos involucrados en este, generando como artefactos la descripción de procesos de negocio y el modelo conceptual.

2.1.1 Descripción de los procesos de negocio.

La gestión del plan de importación en la empresa Tecnotex abarca diferentes subprocesos que permiten llevar a cabo el control sobre los presupuestos de las diferentes empresas a la hora de realizar una importación. A continuación, se describen cómo se lleva a cabo los procesos relacionados con la investigación:

- **Presentar solicitud del PI**

El proceso inicia cuando el cliente llega a la empresa y presenta su solicitud, ya sea para un Plan de Importación o para un aumento del mismo (extraplan), al especialista en gestión económica. Este especialista recepciona el documento entregado por el cliente y revisa que estén correctos y completos los datos del cliente, el desglose de los planes (inversiones y capital de trabajo) y la fundamentación de la propuesta, si son detectados errores en el documento se le envía al cliente para su corrección, de lo contrario se insertan los datos en el sistema. A continuación, se muestra el diagrama de negocio correspondiente

PROPUESTA DE SOLUCIÓN.

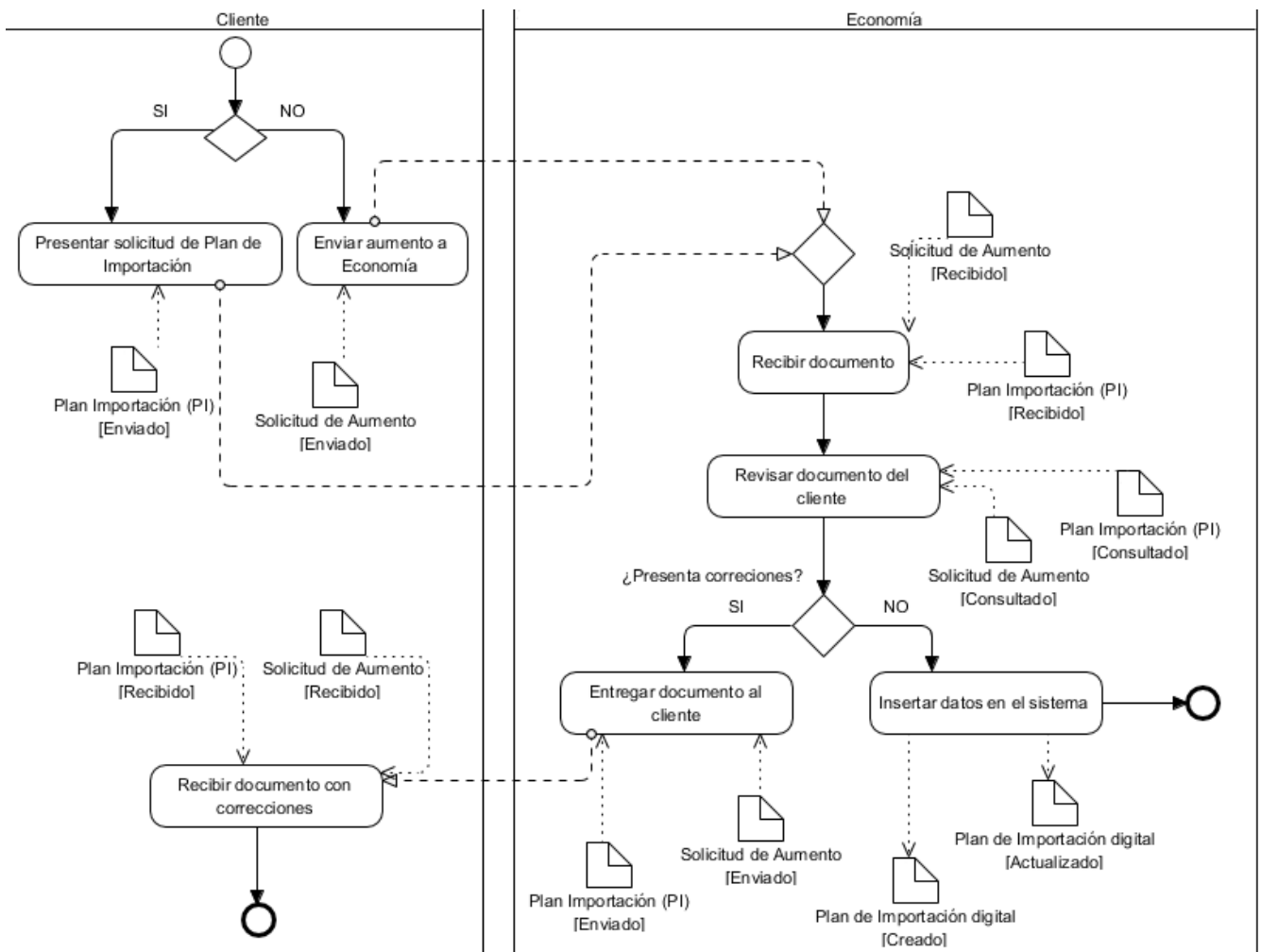


Figura 1: Descripción del proceso de negocio Presentar Solicitud de PI

El modelado del negocio comprende además los subprocesos presentar forma de financiamiento, revisar financiamiento y registrar aprobación. Las descripciones de estos pueden ser consultada en el **Anexo 1**

Para una mayor comprensión de los términos tratados en las descripciones de procesos de negocio, se realizó el modelo conceptual.

2.1.2 Modelo conceptual

El modelo conceptual es una representación de los conceptos del dominio del problema. Este modelo muestra asociaciones entre conceptos y atributos de conceptos. Se puede ver como un modelo que comunica los términos importantes y cómo se relacionan entre sí. (8)

La realización del modelo conceptual del proceso de gestión del plan de importación permitió tener una representación gráfica de los conceptos que se manejan, así como las relaciones que existen entre ellos. A continuación, se muestra el modelo conceptual:

PROPUESTA DE SOLUCIÓN.

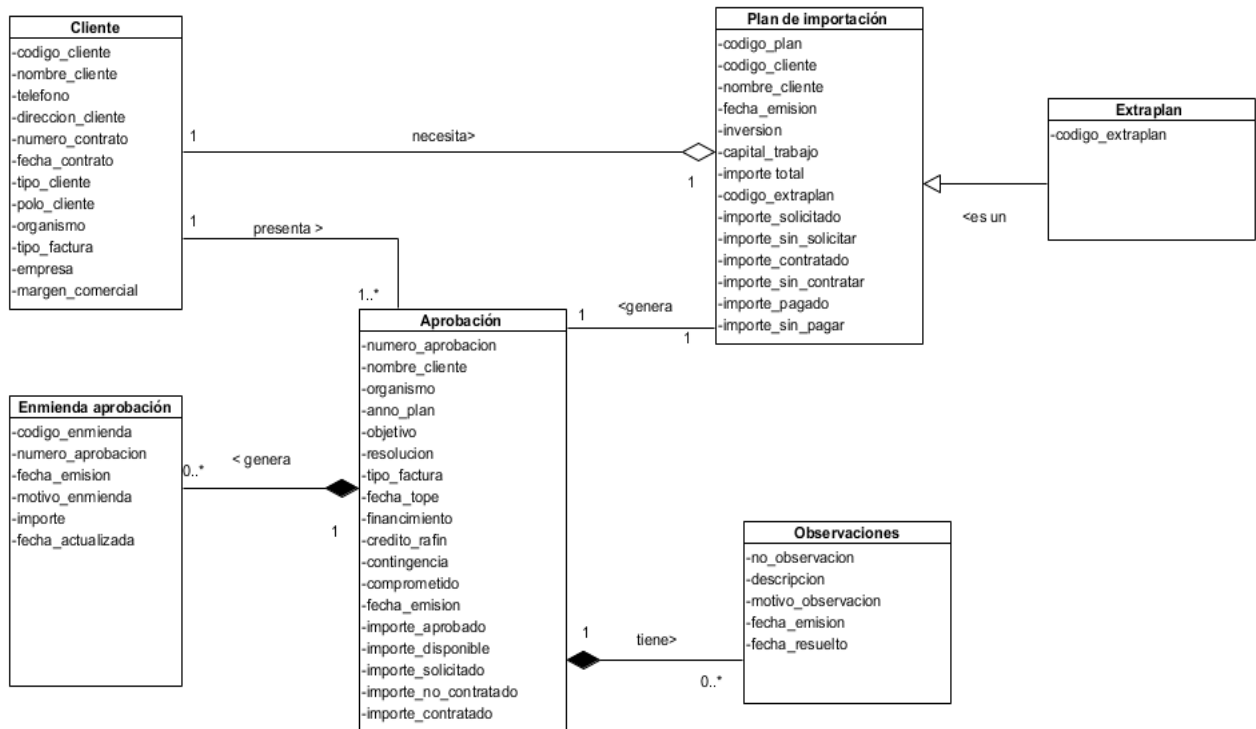


Figura 2: Modelo Conceptual

2.2 Requisitos del sistema

Los requerimientos o requisitos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes(33)

Una vez realizadas las descripciones de procesos y el modelo conceptual es posible identificar y describir los requisitos funcionales del sistema.

2.2.1 Técnicas utilizadas para la obtención de requisitos

La obtención de requisitos es el proceso de recoger información sobre el sistema propuesto y otros sistemas existentes a través de técnicas que determinan el dominio de la aplicación y analizan los servicios que debe proporcionar el sistema(34).

Entre las principales técnicas para la obtención de requisitos se encuentran:

- **Entrevista:** Las entrevistas sirven para obtener una comprensión general de lo que hacen los stakeholders, como podría interactuar con el sistema y las dificultades a las que se enfrentan con los sistemas actuales.
- **Escenarios:** Pueden comprender y criticar un escenario de cómo podrían interactuar con un sistema de software. Esta información se puede utilizar para formular los requerimientos reales del sistema.

PROPUESTA DE SOLUCIÓN.

- **Casos de uso:** Un caso de uso identifica a los actores implicados en una interacción, y nombra el tipo de interacción. Esto se complementa con información adicional que describe la interacción con el sistema.
- **Etnografía:** Es una técnica de observación que se usa para entender los procesos operacionales y ayudar a derivar requerimientos de apoyo para dichos procesos.

Para la obtención de requisitos en la presente investigación se utilizó como técnica la entrevista. Se aplicó realizando preguntas al especialista de economía, en su mayoría relacionadas con el proceso de gestión de los planes de importación en la empresa, obteniendo así la información necesaria para el levantamiento de requisitos.

2.2.2 Requisitos funcionales

Con el fin de identificar que debe hacer el sistema y entender su funcionamiento, es fundamental conocer los requisitos funcionales con los que el software debe cumplir, estos requisitos son la descripción de los servicios proporcionados por el sistema y describen lo que el sistema debe hacer definiendo una función del software o sus componentes.

En la presente investigación se identificaron un total de 57 requisitos funcionales encapsulados en las siguientes agrupaciones (los requisitos correspondientes a cada agrupación se encuentran en el **Anexo 2**):

- Gestionar aprobaciones
- Gestionar desglose por PI
- Gestionar enmienda de aprobación
- Gestionar extraplan
- Gestionar fecha tope de PI
- Gestionar PI
- Gestionar nomenclador objeto
- Gestionar nomenclador tipo de aprobación

Salidas del sistema

Las salidas del sistema es la capacidad de un sistema para generar la información procesada hacia el exterior, bien sean datos de entrada o resultados(35)(36). A continuación, se listan las salidas del sistema identificadas:

- Estado de la aprobación por códigos
- Listado de Aprobación de mercancías
- Reporte del plan de importación
- Listado Plan de importación por cliente

2.2.3 Descripción de requisitos funcionales

La descripción de los requisitos permitió obtener una versión completa del comportamiento del sistema a desarrollar. En dichas descripciones se reflejan los flujos por los que transita el requisito, la información que muestra, así como las restricciones que posee. A continuación, se muestra un ejemplo de la descripción del requisito Crear plan de importación de la agrupación Gestionar PI.

Tabla 2: Descripción del requisito Crear Plan de importación

Precondiciones	<p>El cliente ha sido contratado por la Empresa TECNOTEX.</p> <p>El usuario debe tener los permisos para adicionar plan de importación.</p> <p>Selecciona el menú Presentación / Plan de importación</p>
Flujo de eventos	
Flujo básico Crear plan de importación	
1	Se selecciona la opción Crear plan de importación
2	<p>Se introducen los datos del plan de importación:</p> <ul style="list-style-type: none"> -*Código PI -*Cliente -*Fecha de emisión <p>Forma de financiamiento:</p> <ul style="list-style-type: none"> -A la vista (Inversiones / Capital de trabajo) -Corto plazo (Inversiones / Capital de trabajo) -Mediano plazo (Inversiones / Capital de trabajo) -Largo plazo (Inversiones / Capital de trabajo) -Listado de extraplan (ver DR Gestionar extraplan)
3	Se selecciona el botón de Guardar.
4	El sistema valida los datos introducidos (ver validaciones).
5	Si los datos son correctos el sistema los registra.
6	El sistema confirma el registro de los datos.
7	Concluye el requisito.
Pos-condiciones	
1	Se registró en el sistema un nuevo plan de importación
Flujos alternativos	
Flujo alternativo 4.a Información errónea	
1	El sistema señala los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.

PROPUESTA DE SOLUCIÓN.

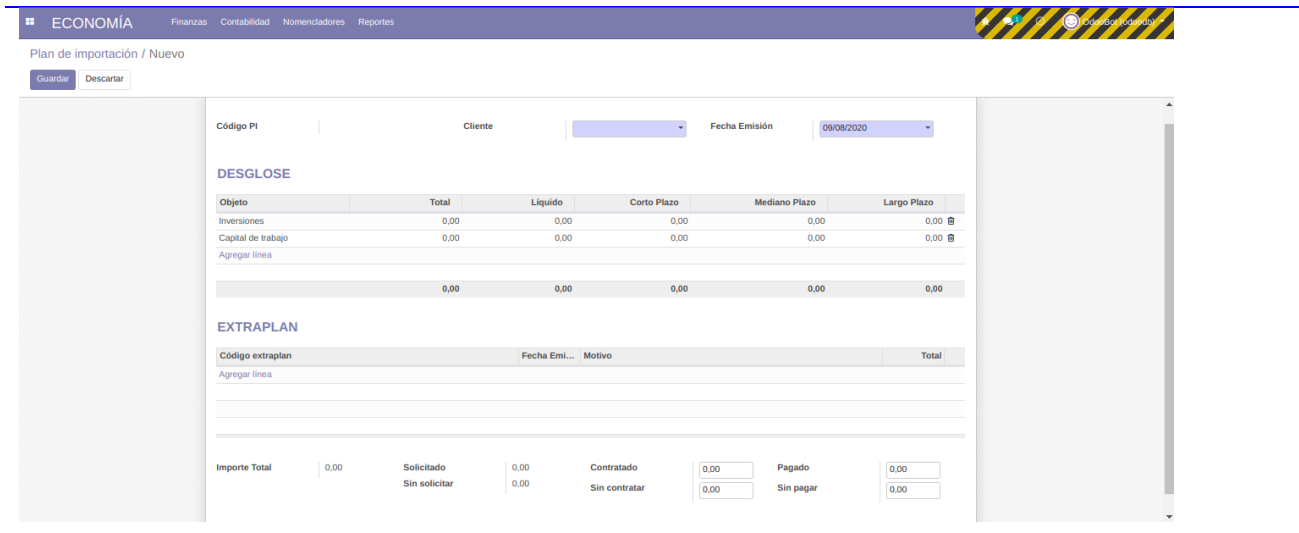
3	Volver al paso 3 del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo 4.b Información incompleta	
1	El sistema señala los datos vacíos y permite corregirlos.
2	El usuario corrige los datos.
1	Volver al paso 3 del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo *.a El usuario cancela la acción	
1	Concluye el requisito.
Pos-condiciones	
1	No se registran los datos.
Validaciones	
1	Se validan los datos según lo establecido en el Modelo conceptual CEIGE_TECNOTEX_Modelo_conceptual.
2	Validar que el código PI sea un dato único y esté conformado de la siguiente manera: el año en que se presenta y cuatro números consecutivos, el sistema lo genera de forma automática tomando la fecha actual. Ej.: 20180001.
3	Validar que el cliente no presente más de un plan de importación en el mismo año.
4	Validar que se encuentre en el rango de fecha permitido para presentar y aprobar PI. En caso que se pase de la fecha que te muestre un mensaje de que se pasó de la fecha tope, pero al final te permita insertar el PI.
5	Validar que el sistema sólo muestre los clientes que presentan contrato con TECNOTEX activo.
6	Validar que el importe total sea la suma de los montos del desglose del plan (Inversiones y capital de trabajo).
7	Validar que el importe solicitado sea la suma de los importes presentados que tienen todos los pedidos de ese cliente. Ver DR Gestionar pedidos.
8	Validar que la suma de las formas de financiamiento (a la vista, corto plazo, mediano plazo, largo plazo) sea igual al desglose de plan que le corresponde.
9	Validar que en caso que los pedidos presentados se pasen del plan de importación, el sistema debe mostrar una alerta y seguido no deja seguir presentando más pedidos al no ser que lo autorice el Director de Economía.
10	Validar que los valores del Extraplan no modifique ningún dato del PI, sólo es necesario

PROPUESTA DE SOLUCIÓN.

	saber la relación que existe entre el extraplan y el PI.		
11	<p>Mensajes que debe mostrar el sistema:</p> <p>1-Cuando hay campos obligatorios y el usuario guarda sin registrar dichos campos: Mensaje: Los siguientes campos son obligatorios: -(muestra con plecas los campos obligatorios que se encuentran vacíos)</p> <p>2-Antes de guardar, si el campo tiene valor incorrecto: Se va aplicar una función de validación, que te va a mostrar el mensaje en dependencia del error. Mensaje: El campo (nombre del campo) solo admite (descripción de la variable definida). Ejemplo: El campo <u>cliente</u> solo admite <u>letras</u>. Cuando se activa la validación del campo el sistema muestra el mensaje ejemplificado anteriormente y limpia el campo. En el caso de los campos que tengan una estructura específica, el sistema muestra en el mensaje un ejemplo válido del campo.</p> <p>3-Eliminar elementos: Cuando se elimina un elemento desde el listar o desde la interfaz del Mostrar, el sistema debe arrojar el siguiente mensaje. Mensaje: ¿Está seguro que quiere eliminar este registro? En caso de eliminar elementos que tiene operaciones asociadas no te va a permitir eliminarlo, pero si desactivarlo. (Si se desactiva no se puede desaparecer sus datos de las operaciones que tiene asociada) Mensaje: El <Elemento> presenta operaciones asociadas por lo que no puede ser eliminado.</p> <p>4-Valores repetidos (Valor único): El sistema no permite que se repitan elementos. Mensaje: Ya existe un (<u>elemento</u>) registrado con ese (valor único de ese elemento). Ejemplo: Ya existe una <u>aprobación</u> registrado con ese No.<u>Aprobación</u>.</p> <p>5-Botón Descartar: A la hora de descartar, si el usuario ha registrado algún dato el sistema debe arrojar el siguiente mensaje: ATENCIÓN El registro ha sido modificado. Sus cambios se descartarán. ¿Desea continuar?</p>		
Conceptos	<table border="1"> <tr> <td>plan de importación</td> <td> Visibles en la interfaz: -Código PI -Cliente </td> </tr> </table>	plan de importación	Visibles en la interfaz: -Código PI -Cliente
plan de importación	Visibles en la interfaz: -Código PI -Cliente		

	<p>-Fecha de emisión</p> <p>Desglose Plan:</p> <p>-Inversión</p> <p>-Capital de trabajo</p> <p>Forma de Financiamiento para cada desglose del plan:</p> <p>- a la vista</p> <p>- corto plazo</p> <p>- mediano plazo</p> <p>- largo plazo</p> <p>-Importe total /Importe solicitado /Importe sin solicita /Importe contratado /Importe sin contratar /Importe pagado /Importe sin pagar</p> <p>-Listado de extraplan (ver DRP Gestionar extraplan)</p>
Requisitos especiales	N/A
Asuntos pendientes	N/A

Prototipo elemental de interfaz gráfica de usuario



2.2.4 Requisitos no funcionales

Los requisitos no funcionales (RnF) detallan las propiedades o cualidades que el producto debe tener, aumentándole funcionalidad al sistema, pues hacen al producto atractivo, fácil de usar, rápido y confiable, los cuales se encuentran separados por categorías. A continuación, se muestran los requisitos identificados para cada categoría.

Tabla 3: Requisitos no funcionales

PROPUESTA DE SOLUCIÓN.

No.	Requisitos no funcionales
Disponibilidad	
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.
Usabilidad	
RnF 4	El componente de la interfaz de usuario debe ejecutarse en un navegador de Internet.
Portabilidad	
RnF 5	La base de datos debe poder ser divisible, cada polo debe poder tener su base de datos.
Interoperabilidad	
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.
Seguridad	
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.
Escalabilidad	
RnF 10	La aplicación debe poder manejar una carga máxima de 300 usuarios simultáneos.
Software	
RnF 11	Navegador Mozilla Firefox versión 71 o superior
RnF 12	Como marco de trabajo se utiliza la plataforma de desarrollo Odoon 13.0
RnF 13	Se empleará como Gestor de Base de Datos, PostgreSQL 12.1
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

PROPUESTA DE SOLUCIÓN.

	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

2.2.5 Validación de requisitos

La validación de requerimientos trata de mostrar que estos realmente definen el sistema que el cliente desea. Se analiza la especificación a fin de garantizar que todos ellos han sido enunciados sin ambigüedades; que se detectaron y corrigieron las inconsistencias, las omisiones y los errores, y que los productos del trabajo se presentan conforme a los estándares establecidos para el proceso, el proyecto y el producto(37). Existen diversas técnicas para la validación de los requisitos, a continuación, se describen unas de las más utilizadas:

- **Revisiones de requerimientos:** Los requerimientos son analizados sistemáticamente por un equipo de revisores e involucra a personas tanto de la organización del cliente como la del contratista.
- **Construcción de prototipos:** En este enfoque de validación se muestra un modelo ejecutable del sistema a los usuarios finales y a los clientes, para experimentar si cumple sus necesidades reales.
- **Generación de casos de pruebas:** Los requerimientos deben poder probarse. Si las pruebas para estos se conciben como parte del proceso de validación, a menudo revela problemas con el requerimiento

Para la validación de los requerimientos se utilizó la técnica de construcción de prototipos. Una vez identificados y descritos cada uno de los requisitos se diseñaron prototipos de interfaces de usuario utilizando como herramienta de modelado Pencil. Mediante los prototipos de interfaz de usuario se obtuvo una visión inicial del sistema a implementar demostrando cómo se van a disponer posteriormente los conceptos que intervienen en el mismo. Estos prototipos fueron revisados, llegando a un entendimiento de la información especificada que permitió la aprobación de cada uno de los requisitos por parte del cliente.

2.3 Diseño de la propuesta de solución

El diseño es un proceso donde a partir de los requerimientos se sintetizan las representaciones de los datos la estructura del programa, características de la interfaz y detalles del procedimiento. En este

PROPUESTA DE SOLUCIÓN.

epígrafe se modela el sistema a partir de los requerimientos definidos para una mayor consistencia en la etapa de implementación.

2.3.1 Diseño arquitectónico

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos (37). El marco de trabajo Odoosigue una arquitectura MVC donde a través del modelo define la estructura de los datos, la vista describe la interfaz con el usuario y el controlador soporta la lógica de negocio de la aplicación (26).

A continuación, se profundiza en la descripción de cada capa de la arquitectura MVC en el marco de trabajo.

- **Modelo:** El modelo está definido por objetos Python cuyos datos son almacenados en una base de datos PostgreSQL. El mapeo de la base de datos es gestionado automáticamente por Odoos, y el mecanismo responsable por esto es el modelo objeto relacional (ORM). Los modelos se encuentran dentro de la carpeta models.
- **Vista:** Las vistas son definidas usando XML, las cuales son usadas por el marco de trabajo del cliente web para generar vistas de datos. Se encuentran dentro de la carpeta views.
- **Controlador:** el controlador está compuesto por las clases controladoras, se pone de manifiesto a través del modelo controller.py, el cuál es el encargado de hacer peticiones al modelo cuando se hace alguna solicitud de la información por parte del cliente.

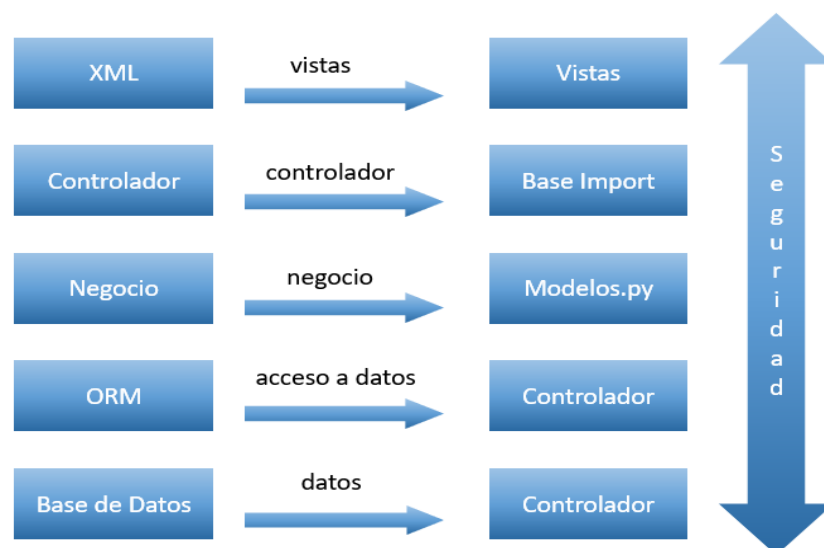


Figura 3: Arquitectura MVC

2.3.2 Diagrama de clases con estereotipos web

Un Diagrama de Clases de Diseño (DCD) representa las especificaciones de las clases e interfaces de software en una aplicación y muestra las relaciones existentes entre las clases del sistema. Éste contiene además la clase servidora, así como los formularios que se muestran en estas. Se utiliza para mostrar una vista más específica de las acciones que realiza el usuario y el flujo de actividades internas del sistema.(8)

A continuación, se muestra el diagrama de clases con estereotipos web para la descripción de requisitos por procesos Gestionar desglose por PI donde la página cliente principal le hace una petición a la página del servidor, esta construye las páginas clientes (crear, editar, suprimir, listar, mostrar, buscar) a partir de la recopilación de elementos de los formularios para mostrarlos en la vista correspondiente.

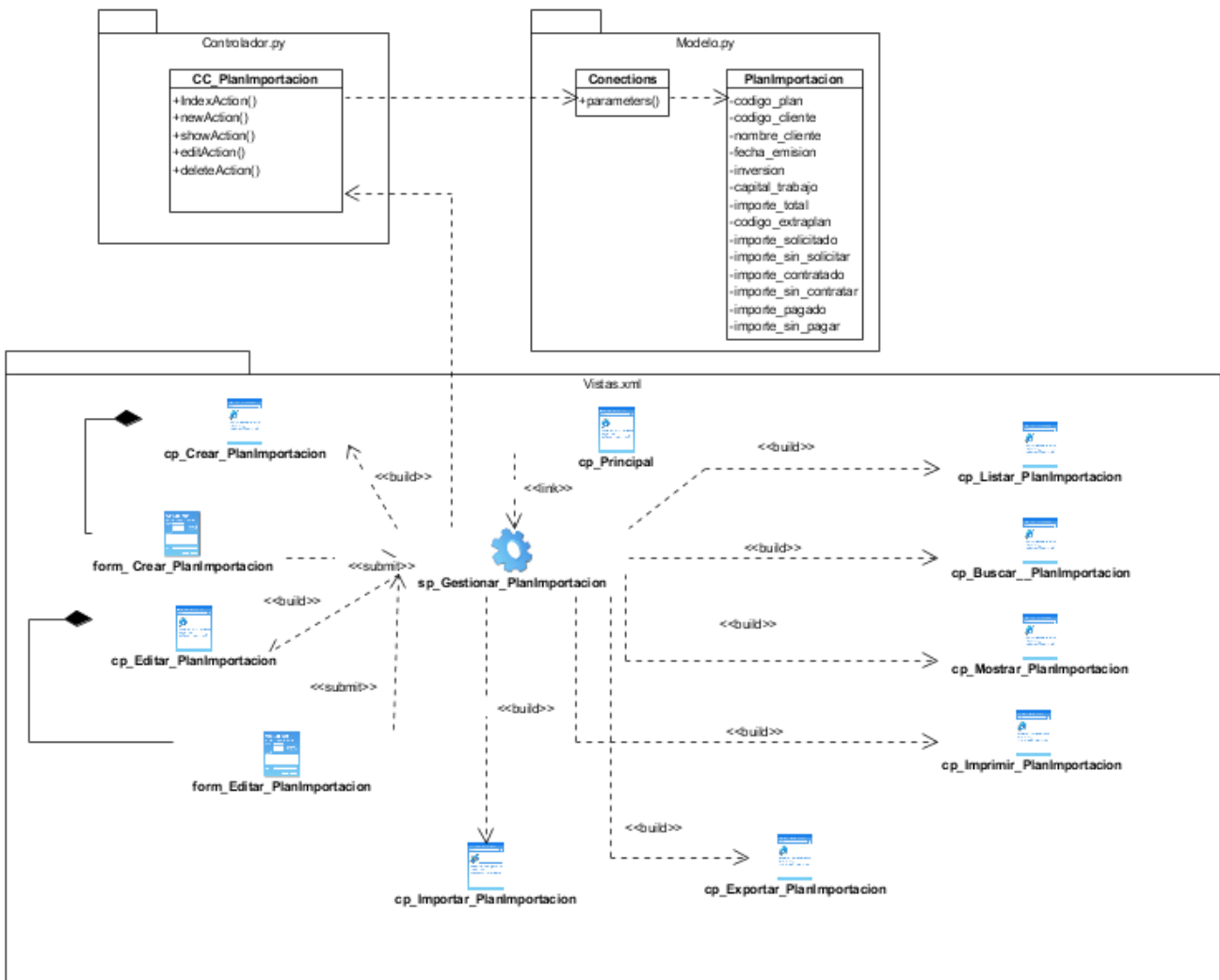


Figura 4: Diagrama de clases del diseño con estereotipos web para la DRP Gestionar_PI:

2.3.3 Patrones de diseño

Un patrón de diseño se caracteriza como “una regla de tres partes que expresa una relación entre cierto contexto, un problema y una solución”(37). A continuación, se describen algunos patrones a tener en cuenta en la implementación de propuesta de solución:

Patrones Generales de Software para Asignación de Responsabilidades (GRASP)

- Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (38). Estos patrones son:
- **Experto:** Experto es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele útil en el diseño orientado a objetos. El comportamiento se distribuye entre las clases que cuentan con la información requerida. El experto en la información es la clase que tiene la información necesaria para cumplir la responsabilidad. Fue usado en la mayoría de las clases ya que a cada una le fue asignada la responsabilidad en dependencia de la información que esta posee.
- **Creador:** Asignar a una clase la responsabilidad de crear una instancia de otra clase cumpliéndose determinados casos. Dicho patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. Este patrón se puede apreciar principalmente en las clases controladoras las cuales necesitan crear objetos de las clases del modelo para acceder a las funcionalidades que estas poseen.
- **Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. Se observa en la mayor parte del diseño pues las clases fueron creadas teniendo en cuenta que las mismas presentan la menor cantidad de relaciones de dependencia posible.
- **Alta cohesión:** asignar responsabilidades de manera que la cohesión se mantenga alta, o sea, cada clase se encargará de realizar solamente las funciones que estén en correspondencia con la responsabilidad que esta posee. Propone que la información que almacena una clase debe de ser coherente y debe estar, en la medida de lo posible, relacionada con la clase.

PatronesGang of four (GOF)

- **Observador:** Los objetos son capaces de suscribirse a una serie de eventos que otro objeto va a emitir, y serán avisados cuando esto ocurra. El patrón Observador puede ser utilizado cuando hay objetos que dependen de otro, se puede apreciar en la relación que existe entre Plan de importación y Aprobación.

```

observacion_ids = fields.One2many('tecnorex.observacion', 'aprobacion_id', string=' Observación')

plan_ids = fields.Many2many('tcx_economia.plan_importacion', string=' Plan', required=True)
objeto id = fields.Many2one('tcx economia.nom.objeto', 'Objeto', required=True)

```

Figura5: Código de la clase Aprobación donde se evidencia el uso del patrón Observador

2.4 Validación del Análisis y Diseño.

Las métricas de diseño permiten medir de forma cuantitativa la calidad de los atributos internos del software. Se centran en cuantificar tanto la complejidad, como la funcionalidad y eficiencia inmersa en el desarrollo de software.(39)

Lorenz y Kidd dividen las métricas basadas en clases en cuatro categorías: tamaño, herencia, valores internos y valores externos. Estas métricas están enfocadas a las características internas del diseño orientado a objeto y de esta manera, contribuyen a asegurar la mantenibilidad de los productos de software. Las métricas orientadas a tamaños para una clase se centran en cálculos de atributos y de operaciones para una clase individual, y promedian los valores para el sistema en su totalidad. Las métricas basadas en herencia se centran en la forma en que se reutilizan las operaciones a lo largo y ancho de la jerarquía de clases. Las métricas para valores internos de clase examinan la cohesión y asuntos relacionados con el código, y las métricas orientadas a valores externos examinan el acoplamiento y la reutilización(40)(41).

Para la validación del diseño se aplicaron las métricas Tamaño Operacional de Clases (TOC) y Relaciones entre Clases (RC).

Métrica de Tamaño Operacional de Clases (TOC): está dado por el número de métodos asignados a una clase. Mediante el cual se calcula el nivel de Responsabilidad de los métodos, la Complejidad de implementación de los mismos y su Reutilización, a fin de inspeccionar la efectividad del diseño, existiendo una relación directa con los dos primeros e inversa con el último antes mencionado.(41)

Tabla 4: Atributos de calidad evaluados por la métrica TOC

Atributo de calidad	Modo en que lo afecta
Responsabilidad	Aumento del TOC provoca un aumento de la responsabilidad asignada a la clase
Complejidad de implementación	Aumento del TOC provoca aumento de la complejidad de implementación de la clase
Reutilización	Aumento del TC 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:

PROPUESTA DE SOLUCIÓN.

Tabla 5: Criterios de evaluación para la métrica TOC

Atributos	Categoría	Criterio
Responsabilidad	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
Complejidad de implementación	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
Reutilización	Baja	$> 2 \times$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	\leq Promedio

Resultados obtenidos al aplicar la métrica TOC:

Responsabilidad

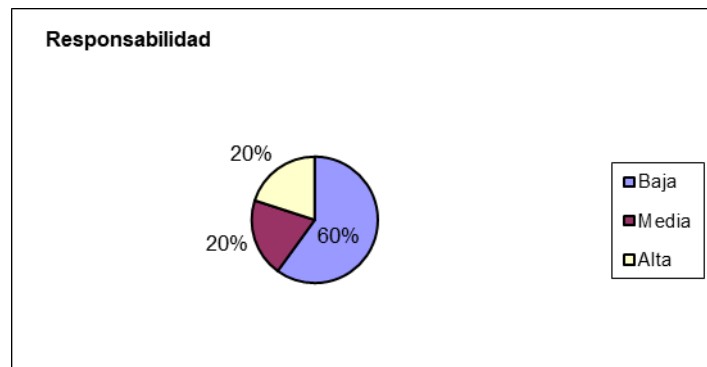


Figura 6: Resultado de la evaluación métrica para el atributo responsabilidad

Complejidad de implementación:

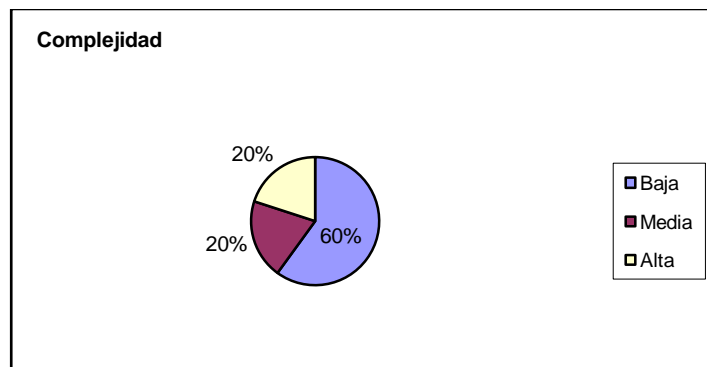


Figura 7: Resultado de la evaluación métrica para el atributo complejidad

Reutilización:

PROPUESTA DE SOLUCIÓN.

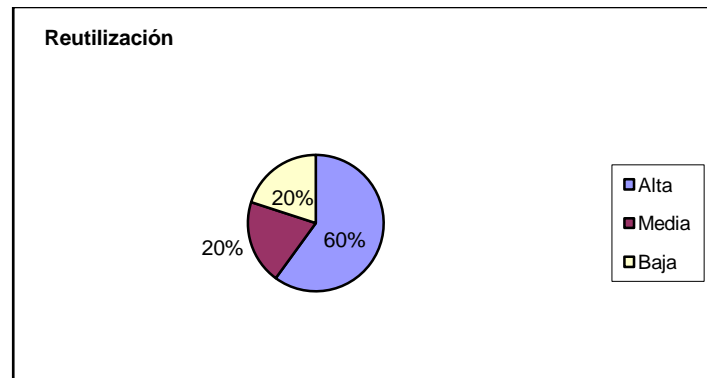


Figura 8: Resultado de la evaluación métrica para el atributo reutilización

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

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(41).

Tabla 6: Atributos de calidad evaluados por la métrica RC

Atributo de calidad	Modo en que los afecta
Acoplamiento	Aumento del RC provoca aumento del acoplamiento de la clase
Complejidad de mantenimiento	Aumento del RC provoca aumento de la complejidad del mantenimiento de la clase
Cantidad de pruebas	Aumento del RC provoca aumento de la cantidad de pruebas de unidad necesarias para probar una clase

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

Tabla 7: Criterios de evaluación para la métrica RC

Atributos	Categoría	Criterio
Acoplamiento	Ninguna	0
	Bajo	1
	Medio	2

PROPUESTA DE SOLUCIÓN.

	Alto	>2
Complejidad de mantenimiento	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
Reutilización	Baja	$> 2 \times$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	\leq Promedio
Cantidad de pruebas	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio

Resultados obtenidos al aplicar las métricas RC:

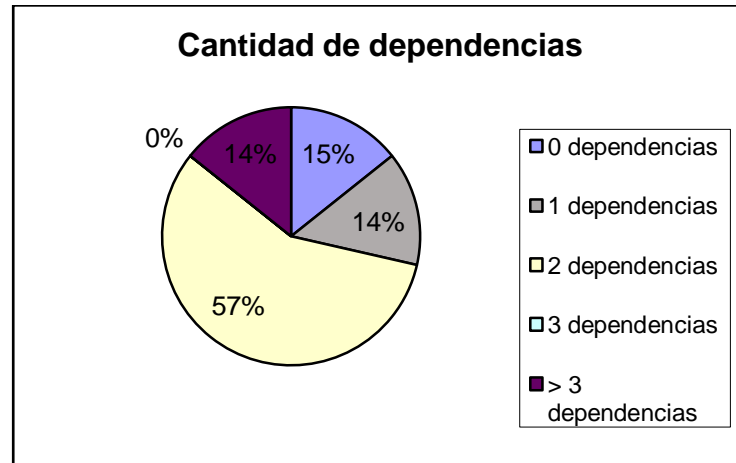


Figura 9: Cantidad de dependencias entre las clases

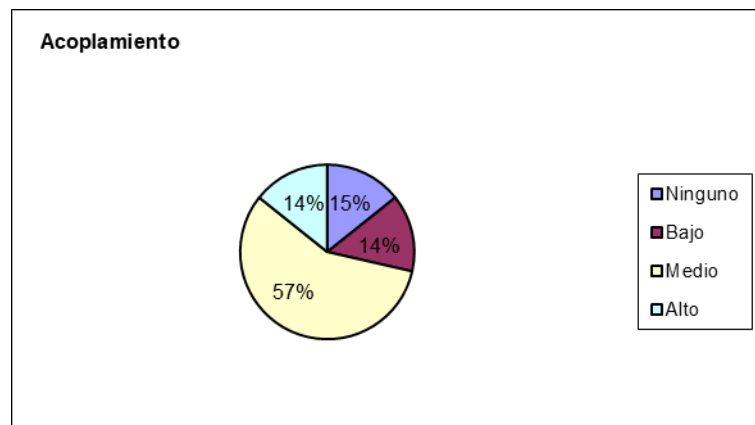


Figura 10: Resultado de la evaluación de la métrica para el atributo acoplamiento

PROPUESTA DE SOLUCIÓN.

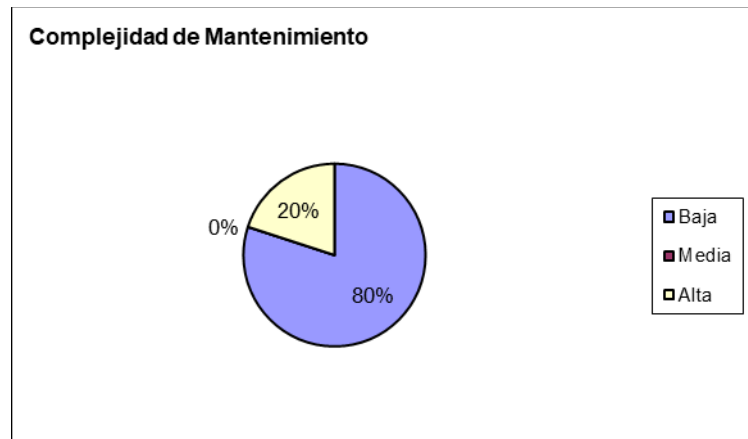


Figura 11: Resultado de la evaluación de la métrica para el atributo complejidad de mantenimiento

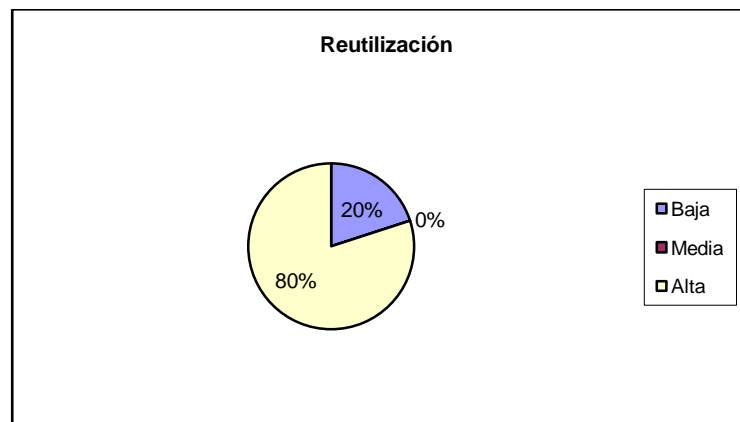


Figura 12: Resultado de la evaluación de la métrica para el atributo reutilización

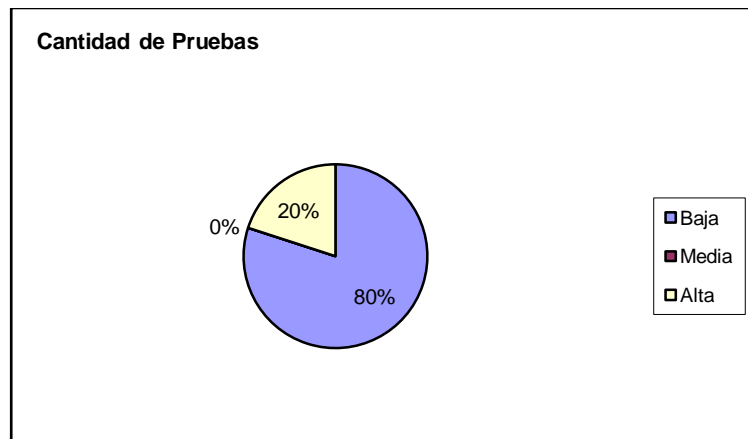


Figura 13: Resultado de la evaluación de la métrica para el atributo cantidad de pruebas

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

2.5 Conclusiones parciales

Se generaron los productos de trabajo correspondientes a las disciplinas modelado de negocio, requisitos, análisis y diseño, que permitieron realizar el diseño de la propuesta de solución. Mediante las técnicas de levantamiento de requisitos se identificaron 57 RF y 17 RnF los cuales fueron descritos y prototipados obteniendo como resultado la validación de estos por parte del cliente. Además, se realizó el diseño de la propuesta de solución, validándolo mediante la métrica TOC donde se evidencia una alta reutilización de las clases, y la métrica RC en la que se obtuvo un valor medio de reutilización y acoplamiento.

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3 Capítulo 3: Implementación y validación de la solución propuesta

Introducción

En el presente capítulo se describen los estándares de código fundamentales usados en la implementación de la solución propuesta. Además, se modela el diagrama de componentes para mostrar la organización y las dependencias lógicas que existen entre los ficheros que contienen código fuente. Una vez culminado el proceso de implementación se procede a aplicar un conjunto de pruebas que permiten validar el componente. Se aplica la técnica se ladov para validar la

3.1 Implementación

En la disciplina implementación se construye el sistema a partir de los resultados obtenidos por el análisis y diseño, su objetivo principal es la obtención del código fuente del componente a realizar, en esta etapa se generan una serie de artefactos. Para modelar los detalles de esta etapa UML define varios diagramas entre estos el diagrama de componentes que describe cómo los elementos del modelo del diseño se implementan en términos de componente, describe la organización de estos componentes y cómo dependen unos de otros.

Antes de comenzar la implementación es fundamental definir el estándar de codificación a utilizar con el fin de especificar la nomenclatura a emplear para el desarrollo del software.

3.1.1 Estándar de codificación

Un estándar de codificación son reglas que se siguen para la escritura del código fuente. Estos permiten que otros programadores puedan identificar las variables, las funciones o métodos al leer los códigos de otras personas. Se definen estándares de codificación porque un estilo de programación homogéneo en un proyecto para un lenguaje de programación permite que todos los participantes lo puedan entender en menos tiempo y que cualquier persona que se desempeñe como codificador de dicho lenguaje pueda interpretar de manera eficiente. (35)

A continuación, se exponen más detalladamente, los estándares de codificación seguidos en el desarrollo del componente:

Identación

Para la identificación se utiliza sólo espacios.

Nombres de clases

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

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

```
class ExtraPlan(models.Model):
    _name = 'tcx_economia.extra_plan'
    _description = 'Listado de extraplanes de un plan de importación'

    plan_id = fields.Many2one('tcx_economia.plan_importacion', 'Plan', required=False)
```

Figura 14: Estándar de codificación Identación y Nombre de clases

Nombres de métodos

Los nombres de los métodos serán en minúsculas y en palabras compuesta se separarán a través de guión bajo.

```
@api.onchange('objeto_id')
def onchange_objeto_id(self):
    if self.plan_id:
        return {'domain': {'objeto_id': [
            ('denominacion', 'not in', self.plan_id.desglose_ids.mapped('objeto_id.denominacion'))]}}
```

Figura 15: Estándar de codificación Nombre de métodos

Nomenclatura de archivos

Los archivos del modelo deben tener nombres cortos formado por letras minúsculas, se puede utilizar también guiones bajos.

Las vistas serán nombradas de la siguiente forma: <nombre_modelo>_view.

Directorios

Este estándar permite la organización de los directorios por los que está integrado el módulo los cuales son:

- controllers/ contiene los controladores de rutas.
- data/ contiene los datos XML.
- models/ contiene los modelos.
- static/ contiene los elementos web, separados en css/,img/, descripción/,
- views/ contiene las vistas.

3.1.2 Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos. Este diagrama proporciona una visión física de la construcción del sistema de información, pueden incluir paquetes que permiten organizar la construcción del sistema de información en subsistemas y que recogen aspectos prácticos relacionados con la secuencia de compilación entre componentes y la agrupación de elementos en librerías.

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

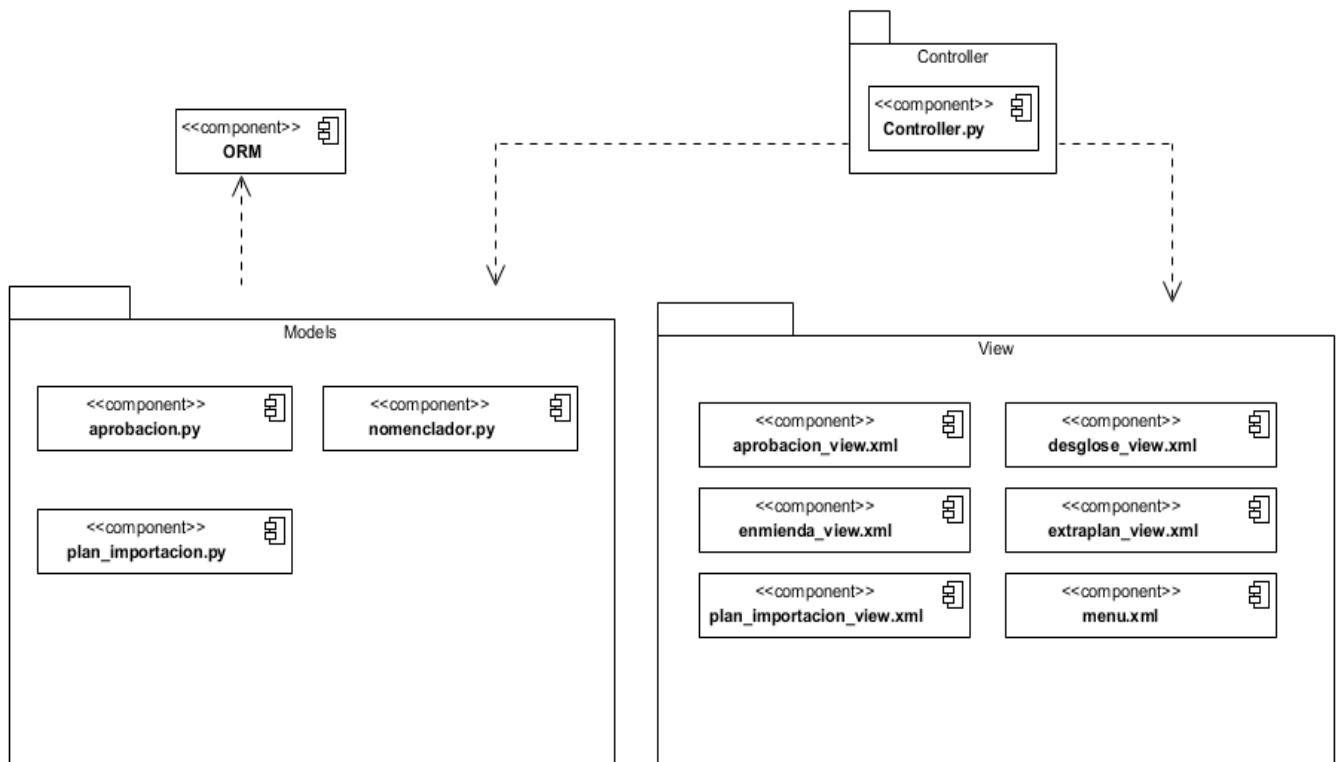


Figura 16: Diagrama de componentes

3.2 Interfaz de usuario funcional

A continuación, se muestran interfaces del componente que responden a los requisitos funcionales ListarPlan de Importación (Figura 17) y Crear Aprobación (Figura 18).

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

<input type="checkbox"/>	Código PI	Cliente	Fecha Emisión	Total	Inversiones	Capital de trabajo
<input type="checkbox"/>	20200001	UCM Cayo coco	21/08/2020	4.000,00	1.000,00	3.000,00
<input type="checkbox"/>	20200002	Delegacion varadero	27/08/2020	35.255,00	688,00	34.567,00

Figura 17: Interfaz de usuario funcional del requisito Listar Plan de Importación

Información de la aprobación Observaciones

No. Aprobación Organismo

Cliente No. R/S

Tipo financiamiento Fecha emisión

Financiero Fecha vencimiento

Tipo aprobación Tipo de factura

Tipo moneda Objeto

Aprobado Solicitado Contratado Pagado
Sin solicitar Sin contratar Sin pagar

PLAN DE IMPORTACIÓN

Código PI	Inversione...	Capital de ...
Agregar línea		

Figura 18: Interfaz de usuario funcional del requisito Crear Aprobación

3.3 Pruebas de Software

Las pruebas de software son realizadas con el objetivo de encontrar errores, verificando de esta manera la calidad del producto (37). En la realización de estas pruebas es importante comprobar la cobertura de los requisitos, dado que su incumplimiento puede comprometer la aceptación del sistema.

En la disciplina de prueba definida por la metodología AUP-UCI se verifica el resultado de la implementación realizada para la construcción del sistema. Entre las pruebas que propone la

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

metodología de desarrollo AUP-UCI se encuentran pruebas internas, pruebas de liberación y pruebas de aceptación(14). Las pruebas aplicadas a la propuesta de solución fueron las pruebas internas.

- **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 liberada. Para esto se utilizarán las pruebas de caja blanca y las pruebas de caja negra, descritas en los siguientes acápite.

3.3.1 Pruebas de caja blanca

La prueba de caja blanca es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que: 1) garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, 2) revisen todas las decisiones lógicas en sus lados verdadero y falso, 3) ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y 4) revisen estructuras de datos internas para garantizar su validez(37).

Técnica de ruta Básica

El método de ruta básica permite al diseñador de casos de prueba derivar una medida de complejidad lógica de un diseño de procedimiento y usar esta medida como guía para definir un conjunto básico de rutas de ejecución. Los casos de prueba derivados para revisar el conjunto básico tienen garantía para ejecutar todo enunciado en el programa, al menos una vez durante la prueba(37).

Antes de considerar el método de ruta básica, debe introducirse una notación simple para la representación del flujo de control, llamado gráfico de flujo. El gráfico de flujo muestra el flujo de control lógico que usa la notación a base de círculos donde cada uno representa enunciados del código fuente.

Luego, para lograr preparar los casos de pruebas de forma efectiva es necesario conocer el número de pruebas necesarias para asegurar que todos los enunciados se ejecutaron al menos una vez, para esto se calcula la complejidad ciclomática. La complejidad ciclomática es una medición de software que proporciona una evaluación cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba de la ruta básica, el valor calculado por la complejidad ciclomática define el número de rutas independientes del conjunto básico de un programa y le brinda una cota superior para el número de pruebas que debe realizar a fin de asegurar que todos los enunciados se ejecutaron al menos una vez(37).

La siguiente figura muestra el código de la funcionalidad *def_onchange_enmienda*

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

```
@api.onchange('enmienda_ids')
def _onchange_enmienda(self):
    aumentar = 0 } 1
    disminuir = 0 }
    for i in self.enmienda_ids: } 2
        aumentar += i.aumento } 3
        disminuir += i.disminucion }
        if i.mv_fecha_vencimiento: } 4
            self.fecha_vencimiento = i.fecha_vencimiento } 5
    self.importe_aprobado = self.importe_aprobado + aumentar - disminuir 6
```

Figura 19: Código de la funcionalidad def_onchange_enmienda

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

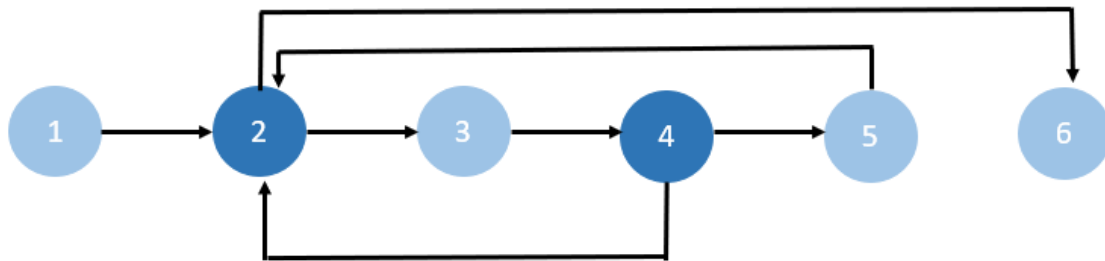


Figura20: Grafo de flujo

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

La complejidad se calcula en una de tres formas:

1. El número de regiones del gráfico de flujo corresponde a la complejidad ciclomática.
2. La complejidad ciclomática $V(G)$ para un gráfico de flujo G se define como $V(G) = E - N + 2$ donde E es el número de aristas del gráfico de flujo y N el número de nodos del gráfico de flujo.
3. La complejidad ciclomática $V(G)$ para un gráfico de flujo G también se define como $V(G) = P + 1$, donde P es el número de nodos predicado contenidos en el gráfico de flujo G .

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

- | | | |
|-----------------------|-------------------|---------------|
| 1. $V(G) = E - N + 2$ | 2. $V(G) = P + 1$ | 3. $V(G) = R$ |
| $V(G) = 7 - 6 + 2$ | $V(G) = 2 + 1$ | $V(G) = 3$ |
| $V(G) = 3$ | $V(G) = 3$ | |

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

Camino básico 1: 126

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Camino básico 2: 123426

Camino básico 3: 1234526

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

Tabla 8: Caso de prueba para el camino 1

Descripción	Analiza los productos faltantes en la instancia del modelo actual para ser registrado en la demanda
Condición de ejecución	Que la aprobación presente cambios
Entrada	Aprobación
Resultado	Ningún Resultado

Tabla 9: Caso de prueba para el camino 2

Descripción	Analiza los productos faltantes en la instancia del modelo actual para ser registrado en la demanda
Condición de ejecución	Que la aprobación presente cambios
Entrada	Aprobación
Resultado	Se registra una enmienda con las modificaciones realizadas

Tabla 10: Caso de prueba para el camino 3

Descripción	Analiza los productos faltantes en la instancia del modelo actual para ser registrado en la demanda
Condición de ejecución	Que la aprobación presente cambios
Entrada	Aprobación
Resultado	Se registra una enmienda con las modificaciones realizadas

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.3.2 Pruebas de caja negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra intentan encontrar errores en las categorías siguientes(37):

- Funciones incorrectas o faltantes
- Errores de interfaz
- Errores en las estructuras de datos o en el acceso a bases de datos externas
- Errores de comportamiento o rendimiento
- Errores de inicialización y terminación

Técnica de Partición de equivalencia

La partición de equivalencia es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. El diseño de casos de prueba para la partición de equivalencia se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Al aplicar los lineamientos para la derivación de clases de equivalencia, pueden desarrollarse y ejecutarse los casos de prueba para cada ítem de datos del dominio de entrada. Los casos de prueba se seleccionan de modo que se revise a la vez el número más grande de atributos de una clase de equivalencia(37).

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Escenario	Descripción	No. Aprobación	Cliente	Tipo financiamiento	Financiero	Tipo aprobación	Tipo moneda	Organismo	No. R/S	Fecha emisión	Fecha vencimiento	Tipo Factura	Objeto	Respuesta del sistema	Flujo central
EC 1.1 Crear una aprobación introduciendo datos validos	El sistema debe permitir crear una aprobación en el sistema	V	V	V	V	V	V	V	V	V	V	V	V	El sistema registra una nueva aprobación	1. Se selecciona la opción de aprobación. 2. Se presiona el botón "Crear". 3. Se introducen los datos necesarios para crear una aprobación. 4. Se selecciona el botón "Guardar".
		202012345	Delegación varadero	TEC-COM	Tecnotex UTE	UTE	USD	FAR	1	20/7/2020	20/7/2023	Sin valor comercial	Inversiones		
EC 1.2 Crear una aprobación introduciendo datos inválidos	El sistema debe permitir crear una aprobación en el sistema	I	V	V	V	V	V	V	V	V	V	V	V	El sistema muestra un mensaje : "El campo	1. Se selecciona la opción de aprobación. 2. Se presiona el botón "Crear". 3. Se introducen los datos necesarios para crear una aprobación. 4. Se selecciona el botón "Guardar".
		abc	Delegación varadero	TEC-COM	Tecnotex UTE	UTE	USD	FAR	1	20/7/2020	20/7/2023	Sin valor comercial	Inversiones		
		V	V	V	V	V	V	V	I	V	V	V	V	El sistema muestra un mensaje : "El campo	
		202012345	Delegación varadero	TEC-COM	Tecnotex UTE	UTE	USD	FAR	as	20/7/2020	20/7/2023	Sin valor comercial	Inversiones		
		I	V	V	V	V	V	V	V	I	V	V	V	V	
as23	Delegación varadero	TEC-COM	Tecnotex UTE	UTE	USD	FAR	asd	20/7/2020	20/7/2023	Sin valor comercial	Inversiones				

Figura 21: Diseño del caso de prueba. Crear Aprobación

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

EC 1.3 Crear un plan de importación dejando campos vacíos	Se dejan campos vacíos al crear una aprobación	V	V	V	V	V	V	V	V	V	V	V	V	El sistema muestra los errores en una alerta indicando que contiene campos vacío	1. Se selecciona la opción de aprobación. 2. Se presiona el botón "Crear". 3. Se introducen los datos necesarios para crear una aprobación. 4. Se selecciona el botón "Guardar".
		(vacío)	Delegación varadero	TEC-COM	Tecnotex UTE	UTE	USD	FAR	1	20/7/2020	20/7/2023	Sin valor comercial	Inversiones		
		V	(vacío)	V	V	V	V	V	V	V	V	V	V		
		202012345	(vacío)	TEC-COM	Tecnotex UTE	UTE	USD	FAR	1	20/7/2020	20/7/2023	Sin valor comercial	Inversiones		
		V	V	I	V	V	V	V	V	V	V	V	V		
		202012345	Delegación varadero	(vacío)	Tecnotex UTE	UTE	USD	FAR	1	20/7/2020	20/7/2023	Sin valor comercial	Inversiones		
		V	V	V	I	V	V	V	V	V	V	V	V		
		202012345	Delegación varadero	TEC-COM	(vacío)	UTE	USD	FAR	1	20/7/2020	20/7/2023	Sin valor comercial	Inversiones		
		V	V	V	V	I	V	V	V	V	V	V	V		
		202012345	Delegación varadero	TEC-COM	Tecnotex UTE	(vacío)	USD	FAR	1	20/7/2020	20/7/2023	Sin valor comercial	Inversiones		
		V	V	V	V	V	I	V	V	V	V	V	V		
		202012345	Delegación varadero	TEC-COM	Tecnotex UTE	UTE	(vacío)	FAR	1	20/7/2020	20/7/2023	Sin valor comercial	Inversiones		
		V	V	V	V	V	V	I	V	V	V	V	V		
		202012345	Delegación	TEC-COM	Tecnotex UTE	UTE	USD	(vacío)	1	20/7/2020	20/7/2023	Sin valor	Inversiones		
		V	V	V	V	V	V	V	I	V	V	V	V		
202012345	Delegación	TEC-COM	Tecnotex UTE	UTE	USD	FAR	(vacío)	20/7/2020	20/7/2023	Sin valor	Inversiones				
V	V	V	V	V	V	V	V	V(N/A)	V	V	V				
202012345	Delegación	TEC-COM	Tecnotex UTE	UTE	USD	FAR	1		20/7/2023	Sin valor	Inversiones				
V	V	V	V	V	V	V	V	V	V	I	V				
202012345	Delegación	TEC-COM	Tecnotex UTE	UTE	USD	FAR	1	20/7/2020	(vacío)	Sin valor	Inversiones				
V	V	V	V	V	V	V	V	V	V	V	I				
202012345	Delegación	TEC-COM	Tecnotex UTE	UTE	USD	FAR	1	20/7/2020	20/7/2023	(vacío)	Inversiones				
V	V	V	V	V	V	V	V	V	V	V	I				
202012345	Delegación	TEC-COM	Tecnotex UTE	UTE	USD	FAR	1	20/7/2020	20/7/2023	Sin valor	(vacío)				
V	V	V	V	V	V	V	V	V	V	V	I				
EC 1.4 Cancelar	Se cancela la operación de crear un producto	V(N/A)	V(N/A)	V(N/A)	V(N/A)	V(N/A)	V(N/A)	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 ventana.	1. Se selecciona la opción de aprobación. 2. Se presiona el botón "Crear". 3. Se introducen los datos necesarios para crear una aprobación. 4. Se

Figura 22: Diseño del caso de prueba. Crear Aprobación

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Con la aplicación de la prueba de caja negra y su método Partición de Equivalencias, se obtuvo, luego de aplicar los Diseños de Casos de Pruebas resultados satisfactorios, demostrando el correcto funcionamiento de todas las funcionalidades implementadas. Se aplicaron las pruebas internas, definida como disciplina de la metodología AUP-UCI, que guía la presente investigación. El método de prueba aplicado demostró resultados satisfactorios desde el punto de vista funcional. Las no conformidades encontradas fueron analizadas y corregidas en el tiempo establecido, logrando un correcto comportamiento ante diferentes situaciones. A continuación, se muestra el número de no conformidades (NC) detectadas en cada iteración de prueba realizada.

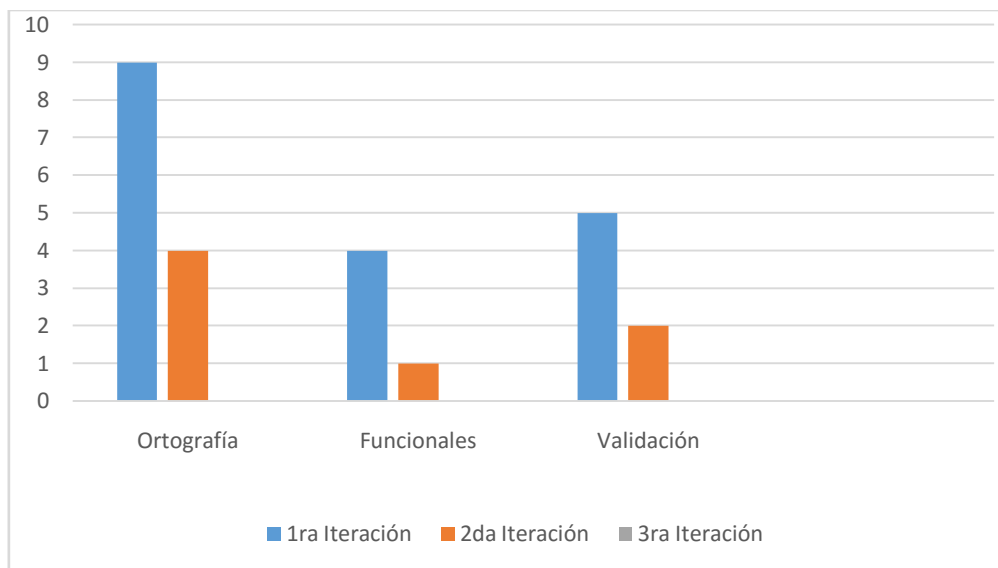


Figura 1: No conformidades detectadas

En la primera iteración fueron detectadas un total de 18 NC: 9 de ortografía, 4 funcionales y 5 de validación. En la segunda iteración fueron detectadas 7NC: 4 de ortografía, 1 funcional y 2 de validación. En la tercera iteración no fueron detectados errores de ningún tipo, para un total de 0 NC, quedando de esta forma validado el sistema.

3.4 Aplicación de la técnica de ladov

Para la validación de la investigación se utilizó la Técnica de ladov. La Técnica de ladov constituye una vía indirecta para el estudio de la satisfacción, pues los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov" (42).

Tabla 11: Cuadro lógico de ladov

7. ¿Le satisface este sistema para su utilización en el área de Economía?	1. ¿Consideraría factible la gestión del plan de importación sin contar con una herramienta que centralice la información relacionada con el proceso y esté disponible
---	--

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

	desde cualquier área que pueda necesitar dicha información?								
	No			No sé			Sí		
	3. ¿Utilizaría usted el componente desarrollado para gestionar el plan de importación, las aprobaciones y sus respectivas modificaciones (extraplan y enmienda)?								
	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 mas 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

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

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definido
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 12: Niveles de satisfacción de la técnica de ladov

+1	Máximo de satisfacción
-----------	------------------------

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

0.5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0.5	Mas satisfecho que insatisfecho
-1	Máxima insatisfacción

La satisfacción grupal se calcula de la siguiente forma:

$$ISG = \frac{A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)}{N}$$

En esta fórmula A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 ó 6; 4; 5 respectivamente y 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:

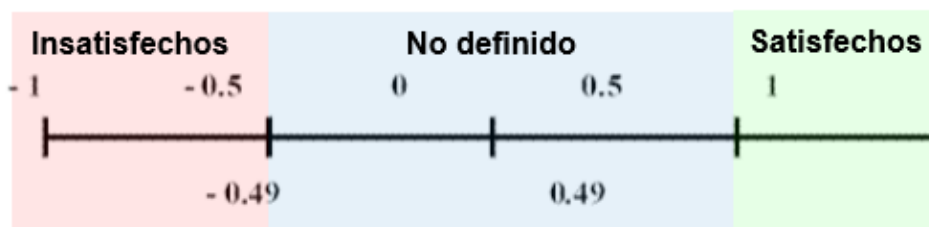


Figura23: Rango de valores para la satisfacción grupal

Resultados de la aplicación de la técnica de ladov:

Para la realización de esta técnica se seleccionaron especialistas del proyecto Tecnotex a los cuales se les aplicó el cuestionario para determinar el grado de satisfacción con la propuesta de solución. En la siguiente tabla se muestran los resultados obtenidos al aplicar el Cuadro lógico de ladov.

Totalde cuestionados	9	Escala
Máximo de satisfacción	5	A
Mas satisfecho que insatisfecho	3	B

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

No definida	1	C
Mas insatisfecho que satisfecho	0	D
Clara insatisfacción	0	E
Contradictoria	0	C

$$ISG = \frac{5(+1) + 3(+0,5) + 1(0) + 0(-0,5) + 0(-1)}{9} = 0.72$$

La valoración de los especialistas del proyecto sobre el sistema confirmó un alto Índice de Satisfacción Grupal (ISG = 0.72), lo cual indica que, según el rango de valores de estos, es un elevado valor de satisfacción de los encuestados respecto a dicho sistema, además dicha información se ratifica en los criterios emitidos en la encuesta, la cual puede ser consultada en el **Anexo 3**

3.5 Beneficios del sistema desarrollado

El desarrollo del Componente para la gestión del plan de importación permite la gestión de los subprocesos Gestión del Plan de importación, extraplanes, aprobaciones y enmiendas. Estos son ingresados en el sistema por el área de Economía para la planificación de las importaciones a realizar en el año y el presupuesto aprobado para estas. Posteriormente esta información es utilizada para controlar las solicitudes de importación, contratación y pago controlando en todas las etapas los porcentos de ejecución del PI y controlando que el monto solicitado en los pedidos no sobrepase el monto aprobado. Entre los principales beneficios que aporta el desarrollo del componente se encuentran:

- Es un componente web que permite la disponibilidad desde todas las áreas de la empresa. Lo que contribuye a un manejo centralizado de los datos referentes al PI y las aprobaciones
- Disminuye los errores de cálculo en los porcentos de ejecución del PI
- Facilita el chequeo de consistencia de la información a través de métodos que validan diferentes campos como la creación de un solo PI en el período de un año y los estados del presupuesto aprobado.
- Disminuye los errores introducidos en la presentación de un plan de importación y sus modificaciones (extraplanes)
- Provee un conjunto de reportes que muestran la información acerca del PI de la empresa, el estado de la aprobación por códigos y el estado del PI por clientes.

IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.6 Conclusiones parciales

En este capítulo se definen temas relacionados con los procesos de implementación y prueba de la aplicación que posibilitan la implementación de los requisitos funcionales en correspondencia con las descripciones realizadas y la verificación del correcto funcionamiento de los mismos. La utilización de los estándares de codificación permitió organizar el código para un mejor entendimiento de todas las clases y métodos desarrollados. Se ejecutaron las pruebas internas, para el control de la calidad del sistema, arrojando resultados satisfactorios que permitieron la corrección en tiempo y forma de las no conformidades detectadas. Se validó la investigación con la aplicación de la técnica de la dov obteniendo resultados satisfactorios.

CONCLUSIONES GENERALES

- Se analizaron las características y limitaciones de las soluciones existentes, determinándose la necesidad de desarrollar un componente para la gestión del plan de importación, debido a que los existentes no reunían todas las funcionalidades que se requieren. Sin embargo, se comprobó la factibilidad de la plataforma Odoó para el desarrollo de un nuevo componente.
- Los artefactos generados correspondientes a las disciplinas Modelado de negocio, Requisitos y Análisis y diseño guiaron el análisis de la solución y diseño de la arquitectura. Además, proporcionaron los cimientos para realizar la disciplina de implementación.
- Se implementó el componente para la gestión del plan de importación de la empresa Tecnotex. La utilización de estándares de codificación permitió obtener un código estructurado, logrando un lenguaje común y comprensible. La implementación del sistema permitió cumplir con las funcionalidades identificadas, utilizando para ello lenguajes, notaciones, herramientas y tecnologías distribuidas bajo licencias de software libre en correspondencia con las políticas de la Universidad y del país.
- La aplicación de pruebas permitió validar el correcto funcionamiento del componente, comprobando los requerimientos y garantizando la calidad del software desarrollado
- La aplicación de la técnica de *lado v* arrojó que los usuarios se sienten satisfechos con el componente para la gestión del plan de importación.

RECOMENDACIONES

Para dar continuidad a la presente investigación se recomienda:

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

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

1. *Una mirada de los impactos negativos del escenario internacional a las importaciones de las economías pequeñas y vulnerables*. **Ramirez Jones, Ivette**. s.l. : CEPEC, 2015, Vol. 73. ISSN.
2. **Koontz, Harlod**. *Administracion* . 12. McGrawHill : s.n., 2016.
3. **Acción contra el hambre** .*Exportación e importación en la empresa*. s.l. : Fondo Social Europeo dentro del Programa Operativo contra la Discriminación, 2012.
4. **Cantos Suarez, nd y Castro Paredes, nd**. Plan de Importación y Logística de comercialización del insecticida Neonicotinoide para el control de plagas en los cultivos en la provincia del Guayas. [En línea] septiembre de 2012. [Citado el: 14 de 10 de 2019.] <https://dspace.ups.edu.ec/bitstream/123456789/3125/1/UPS-GT000325.pdf>.
5. **MINCEX**. Ministerio de comercio exterior. [En línea] 2019. [Citado el: 2 de 12 de 2019.] <https://www.mincex.gob.cu/files/Reglamento-General-sobre-la-Actividad-de-Importacion-y-Exportacion-Res-50.pdf>.
6. **Mistral Import**. Mistral Caribe Holding S.A. - Mistral Import. *Mistral Import*. [En línea] 2019. [Citado el: 14 de 10 de 2019.] http://www.mistralcaribe.com/index.php?option=com_content&view=article&id=124&Itemid=49.
7. **Orbit Consulting**. Orbit Consulting. [En línea] 2019. [Citado el: 14 de 10 de 2019.] <https://www.orbit.es/soluciones-y-servicios/soluciones-de-negocio/crm->.
8. **Figueroa Garcia, Anabel**. *sistema de información sobre las ventas de los productos de la cartera de Desoft*. La Habana : s.n., 2016.
9. **Six Degrees Planning**. Six Degrees Planning. [En línea] 2020. [Citado el: 08 de 02 de 2020.] <http://sixdegreesplanning.com/>.
10. **Sistrade**. Sistrade. [En línea] 2020. [Citado el: 08 de 01 de 2020.] <https://www.sistrade.com/es/soluciones/gestion-comercial-presupuestos/presupuestos/>.

REFERENCIAS BIBLIOGRÁFICAS

11. **Amso consulting.** Amso consulting. [En línea] 2019. [Citado el: 08 de 01 de 2020.] <https://amsogroup.com/corporate-performance-management-cpm/presupuesto-y-planificacion/>.
12. **Areba, J.B.D.** *Metodología del análisis estructurado de sistemas*. Madrid : s.n., 2001.
13. **Laboratorio Nacional de Calidad del Software de INTECO.** *INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA*. 2009.
14. **Sanchez Rodriguez, Tamara.** *Metodología de desarrollo para la Actividad productiva de la UCI v1.1*. La Habana : s.n., 2015.
15. *A comparative analysis between BPMN and SPEM modeling standards in the software processes context*. **Portela, Carlos, y otros.** s.l. : Journal of Software engineering and applications, 2012, Vol. 5. 5.
16. **Rioseco Reinoso, Cristian Andres.** *Modelado y mejora de procesos de software*. s.l. : Pontificia Universidad Católica de Valparaíso , 2012.
17. **Booch, Grady.** *Object-Oriented Analysis and Design with Applications*. s.l. : Addison-Wesley, 2004.
18. **Mendoza Peña, Dayana y Baquero Hernandez, Lionel Rodolfo.** *Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso*. Habana : s.n., 2016.
19. **ISO 8879.** *Information processing— Text and Office Systems — Standard Generalized Markup Language*. 1986.
20. **Sosa Lopez, Daylin.** *Sistema para el control del uso de los softwares educativos*. s.l. : EUMED, 2009.
21. **Perez Porto, Julian y Gardey, Ana.** Definición . [En línea] 2013. [Citado el: 23 de 11 de 2019.] <https://definicion.de/xml>.
22. **Oliphant, Travis E.** *Python for Scientific Computing Publisher* . s.l. : IEEE, 2007.
23. **Summerfield, Mark.** *Rapid GUI Programming with Python and Qt*.

REFERENCIAS BIBLIOGRÁFICAS

24. **Python Software Foundation.** Python a programming language changes the world. [En línea] 2012. [Citado el: 23 de 11 de 2019.] <https://brochure.getpython.info/media/releases/prereleases/psf-python-brochure-vol-1-final-content-preview>.
25. **Postgre SQL.** Postgre SQL. [En línea] 2019. [Citado el: 23 de 11 de 2019.] <https://www.postgresql.org/about/>.
26. **Reis, Daniel.** *Odoo Development Essentials*. Birminham : Packt publishing, 2015.
27. **Olivan Crego, Xavier y Sanchez Castello, JM.** *Estudio para la implantacion de un RP en una PyME*. Universitat Autonoa de Barcelona. Barcelona : s.n., 2015.
28. **Quintana Rondon, Yoandri, Camejo Dominguez, Lianet y Diaz Berenguer, Abel.** 15, s.l. : *Revista Informatica Educativa y Medios AudioVisuales*, 2011, Vol. 8.
29. **adrearrrs.** *Hipertextual*. [En línea] 2014. [Citado el: 14 de 10 de 2019.] <https://hipertextual.com/archivo/2014/06/pycharm-ide-python>.
30. **JetBrains.** *JetBrains*. [En línea] 2019. [Citado el: 24 de 10 de 2019.] <https://jetbrains.com/pycharm>.
31. **PgAdmin.** *PgAdmin*. [En línea] 2019. [Citado el: 02 de 12 de 2019.] <https://www.pgadmin.org>.
32. **Losavio, Francisca y Estevez, Yuly.** *Modelado del Negocio como Técnica Centrada en la Calidad del Software para el Análisis del Dominio del Aprendizaje Electrónico*. Caracas, Venezuela : IV Simposio Científico y Tecnológico en Computación / SCTC 2016 , 2016.
33. **Summerville, Ian.** *Ingeniería de software*. 9. Mexico : Pearson Education, 2011.
34. —. *Ingeniería de Software* . 7. madrid : Pearson, 2005.
35. **Pressman, Roger.** *Ingeniería de software: un enfoque práctico*. 5. s.l. : McGrawHill, 2002.
36. **Constanza Pavas, Adriana, y otros.** *Diseño de salida del sistema*. s.l. : Universidad del Quindío, 2016.

REFERENCIAS BIBLIOGRÁFICAS

37. **Pressman, Roger.** *Ingeniería de software. Un enfoque práctico . 7ma.* New York : McGrawHill, 2010.
38. *Patrones de Diseño .* **Visconti, Marcello y Astudillo, Hernan.** Colombia : Universidad Técnica Federico Santa María , 2015.
39. **Vazquez Escudero, Pedro Jesus, Moreno Garcia, Maria N. y Garcia Peñalvo, Francisco J.** *Metrica Orientada a Objetos .* Salamanca : s.n., 2001.
40. **Lorenz, M y Kidd, J.** *Object Oriented Metrics. 1.* New Jersey : Prentice hall, 1994.
41. **Rogriguez Rodriguez, Carlos Rafael.** *Herramienta para aplicar métricas al Diagrama de Clases del Diseño Orientado a Objetos. .* Matanzas : Revista Avanzada científica, 2012.
42. **Lopez Rodriguez, Alejandro.** *La técnica de ladov. Una aplicación para el estudio de la satisfacción de los alumnos por las clases de educación física.* 2002.
43. **Alvarez de Zayas, Carlos.** *Metodología de la investigación científica.* Santiago de Cuba : s.n., 1995.
44. **Nocedo, Irma.** *Metodología de la investigación educacional.* La Habana : Felix Varela, 2001.
45. **Pencil Project.** *Pencil Project.* [En línea] 2019. [Citado el: 24 de 10 de 2019.] <https://pencl.evolus.vn>.
46. *Introducción a la ingeniería de software.* **Gracia Peñalvo, Francisco Jose y Garcia Holgado, Alicia.** Salamanca : GRIAL Research group University of Salamanca, 2018.
47. *Mantenimiento Perfectivo en un Gestor de Contenidos: Una Experiencia para la Especificación de Consultas Dinámicas.* **Godoy, Maria V, y otros.** ISSN 2314-2642, s.l. : Revista Latinoamericana de Ingenieria de software, 2014, Vol. 2(5), págs. 310-312. Referenciado por: Gomez Codutti, Ana; Mariño, Sonia I en *Una propuesta de proceso software para implementar una revista digital.*
48. **Real Academia Española.** RAE. [En línea] 2019. [Citado el: 04 de 10 de 2019.] <https://dej.rae.es/lema/importaci%C3%B3n>.

REFERENCIAS BIBLIOGRÁFICAS

49. **Thompson Reuters.** *Import Compliance and Management Software from ONESOURCE Global Trade.* Thompson Reuters. [Online] 2019. [Cited: 10 14, 2019.] <https://tax.thomsonreuters.com/en/onesource/global-trade-management/import-compliance>.
50. **Yéminus.** *Liquidación de Importaciones – Yéminus.* Yéminus. [En línea] 2019. [Citado el: 14 de 10 de 2019.] <https://www.yeminus.com/importaciones/>.
51. **Fundación contra el hambre.** *Exportación e importación en la empresa. s.l. : Fondo Social Europeo dentro del Programa Operativo contra la Discriminación.*
52. **Real Academia Española.** RAE. [En línea] 2019. [Citado el: 04 de 10 de 2019.] <https://dej.rae.es/lema/importaci%C3%B3n>.
53. **GONZÁLEZ NAVARRO, DANIELA.** *EXPRESIVIDAD DE BPMN 2.0 DESDE LA PERSPECTIVA DEL PROBLEMA DE SECUENCIAMIENTO.* Chile : s.n., 2018.

BIBLIOGRAFÍA

1. *Una mirada de los impactos negativos del escenario internacional a las importaciones de las economías pequeñas y vulnerables*. **Ramirez Jones, Ivette**. s.l. : CEPEC, 2015, Vol. 73. ISSN.
2. **Koontz, Harlod**. *Administracion* . 12. McGrawHill : s.n., 2016.
3. **Acción contra el hambre** . *Exportación e importación en la empresa*. s.l. : Fondo Social Europeo dentro del Programa Operativo contra la Discriminación, 2012.
4. **Cantos Suarez, nd y Castro Paredes, nd**. Plan de Importación y Logística de comercialización del insecticida Neonicotinoide para el control de plagas en los cultivos en la provincia del Guayas. [En línea] septiembre de 2012. [Citado el: 14 de 10 de 2019.] <https://dspace.ups.edu.ec/bitstream/123456789/3125/1/UPS-GT000325.pdf>.
5. **MINCEX**. Ministerio de comercio exterior. [En línea] 2019. [Citado el: 2 de 12 de 2019.] <https://www.mincex.gob.cu/files/Reglamento-General-sobre-la-Actividad-de-Importacion-y-Exportacion-Res-50.pdf>.
6. **Mistral Import**. Mistral Caribe Holding S.A. - Mistral Import. *Mistral Import*. [En línea] 2019. [Citado el: 14 de 10 de 2019.] http://www.mistralcaribe.com/index.php?option=com_content&view=article&id=124&Itemid=49.
7. **Orbit Consulting**. Orbit Consulting. [En línea] 2019. [Citado el: 14 de 10 de 2019.] <https://www.orbit.es/soluciones-y-servicios/soluciones-de-negocio/crm->.
8. **Figueroa Garcia, Anabel**. *Sistema de información sobre las ventas de los productos de la cartera de Desoft*. La Habana : s.n., 2016.
9. **Six Degrees Planning**. Six Degrees Planning. [En línea] 2020. [Citado el: 08 de 02 de 2020.] <http://sixdegreesplanning.com/>.
10. **Sistrade**. Sistrade. [En línea] 2020. [Citado el: 08 de 01 de 2020.] <https://www.sistrade.com/es/soluciones/gestion-comercial-presupuestos/presupuestos/>.

BIBLIOGRAFÍA

11. **Amso consulting.** Amso consulting. [En línea] 2019. [Citado el: 08 de 01 de 2020.] <https://amsogroup.com/corporate-performance-management-cpm/presupuesto-y-planificacion/>.
12. **Areba, J.B.D.** *Metodología del análisis estructurado de sistemas*. Madrid : s.n., 2001.
13. **Laboratorio Nacional de Calidad del Software de INTECO.** *INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA*. 2009.
14. **Sanchez Rodriguez, Tamara.** *Metodología de desarrollo para la Actividad productiva de la UCI v1.1*. La Habana : s.n., 2015.
15. *A comparative analysis between BPMN and SPEM modeling standards in the software processes context*. **Portela, Carlos, y otros.** s.l. : Journal of Software engineering and applications, 2012, Vol. 5. 5.
16. **Rioseco Reinoso, Cristian Andres.** *Modelado y mejora de procesos de software*. s.l. : Pontificia Universidad Católica de Valparaíso , 2012.
17. **Booch, Grady.** *Object-Oriented Analysis and Design with Applications*. s.l. : Addison-Wesley, 2004.
18. **Mendoza Peña, Dayana y Baquero Hernandez, Lionel Rodolfo.** *Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso*. Habana : s.n., 2016.
19. **ISO 8879.** *Information processing— Text and Office Systems — Standard Generalized Markup Language*. 1986.
20. **Sosa Lopez, Daylin.** *Sistema para el control del uso de los softwares educativos*. s.l. : EUMED, 2009.
21. **Perez Porto, Julian y Gardey, Ana.** Definicion . [En línea] 2013. [Citado el: 23 de 11 de 2019.] <https://definicion.de/xml>.
22. **Oliphant, Travis E.** *Python for Scientific Computing Publisher* . s.l. : IEEE, 2007.
23. **Summerfield, Mark.** *Rapid GUI Programming with Python and Qt*.

BIBLIOGRAFÍA

24. **Python Software Foundation.** Python a programming language changes the world. [En línea] 2012. [Citado el: 23 de 11 de 2019.] <https://brochure.getpython.info/media/releases/prerelases/psf-python-brochure-vol-1-final-content-preview>.
25. **PostgreSQL.** PostgreSQL. [En línea] 2019. [Citado el: 23 de 11 de 2019.] <https://www.postgresql.org/about/>.
26. **Reis, Daniel.** *Odoo Development Essentials*. Birminham : Packt publishing, 2015.
27. **Olivan Crego, Xavier y Sanchez Castello, JM.** *Estudio para la implantación de un RP en una PyME*. Universitat Autonoa de Barcelona. Barcelona : s.n., 2015.
28. **Quintana Rondon, Yoandri, Camejo Dominguz, Lianet y Diaz Berenguer, Abel.** 15, s.l. : *Revista Informatica Educativa y Medios AudioVisuales*, 2011, Vol. 8.
29. **adrearrrs.** *Hipertextual*. [En línea] 2014. [Citado el: 14 de 10 de 2019.] <https://hipertextual.com/archivo/2014/06/pycharm-ide-python>.
30. **JetBrains.** *JetBrains*. [En línea] 2019. [Citado el: 24 de 10 de 2019.] <https://jetbrains.com/pycharm>.
31. **PgAdmin.** *PgAdmin*. [En línea] 2019. [Citado el: 02 de 12 de 2019.] <https://www.pgadmin.org>.
32. **Losavio, Francisca y Estevez, Yuly.** *Modelado del Negocio como Técnica Centrada en la Calidad del Software para el Análisis del Dominio del Aprendizaje Electrónico*. Caracas, Venezuela : IV Simposio Científico y Tecnológico en Computación / SCTC 2016 , 2016.
33. **Summerville, Ian.** *Ingeniería de software*. 9. Mexico : Pearson Education, 2011.
34. —. *Ingeniería de Software* . 7. madrid : Pearson, 2005.
35. **Pressman, Roger.** *Ingeniería de software: un enfoque práctico*. 5. s.l. : McGrawHill, 2002.
36. **Constanza Pavas, Adriana, y otros.** *Diseño de salida del sistema*. s.l. : Universidad del Quindío, 2016.

BIBLIOGRAFÍA

37. **Pressman, Roger.** *Ingeniería de software. Un enfoque práctico . 7ma.* New York : McGrawHill, 2010.
38. *Patrones de Diseño .* **Visconti, Marcello y Astudillo, Hernan.** Colombia : Universidad Técnica Federico Santa María , 2015.
39. **Vazquez Escudero, Pedro Jesus, Moreno Garcia, Maria N. y Garcia Peñalvo, Francisco J.** *Metrica Orientada a Objetos .* Salamanca : s.n., 2001.
40. **Lorenz, M y Kidd, J.** *Object Oriented Metrics. 1.* New Jersey : Prentice hall, 1994.
41. **Rogríguez Rodriguez, Carlos Rafael.** *Herramienta para aplicar métricas al Diagrama de Clases del Diseño Orientado a Objetos. .* Matanzas : Revista Avanzada científica, 2012.
42. **Lopez Rodriguez, Alejandro.** *La técnica de ladov. Una aplicación para el estudio de la satisfacción de los alumnos por las clases de educación física.* 2002.
43. **Alvarez de Zayas, Carlos.** *Metodología de la investigación científica.* Santiago de Cuba : s.n., 1995.
44. **Nocedo, Irma.** *Metodologia de la investigación edicacional.* La Habana : Felix Varela, 2001.
45. **Pencil Project.** *Pencil Project.* [En línea] 2019. [Citado el: 24 de 10 de 2019.] <https://pencl.evolus.vn>.
46. *Introducción a la ingeniería de software.* **Gracia Peñalvo, Francisco Jose y Gracia Holgado, Alicia.** Salamanca : GRIAL Research group University of Salamanca, 2018.
47. *Mantenimiento Perfectivo en un Gestor de Contenidos: Una Experiencia para la Especificación de Consultas Dinámicas.* **Godoy, Maria V, y otros.** ISSN 2314-2642, s.l. : Revista Latinoamericana de Ingenieria de software, 2014, Vol. 2(5), págs. 310-312. Referenciado por: Gomez Codutti, Ana; Mariño, Sonia I en *Una propuesta de proceso software para implementar una revista digital.*
48. **Real Academia Española.** RAE. [En línea] 2019. [Citado el: 04 de 10 de 2019.] <https://dej.rae.es/lema/importaci%C3%B3n>.

BIBLIOGRAFÍA

49. **Thompson Reuters.** *Import Compliance and Management Software from ONESOURCE Global Trade.* Thompson Reuters. [Online] 2019. [Cited: 10 14, 2019.] <https://tax.thomsonreuters.com/en/onesource/global-trade-management/import-compliance>.
50. **Yéminus.** *Liquidación de Importaciones – Yéminus.* Yéminus. [En línea] 2019. [Citado el: 14 de 10 de 2019.] <https://www.yeminus.com/importaciones/>.
51. **Fundación contra el hambre.** *Exportación e importación en la empresa. s.l. : Fondo Social Europeo dentro del Programa Operativo contra la Discriminación.*
52. **Real Academia Española.** RAE. [En línea] 2019. [Citado el: 04 de 10 de 2019.] <https://dej.rae.es/lema/importaci%C3%B3n>.
53. **GONZÁLEZ NAVARRO, DANIELA.** *EXPRESIVIDAD DE BPMN 2.0 DESDE LA PERSPECTIVA DEL PROBLEMA DE SECUENCIAMIENTO.* Chile : s.n., 2018.

ANEXOS

Anexo 1: Descripción de los subprocesos

- **Presentar forma de financiamiento**

El cliente presenta a economía la forma de financiamiento (FF) del plan de importación. El especialista (Esp.) B es gestión económica se encarga de una vez recibido el fondo del cliente distribuirlo en el sistema teniendo en cuenta como de encuentra desglosado el PI del cliente. A medida que se van presentando los pedidos del cliente, este puede ir modificando la fuente de financiamiento relacionada con la FF definida en el PI.

- **Revisar financiamiento**

El Esp. B en gestión económica recepciona la solicitud de importación (SI) enviada por el cliente y busca a dicho cliente en el sistema CIS donde mediante la SI entregada identifica la FF que se va a emplear. El Esp. B verifica si la aprobación se encuentra registrada, de ser así verifica el monto y la garantía de la aprobación. En el sistema el Esp. B relaciona la SI con el PI que le corresponde y se controla que no se sobregire el monto aprobado el en PI.

- **Registrar aprobación**

La casa financiera RAFIN envía a TECNOTEX las notificaciones de las aprobaciones, en la empresa el Esp. B en gestión financiera se encarga de recepcionar dichas notificaciones y las registra en el sistema CIS.

Anexo 2: Listado de requisitos funcionales

No	Requisitos funcionales
	Gestionar Aprobación
RF1	Crear Aprobación
RF2	Editar Aprobación
RF3	Listar Aprobación
RF4	Mostrar Aprobación
RF5	Buscar Aprobación
RF6	Exportar Aprobación
RF7	Importar Aprobación
RF8	Activar Aprobación
RF9	Desactivar Aprobación
	Gestionar Desglose por PI
RF10	Crear Desglose por PI
RF11	Editar Desglose por PI
RF12	Suprimir Desglose por PI

RF13	Listar Desglose por PI
RF14	Mostrar Desglose por PI
RF15	Buscar Desglose por PI
	Gestionar enmienda de aprobación
RF16	Crear Enmienda de aprobación
RF17	Editar Enmienda de aprobación
RF18	Suprimir Enmienda de aprobación
RF19	Listar Enmienda de aprobación
RF20	Mostrar Enmienda de aprobación
RF21	Buscar Enmienda de aprobación
	Gestionar Extraplan
RF22	Crear Extraplan
RF23	Editar Extraplan
RF24	Suprimir Extraplan
RF25	Listar Extraplan
RF26	Mostrar Extraplan
RF27	Buscar Extraplan
	Gestionar fecha tope de presentar Plan de importación
RF28	Crear fecha tope de presentar Plan de importación
RF29	Editar fecha tope de presentar Plan de importación
RF30	Listar fecha tope de presentar Plan de importación
RF31	Buscar fecha tope de presentar Plan de importación
RF32	Activar fecha tope de presentar Plan de importación
RF33	Desactivar fecha tope de presentar Plan de importación
RF34	Mostrar fecha tope de presentar Plan de importación
	Gestionar nomenclador objeto
RF35	Crear nomenclador objeto
RF36	Editar nomenclador objeto
RF37	Listar nomenclador objeto
RF38	Buscar nomenclador objeto
RF39	Activar nomenclador objeto
RF40	Desactivar nomenclador objeto
RF41	Mostrar nomenclador objeto
	Gestionar nomenclador tipo de aprobación

RF42	Crear nomenclador tipo de aprobación
RF43	Editar nomenclador tipo de aprobación
RF44	Listar nomenclador tipo de aprobación
RF45	Buscar nomenclador tipo de aprobación
RF46	Activar nomenclador tipo de aprobación
RF47	Desactivar nomenclador tipo de aprobación
RF48	Mostrar nomenclador tipo de aprobación
	Gestionar plan de importación
RF49	Crear plan de importación
RF50	Editar plan de importación
RF51	Suprimir plan de importación
RF52	Listar plan de importación
RF53	Mostrar plan de importación
RF54	Buscar plan de importación
RF55	Exportar plan de importación
RF 56	Imprimir plan de importación
RF57	Importar plan de importación

Anexo 3: Cuestionario de la técnica de ladov

1.	¿Consideraría factible la gestión del plan de importación sin contar con una herramienta que centralice la información relacionada con el proceso y esté disponible desde cualquier área que pueda necesitar dicha información?	Sí ____ No ____ No sé ____
2.	¿Considera usted que el componente propuesto facilita el acceso a la información desde todas las áreas de la empresa?	Sí ____ No ____ No sé ____
3.	¿Utilizaría usted el componente desarrollado para gestionar el plan de importación, las aprobaciones y sus respectivas modificaciones (extraplan y enmienda)?	Sí ____ No ____ No sé ____
4.	¿Considera usted que la solución propuesta permite agilizar el	Sí ____

ANEXOS

	proceso de Gestión del Plan de Importación?	No ____ No sé ____
5.	¿Considera que el componente realiza un seguimiento adecuado sobre los estados del presupuesto aprobado?	Sí ____ No ____ No sé ____
6.	¿Considera que la solución propuesta contribuye a disminuir los errores de presentación de los planes de importación y extraplanes?	Sí ____ No ____ No sé ____
7.	¿Le satisface este sistema para su utilización en el área de Economía?	____ 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
8. ¿Qué elementos considera positivo del Componente para la gestión del Plan de importación?		
9. ¿Qué elementos considera negativo del Componente para la gestión del Plan de importación?		
10. ¿Qué importancia le concede al componente del plan de importación?		