Universidad de las Ciencias Informáticas

Facultad 3

Módulo de Seguridad para el Sistema de Importación de Tecnotex en Odoo v13.0

Trabajo de diploma para optar por el título de

Ingeniero en CienciasInformáticas

Autor:

Victor Manuel Sarduy Horta

Tutores:

Ing. Boris Luis Correa Frías

Ing. Lisardo García Jane

Ing. Roberto Bandera Gómez

La Habana, 2020

Declaración de autoría

Declaro ser el único autor del trabajo de diploma y otorgo a la Universidad de lasCiencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

_____

Firma del autor
Victor Manuel Sarduy Horta

_____	_____	_____
Firma del tutor	**Firma del tutor**	**Firma del tutor**
Ing. Boris Luis Correa Frías	Ing. Lisardo García Jane	Ing. Roberto Bandera Gómez

Agradecimientos

Agradezco a mi familia por todo el apoyo que me han brindado siempre, no puedo desear una mejor. A mis tutores por las noches de desvelo, la dedicación y entrega. A mis amigos de la universidad. A mis profesores, especialmente los profesores Zénel, que nunca dejó de ser nuestro guía durante toda la universidad. A mis compañeros de la UCI que durante estos años nos hemos convertido en una familia y con los que siempre podré contar. A mis compañeros de aula por el apoyo en las horas de estudio y los buenos momentos que pasamos juntos que, por mi parte, nunca serán olvidados.

Dedicatoria

Dedico este triunfo a mis padres, que más que padres lo han sido todo para mí, a sus cientos de lágrimas derramadas por mi ausencia, a sus sufrimientos por no tenerme en los momentos especiales para nosotros, por aguantar mi ausencia, por desvivirse para que pudiera estar bien en la universidad, por luchar por mi y creer en este momento desde el día uno, en el que me dejaron marchar con sus ojos convertidos en mares, a mi tutor Lisardo García James que estuvo siempre ahí para mí, a cualquier hora, en cualquier situación. A todos los cientos de amigos que hice y conocidos, a Alejandro MendezAchón, que me levantó todos los días para poder asistir a clases, a las personas que vivieron conmigo todos estos años, que fueron más que familia, que profesores, más que amigos. A Malcom y Yeni que me dieron una percepción diferente de la universidad, de la carrera, de la vida, a ellos que me enseñaron lo lindo que es todo y como lo podemos plasmar en lo que hacemos, se la dedico a todos esos profesores que me hicieron quien soy ahora, a todos; a la UCI, gracias.

Resumen

La relación entre las TIC y los sistemas informáticos generan una cantidad significativa de activos a proteger, así como vulnerabilidades que aumentan diariamente. La Seguridad Informática es una disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático. También se ocupa de diseñar los procedimientos, métodos y técnicas, orientados a proveer condiciones seguras y confiables, para el procesamiento de datos en sistemas informáticos. La presente investigación tiene como objetivo fundamental desarrollar un módulo de seguridad que permita garantizar la administración, autenticación, autorización y auditoría del Sistema de Importación de TECNOTEX. Se aplicaron diferentes técnicas para la obtención de requisitos se favoreció la especificación y descripción de los mismos, documentándose toda la información relativa a la propuesta de solución. Para el desarrollo de la aplicación se utilizaron técnicas y herramientas de software libre, lo que posibilitó una mayor soberanía tecnológica. La solución fue sometida a un proceso de pruebas de funcionamiento guiado por casos de pruebas, donde la realización de las mismas corroboró la solidez del sistema, cumpliendo con las características pactadas por el cliente.

Palabras claves: Seguridad, Tecnotex, Módulo

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Introducción

El rápido crecimiento de las Tecnologías de la Información y la Comunicación (TIC) y las mejoras de seguridad, organización y gestión que traen consigo, promueven su uso y la necesidad de su aplicación en diferentes áreas de la sociedad. Esto trae como ventajas: el fácil acceso a la información, el procesamiento de forma rápida e integrar los datos, una amplia capacidad de almacenamiento y automatización de trabajos, así como gran interactividad con los usuarios que la utilizan.Dentro de sus amplias categorías juegan un papel fundamental los sistemas informáticos ya que permiten almacenar y procesar información,como un conjunto de partes interrelacionadas: hardware, software y personal informático.

La relación entre las TIC y los sistemas informáticos generan una cantidad significativa de activos a proteger, así como vulnerabilidades que aumentan diariamente. Estos activos a protegerson información de todo tipo, tanto personal como empresarial. Por esto es necesario el control interno del sistema, o sea, establecerciertos privilegios para darle acceso al personal que interactúan con estos sistemas. Es por esto que es crucial la verificación y control de cambios para evitar la pérdida o filtrado de información no deseada.Teniendo en cuenta que esta información puede ser utilizada para el control de decisiones (por vía directa o indirecta) sobre individuos o sociedades, es importante el correcto manejo de la información y disponer de módulos de seguridad que puedan proteger los archivos en dichos sistemas.En tal sentido se definen como privilegios los permisos de acceso, lectura o escritura de alguna parte o toda la información que en ellos se manejan.

La Seguridad Informática es una disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático. También se ocupa de diseñar los procedimientos, métodos y técnicas, orientados a proveer condiciones seguras y confiables, para el procesamiento de datos en sistemas informáticos. Para lograr sus objetivos la seguridad informática se fundamenta en tres principios, que debe cumplir todo sistema informático: confidencialidad, integridad y disponibilidad (López 2010).

Cuba tiene un tratamiento muy específico y riguroso con respecto a la seguridad de sus sistemas informáticos. En el año 2019 se emiten por primera vez normas jurídicas de rango superior que ordenan el proceso de informatización de la sociedad. A raíz de estas normas El Ministerio del Interior (MININT) dicta normativas para la producción de software soberano y seguro, que propone estándares de seguridad los cuales todo sistema cubano debe cumplir para poder ser utilizado en el país.

La estrategia de informatización de Cuba se ha adoptado para proveer soberanía tecnológica, a pesar de no ser un país desarrollado, ha dedicado una importante parte de sus recursos al

desarrollo de la informática en diferentes esferas de la sociedad. Una de las instituciones que contribuye a este proceso gracias a las TIC, es la Universidad de las Ciencias Informáticas (UCI), que además de ser un centro de estudios universitarios, tiene como uno de sus objetivos la informatización del país, por lo que trata de desarrollar al máximo la industria del software en la mayoría de las ramas de la sociedad, contribuyendo de esta forma con el avance económico del país.En correspondencia a esto, 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 sistemas informáticos encaminados a gestionar procesos empresariales.

Uno de los marcos de trabajo que utiliza para la gestión empresarial es Odoo, como sistema de planificación de recursos empresariales, provee a las empresas una mejor gestión de todos sus recursos como, por ejemplo: las compras, ventas, facturas, finanzas, clientes y productos. En este, el control interno y el control de cambios es esencial ya que maneja información sensible para las empresas. Es por ello que la seguridad puede ser clave para la resolución de un conflicto dentro de una organización. En tal sentido se deben incluir un conjunto de requerimientos de seguridad cuando se especifican los requisitos del sistema informático a desarrollar, estos deben ser implementados en el ciclo de desarrollo y despliegue de dicho sistema. Este marco de trabajo contiene un conjunto de funcionalidades implementadas que responden de manera general a un conjunto de requisitos de seguridad, pero para la autenticación, autorización y administración se debe realizar nuevas configuraciones.

Actualmente CEIGE se encuentra desarrollando elSistema de Importación y Suministro para la empresa TECNOTEX, utilizando como marco de trabajo Odoo en su versión 13.0, dicho sistema está dividido en módulos para facilitar su desarrollo, siendo el Módulo de Seguridad para el Sistema de Importación de TECNOTEX uno de estos. Odoocuenta entre sus aplicaciones con un módulo de seguridad, el cual a pesar de no adaptarse completamente al escenario actualpuede ser reutilizado. Dicho esto, los aspectos que deben ser modificados se listan a continuación:

- El manejo del mecanismo de sesiones teniendo en cuenta los identificadores de sesión.
- La protección de la comunicación utilizando protocolos seguros.
- El mecanismo de recolección de trazas, afectando el control interno del sistema.

De igual forma no contempla el conjunto de requisitos de seguridad, establecidos en las normas jurídicas mencionadas previamentey que los sistemas empresariales cubanos deben cumplir.

A partir de la problemática antes planteada se define como problema a resolver: ¿Cómo garantizar la seguridad en la administración, autenticación, autorización y auditoría del Sistema de Importación para la empresa TECNOTEX?

Se identifica, como **objeto de estudio:** la seguridad de la información.

Enmarcándose como **campo de acción:** la administración, autenticación, autorización y auditoría en los sistemas de gestión empresarial.

Para dar solución al problema planteado, se define como **objetivo general:** desarrollar un módulo de seguridad que cumpla con los requisitos definidos por el MININT para garantizar la administración, autenticación, autorización y auditoría del Sistema de Importación de TECNOTEX.

A partir del análisis del objetivo general se definieron 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 módulo para la gestión de la seguridad del sistema.
4. Validar la propuesta de solución a través de pruebas y métodos científicos.

Por lo tanto, se planteacomo **idea a defender:**si se desarrolla un módulo para la gestión de la seguridad, entoncesse garantiza la administración, autenticación, autorización y auditoría del Sistema de Importación de TECNOTEX.

Se identifica como **variable independiente:** módulo de seguridad, y como **variable dependiente:** garantizar la administración, autenticación, autorización y auditoría.

Para el desarrollo de la investigación se utilizan los siguientes **métodos científicos**,clasificados en teóricos y empíricos de los cuales se emplearon:

- **Métodos teóricos:**
  - ✓ **Método Analítico-Sintético:** se utilizó para descomponer el objeto de estudio en partes más pequeñas para ser estudiadas y luego analizarlo de manera integral. De igual forma, facilitó el análisis de la seguridad en la versión de Odoo a trabajar, en comparación con otras versiones.
  - ✓ **Modelación:** se utilizó para la realización de los artefactos necesarios en el proceso de desarrollo de software, haciendo una representación abstracta de la solución, facilitando así el desarrollo de la misma.
- **Métodos empíricos:**
  - ✓ **Revisión bibliográfica:** facilitó la elaboración del marco teórico de la investigación a partir de la consulta de documentos relacionados con la seguridad de la información, así como una caracterización de las técnicas, herramientas y lenguajes a emplear en el desarrollo de la investigación.

- ✓ **Entrevista:** permitió el intercambio verbal con el cliente para obtener la mayor cantidad de información posible, así como las deficiencias existentes que permitieron definir el problema a resolver.

El presente trabajo de diploma se encuentra distribuido en 3 capítulos:

Capítulo 1: Fundamentación teóricade la investigación

En este capítulo se realiza un estudio del estado del arte como punto de partida para la investigación. El estudio parte del análisis de la seguridad de la información en los sistemas de gestión empresarial, para un correcto control interno y seguimiento de los documentos sensibles. Luego se establecen los conceptos esenciales relacionados con el tema de la investigación. Se realiza, además, un estudio de la metodología de desarrollo de software que serve como guía para dar solución al problema planteado. Por último, se describen las herramientas y lenguajes a utilizar durante el desarrollo de la propuesta de solución.

Capítulo 2: Propuesta de solucióne Implementación

En este capítulo se plasma la propuesta de solución, abarcando cada una de las disciplinas que establece la metodología de desarrollo de software utilizada en la presente investigación. Para ello se partede la descripción de la solución, luego se especifican los requisitos funcionales y no funcionales a desarrollar. De igual forma se realiza el análisis y diseño, a través de varios artefactos utilizados para modelar el sistema. Por último, se establecen los estándares de codificación a tener en cuenta para la implementación, así como una vista del modelo de la base de datos utilizada para describir la estructura lógica y física de la información persistente gestionada por el módulo propuesto.

Capítulo 3:Validación de la propuesta de solución

En el presente capítulo, una vez concluida la fase de análisis y diseño de la propuesta de solución se procede a la implementación de las clases y ejecución de casos de prueba que evalúen las funcionalidades. Se determina si las funcionalidades implementadas cumplen con las características establecidas y con las descripciones de las HU anteriormente expuestas, realizando las iteraciones necesarias para cumplir satisfactoriamente los casos de pruebas elaborados.

Capítulo 1: Fundamentación teórica de la investigación

1.1. Introducción

Este capítulo contiene la base teórica de la propuesta de solución, que está sustentada en la seguridad del sistema para un correcto control interno y seguimiento de los documentos sensibles, que sirven de punto de partida para la investigación. Se realizará, además, un estudio de la metodología de desarrollo de software que servirá como guía para dar solución al problema planteado. Por último, se describirán las herramientas y lenguajes a utilizar durante el desarrollo de la propuesta de solución.

1.2. Conceptos asociados a la investigación

Con el objetivo de lograr un entendimiento del objeto de estudio de la presente investigación, se requiere el establecimiento de los conceptos enunciados a continuación:

Seguridad de la información:

La información es un activo que, como otros importantes activos de negocios, tiene valor para una organización y en consecuencia necesita ser debidamente protegido. Puede existir en muchas formas: impresa o escrita en papel, almacenada electrónicamente, trasmitida por correo o usando medios electrónicos, impresa en películas o hablado en conversación. No importa la forma que tome, el medio por el que se comparta o en el que se almacene, siempre debe ser correctamente protegida (Ponjuan Dante, 2012).

En términos generales, la seguridad puede entenderse como aquellas reglas técnicas y/o actividades destinadas a prevenir, proteger y resguardar lo que es considerado como susceptible de robo, pérdida o daño, ya sea de manera personal, grupal o empresarial (HUGO, 2008).

Esta se logra mediante la implementación de un apropiado sistema de controles, que pudieran ser políticas, prácticas, procedimientos, estructuras organizacionales y funciones de software. Estos controles necesitan ser establecidos para asegurar que los objetivos específicos de seguridad se cumplan (Yudith, 2019).

En este sentido, es la información el elemento principal a proteger, resguardar y recuperar dentro de los sistemas informáticos, por tal sentido surge la seguridad informática.

Seguridad informática:

La seguridad informática protege la información de un amplio rango de amenazas con el objetivo de asegurar la continuidad de negocios, minimizar el daño comercial y maximizar el reembolsode las inversiones y oportunidades comerciales (NTP-ISO/IEC-17799, 2007).

Es por ello que se define como una disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático.(Merino, 2008)

Esta se caracteriza como la protección de:

- La confidencialidad: se refiere a la privacidad de la información almacenada y procesada en un sistema informático, las herramientas de seguridad informática deben proteger el sistema de intrusos y accesos por parte de personas o programas no autorizados. Este principio es importante en aquellos sistemas en los que los usuarios, computadoras y datos residen en lugares diferentes, pero están física y lógicamente interconectados.
- La integridad: se refiere a la validez y consistencia de los elementos de información almacenados y procesador en un sistema informático. Basándose en este principio, las herramientas de seguridad deben asegurar que los procesos de actualización estén bien sincronizados y no se dupliquen, de forma que todos los elementos del sistema manipulen adecuadamente los mismos datos. Este principio es importante en aquellos sistemas en los que diferentes usuarios, computadoras y procesos comparten la misma información.
- La disponibilidad: se refiere a la continuidad de acceso a los elementos de información almacenados y procesados en un sistema informático. Basándose en este principio, las herramientas de seguridad informática deber reforzar la permanencia del sistema informático, en condiciones de actividad adecuadas para que los usuarios accedan a los datos con la frecuencia y dedicación que requieran. (Yudith, 2019)

Administración:

El concepto de administración hace referencia al funcionamiento, la estructura y el rendimiento de las organizaciones. El término proviene del latín *ad-ministrare* (“servir”) o *ad manustrahere* (“manejar” o “gestionar”). La administración puede ser entendida como la disciplina que se encarga de realizar una gestión de los recursos (ya sean materiales o humanos) en base a criterios científicos y orientada a satisfacer un objetivo concreto (Definición.de, 2019).

En el ámbito de la informática diferentes tipos de aplicaciones necesitan la gestión o configuración de los usuarios, las estructuras donde operan sus funcionarios, gestión de grupos de usuarios, definición de dominios, regulación de acciones y control de acceso restringido para cada estructura.

Los sistemas informáticos actuales son cada vez más flexibles y adaptables en dependencia de las necesidades de los clientes. Por esta razón generalmente la mayoría de los sistemas cuentan con un rector informático. El mismo se encarga de permitir el control de la estructura organizativa del sistema, garantizando eficiencia en todos sus módulos y procesos, posibilitando la centralización de algunos procedimientos y configuraciones globales que garantizan el buen funcionamiento del mismo. Manejando diferentes conceptos en dependencia de las necesidades finales del cliente, por ejemplo: usuarios, puestos de trabajo, gestión de configuración de archivos y variables, así como el control del flujo de eventos (La administración de sistemas informáticos,

una alternativa a la formación del profesional en tecnologías de información y comunicaciones, 2018).

Autenticación:

En informática se refiere al proceso electrónico que permite la identificación de una persona física o jurídica, o del origen de la integridad de los datos en formato electrónico. En otras palabras, es el procedimiento de identidad de un usuario en un recurso informático(RAE - Diccionario Real de la Academia Española, 2017).

Los métodos de autenticación están en función de lo que utilizan para la verificación y estos se dividen en tres categorías(RedIRIS, 2008):

- Sistemas basados en algo conocido: ejemplo, una contraseña
- Sistemas basados en algo poseído: ejemplo, una tarjeta de identidad o una tarjeta inteligente
- Sistemas basados en una característica física delante usuario o un acto involuntario del mismo: ejemplo, verificación de voz, de escritura, de huellas, de patrones oculares.

En la presente investigación se tendrá en cuenta la autenticación en sistemas basado en algo conocido. Procedimiento por el cual se decide si un usuario es quien dice ser simplemente basándose en una prueba de conocimiento, una contraseña, que en principio es secreta por las personas que interactuarán en el Sistemas de Importación de TECNOTEX.

Autorización:

Es la acción y efecto de autorizar (reconocer la facultad o el derecho de una persona para hacer algo). Para la informática, la autorización es la parte de un sistema operativo que protege los recursos del sistema, de modo tal que sólo puedan ser utilizados por los usuarios que cuentan con permiso para eso. Consiste en dar consentimiento para que otros hagan o dejen de hacer algo (Definición.de, 2019).

En la presente investigación, la autorización se basará en una jerarquía de usuarios en correspondencia con las acciones que puedan realizar, dotando a los usuarios solo los privilegios que necesitan. Los permisos seránlimitados basados en los roles definidos.

Auditoría:

Las auditorías han de ser consideradas como procesos informacionales, desde las económico-financieras hasta las de conocimiento; de ahí que pueden ser tratadas de manera conjunta si se poseen las herramientas y metodologías apropiadas. Éstas interactúan con la estrategia de la organización, tomando los elementos primarios para comenzar la revisión de los procesos, los recursos y la propia estrategia, y una vez cumplidos sus objetivos, enriquecen esta última, formulan recomendaciones, acciones correctivas, cronogramas y planes de implementación,

seguimiento y control(Mirada contextual a los nexos entre las auditorías de información y las auditorías de conocimiento, 2011).

Para lapropuesta de solución la auditoría en el Sistema de Importación de TECNOTEX se tendrá en cuenta mediante un sistema de gestión de trazas y auditoría, donde se deben guardar trazas de todas las operaciones realizadas por el usuario sobre la solución, así como otros eventos de interés, que permitirán determinar funcionamientos incorrectos de la solución, así como resolver incidentes de seguridad.

En Cuba, de obligatorio cumplimiento para todas las entidades, está la resolución 127 de 2007 del Ministerio de la Informática y las Comunicaciones (MIC) actualmente Ministerio de las Comunicaciones. La resolución cuenta con 100 artículos. La sección octava del Capítulo III trata sobre la seguridad de redes con tres artículos. Concretamente, los artículos 58 inciso b, 62 inciso b y 83 inciso b tratan sobre la generación, revisión periódica y específicamente el registro de las conexiones remotas(MIC, 2007).

De las guías y documentos de buenas prácticas descritas se pueden extraer tres categorías asociadas a la gestión de trazas. Una categoría general asociada a los componentes necesarios en la implementación de un sistema de gestión de trazas, los registros de trazas de mayor importancia para la seguridad de los sistemas monitorizados y a un nivel más profundo, los campos que deben contener estos registros(Marco de trabajo para la gestión centralizada de trazas de seguridad usando herramientas de código abierto, 2015).

Para el caso de la propuesta de solución se establece un mecanismo de gestión de trazas basado en:

- Traza para auditoria funcional: registro de la identidad del usuario autenticado que invocó el sistema, tipo de operación realizada, fecha, hora, datos de la máquina desde donde fue realizado (dirección IP).
- Traza para la auditoria de seguridad: además de los datos de la auditoría funcional, se registran otros datos como: si el evento tuvo éxito o fallo, descripción del evento, nivel de seguridad del evento.

El autor de la presente investigación, una vez establecidos los referentes teóricos que engloban la propuesta de solución, muestra la relación entre el objetivo que se persigue, con los principios de la seguridad informática.

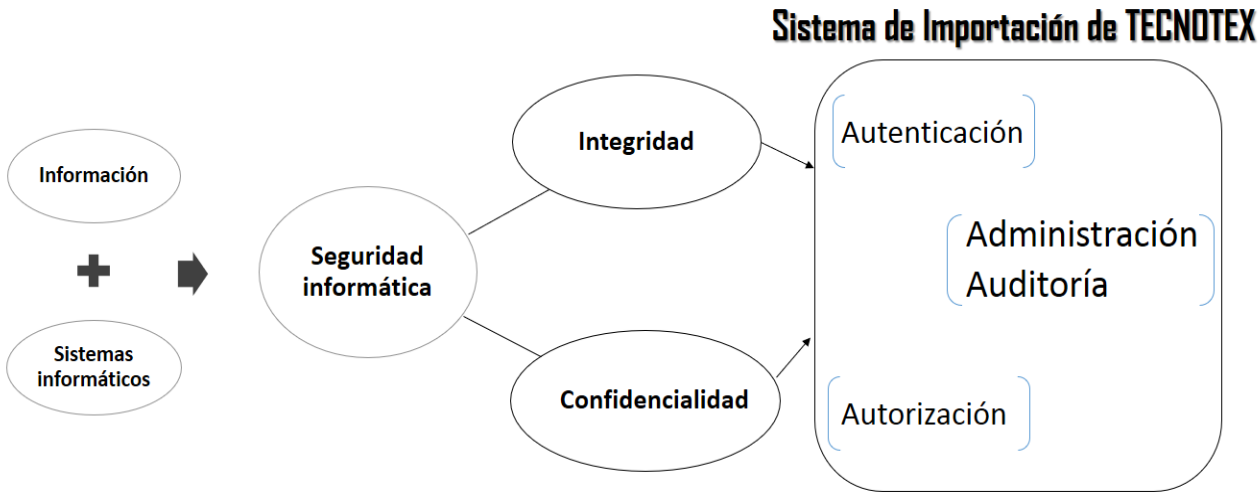


Figura 1 Seguridad en el Sistema de Importación de TECNOTEX

Fuente: elaboración propia

Como se muestra en la figura anterior, una vez que se logre la administración, autenticación, autorización y auditoría en el Sistema de Importación de TECNOTEX, se proveerá de políticas, prácticas y procedimientos que aseguraran la integridad y confidencialidad de la información del mismo.Para lograr este objetivo, se necesita una personalización del marco de trabajo utilizado por dicho sistema:de forma tal que se adapte a las necesidades del cliente. Por tal motivo a continuación se describe cómo se lleva a cabo la personalización de módulos en Odoo.

1.3. Personalización de módulos en Odoo

La personalización es el desarrollo de sistemas, aplicaciones o programas, de acuerdo con las especificaciones del cliente. No es más que adaptar algo a las características, al gusto o a las necesidades de una persona, cliente o empresa(Gesoft-Informática, 2019).

Esta puede ser total o parcial:

- El software puede ser 100% personalizado, totalmente desarrollado específicamente para el cliente en cuestión.
- El software se puede adaptar a las necesidades del cliente, a partir de un software básicoexistente (Gesoft-Informática, 2019).

Para el caso de la presente investigación se realizará una personalización del módulo de seguridad de forma parcial, ya que se adaptará a las necesidades del cliente, según las normas jurídicas establecidas a nivel nacional.

Para ello, existen dos maneras de llevar a cabo la personalización en Odoo:mediante programación de los archivos de código Python y XML o a través de la misma interfaz web (Yeray Fernández, 2014). A continuación, se detallan ambos procederes:

1.3.1. Programación de los archivos mediante código Python y XML

Este método consiste en la alteración del código fuente de los archivos correspondientes al módulo o a los módulos que se quieran cambiar. Para efectuar cambios en el sistema basta con realizar las modificaciones pertinentes en el código del módulo para cargar después la actualización vía web. Se eliminan, añaden y/o modifican los elementos existentes en el archivo correspondiente del módulo y se realizan los cambios pertinentes en los archivos XML para que quede reflejado en las vistas. Estas alteraciones se realizan manualmente, como, por ejemplo, con la herramienta IDE PyCharm. El software reconoce los archivos alterados y aparecen en el apartado “Actualizar lista de módulos”, que está en la sección “Módulos” del menú “Configuración”. A través del código se pueden añadir nuevas variables, nuevos menús, nuevos campos, entre otros elementos. Todos esos cambios deben ser consistentes y ser declarados por igual en los distintos archivos para que el programa los reconozca y se puedan implementar en la solución final. Entre las ventajas de realizar el cambio mediante la programación se encuentran las siguientes (FERNÁNDEZ ALONSO, 2014.):

- Las posibilidades de modificación son enormes. Se pueden añadir los elementos que seconsideren necesarios, declarar su tipo y función de una sola vez.
- Se pueden trasformar los campos bases, que de otra manera permanecen inmutables.
- Se da la posibilidad de aplicar numerosos cambios al programa simultáneamente.
- Ofrece un alto grado de versatilidad.

1.3.2. A través de la interfaz

El proceso consiste básicamente en definir nuevas variables o campos y coordinarlo con las vistaspara que se puedan utilizar de manera funcional. De la misma manera se pueden modificar losmenús y las acciones ligadas a estos.

Para comenzar a trabajar con este método en la modificación del módulo, es necesario activar elmodo desarrollador. Esta opción se encuentra en “Acerca de Odoo”. Tras activar estemodo desarrollador, en cada pantalla de Odoo aparecerá un menú desplegable en la esquina superior, a la izquierda del nombre de la sección. Lo más importante que se puede llevar a cabo desde él es la edición de las vistas, donde se puede añadir y quitar campos, introducir nuevos tipos de vistas, entre otras modificaciones.

Las ventajas que presenta esta forma de modificación son:

- Es un método intuitivo y rápido.
- No es necesario salir del ERP.
- Se puede ir observando el resultado mientras se van realizando los cambios.
- Los campos se ven de manera compacta, de forma que en una misma ventana se puedenobservar sus propiedades y modificarlas, asignar grupos, añadir menús.Por el contrario,presenta los siguientes inconvenientes:

- Hay que coordinar la coherencia con las vistas para evitar errores y fallos.
- No se pueden cambiar los campos base. En este sentido, la personalización esmenosprofunda.

Para el caso de la presente investigación, se reutilizarán funcionalidades de otros módulos que cumplan con las necesidades del sistema, se realizará la personalización y reutilización de módulos de seguridad mediante programación de archivos de código Python y XML. Además, se utilizará la personalización a través de la interfaz, ya que existen módulos que solo con configuración mediante la interfazcumplirán con las exigencias del sistema. Por lo que se utilizarán los dos procederes.

1.4. Análisis de la seguridad en Odoo v13.0

Odoo en su versión 13.0 cumple con muchas de las características para la realización del Sistema de Importación de TECNOTEX. Para el caso de la presente investigación se realizó un análisis en el cual se valoró las necesidades de seguridad para este sistema con respecto a las exigencias normadas por el MININT, el cual arrojó que Odoo v13.0 no cumple con algunas funcionalidades y necesidades de seguridad para este sistema como son:

- Autenticación por servicios LDAP.
- Bloqueo de autenticación por IP.
- Cierre de cesión por inactividad.
- Auditoria y gestión de trazas.
- Establecimiento de un protocolo seguro.
- Suma de verificación en importación y exportación de archivos.
- Suma de verificación en salvas y restauración de bases de datos.

1.5. Metodología de desarrollo de software

Una metodología de desarrollo de software describe cómo llevar a cabo dicho proceso,definiendo un conjunto de pasos y procedimientos,que se deben llevar a cabo en el desarrollo del software. Constituye un medio deestandarización que cubre por completo el proceso de desarrollo de software, además provee un lenguaje común entre los analistas, programadores, clientes y usuarios (Metodologías actuales de desarrollo de software, 2015).

Para el desarrollo de la solución planteada se empleará la metodologíade desarrollopara la actividad productiva de la UCI, debido a que está definida por la universidad para el entorno productivo. Esta metodología es una variación de la metodología Proceso Unificado Ágil (AUP por sus siglas en inglés de *Agile UnifiedProcess*)

1.5.1. Metodología Proceso Unificado Ágil variación para la UCI

La variación de la metodología AUP para la UCI (AUP-UCI) se crea porque, no existía una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto exigiéndose así que el proceso sea configurable. Por lo que se

decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI(Rodríguez Sánchez, 2015).

Entre las principales diferencias presentes en esta versión se encuentran:

- AUP define cuatro (4) fases de desarrollo (Inicio, Elaboración, Construcción, Transición), y su versión para la UCI mantiene la fase de Inicio, pero se modifica el objetivo, agrupa las otras tres (3) fases en una llamada Ejecución e incorpora una llamada Cierre.
- AUP propone siete (7) disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), la metodología definida en la UCI tiene ocho (8) disciplinas.
- Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran cada uno de ellos disciplinas. Específicamente en la disciplina Requisitos plantean cuatro (4) posibles escenarios para la identificación y descripción de requisitos. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en tres (3) disciplinas: Pruebas Internas, de Liberación y Aceptación y la disciplina Despliegue se considera opcional(Rodríguez Sánchez, 2015).

En esta variante se definen cuatro (4) escenarios:

1. Proyectos que modelen el negocio con CUN (Caso de Uso del Negocio) solo pueden modelar el sistema con CUS (Casos de Uso del Sistema).
2. Proyectos que modelen el negocio con MC (Modelo Conceptual) solo pueden modelar el sistema con CUS (Casos de Uso del Sistema).
3. Proyectos que modelen el negocio con DPN (Descripción de Proceso de Negocio) solo pueden modelar el sistema con DRP (Descripción de Requisitos por Proceso).
4. Proyectos que no modelen negocio solo pueden modelar el sistema con HU (Historias de usuario)(Rodríguez Sánchez, 2015).

Una vez evaluada la propuesta de solución, teniendo en cuenta que solo se centraría en personalizar y reutilizar módulos de manera tal que cubriera las necesidades del cliente, se obtuvo un negociobien definido y se concibió que la propuesta no modela negocio, ya que se obvian los procesos de gestión empresarial que engloba la empresa TECNOTEX y solo se trata la seguridad del sistema en si. Por otra parte, el cliente estará acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos, por lo que se puede desarrollar el sistema a través de las HU utilizando el escenario cuatro (4) que define AUP-UCI.

1.6. Herramientas y lenguajes de desarrollo

En el presente epígrafe se caracterizan las herramientas y lenguajes utilizados para el desarrollo del Sistema de Importación de TECNOTEX, y por ende para la propuesta de solución:

1.6.1. Herramienta para el modelado

Visual Paradigmv8.0: es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenierosde Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados enconstrucción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrolloorientado a objetos. Ofrece un entorno de creación de diagramas para UML. Entre muchas de sus ventajas permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generardocumentación. Presenta licencia gratuita y comercial, además de ser multiplataforma, fácil de instalar y actualizar(Visual Paradigm, 2016).

Para el modelado de la solución se seleccionó VisualParadigm por ser una herramienta multiplataforma que no se inclina por ningunametodología específica.

1.6.2. Lenguaje de modelado

UML v2.4: el Lenguaje Unificado de Modelado (UML por sus siglas en inglés de *UnifiedModelingLanguage*) prescribe un conjunto de notaciones y diagramasestándares para modelar sistemas orientados a objetos, y describe la semántica esencial de lo queestos diagramas y símbolos significan. Posibilitando así visualizar, especificar y documentar losartefactos o toda información que se obtiene o modifica durante un proceso de desarrollo desoftware. Además de poder utilizarse para modelar distintos tipos de sistemas de software, hardwarey organizaciones del mundo real (Sierra, 2013).

1.6.3. Marco de trabajo

Odoo v13: es un sistema integrado de gestión empresarial para la planeación de recursos empresariales. Es de código abierto, es capaz de cubrir las necesidades de las áreas de grandes, medianas y pequeñas empresas. A continuación, se detallan algunas característicasde manera general: (Odoo Community Hub, 2019)

- Presenta un enfoque modular, lo que permite ir añadiendo módulos progresivamente a una aplicación.
- Es un *framework* multiplataforma, pues a través de la interfaz web se puede acceder a cualquier ordenador, independientemente del sistema operativo, incluso desde tabletas y *Smartphone*[1].
- Es totalmente gratuito su uso y licencia.
- Toda la información se encuentra disponible en la página oficial de Odoo.

[1] Teléfonos inteligentes

- Odoo utiliza una estructura cliente-servidor, es decir el servidor maneja la lógica, se comunica con la basa de datos independientemente del cliente que muestra la información a los usuarios y les permite comunicarse con el servidor (dicho servidor lo trae por defecto Odoo).
- Este sistema ERP utiliza exclusivamente *PostgresSQL* como gestor de base de datos.
- El lenguaje de programación en el que se desarrolla el servidor web de Odoo es *Python*.
- Su arquitectura consta de tres niveles o capas (datos, negocio y presentación).
    - El servidor de base de datos*PostgresSQL*, que contiene todos los datos de la aplicación y la mayoría de los elementos de configuración del sistema Odoo.
    - El servidor Odoo, que contiene toda la lógica de la empresa y asegura que el sistema funcione de manera óptima. Este servidor tiene dos capas a su vez: una dedicada a la comunicación y la interfaz con la base de datos*PostgresSQL* (ORM *ObjectRelationalMapping*), y otra denominada capa Web, que permite las comunicaciones entre el servidor y un navegador web.
    - El cliente, que se ejecuta de forma local a través de un navegador web como una aplicación *JavaScript*. Este cliente se comunica en red con el servidor a través del protocolo XML-RPC.
- El servidor de Odoo se basa en una arquitectura donde el acceso a los datos y la lógica de negocio (Modelo) están separados de la presentación de los datos y la interfaz de usuario (Vista) a través de un componente de intermedio con acceso a ambos denominado Controlador. Este tipo de diseño se llama Modelo-Vista-Controlador (MVC), a continuación, se muestra dicha arquitectura.

1.6.4. Lenguajes de programación

Python:dentro de los lenguajes informáticos, *Python*, pertenece al grupo de los lenguajes de programación y puede ser clasificado como un lenguaje interpretado, de alto nivel, multiplataforma, de tipo dinámico y multiparadigma. A diferencia de la mayoría de los lenguajes de programación, *Python* provee de reglas de estilos, a fin de poder escribir código fuente más legible y de manera estandarizada. (Bahit, 2012)

A continuación, se describen algunas características importantes del lenguaje *Python*: (Python Software Foundation, 2010)

- Corre en múltiples plataformas, incluyendo *Windows*, *Mac OS* y Linux.
- Su sintaxis y semántica es sencilla y consistente.
- Utiliza tipos dinámicos.
- Es adecuado tanto para programar scripts como aplicaciones de gran tamaño.
- Es muy modular.
- Cuenta con administración automática de memoria a través de recolección de basura.

- Incluye una poderosa y extensa biblioteca de clases.
- Cuenta con una gran comunidad que se dedica a promover su desarrollo y adopción.

Para el desarrollo de la aplicación seutilizó **Python v3.7.4** debido a que es el lenguaje de programación que utiliza la versión de Odoo escogida.

XML: elLenguaje de marcas extensible (XML, siglas en inglés de *XtensibleMarkupLanguage*), es unlenguaje de marcas desarrollado por el *World Wide Web Consortium* (W3C) utilizado paraalmacenar datos en forma legible. Permite representar información estructurada en la web (todos documentos), de modo que esta información pueda ser almacenada, transmitida, procesada, visualizada e impresa, por muy diversos tipos de aplicaciones y dispositivos (XML, 2016).

Para el desarrollo dela aplicación se utilizó **XML v1.2** debido a que es el lenguaje de marcas extensibles que utiliza la versión de Odoo escogida.

1.6.5. Entorno de desarrollo integrado

Un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés de *IntegratedDevelopmentEnvironment*) es una aplicación compuesta por un conjunto de herramientas útiles para un programador. Un IDE puede ser exclusivo para un lenguaje de programación o bien, poder utilizarse para varios.(Alegsa, 2010)

PyCharmv4.5: es un IDE o entorno de desarrollo integrado multiplataforma utilizado para desarrollar enel lenguaje de programación Python. Entre las características fundamentales que posee se encuentran el autocompletado, resaltador de sintaxis, herramientas de análisis y refactorización. Posee un depurador avanzado, además de la integración con lenguajes de plantillas como Mako, Jinja2, Django. Soporta entornos virtuales e intérpretes de Python 2.x, 3.x, PyPy, Iron Python y Jython (JetBrains, 2020).

Se utilizó el Pycharm como IDE porque es el utilizado para la programación en Python.

1.6.6. Herramientas para la administración y gestión de base de datos

PgAdminv1.8.1: es una aplicación gráfica para gestionar y administrar las bases de datos PostgreSQL. PgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, Mac OSX y Solaris (Levin, 2019).

PostgreSQL v10: es un sistema de gestión de bases de datos relacional orientado a objetos, el cual incluye características como herencia, restricciones, tipos de datos, reglas e integridad transaccional. Tienesoporte total para transacciones, disparadores, vistas, procedimientos

almacenados,almacenamientode objetos de gran tamaño. Se destaca en ejecutar consultas complejas, consultassobre vistas, sub-consultas y *joins*[2]. Permite la definición de tipos de datos personalizados e incluyeun modelo deseguridad completo. Utiliza el modelo cliente servidor y es un manejador de base dedatos decódigo abierto liberado bajo la licencia BSD[3]. Postgresql está diseñado para administrargrandesvolúmenes de datos (PostgresSQL, 2010).

Se utilizó como sistema gestor de base de datosdebido a que es el que utiliza la versión de Odoo escogida.

1.7. Conclusiones parciales

- El establecimiento de los diferentes conceptos referentes a la seguridad de la información en sistemas informáticos permitió una mejor comprensión de la investigación.
- La caracterización de la metodología AUP en su variación para la UCI, sirvió como guíadel proceso de desarrollo de software;estandarizando así el ciclo de vida del software, dando cumplimiento, además, a las buenas prácticas que define el proceso de mejora de la universidad.
- El estudio de las características de las herramientas y tecnologías a utilizar sentaron las bases para el desarrollo de la propuesta de solución.

[2]La sentencia join de SQL permite combinar registros de una o más tablas en una base de datos.

[3]Licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution), un tipo del sistema operativo Unix-like.

Capítulo 2: Propuesta de solución e Implementación

2.1. Introducción

En este capítulo se plasma la propuesta de solución, abarcando cada una de las disciplinas que establece la metodología de desarrollo de software utilizada en la presente investigación. Para ello se parte de la descripción de la solución, luego se especifican los requisitos funcionales y no funcionales a desarrollar. De igual forma se realiza el análisis y diseño, a través de varios artefactos utilizados para modelar el sistema. Por último, se establecen los estándares de codificación a tener en cuenta para la implementación, así como una vista del modelo de la base de datos utilizada para describir la estructura lógica y física de la información persistente gestionada por el módulo propuesto.

2.2. Descripción de la propuesta de solución

El Sistema de Importación de TECNOTEX facilitará la importación, exportación y suministro en dicha empresa.Esta empresa tiene distribuidas sus acciones en tres procesos fundamentales, Presentación y Revisión de pedidos, Contratación y Suministro de mercancía.Estos se automatizarán permitiendo que cada proceso y sus subprocesos se puedan ejecutar de manera ágil y con la calidad requerida. Para el desarrollo de dicho sistema es necesario un módulo de seguridad, el cual creará las condiciones necesarias que debe poseer todo sistema informático desarrollado en Cuba en cuanto a la seguridad informática, teniendo en cuenta las normativas del Ministerio del Interior (MININT). Con la implantación de este módulo se garantizará la administración, autenticación, autorización y auditoría en dicho sistema.

2.3. Disciplina de Requisitos

La tarea principal de la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto (Rodríguez Sánchez, 2015).

Para la obtención de los requisitos en la propuesta de solución se emplearon las siguientes técnicas:

- **Entrevista:** es una técnica de gran utilidad para obtener información cualitativa como opiniones o descripciones subjetivas de actividades. Es una técnica muy utilizada, y requiere una mayor preparación y experiencia por parte del analista (Sommerville, 2011). Esta técnica se aplicó durante las reuniones ejecutadas con el cliente, donde se formularon un conjunto de preguntas con el objetivo de entender las necesidades del mismo y concebir un lenguaje común entre el cliente y el equipo de desarrollo de la propuesta de solución.
- **Tormenta de ideas:** es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios (Pressman, 2010). Esta técnica se aplicó durante las reuniones de proyecto, luego de un diálogo entre los implicados, se obtuvo como resultado un conjunto de ideas con el fin de refinar las necesidades del cliente.

- **Revisión documental**: varios tipos de documentación, como manuales, reglamentos o cualquier información legal que defina como se realizará un proceso en el futuro sistema y reportes, pueden proporcionar al analista información valiosa con respecto a las organizaciones y a sus operaciones(Fuentes Castillo, y otros, 2016). Esta técnica se aplicó durante el estudio de la documentación entregada por el cliente para el análisis de los procesos. Además de las diferentes normas jurídicas establecidas a nivel de país, a tener en cuenta en el desarrollo de sistemas informáticos en cuanto a la seguridad de la información.

Una vez aplicadas estas técnicas se identificaron un conjunto de requisitos funcionales y no funcionales para la propuesta de solución, los mismos se describen en los siguientes epígrafes.

2.4. Requisitos funcionales

Los requisitos funcionales (RF) de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones(Pressman, 2010).A continuación, se numeran los RF detectados:

RF1. Modificar la función DELETE en el módulo de ajustes.
RF2. Modificar la función ADD en el módulo ajustes.
RF3. Modificar la función autenticación en el módulo LDAP.
RF4. Crear la función bloqueo por IP en el módulo LDAP.
RF5. Crear la función fortaleza de contraseña en el módulo LDAP.
RF6. Agregar la función cierre de sesión por inactividad en el módulo LDAP.
RF7. Modificar el sistema de guardado de trazas.
RF8. Modificar el sistema de trazas para que admita selecciones múltiples de modelos.
RF9. Configurar protocolo seguro para la administración de la seguridad.
RF10. Configurar módulo suma verificación para la salva y restauración de bases de datos.
RF11. Modificar módulo suma verificación para la importación y exportación de archivos.

2.5. Requisitos no funcionales

Los Requisitos no funcionales (RnF)son limitaciones sobre servicios o funciones queofrece el sistema. Incluyen restricciones tanto de temporización y del proceso dedesarrollo, como impuestas por los estándares. Los requerimientos no funcionalesse suelen aplicar aI sistema como un todo, más que a características o a serviciosindividuales del sistema (Sommerville, 2011). Teniendo en cuenta el objetivo que se persigue con la propuesta de solución a continuación, se detallan los RnFidentificados para el módulo de seguridad, para ello se agruparon de la siguiente forma:

Autenticación

RnF 1. El sistema debe garantizar la fortaleza de la autenticación sobre la base de que:

- Se debe solicitar la autenticación de acceso a través de usuario y contraseña.

- Todas las páginas y recursos requieren autenticación.
- Ante ochointentos fallidos de acceso el usuario queda inhabilitado.
- Solo puede acceder al sistema un mismo usuario a la vez desde lugares diferentes.
- Todos los controles de autenticación se deben realizan en el servidor.
- La re-autenticación es requerida antes de realizar cualquier operación crítica, definidas en el levantamiento de requisitos.
- El tiempo de vida de las sesiones de la base de datos es de35min.
- Las cuentas generadas por defecto y sus contraseñas son conocidas, se establece una contraseña segura para dichas cuentas, se bloquean y expiran.

Contraseñas

RnF 2. El sistema debe garantizar la fortaleza de las contraseñas sobre la base de que:

- Las contraseñas no se almacenarán en texto plano o en espacios lógicos que permitan el acceso o modificación por personas no autorizadas.
- La longitud mínima debe ser de ocho caracteres y debe contener combinaciones de letras minúsculas, letras mayúsculas, números y caracteres especiales.
- La contraseña se cambiará la primera vez de uso.
- Se debe controlar el ciclo de vida de la contraseña, con un tiempo máximo de 90 días.
- Se debe controlar el historial de contraseñas, con un mínimo de 24 contraseñas.

Autorización

RnF 3. Se debe definir una jerarquía de usuarios en correspondencia con las acciones que puedan realizar, dotando a los usuarios solo los privilegios que necesitan. No deben existir usuarios con privilegios de DBA, o sea, superadministrador

RnF 4. Las políticas de acceso deben ser definidas por el administrador del sistema, los permisos seránlimitados basados en los roles definidos y el usuario no podrá gestionarlos. Dichos permisos seharán corresponder con los niveles de acceso a la información que se define como clasificada, y se especifican y gestionan desde la etapa de diseño (estáticos), o en la etapa de explotación (dinámicos).

RnF 5. La creación de los roles se debe realizar bajo la política de mínimo privilegio, siguiendo el principio de menor privilegio, que permitan operaciones DML (INCERT, UPDATE, DELETE) y de selección (SELECT). En caso que se necesiten privilegios avanzados (CREATE, DROP, ANY), se justificaron.

RnF 6. El privilegio de borrado de trazas, no lo debe poseer ningún usuario, incluyendo el administrador del sistema.

RnF 7. La navegación por los directorios debe ser deshabilitada por defecto.

RnF 8. Todo el control de acceso debe ser asegurado del lado del servidor.

RnF 9. Los ficheros que se generan con la instalación del sistema gestor de base de datos se deben proteger contra usos no autorizados.

RnF 10. Se deberá garantizar que exista un correcto mecanismo de manejo de sesiones, lo que implicará:

- Se debe utilizar la implementación del control de sesiones que posee Odoo como *frameworks*.
- Las sesiones en el servidor deben ser destruidas cuando el usuario se desconecta.
- Las sesiones después de diez minutos(10min) de inactividad se deben deshabilitar.
- Todo acceso que requiera autenticación, debe poseer un vínculo que permite cerrar sesión.
- Solo se debe permitir el envío del identificador de sesión en cookies, prohibiéndose el uso de URLs[4], mensajes de error o trazas.
- No se debe soportar restructura de URL de las cookies de sesión.
- El indicador de sesión se debe cambiar en cada sesión indicada por el usuario.
- Los identificadores de las sesiones se deben cambiar en la re-autenticación.
- El identificador de la sesión se debe cambiar o blanquear después de terminar sesión.

Confidencialidad e integridad de los datos

RnF 11. Se debe garantizar el uso de servicios de red, puertos y protocolos acordes con las políticas deseguridad trazadas por la entidad que soportará el despliegue de la aplicación.

Trazas y auditorías

RnF 12. Se debe crear un rol de súper administrador con los permisos necesarios para examinar y procesar las trazas generadas en el sistema.

RnF 13. Se deben guardar trazas de todas las operaciones realizadas por el usuario sobre la solución, así como otros eventos de interés, que permitirán determinar funcionamientos incorrectos de la solución, así como resolver incidentes de seguridad.

RnF 14. Se debe garantizar para cada usuario un mecanismo de recolección de trazas mediante:

- Traza para auditoria funcional: registro de la identidad del usuario autenticado que invocó el sistema, tipo de operación realizada, fecha, hora, datos de la máquina desde donde fue realizado (dirección IP).
- Traza para la auditoria de seguridad: además de los datos de la auditoría funcional, se registran otros datos como: si el evento tuvo éxito o fallo, descripción del evento, nivel de seguridad del evento.

4 Uniform Resource Locator (Localizador Uniforme de Recursos). Se trata de la secuencia de caracteres que sigue un estándar y que permite denominar recursos dentro del entorno de Internet para que puedan ser localizados.

RnF 15. La ubicación de las trazas de auditoría debe ser protegidas para impedir violaciones. No se debe asignar el privilegio de borrado de trazas a ningún usuario.

RnF 16. Se debe garantizar en el sistema una política de rotación de salvas o histórico de trazas cada cierto tiempo, que no permiten el acceso y/o modificación por personas no autorizadas.

RnF 17. Se deben generar trazas obligatoriamente para los siguientes eventos:

Tabla 1 Eventos para los cuales se deben generar trazas

Evento	Tipo de traza
➢ **Gestión de usuarios y permisos.** ➢ **Inicio/cierre de sesión.**	Auditoría de Seguridad Auditoría Funcional.
➢ **Manejo de datos que incluyen creación, modificación y eliminación de datos.**	Auditoría Funcional.
➢ **Errores de la solución informática.** ➢ **Errores de conexión a la Base de Datos.** ➢ **Intento de operaciones incorrectas en la Base de Datos.** ➢ **Intento de acceso a módulos no asignado al usuario.** ➢ **Fallos de validación de entradas.**	Auditoría de Seguridad

Una vez identificados los requisitos de la propuesta de solución, se procedió a la encapsulación de los mismo a través de las Historias de usuario, según lo que establece la metodología AUP-UCI en su escenario cuatro (4). A continuación, se describe esta actividad.

2.6. Historias de usuarios

Las Historias de Usuario (HU), son pequeñas descripciones de los requerimientos de un cliente. Su utilización es común cuando se aplica marcos de entornos ágiles. Además, construyen entendimiento compartido a través de la colaboración, con palabras e imágenes. En otras palabras, son discusiones acerca de las soluciones a problemas para la organización, los clientes y los usuarios, que llevan a acuerdos sobre los cuales construir lo deseado(TENSTEP.INC, 2016).

A continuación, se describe una HU de prioridad Alta para el cliente, obtenida en el desarrollo de la propuesta de solución, el resto se pueden consultar en el artefacto creado por el autor de la presente investigación:

Tabla 2 Historia de usuario RF4 Crear la función bloqueo por IP en el módulo LDAP

Número: 1	**Requisito:**Crear la función bloqueo por IP en el módulo LDAP
Programador: Victor Manuel Sarduy Horta	**Iteración Asignada:** 1

Prioridad: Alta	**Tiempo Estimado:**6 horas
Riesgo en Desarrollo:	Tiempo Real: 2 horas
Descripción: El administrador da clic en configuración, selecciona el módulo Autenticación LDAP, una vez en el módulo selecciona la opción ´Crear¨ y podrá editar los parámetros para la autenticación de los usuarios, entre ellos dirección de IP y puerto del servidor LDAP, usuario y contraseña del servidor LDAP, permite crear usuarios y además la descripción IP del usuario autenticado que puede ser detectada automáticamente. De esta manera se asegura que un usuario solo se autentique en una única dirección de IP.	
Observaciones: N/A	
Prototipo elemental de interfaz gráfica de usuario: 	

Tabla 3 Historia de usuario RF6 Agregar la función cierre de sesión por inactividad en el módulo LDAP

Número: 2	**Requisito:**Agregar la función cierre de sesión por inactividad en el módulo LDAP.
Programador: Victor Manuel Sarduy Horta	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo Estimado:**7 horas
Riesgo en Desarrollo:	Tiempo Real:3 horas
Descripción: Una vez autenticado el usuario en el sistema y transcurridos los 10 minutos de inactividad se cierra automáticamente la sesión. De esta forma se garantiza menos tráfico de red, robo o perdida de información. Para entrar al sistema el usuario debe autenticarse nuevamente.	
Observaciones: N/A	

Prototipo elemental de interfaz gráfica de usuario:

2.7. Disciplina de Análisis y diseño

En esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales (Rodríguez Sánchez, 2015).

2.7.1. Arquitectura de software

El diseño arquitectónico es la primera etapa en el proceso de construcción del software. Constituye el enlace crucial entre el diseño y la ingeniería de requerimientos, ya que identifica los principales componentes estructurales en un sistema y la relación entre ellos. Su salida consiste en un modelo que describe la forma en que se organiza el sistema como un conjunto de componentes en comunicación(Sommerville, 2011).La solución propuesta tiene como base la arquitectura de Odoo, la cual se basa en el uso del patrón arquitectónico Modelo Vista Controlador (MVC).

Este patrón separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.El modelo contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia. La vista, o interfaz de usuario, compone la información que se envía al cliente y los mecanismos de interacción con éste. El controlador, actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno (Álvarez, 2014).

En el caso particular de Odoo (GitHub, 2020):

- La capa modelo es definida por objetos Python cuyos datos son almacenados en una base de datos PostgreSQL. El mapeo de la base de datos es gestionado automáticamente por Odoo, y el mecanismo responsable por esto es el Modelo Objeto Relacional, (ORM por sus siglas en inglés deObjectRelationalModel)
- La capa vista describe la interfaz con el usuario. Las vistas son definidas usando XML, las cuales son usadas por el marco de trabajo del cliente web para generar vistas HTML de datos.
- Las vistas del cliente web ejecutan acciones de datos persistentes a través de la interacción con el servidor ORM. Estas pueden ser operaciones básicas como escribir o eliminar, pero pueden también invocar métodos definidos en los objetos Python del ORM.A esto se le refiere la lógica de negocio, o sea la capa controladora.

A continuación, se muestra de forma más detalla esta arquitectura:

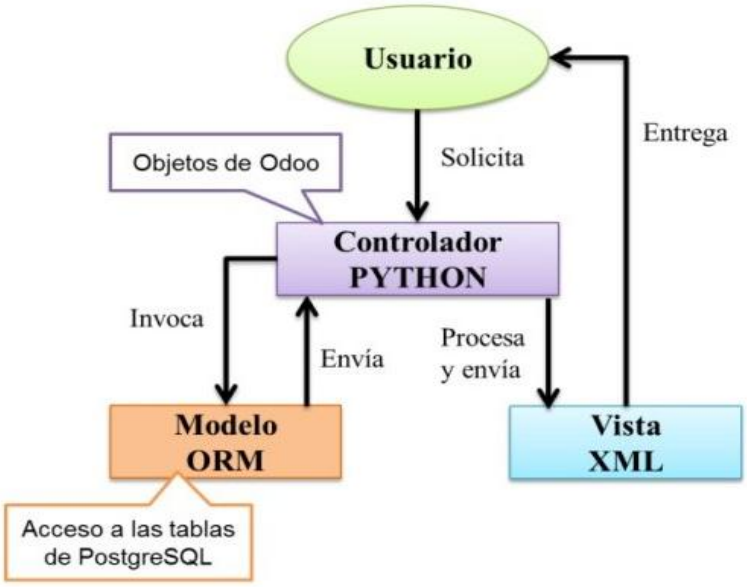


Figura 2:Arquitectura Modelo-Vista-Controlador del sistema Odoo

2.7.2. Diagramas de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. A diferencia del modelo conceptual, un diagrama de este tipo contiene las definicionesde las entidades del software en vez de conceptos del mundo real(Larman, 2002).

A continuación, se muestra el diagrama de clases de diseño

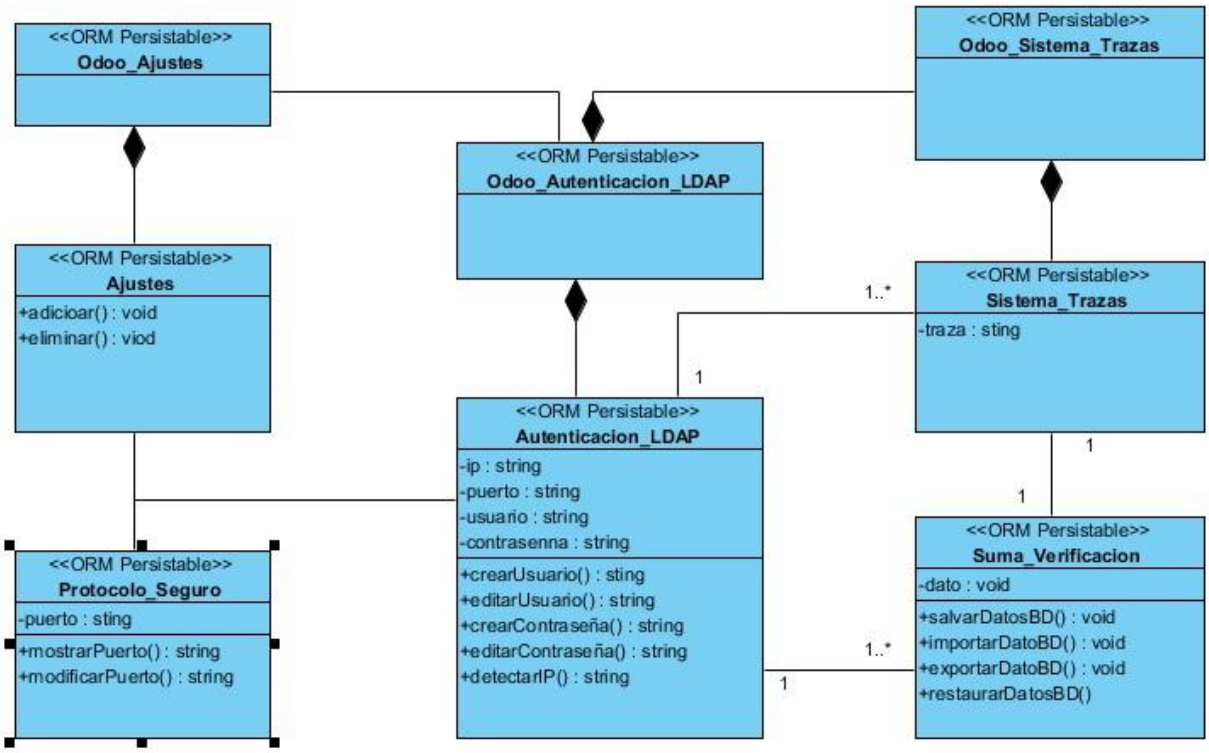


Figura 3 Diagrama de clases de diseño

2.7.3. Patrones de diseño

Un patrón de diseño describe una estructura que resuelve un problema de diseño en particular dentro de un contexto específico y en medio de fuerzas que pueden tener un impacto en la manera en que se aplica y utiliza el patrón. (Pressman, 2007)

2.8. Patrones Generales de Software para Asignación de Responsabilidades

Los Patrones Generales de Software para Asignación de Responsabilidades (GRASPpor sus siglas en inglés *General ResponsibilityAssignment Software Patterns*) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones(Larman, 2002). En este caso, los patrones GRASP utilizados son:

- **Experto:** este patrón establece la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo.
- **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.

Patrones GangofFour

En el libro "*DesignPatterns: Elementsof Reusable ObjectOriented Software*" escrito por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, se realiza una recopilación de 23

patrones de diseño aplicados usualmente por expertos diseñadores de software orientado a objetos. Desde luego que estos no son los inventores ni los únicos involucrados, pero fue el punto de partida para difundirse con más fuerza la idea de patrones de diseño**GangofFour** (GoF, que en español es la pandilla de los cuatro)(Gamma, y otros, 1994).

Estos patrones se dividen en diferentes categorías, para la propuesta de solución se tuvieron en cuanta los siguientes:

- **Patrones creacionales:** solucionan problemas de creación de instancias, ayudan a encapsular y abstraer dicha creación
  - **Factory Method***(*Método de fabricación*):* Expone un método de creación, delegando en las subclases la implementación de este método. Esto se evidencia en cada una de las clases del modelo que son las encargadas de inicializar y configurar cada uno de sus objetos.
- **Patrones de comportamiento:** son soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.
  - **Mediator (mediador):** Objeto que encapsula cómo otro conjunto de objetos interactúa y se comunican entre sí.Se evidencia en la comunicación entre dos objetos, al utilizar el patrón mediador se disminuye la dependencia entre estos objetos evitando que se relacionen entre ellos y garantizando mantenibilidad al módulo.

2.9. Diseño de la base de datos

El diseño de una base de datos consiste en definir la estructura de los datos que debe tener un sistema de informacióndeterminado. Para ello se suelen seguir por regla general unas fases en el proceso de diseño, definiendo para ello el modelo conceptual, el lógicoy el físico(Krasis Consulting S.L.U., 2020).

- En el diseño conceptual se hace una descripción de alto nivel de la estructura delabase de datos, independientemente del SGBD (Sistema Gestor de Bases de Datos) que se vaya a utilizar para manipularla. Su objetivo es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar dicha información.
- El diseño lógico parte del resultado del diseño conceptual y da como resultado una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de SGBD. El diseño lógico depende del tipo de SGBD que se vaya a utilizar, se adapta a la tecnología que se debe emplear, pero no depende del producto concreto. En el caso de bases de datos convencionales relacionales (basadas en SQL para entendernos), el diseño lógico consiste en definir las tablas que existirán, las relaciones entre ellas, así como sus atributos, llaves primarias y llaves foráneas.
- El diseño físico parte del lógico y da como resultado una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos.

Con el objetivo de describir el contenido de información de la base de datos, se muestra el modelo conceptual de dicha propuesta:

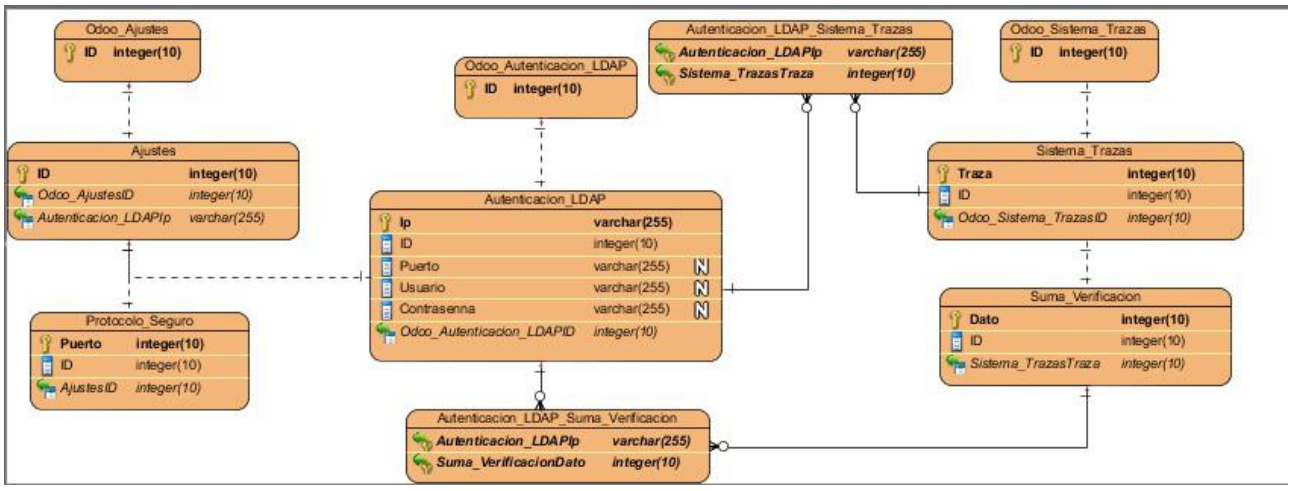


Figura 4 Modelo conceptual

2.10. Disciplina de Implementación

A partir de los resultados del análisis y diseño se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares. Al reutilizar componentes software ya implementados se lleva a cabo el desarrollo necesario para ajustar a los requisitos actuales y posteriormente realizar la integración de los componentes (Rodríguez Sánchez, 2015).

La implementación se basa en cómo organizar y desarrollar los componentes basándose en la fase de análisis y diseño, tomando como referencia los artefactos generados en la misma. En este sentido, se definen estándares de codificación con el objetivo de obtener un estilo de programación homogéneo.

Según(Metodología de desarrollo para la Actividad Productiva de la UCI, 2015) en la implementación se construye el sistema a partir de los resultados del Análisis y Diseño.

2.10.1. Estándares de codificación

Un estándar de codificación son reglas que se siguen para la escritura del código fuente. De tal manera que otros programadores puedan identificar las variables, las funciones o métodos. 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 (Álvarez, 2010).

A continuación, se exponen más detalladamente, los estándares de codificación seguidos en el desarrollo de la propuesta de solución:

- Se usará 4 (cuatro) espacios por indentación.
- Las líneasdecontinuación se alinearán verticalmenteconelcarácterquesehautilizadoparéntesis, llaves, corchetes.

- El paréntesis / corchete / llave que cierre una se alineará con el primer carácter que no sea un espacio en blanco
- Todas las líneas tendrán un máximo de 79 caracteres.
- Se separará las funciones de alto nivel y definiciones de clase con dos líneasen blanco.
- Las definiciones de métodos dentro de una clase se separarán por unalínea en blanco.

2.11. Validación del diseño

Una métrica es un instrumento que cuantifica un criterio y persigue comprender mejor la calidad del producto, estimar la efectividad del proceso y mejorar la calidad del trabajo realizado al nivel del proyecto.

Para la evaluación de la calidad del diseño propuesto para la solución se hizo un estudio de las métricas básicas inspiradas en la calidad del diseño orientado a objeto, en el mismo se abarcan atributos de calidad que permiten medir la calidad del diseño propuesto. Dentro de estos se encuentran (Sifontes y Avila 2015):

- **Responsabilidad:** consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- **Complejidad de implementación:** consiste en el grado de dificultad que implica la implementación de un diseño de clases determinado.
- **Reutilización:** consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.
- **Acoplamiento:** consiste en el grado de dependencia o interconexión de una clase o estructura de clase con otras, está muy ligada a la característica de Reutilización.
- **Complejidad del mantenimiento:** consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costos y la planificación del proyecto.
- **Cantidad de pruebas:** consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad del producto diseñado. Las métricas concebidas como instrumento para evaluar la calidad del diseño y su relación con los atributos de calidad definidos son las siguientes:

2.11.1. TOC (Tamaño Operacional de Clase)

Se refiere al número de métodos pertenecientes a una clase. La siguiente tabla muestra los atributos que forman parte de esta métrica y el modo en que se afectan.

Tabla 4. Tamaño operacional de la clase (TOC)

Atributo que afecta	Modo en que lo afecta
Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de Implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.

Reutilización	Un aumento del TOC implica una disminución en el grado de reutilización de la clase.

Esta métrica está determinada por los atributos: Responsabilidad, Complejidad de implementacióny la Reutilización, existiendo una relación directa con los dos primeros e inversa con el últimoantes mencionado.

La tabla que se muestra a continuación contiene el rango de valores para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización). La variable “Prom" indica el promedio de operaciones por cada clase.

Tabla 5. Rango de valores para la evaluación técnica del TOC

Atributos	Categoría	Criterio
Responsabilidad	Baja	<= Prom
	Media	Entre Prom y 2*Prom
	Alta	> 2* Prom
Complejidad de Implementación	Baja	<= Prom
	Media	Entre Prom y 2*Prom
	Alta	> 2* Prom
Reutilización	Baja	<= Prom
	Media	Entre Prom y 2*Prom
	Alta	> 2* Prom

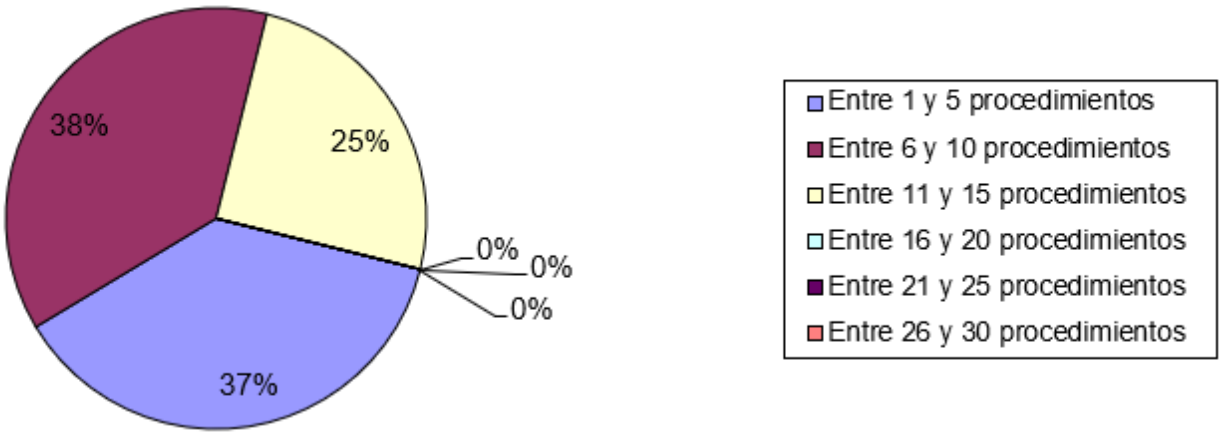
La siguiente tabla muestra los umbrales de la métrica TOC.

Tabla 6. Umbrales de la métrica TOC

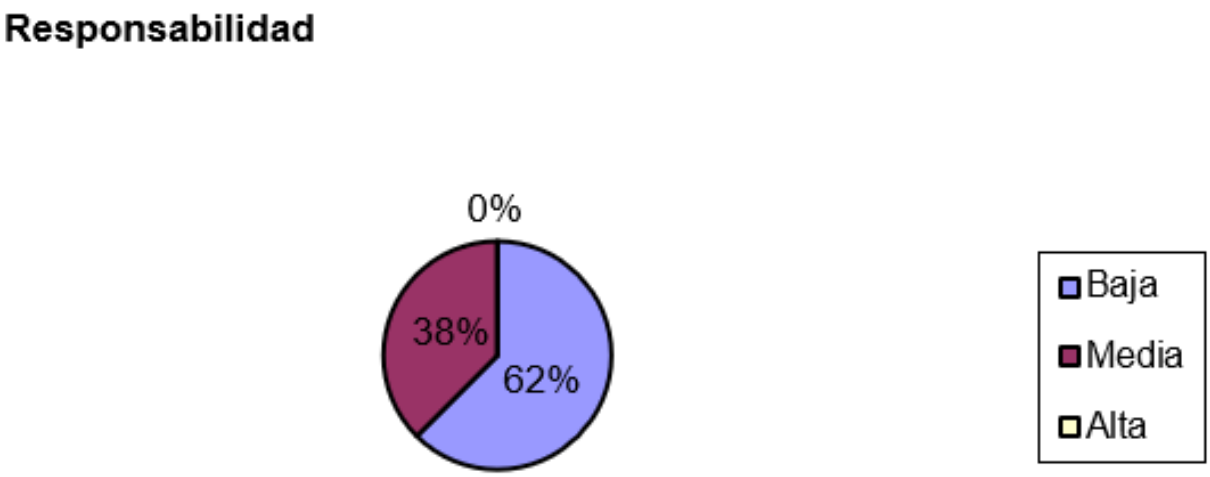
Tamaño operacional de clase	Criterio
Pequeño	<= Prom
Medio	Entre Prom y 2*Prom
Grande	> 2* Prom

2.11.2. Resultados del instrumento de evaluación de la métrica TOC.

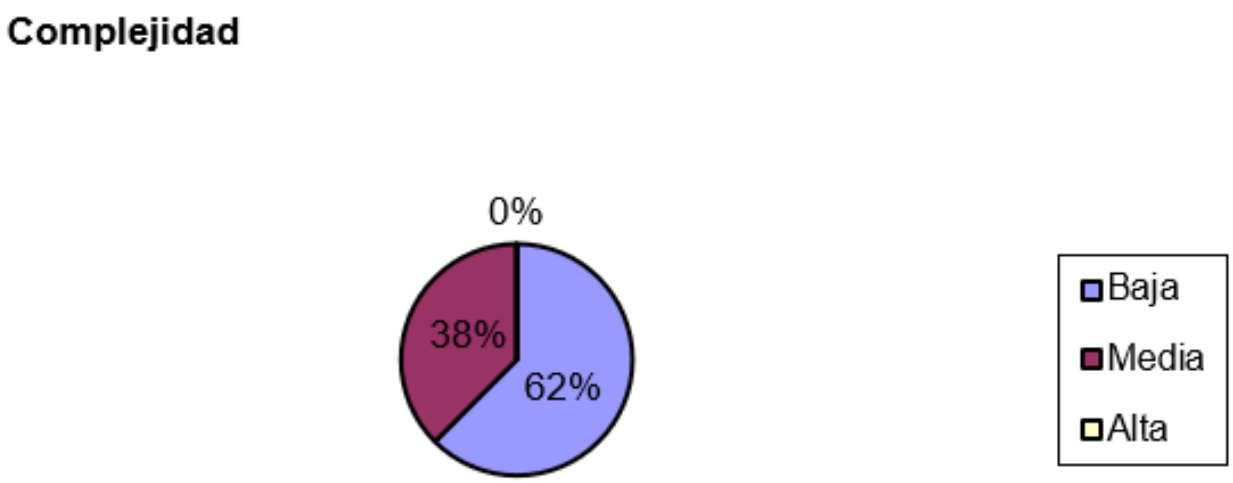
Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.



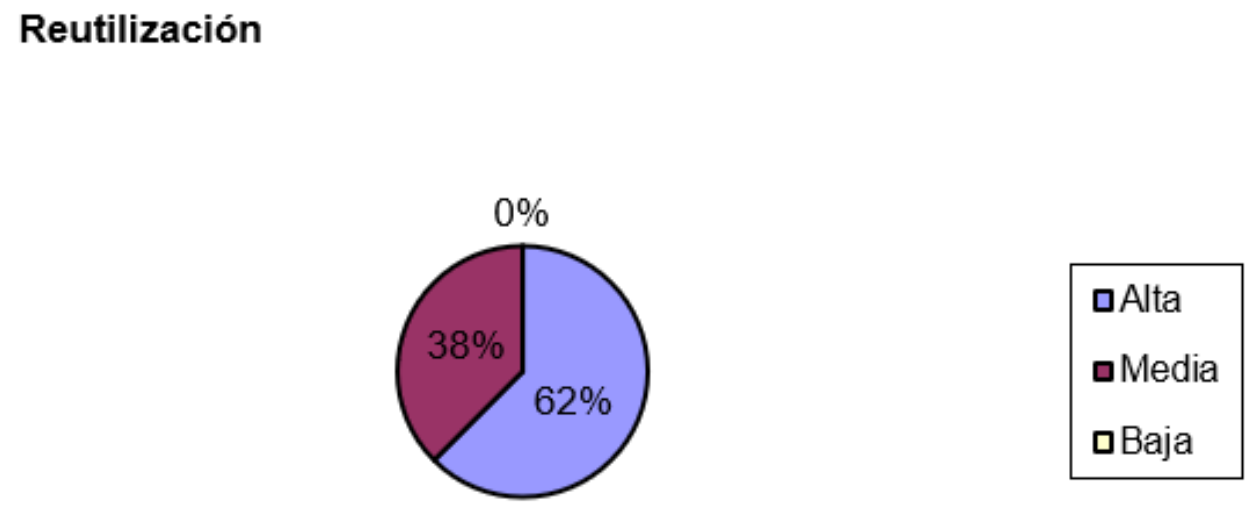
Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.



Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación.



Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización



Al analizar los resultados obtenidos luego de aplicar el instrumento de medición de la métrica TOC, se puede concluir que el diseño propuesto para el sistema es simple y tiene una calidad aceptable, teniendo en cuenta que la mayoría de las clases (75%) posee menos cantidad de operaciones que la media registrada en las mediciones. Los atributos de calidad se encuentran en un nivel satisfactorio en el 62% de las clases, de manera que se puede observar cómo se fomenta la Reutilización (elemento clave en el proceso dedesarrollo de software) y cómo están reducidas en menor grado la Responsabilidad y la Complejidad de implementación.

2.11.3. RC (Relaciones entre Clases)

Esta métrica está dada por el número de relaciones de uso de una clase. La siguiente tabla muestra los atributos pertenecientes a esta métrica y el modo en que se afectan.

Tabla 7. Relaciones entre las clases (RC)

Atributo que afecta	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del Acoplamiento de la clase.
Complejidad del mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.
Reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la clase.

Esta métrica está determinada por los atributos: Acoplamiento, Complejidad de mantenimiento, Cantidad de pruebas y Reutilización, existiendo una relación directa con los tres primeros e inversa con el último antes mencionado.

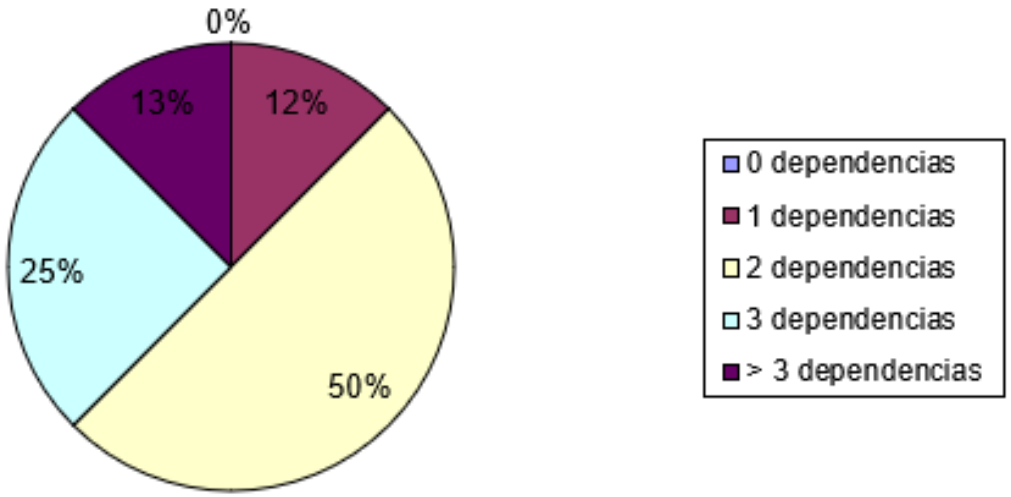
La siguiente tabla muestra el Rango de valores para la evaluación técnica de los atributos de calidad(Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de Pruebas) relacionadoscon la métrica RC. La variable Prom indica el promedio de relaciones entre clases.

Tabla 8. Rango de valores para la evaluación técnica de RC

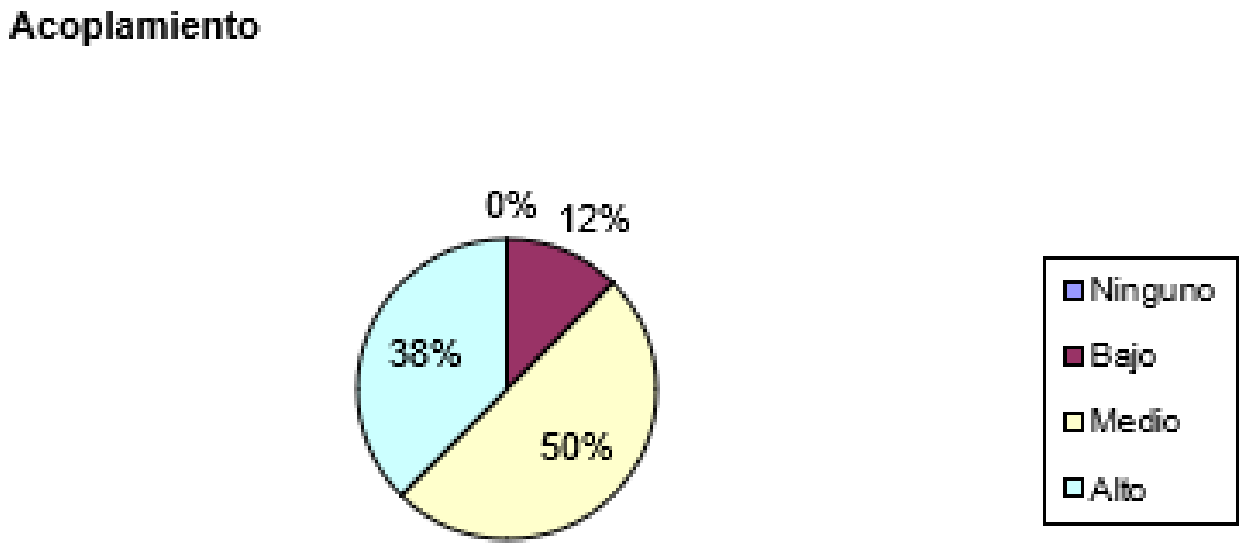
Atributos	Categoría	Criterio
Acoplamiento	Baja	<= Prom
	Media	Entre Prom y 2*Prom
	Alta	> 2* Prom
Complejidad del mantenimiento	Baja	<= Prom
	Media	Entre Prom y 2*Prom
	Alta	> 2* Prom
Cantidad de pruebas	Baja	<= Prom
	Media	Entre Prom y 2*Prom
	Alta	> 2* Prom
Reutilización	Baja	<= Prom
	Media	Entre Prom y 2*Prom
	Alta	> 2* Prom

2.11.4. Resultados del instrumento de evaluación de la métrica Relaciones entre Clases

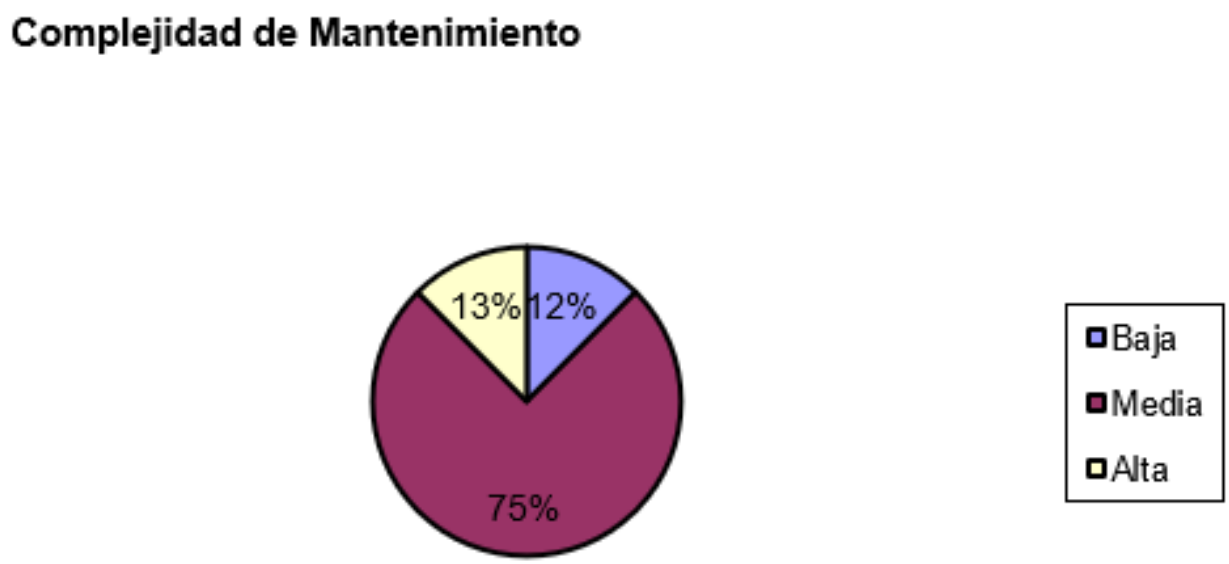
Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.



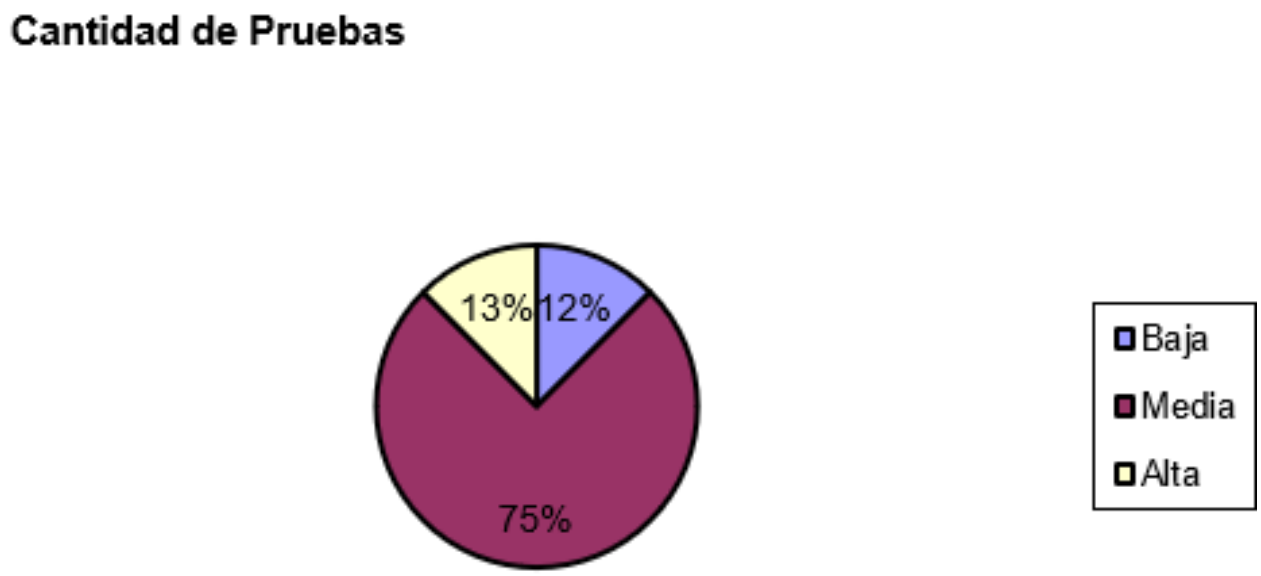
Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.



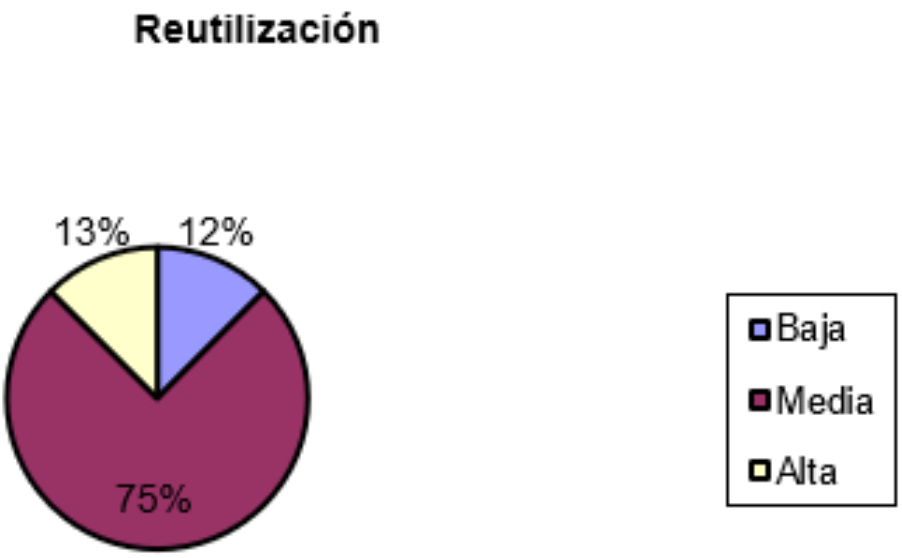
Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento.



Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas.



Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.



Al analizar los resultados obtenidos luego de aplicar el instrumento de medición de la métrica RC, se puede concluir que el diseño propuesto para el sistema es simple y tiene una calidad aceptable, teniendo en cuenta que la mayoría de las clases (62%) poseen 2 o menos dependencias respecto a otras. Los atributos de calidad se encuentran en un nivel satisfactorio, en el 62% de las clases el grado de acoplamiento es mínimo, la Complejidad de mantenimiento, la Cantidad de pruebas y la Reutilización se comportan favorablemente para un 87% de las clases.

2.12. Conclusiones parciales

- La descripción de la propuesta de solución facilitó abstraerse del problemaen el que se enmarca la presente investigación, y a su vez, identificar cómo interactúa el sistema en el cual se desenvuelve la solución.
- La aplicación de diferentes técnicas para la obtención de requisitos favoreció suespecificación y descripción,permitiendo documentarsetoda la información relativa a la propuesta de solución.
- El diseño de la propuesta de solución permitió concebir los elementos necesarios para la implementación del módulo de seguridad para el Sistema de Importación TECNOTEX.
- Los estándares de codificación definidos posibilitaron obtener un estilo de programación homogéneo, permitiendo a los participantes un mayor entendimiento e interpretación.

Capítulo 3: Validación de la propuesta de solución

En el presente capítulo, una vez concluida la fase de análisis y diseño de la propuesta de solución se procede a la implementación de las clases y ejecución de casos de prueba que evalúen las funcionalidades de la herramienta de configuración. Se determina si las funcionalidades implementadas cumplen con las características establecidas y con las descripciones de las HU anteriormente expuestas, realizando las iteraciones necesarias para cumplir satisfactoriamente los casos de pruebas elaborados.

3.1.Pruebas al sistema

Los sistemas de software hoy en día son parte importante e integral en la gran mayoría de las actividades diarias, es por ello que se debe tener en cuenta que los sistemas o aplicaciones son creadas, desarrolladas e implementadas por seres humanos y, por ende, en cualquiera de sus etapas de creación se puede presentar una equivocación que puede llevar a defectos en las aplicaciones. Las pruebas son necesarias porque con ellas se puede ayudar a reducir los riesgos en las aplicaciones y lograr de esta manera que se identifiquen los defectos antes de que se ejecuten (Paz 2016).

Una vez concluida la disciplina de implementación y con el objetivo de validar el correcto funcionamiento de los requisitos implementados se realizan pruebas de caja negra, caja blanca y aceptación.

3.1.1 Pruebas de Caja Blanca

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo(Pressman, 2010).

3.1.2 Técnica del camino básico

Es una técnica de prueba de caja blanca propuesta inicialmente por Tom McCabe. El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución(Pressman, 2002)

La complejidad ciclomática es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba del camino básico, el valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y nos da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez(Pressman, 2002)

```
sessions = {} 1
for name in conf.sections(): 2
    sessions[name] = {
        'type': conf.get(name, 'type'),
        'host': conf.get(name, 'host'),
        'protocol': conf.get(name, 'protocol'),
        'port': conf.getint(name, 'port'),
        'timeout': conf.getfloat(name, 'timeout'),
        'user': conf.get(name, 'user'),
        'passwd': conf.get(name, 'passwd'),
        'database': conf.get(name, 'database'),
    }
return sessions 4
```

3

Figura 5 Funcionalidad del método sesiones

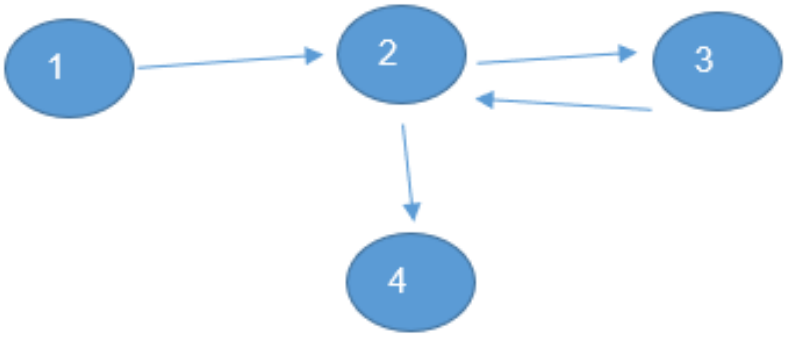


Figura 6 Grafo resultante de aplicar técnica Camino Básico

V (G) = A – N + 2 V (G) = Complejidad ciclomática

= 4 – 4 +2 A = Cantidad de aristas del grafo

= 2 N = Cantidad de nodos del grafo

V (G) = P +1 P= Nodos predicados.

= 1+1

= 2

V (G) = Regiones Regiones= Áreas delimitadas por nodos

= 2 y aristas del grafo

Cantidad de caminos básicos = 1

Camino 1: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12.

Tabla 9. Caso de prueba para el camino 1

Descripción	Verificar los resultados de la función del módulo sesiones
Condición de ejecución	for name in conf.sections () :
Entrada	sessions{}
Resultado	Devuelve la sesión autenticada con todos sus parámetros necesarios para el sistema.

Luego de haberle aplicado el método del camino básico a la funcionalidad anterior se comprobó que la sentencia es ejecutada al menos una vez. El valor calculado como complejidad ciclomática definió el número de caminos independientes del conjunto básico, lo que facilitó el límite superior para el número de pruebas que se deben realizar.

3.2.1. Pruebas de Caja Negra

Las pruebas de caja negra son una forma de derivar y seleccionar condiciones, datos y casos de prueba a partir de los requisitos del sistema. Las pruebas de caja negra no utilizan ninguna información interna de los componentes de software o sistemas que se van a probar, sino que consideran el comportamiento del software desde el punto de vista de un observador externo (Como los usuarios del sistema). Son utilizadas para realizar pruebas funcionales, basadas en las funciones o características del sistema y su interacción con otros sistemas o componentes. Las funciones del software son descritas en los documentos de especificación de requisitos y en las HU (pmoinformatica.com 2016).

3.2.2. Partición de Equivalencia

Esta es una técnica de prueba de Caja Negra que divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. El diseño de estos casos de prueba para la partición equivalente se basa en la 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 la cual regularmente es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (Pressman 2005). A continuación, se muestran algunos de los casos de prueba empleados para validar el correcto funcionamiento del sistema:

Tabla 10.Diseño de caso de prueba RF Bloqueo por IP

Escenario	Descripción	Variables	Respuestadel sistema	Flujo central
EC 1.1	Se intenta	direcconIP	1.1 El sistema	1.1 El usuario introduce las

Bloqueo por dirección IP	bloquear la dirección ip del usuario autenticado en el sistema	192.168.123.45	autentica al usuario y bloquea la dirección IP de la PC.	credenciales de autenticación: usuario y contraseña. Luego el sistema verifica que estén correctos y detecta la dirección IP de la PC, una vez hecho esto bloquea esa dirección IP imposibilitando que el usuario autenticado pueda entrar al sistema en una nueva PC con diferente dirección IP mientras tenga su sesión activa en otra.
EC 1.2 Bloqueo por IP / Dirección IP no detectada	Se intenta bloquear la dirección ip del usuario autenticado en el sistema, pero no se detecta la misma	direcconIP N/A	1.2 El sistema autentica al usuario y no bloquea la dirección IP de la PC.	1.2 El usuario introduce las credenciales de autenticación: usuario y contraseña. Luego el sistema verifica que estén correctos y no detecta la dirección IP de la PC.
EC 1.3 Bloqueo por IP / Usuario autenticado en dos PC	Se intenta bloquear la dirección ip del usuario autenticado en el sistema en dos PC diferentes	direcconIP 192.168.123.45	1.3 El sistema autentica al usuario y bloquea la dirección IP de la PC en que primero introdujo las credenciales de autenticación	1.3 El usuario introduce las credenciales de autenticación: usuario y contraseña. Luego el sistema verifica que estén correctos y detecta la dirección IP de la PC, una vez hecho esto bloquea esa dirección IP imposibilitando que el usuario autenticado pueda acceder al sistema en la segunda PC. El sistema muestra un mensaje de alerta al usuario informándole que ya está autenticado en el sistema en otra PC

Tabla 11 Diseño de caso de prueba RF Cierre de cesión por inactividad

Escenario	Descripción	Variables	Respuesta del sistema	Flujo central
EC 2.1 Cierre de Sesión por inactividad	Se intenta cerrar la sesión del usuario autenticado transcurridos los 10 minutos de inactividad.	CecionLog Falce	2.1 El sistema transcurridos 10 minutos de inactividad cierra automáticamente la sesión.	2.1 El usuario ya autenticado pasa un periodo de 10 minutos de inactividad en él sistema, transcurrido este tiempo, su sesión cierraautomáticamente. El sistema muestra un mensaje de alerta al usuario informándole que su sesión expiró por inactividad.
EC 2.2	Se intenta cerrar	CecionLog	2.2 El sistema	2.2 El usuario ya autenticado

Cierre de Sesión por inactividad/ Usuario inactivo Sesión no cerrada	la sesión del usuario autenticado transcurridos los 10 minutos de inactividad. Pero no se logra.	True	transcurridos 10 minutos de inactividad no cierra automáticamente la sesión.	pasa un periodo de 10 minutos de inactividad en él sistema, transcurrido este tiempo, su sesión cierra automáticamente.

3.3.Valoración del procedimiento mediante el criterio de expertos

El método de expertos permite valorar la factibilidad y viabilidad de aplicación del procedimiento general, sus componentes y los procedimientos específicos propuestos, para su implementación en lasorganizaciones orientadas a proyectos en Cuba.
La aplicación se llevó a cabo a través del cumplimiento de los pasos siguientes:

- Identificación de los posibles expertos.
- Selección de los expertos.
- Realización de la consulta a los expertos.
- Procesamiento y valoración de la información obtenida.

Para identificar los posibles expertos se tuvieron en cuenta, la experiencia profesional en relación con el objeto de investigación, la participación en investigaciones relacionadas con esta temática, el dominio teórico de la temática, la preparación académica y científica, y la experiencia; de modo que estuvieran en capacidad de ofrecer valoraciones y hacer recomendaciones pertinentes. Para la aplicación del método se identificaron 13 posibles expertos de los que fueron seleccionados 10. A continuación se recoge una breve caracterización de los expertos seleccionados.

Tabla 12 Caracterización del grupo de expertos

Relevancia	**Cantidad**
Doctores en ciencias	6
Máster en ciencia	3
Especialista de calidad de software	1
Total	**10**

- De ellos, 5 tienen 7 o más años de experiencia en el proceso de administración y control de sistemas.
- Tres poseen la categoría de Investigador Titular.
- Se encuentran representados diversos centros orientados a proyectos entre los que están CEIGE y CEGEL.

Figura. 1Caracterización del grupo de experto

Como resultado del análisis de la población objeto de aplicación de la encuesta, se decidió aplicar un muestreo a conveniencia. Consiste en una técnica de muestreo no probabilístico donde los sujetos son seleccionados, dada la conveniente accesibilidad y proximidad de los sujetos para el investigador.

En correspondencia con estos elementos se diseñó un cuestionario orientado a obtener información sobre la valoración de los expertos, para ello se propone un cuestionario de un total de 5 preguntas de respuesta cerrada, facilitándose de esta forma el procesamiento y la cuantificación. La vía utilizada para la aplicación fue el cuestionario impreso.

Las respuestas a las preguntas están representadas en una escala de Likert. Es una escala psicométrica comúnmente utilizada en cuestionarios y se considera la escala de uso más amplio en encuestas para la investigación, principalmente en ciencias sociales.

Para el procesamiento de la información obtenida en las encuestas se hizo corresponder un valor numérico a cada posible resultado de la escala utilizada. Estos valores utilizados se muestran en la siguiente tabla:

Respuesta	Valor
Mucho	5
Bastante	4
Poco	3
Muy poco	2
Nada	1

Figura. 2 Valores asociados a cada respuesta

Procedimiento para determinar el grado de consenso entre expertos:

A partir de la tabla siguiente se puede afirmar que existe un alto grado de concordancia en el criterio de los expertos en todas las preguntas realizadas. En todos los casos el coeficiente de variación muestra valores por debajo de 0,22 lo que sustenta la afirmación anterior. De este modo se puede concluir que, de acuerdo al consenso en el juicio de expertos, la solución propuesta contribuye a mejorar el proceso de Sistema de Trazas. Destacando las preguntas 2 y 3 con un coeficiente de variación por debajo de 0,15.

Expertos	Preguntas				
	1	2	3	4	5
1	4	4	4	3	5
2	5	5	5	5	3
3	3	4	5	5	4
4	4	5	5	5	3
5	5	5	4	5	5
6	5	4	4	5	4
7	4	5	4	4	5
8	4	5	5	4	3
9	4	4	4	4	4
10	5	4	4	5	3
Cj media	4,30	4,5	4,4	4,5	3,9
Varianza	0,46	0,28	0,27	0,50	0,77
Coef. De Variación	0,16	0,12	0,12	0,16	0,22

Figura. 3 Resultados de la aplicación

3.4. Conclusiones parciales

Una vez culminada la etapa de implementación y haberle realizado las pruebas de caja negra a la herramienta de configuración, se puede concluir que:

- Se desarrollaron diferentes casos de prueba que permitieron darle cumplimiento a las HU especificadas (en el Capítulo 2), cumpliendo de manera satisfactoria las pruebasrealizadas.
- Se realizaron las pruebas de caja blanca y de caja negra, arrojando resultados satisfactorios con respecto a el cumplimiento de los requisitos funcionales y a la implementación de la propuesta de solución.

Conclusiones generales

- La aplicación de diferentes técnicas para la obtención de requisitos favoreció su especificación y descripción, permitiendo documentarse toda la información relativa a la propuesta de solución.
- El diseño de la propuesta de solución permitió concebir los elementos necesarios para la implementación del módulo de seguridad para el Sistema de Importación TECNOTEX.
- Los estándares de codificación definidos posibilitaron obtener un estilo de programación homogéneo, permitiendo a los participantes un mayor entendimiento e interpretación.
- Se obtuvo una aplicación desarrollada a partir del uso de tecnologías y herramientas actualizadas y en versiones libres
- La solución fue sometida a un proceso de pruebas de funcionamiento guiado por casos de pruebas, donde la realización de las mismas corroboró la solidez del sistema, cumpliendo con las características pedidas por el cliente.

Recomendaciones

Los objetivos generales de este trabajo fueron alcanzados, pero durante su desarrollo, han surgido ideas que sería recomendable tener en cuenta para su futuro perfeccionamiento:

- Continuar el desarrollo de este módulo, adicionándole nuevas funcionalidades y servicios que puedan satisfacer necesidades futuras de los clientes.
- Presentar los resultados de la investigación en eventos científicos.

Referencias bibliográficas

Alegsa. 2010. Definición de IDE . *Diccionario de Informática y Tecnología .* [En línea] 2010. http://www.alegsa.com.ar/Dic/ide.php.

Álvarez, Daniel José Salas. 2010. Estándares de codoficacion Java. [En línea] 2010. http://www.aves.edu.co/ovaunicor/recursos/view/265.

Álvarez, Miguel Angel. 2014. desarrolloweb.com. [En línea] 2014. https://desarrolloweb.com/articulos/que-es-mvc.html.

Bahit, Eugenia. 2012.*Python para principiantes.* Buenos Aires, Aregentina : Creative Commons, 2012.

Definición.de. 2019. Definición.de. [En línea] 2019. https://definicion.de/autorizacion/.

—. 2019. Definición.de. [En línea] 2019. https://definicion.de/administracion/.

FERNÁNDEZ ALONSO, Yeray, et al. 2014..*Personalización de módulos en OpenERP 7.0.* s.l. : PYME. , 2014.

—. 2014..*Personalización de módulos en OpenERP 7.0.* s.l. : PYME. , 2014.

Fuentes Castillo, Yordanka, Núñez de los Ríos, Madielennis y Rodríguez Lemus, Marianela. 2016.*Guía para aplicar técnicas para el levantamiento de información.* La Habana : Universidad de las Ciencias Informáticas, 2016.

Gamma, Erich, y otros. 1994.*"Design Patterns: Elements of Reusable Object Oriented Sofware".* s.l. : Grady Booch, 1994.

Gesoft-Informática. 2019. Gesoft Informática. [En línea] 2019. [Citado el: 15 de Noviembre de 2019.] http://www.gesoft.com.br/vista/desarrollo/personalizacion-de-software.jsp.

GitHub. 2020.*Odoo Development Essentials.* 2020.

HUGO. 2008.HUGO TESIS. [En línea] 2008. https://problema.blogcindario.com/2008/10/00014-marco-teorico.html.

Isabel Ramos. 2007. Técnicas cuantitativas para la gestión en la Ingeniería del Software. 2007.

JetBrains. 2020. JetBrains. *PyCharm.* [En línea] 2020. https://www.jetbrains.com/es-es/pycharm/.

Krasis Consulting S.L.U. 2020. campusMVP. [En línea] 2020. https://www.campusmvp.es/recursos/post/Disenando-una-base-de-datos-en-el-modelo-relacional.aspx.

La administración de sistemas informáticos, una alternativa a la formación del profesional en tecnologías de información y comunicaciones. **Valencia-Duque, Francisco Javier y Bermón-Angarita, Leonardo. 2018.** 25, Colombia : s.n., 2018, Revista de Educación en Ingeniería, Vol. XIII, págs. 44-49. ISSN 1900-8260.

Larman, Craig. 2002.*UML y Patrones: Introducción al análisis y diseño orientado a objetos.* 2ra. Mexico : Prentice Hall Hispanoamerica, S.A, 2002. 84-205-3438--2.

Levin, Jonathan. 2019. Pgadmin. [En línea] 2019. [Citado el: 30 de noviembre de 2019.] https://www.pgadmin.org.

Marco de trabajo para la gestión centralizada de trazas de seguridad usando herramientas de código abierto. **Porven Rubier, Rubier y Montesino Perurena, Raydel. 2015.** 3, s.l. : Revista Cubana de las Ciencias Informáticas, 2015, Vol. IX. ISSN: 2227-1899 | RNPS: 2301.

Merino, Julián Pérez Porto y María. 2008. Definición de. *Definición de Seguridad Informática.* [En línea] 2008. https://definicion.de/seguridad-informatica/.

Metodología de desarrollo para la Actividad Productiva de la UCI. 2015.*Metodología de desarrollo para la Actividad Productiva de la UCI.* 2015.

Metodologías actuales de desarrollo de software. **Rivas, Carlos Ignacio, y otros. 2015.** 5, México : ECORFAN, 2015, Revista de Tecnología e Innovación, Vol. II. ISSN 2410-3993.

MIC. 2007.*Resolución 127/2007 MIC. Reglamento de seguridad para las tecnologías de la información. Ministerio de la informática y las comunicaciones.* La Habana : MInisterio de las Comunicaciones, 2007.

Mirada contextual a los nexos entre las auditorías de información y las auditorías de conocimiento. **Ponjuán, Gloria. 2011.** 1, La Habana : s.n., 2011, Ciencias de la Información, Vol. 42. ISSN 0864 4659.

NTP-ISO/IEC-17799. 2007.*NORMA TÉCNICA PERUANA.* Lima : Comisión de Reglamentos Técnicos y Comerciales - INDECOPI, 2007.

Odoo Community Hub. 2019. Odoo Community Hub. [En línea] 2019. [Citado el: 1 de noviembre de 2019.] https://odoohub.wordpress.com/.

Ponjuan Dante, Gloria. 2012.*INTRODUCCIÓN A LA GESTIÓN DE INFORMACIÓN.* La Habana : Facultad de Comunicación. Universidad de la Habana, 2012.

PostgresSQL. 2010. PostgresSQL. [En línea] 2010. [Citado el: 16 de Noviembre de 2019.] http://www.postgresql.org.es/sobre_postgresql#intro.

Pressman, Roger. 2002.*Ingeniería del Software. Un enfoque práctico.* s.l. : Quinta Edición, 2002.

Pressman, Roger S. 2010.*Ingeniería del Software. Un Enfoque Práctico.* 2010.

Pressman, Roger S. 2010.*Ingeniería de Software. Un enfoque práctico.* Septima . México DF : McGraw-Hill INTERAMERICA EDITORES, 2010.

Pressman, RS. 2007.*Ingeniería de software. Un enfoque práctico.* 6ta. Nueva York : McGraw-Hill, 2007.

Python Software Foundation. 2010. Python Programming Language. *Python Programming Language-Official Website.* [En línea] 2010. http://www.python.org/.

RAE - Diccionario Real de la Academia Española. 2017. Real de la Academia Española. [En línea] 23 de Julio de 2017. https://dej.rae.es/lema/autenticaci%C3%B3n.

RedIRIS. 2008. RedIRIS. [En línea] 12 de Noviembre de 2008. [Citado el: 26 de Octubre de 2019.] http://www.rediris.es/cert/doc/unixsec/node14.html.

Rodríguez Sánchez, Tamara. 2015.*Metodología de desarrollo para la Actividad productiva UCI.* La Habana : Universidad de las Ciencias Informáticas, 2015.

Sierra, Antonio. 2013.*UML (Unified Medling Language) Lenguaje Unificado de Modelado.* 2013.

SIFONTES, R.S. y AVILA, Y.F., . 2015.*Desarrollo del Módulo Procesamiento de Audiovisuales para el sistema XABAL Arkheia 2.1 para la OAHCE. .* La Habana: Universidad de las Ciencias Informáticas. : s.n., 2015.

Sommerville, Ian. 2011.*Ingeniería de Software.* 9na. México : Addison-Wesley, 2011. ISBN: 978-607-32-0603-7.

TENSTEP.INC. 2016. [En línea] 2016. https://www.tenstep.ec/portal/articulos-boletin-tenstep/41-scrum/253-scrum-como-escribir-historias-de-usuarios-sin-morir-en-el-intento.

Visual Paradigm. 2016. Visual Parading Web site. [En línea] 2016. [Citado el: 30 de noviembre de 2019.] https://www.visual-paradigm.com/.

XML. 2016. XML.com. [En línea] 2 de marzo de 2016. http://www.xml.com/.

Yudith, Doina. 2019. Instituciones. *Direccion Nacional Seguridad y Proteccion MINSAP.* [En línea] 2019. [Citado el: 26 de Octubre de 2019.] https://instituciones.sld.cu/dnspminsap/seguridad-informatica/.

Anexos

Anexo 1: Valoración de las preguntas

Preguntas	Valoración de expertos
1. La solución propuesta contribuirá a garantizar la integridad de los datos en el Sistema de Importación de TECNOTEX.	4,3
2. La solución propuesta contribuirá a mejorar el control interno en el Sistema de Importación de TECNOTEX.	4,5
3. La solución propuesta contribuirá a gestionar las trazas en el Sistema de Importación de TECNOTEX.	4,4
4. La solución propuesta contribuirá a mejorar el método de autenticación en el Sistema de Importación de TECNOTEX.	4,5
5. La solución propuesta contribuirá a facilitar la importación, exportación y restauración de archivos en el Sistema de Importación de TECNOTEX.	3,9

Anexo 2: Encuesta de satisfacción

Estimado compañero(a):

Usted a sido seleccionado(a) como posible experto, la siguiente encuesta ha sido elaborada con el objetivo de recoger su juicio acerca del impacto de la solución del Módulo de Seguridad para el Sistema de Importación de TECNOTEX sobre los procesos relacionados con este módulo. Después de tener conocimiento de las funcionalidades del sistema, por favor responda cada pregunta con el mayor nivel de objetividad posible. Usted debe marcar solo un ítem por pregunta.

Preguntas:

1. En qué medida considera usted que la solución propuesta contribuirá a garantizar la integridad de los datos en el Sistema de Importación de TECNOTEX.
   Mucho ___ Bastante ___ Poco ___ Muy poco ___ Nada ___
2. Considera usted que la solución propuesta contribuirá a mejorar el control interno en el Sistema de Importación de TECNOTEX.
   Mucho ___ Bastante ___ Poco ___ Muy poco ___ Nada ___

3. En qué medida cree que la solución propuesta contribuirá a gestionar las trazas en el Sistema de Importación de TECNOTEX.
   Mucho ___ Bastante ___ Poco ___ Muy poco ___ Nada ___
4. Considera usted que la solución propuesta contribuirá a mejorar el método de autenticación en el Sistema de Importación de TECNOTEX.
   Mucho ___ Bastante ___ Poco ___ Muy poco ___ Nada ___
5. En qué medida cree que la solución propuesta contribuirá a facilitar la importación, exportación y restauración de archivos en el Sistema de Importación de TECNOTEX.
   Mucho ___ Bastante ___ Poco ___ Muy poco ___ Nada ___

¡Muchas Gracias por su colaboración!