



Facultad 1

Sistema para la planificación y control de inventarios en el proceso de personalización de documentos de identificación

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

Autor

Eduardo Baratutes González

Tutores

Ing. Vladimir Campos Kindelán

MSc. José Carlos Pérez Zamora

La Habana, Cuba. Junio de 2018

DECLARACION DE AUTORIA

Declaro por este medio que yo, Eduardo Baratutes González, con carnet de identidad 94021229861 soy el autor principal del trabajo de diploma titulado “Sistema para la planificación y control de inventarios en el proceso de personalización de documentos de identificación” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste se firma la presente declaración jurada de autoría en La Habana a los ____ días del mes de _____ del año 2018.

Firma del autor

Eduardo Baratutes González

Firma del tutor

Ing. Vladimir Campos Kindelán

Firma del tutor

MSc. José Carlos Pérez Zamora

AGRADECIMIENTOS

Primero quiero agradecerle a mi mamá, hermana y mi papá por todo lo que hacen por mí, en especial todo el sacrificio que han hecho en estos 5 años, sin ustedes nada de esto sería posible, pues lo son todo. A mi familia entera, a mi hermano Ale que siempre me apoyó a pesar de la distancia y que me ayudó en todo lo que pudo, muchas gracias mi hermano. A mi pulga que siempre me apoyó y me presionó bastante para que aprovechara el tiempo, muchas gracias mi amorcito. A mis tutores... Vladimir porque siempre estuvo para mí y siempre me dedicó todo el tiempo que no tenía para aclarar todas mis dudas y ser mi guía en este trabajo de diploma, muchas gracias profe por todo, a Jose Carlos porque en cuanto me fue asignado enseguida me brindó toda su ayuda y me dio todo el tiempo y todo el aliento del mundo para poder llegar hasta el final, muchas gracias profe, muchas gracias a los dos por ser tan pacientes y por ser tan exigentes conmigo. Muchas gracias también a los “tipos duros” de Django que me ayudaron en todo. También quiero agradecer a mis primos Irving y Dayana que fueron fundamentales y que estuvieron siempre para mí, aunque la distancia que nos separa es muy grande, gracias y sobre todo por hacerme parte de la familia UCI conformada por Alina, Annia, Elisabet, Guille, Reynier, Abel y Yadier, muchas gracias a ustedes por haberme ayudado en todo, desde que entré en primer año me ayudaron mucho e hicieron que en un lugar desconocido por completo me sintiera como si estuviera en casa, gracias a ustedes por todo. A todos los profesores que de una manera u otra ayudaron a mi formación como ingeniero, gracias a ustedes por todos los valores y por haber sido fundamental en toda la carrera. A mis compañeros de cuarto que hicieron que este viaje fuera inolvidable, especialmente a los caracoles, el equipo más completo de la universidad en todos los aspectos.

DEDICATORIA

Quiero dedicar este logro a mi mamá, hermana y mi papá por su apoyo incondicional, por confiar siempre y creer siempre que esto sería posible, muchas gracias por todo y por estar siempre para mí cuando más lo necesito. También a toda mi familia y a todas las personas que confiaron en mí y nunca me dijeron que no.

RESUMEN

El presente trabajo consiste en el desarrollo de un sistema de planificación y control de inventarios que mejore el proceso de personalización de documentos en la Oficina de Identificación de la Universidad de las Ciencias Informáticas (UCI). Se estudiaron varias aplicaciones que tienen estrecha relación con la propuesta de solución, y su análisis permitió determinar varios aspectos importantes a tener en cuenta a la hora de desarrollar la misma. El sistema de planificación y control de inventarios gestiona todo lo relacionado con los inventarios que se utilizan en dicho proceso, además de incorporar nuevas características como generar reportes y graficar el gasto de materiales. El desarrollo de la propuesta de solución estuvo guiado por la metodología AUP-UCI, se utilizó como herramienta para la implementación el marco de trabajo web Django en su versión 1.10.3, lenguaje de programación Python 3.4, implementado en Pycharm 2017.2.2, como Sistema Gestor de Bases de Datos PostgreSQL 9.4.4, Visual Paradigm como herramienta de modelado utilizando el lenguaje UML, para la maquetación se utilizó HTML 5 y CSS 3. Las pruebas de software arrojaron que la propuesta de solución es funcional y segura. Los resultados obtenidos mediante la aplicación del método diagnóstico mostraron un sistema que mejora la planificación y el control de los inventarios utilizados en el proceso de personalización de documentos de identificación.

Palabras claves: control, gestión, inventarios, planificación, sistema

Índice

Introducción	1
Capítulo 1: Estudio de sistemas de gestión de inventarios para la realización del sistema de planificación y control de inventarios, funcionalidades y tecnologías para su desarrollo.	6
1.1 Conceptos fundamentales relacionados con la gestión de inventarios	6
1.2 Estudio de sistemas homólogos	8
1.3 Metodología de desarrollo, herramientas y tecnologías	13
1.4 Conclusiones parciales del capítulo.....	28
Capítulo 2: Caracterización del sistema para la planificación y control de inventarios.....	29
2.1 Descripción de la propuesta de solución	29
2.2 Descripción del proceso actual	30
2.3 Requisitos de la propuesta de solución	31
2.4 Historias de usuario (HU)	33
2.5 Descripción de la arquitectura de software y los patrones de diseño	35
2.6 Diagrama de clases del diseño.....	39
2.7 Diagrama de componentes.....	40
2.8 Modelo de datos.....	42
2.9 Diagrama de despliegue.....	42
2.10 Conclusiones parciales del capítulo.....	43
Capítulo 3: Implementación y pruebas del sistema de planificación y control de inventarios.....	45
3.1 Estándar de codificación.....	45
3.2 Estrategia de pruebas de software	49
3.2.1 Pruebas funcionales al sistema.....	50

3.2.2 Pruebas de seguridad	54
3.3 Validación de la propuesta	56
Conclusiones parciales del capítulo.....	60
Conclusiones	61
Recomendaciones	62
Referencias bibliográficas	63
Anexos.....	67

Índice de tablas

Tabla 1 Comparación de sistemas.....	12
Tabla 2 Descripción del proceso actual	30
Tabla 3 Requisitos Funcionales.....	32
Tabla 4 HU 04 Gestionar Entrada de Materiales	34
Tabla 5 Estándar de codificación	45
Tabla 6 Variables para el caso de prueba funcional Solicitud de productos.....	50
Tabla 7 Caso de prueba funcional Insertar Solicitud	51
Tabla 8 Caso de prueba funcional Eliminar Solicitud	52
Tabla 9 Diagnóstico para validar la propuesta	59
Tabla 10 HU 01 Gestionar Usuarios.....	67
Tabla 11 HU 02 Gestionar Unidad de Medida.....	68
Tabla 12 HU 03 Registrar Tipo de Material	69
Tabla 13 HU 05 Gestionar Salida de Materiales	71
Tabla 14 HU 06 Gestionar Producto	72
Tabla 15 HU 07 Gestionar Norma de Consumo	73
Tabla 16 HU 08 Solicitud de Producto.....	74
Tabla 17 HU 10 Gestionar Presupuesto.....	76
Tabla 18 HU 11 Generar Reporte	77
Tabla 19 HU 12 Graficar comportamiento de los gastos de materiales	78
Tabla 20 Variables para los casos de prueba funcional Registrar Material.....	79
Tabla 21 Caso de prueba funcional Registrar Material	80
Tabla 22 Caso de prueba funcional Modificar Material	81

Tabla 23 Variables para los casos de prueba funcional Entrada Material	82
Tabla 24 Caso de prueba funcional Insertar Entrada Material	83
Tabla 25 Caso de prueba funcional Eliminar Entrada Material	84
Tabla 26 Variables para los casos de prueba funcional Salida Material	85
Tabla 27 Caso de prueba funcional Salida Material	85
Tabla 28 Caso de prueba funcional Eliminar Salida Material.....	86
Tabla 29 Variables para los casos de prueba funcional Gestionar Presupuesto	87
Tabla 30 Caso de prueba funcional Insertar Presupuesto	87
Tabla 31 Caso de prueba funcional Modificar Presupuesto	88
Tabla 32 Caso de prueba funcional Eliminar Presupuesto	89
Tabla 33 Variables para los casos de prueba funcional Insertar Producto	89
Tabla 34 Caso de prueba funcional Insertar Producto.....	90
Tabla 35 Caso de prueba funcional Modificar Producto	90
Tabla 36 Variables para los casos de prueba funcional Insertar Unidad de Medidas.....	91
Tabla 37 Caso de prueba funcional Insertar Unidad de Medida	92
Tabla 38 Caso de prueba funcional Modificar Unidad de Medida	92
Tabla 39 Variables para los casos de prueba funcional Insertar Norma de Consumo	93
Tabla 40 Caso de prueba funcional Insertar Norma de Consumo.....	94
Tabla 41 Caso de prueba funcional Modificar Norma de Consumo	95

Índice de figuras

Figura 1: Fases e iteraciones de AUP para la UCI. (Rodríguez Sánchez , 2015)	19
Figura 2 Modelo-Vista-Plantilla	36
Figura 3 Patrón Experto presente en la clase Material.....	37
Figura 4 Patrón Creador presente en la clase registrarMaterial	38
Figura 5 Patrón Bajo acoplamiento presente en la clase Producto	38
Figura 6 Patrón Alta cohesión presente en método eliminarMaterial	39
Figura 7 Patrón Decorador presente en la clase insertarSolicitud.....	39
Figura 8 Diagrama de Clases del Diseño HU 10 Gestionar Solicitud	40
Figura 9 Diagrama de Componentes	41
Figura 10 Modelo de Datos.....	42
Figura 11 Diagrama de Despliegue.....	43
Figura 12 Resultados de las Pruebas Funcionales	53
Figura 13 Resultado de la prueba de seguridad 1ra iteración	55
Figura 14 Resultado de la prueba de seguridad 2da iteración	55
Figura 15 Resultado de la prueba de seguridad 3ra iteración	56

Introducción

Actualmente en el mundo la información se genera cada segundo de forma instantánea en todas las organizaciones y en cada uno de sus niveles, por esto lograr la informatización en todos los sectores se ha convertido en una necesidad en cada rincón del mismo. Esto gracias al uso de las Tecnologías de la Información y las Comunicaciones (TIC), las TIC posibilitan poner en práctica estrategias comunicativas y educativas para establecer nuevas formas de enseñar y aprender, han facilitado el acceso y el uso a todo tipo de información, contribuyendo a que en el mundo exista y se fortalezca la comunicación y el flujo de la misma.

En el ámbito empresarial, tener a mano la información necesaria puede significar una ganancia o una pérdida monetaria. A través de las últimas décadas, han aparecido y evolucionado los sistemas de planeación de los recursos empresariales para ayudar en este sector, estos son los sistemas de planificación de recursos empresariales (ERP, por sus siglas en inglés, *enterprise resource planning*), según (Rosemann, 1999) es una aplicación de software estándar que incluye soluciones de negocio para los procesos básicos y las principales funciones administrativas (contabilidad, recursos humanos, logística, etc.), de una empresa.

En Cuba la informatización de la sociedad es un proceso vital que se ha lanzado con mucha fuerza, esto se evidencia en la formación de profesionales que sean capaces de llevar a cabo la estrategia que se ha propuesto el país en aras de lograr el desarrollo para el mejor funcionamiento de todos los sectores de la sociedad, en los cuales será aplicada para así permitir un mejor manejo de la información y las tecnologías. En una entidad es fundamental realizar la supervisión de todos los inventarios que se utilizan en la realización o en la confección de un producto, la Oficina de Identificación de la Universidad de las Ciencias Informáticas (UCI), no es la excepción, dentro de los procesos que se realizan en esta oficina se encuentra la confección de los solapines de identificación de todo el personal de la universidad. En la actualidad el control de inventarios se realiza totalmente manual, esto implica mayor probabilidad de cometer errores que afecten la integridad de dichos inventarios, por lo tanto, se ha tomado la decisión de informatizarlo, de aquí la necesidad de desarrollar un sistema de planificación y control de inventarios utilizados en el proceso de personalización de documentos para la Oficina de Identificación.

La UCI para la emisión de las credenciales de las personas que conviven en su campus, cuenta con esta oficina. Para la emisión de estas credenciales son utilizadas grandes sumas de inventarios entre los que se encuentran hojas de pre-impresos, micas para plasticado y tonel para impresoras a color. Desde el instante

en que comienza el proceso de elaboración de un documento de identificación hasta que termina en el poder de la persona que lo solicita, los inventarios necesarios para su confección transitan por una gran cantidad de trabajadores, quienes pueden ser desde la persona que organiza los inventarios por lotes hasta el trabajador encargado de la confección del documento de identificación. Durante todo este proceso por el que transitan los inventarios, si no son controlados de una manera rigurosa, pueden sufrir serias afectaciones, tales como su pérdida, robo y la degradación o deterioro por causa de acumulación de los inventarios sin base de planificación real. Vale señalar también que no se tiene control sobre la evidencia de que el servicio se ejecute de manera satisfactoria, o sea que a la vez que el servicio se realiza no se sabe cuál fue la cantidad de inventarios por su tipo que fueron utilizados para confeccionar una cantidad determinada de solapines. No existe una correspondencia entre la planificación de inventarios y la demanda real ejecutada del servicio de personalización de documentos de identificación. Actualmente la planificación es insuficiente y para realizar el control no se cuenta con un sistema capaz de brindar tal seguridad a los trabajadores, el proceso es realizado completamente de forma manual, permitiendo la ocurrencia de los factores antes mencionados. A partir de los elementos planteados anteriormente se define como **problema de investigación**:

¿Cómo mejorar la planificación y el control de inventarios en la Oficina de Identificación de la Universidad de las Ciencias Informáticas?

Para dar solución a dicho problema se define como **objeto de estudio**:

La gestión de inventarios en los procesos productivos y los servicios, siendo el **campo de acción** la planificación y el control de inventarios en función del proceso de personalización de documentos de identificación en la Oficina de Identificación de la UCI.

Se plantea como **objetivo general** desarrollar un sistema para mejorar la planificación y el control de los inventarios utilizados en el proceso de personalización de documentos de identificación de la Oficina de Identificación de la UCI.

Para darle cumplimiento al objetivo planteado se responderán las siguientes preguntas científicas:

1. ¿Cuáles son los referentes teóricos de la investigación que sustentan el desarrollo del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI?
2. ¿Qué características debe tener el Sistema para la planificación y el control de inventarios en la

Oficina de Identificación de la UCI?

3. ¿Qué fundamentos presenta el análisis del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI?
4. ¿Qué fundamentos presenta el diseño del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI?
5. ¿Cómo implementar el Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI?
6. ¿Cómo aprobar el Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI?

Para cumplir con los objetivos específicos de este trabajo las **tareas de investigación** propuestas son:

1. Caracterización de los fundamentos teóricos que sustentan la investigación relacionados con la planificación y el control de los inventarios en función del proceso de personalización de documentos de identificación en la Oficina de Identificación de la UCI.
2. Selección de las tecnologías, herramientas y estándares que se necesitan para implementar el Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.
3. Descripción del proceso de planificación y control de los inventarios en función del proceso de personalización de documentos de identificación en la Oficina de Identificación de la UCI.
4. Identificación de los requerimientos del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.
5. Elaboración del modelo de datos persistentes del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.
6. Diseño arquitectónico del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.
7. Implementación del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.
8. Aprobación de las funcionalidades del Sistema para la planificación y el control de inventarios en la

Oficina de Identificación de la UCI.

9. Validación de los resultados de la investigación.

Métodos teóricos utilizados:

Método Analítico – Sintético: empleado para procesar toda la información consultada durante el desarrollo de la investigación, lo que permitió obtener una visión general del objeto de la investigación, desde aquellos elementos que lo sustentan hasta aspectos que posibilitan entender su comportamiento y, por ende, buscar vías para solucionar los problemas que se generen.

Método Histórico – Lógico: permitió estudiar la evolución y desarrollo de sistemas utilizados a nivel internacional, que de cierta manera dan solución a la problemática planteada.

Modelación: utilizado para modelar los procesos y actividades relacionadas con el objeto de estudio y la solución que se propone.

Con la realización de esta investigación se pretende:

- ✓ Obtener un sistema que permita controlar los inventarios utilizados en el proceso de personalización de documentos de identificación.
- ✓ Brindar información sobre el estado de los inventarios y las operaciones realizadas con los mismos.

En aras de lograr un mejor entendimiento la investigación ha quedado estructurada de la siguiente forma:

Capítulo 1. Estudio de sistemas de gestión de inventarios para la realización del sistema de planificación y control de inventarios, funcionalidades y tecnologías para su desarrollo:

Se analiza y se fomenta la base metodológica y teórica, los conceptos que se utilizaron, lenguaje de programación y las herramientas que se utilizaron para la realización del sistema. También se hace un estudio de los diferentes sistemas homólogos que se utilizan a nivel mundial para poder establecer una comparación y se realiza un estudio profundo acerca de las tecnologías que se emplearán en la implementación del sistema.

Capítulo 2. Caracterización del sistema para la planificación y control de inventarios:

En este capítulo atendiendo a la situación problemática enunciada y la solución que se propone se plasma todo el proceso de Modelación del Negocio, así como el levantamiento de requisitos que incluye la definición

de requisitos funcionales y no funcionales del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.

Capítulo 3. Implementación y pruebas al sistema de planificación y control de Inventarios:

Se le realizarán pruebas a la aplicación para así encontrar errores y valorar su funcionamiento y con esto mitigar los posibles errores que se le pueda encontrar en aras de lograr el mejor funcionamiento posible para alcanzar el objetivo principal que es obtener un sistema con calidad y que sea sostenible.

Capítulo 1: Estudio de sistemas de gestión de inventarios para la realización del sistema de planificación y control de inventarios, funcionalidades y tecnologías para su desarrollo.

En este capítulo se abarca una serie de conceptos, así como un estudio de diferentes sistemas de gestión de inventarios, se realiza un análisis de las distintas herramientas, tecnologías y la metodología a utilizar, de todos se mencionan sus características principales y el porqué de su uso en la investigación.

1.1 Conceptos fundamentales relacionados con la gestión de inventarios

Para lograr un buen entendimiento sobre la investigación que se lleva a cabo se definirán algunos conceptos sobre los aspectos y herramientas fundamentales de la misma.

El **control** es el proceso que consiste en supervisar las actividades para garantizar que se realicen según lo planeado y corregir cualquier desviación significativa. Todos los directivos deben participar en la función de control, aunque sus unidades se estén desempeñando según lo planeado. (Robbins, et al., 2005)

El **control** ha sido definido bajo dos grandes perspectivas, una perspectiva limitada y una perspectiva amplia. Desde la perspectiva limitada, el control se concibe como la verificación a posteriori de los resultados conseguidos en el seguimiento de los objetivos planteados y el control de gastos invertidos en el proceso realizado por los niveles directivos, donde la estandarización en términos cuantitativos forma parte central de la acción de control. Bajo la perspectiva amplia, el control es concebido como una actividad no sólo a nivel directivo, sino de todos los niveles y miembros de una entidad, orientando a la organización hacia el cumplimiento de los objetivos propuestos bajo mecanismos de medición cualitativos y cuantitativos. Este enfoque hace énfasis en los factores sociales y culturales presentes en el contexto institucional ya que parte del principio de que es el propio comportamiento individual quien define en última instancia la eficacia de los métodos de control elegidos en la dinámica de gestión. (Cabrera, 2003)

El control como proceso se lleva a cabo con el objetivo de lograr mayor seguridad sobre los productos que verifica. La **seguridad** es un conjunto de medidas tomadas para protegerse contra robos, ataques, crímenes y espionajes o sabotajes. La seguridad implica la cualidad o estado de estar seguro, es decir, evitar exposiciones a situaciones de peligro y la actuación para quedar encubierto frente a contingencias adversas. (Colectivo de autores, 2017)

La **planificación** consiste en definir las metas de la organización, establecer una estrategia general para alcanzarlas y trazar planes exhaustivos para integrar y coordinar el trabajo de la organización. Se ocupa

tanto de los fines (qué hay que hacer) como de los medios (cómo hay que hacerlo). (Robbins, et al., 2005)

Un **insumo** es un bien consumible utilizado en el proceso productivo de otro bien. Los insumos usualmente son denominados factores de la producción o recursos productivos, que pierden sus propiedades y características para transformarse y formar parte de un producto final. (Robbins, et al., 2005)

De acuerdo a lo anteriormente planteado se puede establecer que el **control de insumos** es un proceso en el que se asegura que los bienes o servicios que son propiedad del cliente o que son suministrados por él para incorporarse a los servicios o procesos, se manejen en base a un acuerdo previo de las condiciones de recepción, manipulación y aprovechamiento óptimo. De esa manera se garantiza para ambas partes que no habrá desorden, desperdicio, pérdidas innecesarias, negligencia o desinformación respecto al manejo de los bienes o servicios que se están entregando al proveedor. (Reyes, 2001)

Un **inventario** es una relación detallada, ordenada y valorada de los elementos que componen el patrimonio de una empresa o persona en un momento determinado. Antes, los inventarios se realizaban por medio físico (se escribían en un papel). Es detallada porque se especifican las características de cada uno de los elementos que integran el patrimonio. Es ordenada porque agrupa los elementos patrimoniales en sus cuentas correspondientes y las cuentas en sus masas patrimoniales. Es valorada porque se expresa el valor de cada elemento patrimonial en unidades monetarias.

El concepto de **inventario** en una empresa se basa en todos los productos y materias primas que posee la empresa y que son potenciales para la futura venta y que proporcionen beneficios a la organización. El inventario está formado por todo el stock de la empresa que no se ha vendido, las materias primas que se poseen que les permiten crear nuevos productos y todos aquellos productos que se encuentran en el proceso de producción de la empresa y próximamente ya estarán disponibles para su venta. (Gavilema)

La **gestión de inventarios** es un procedimiento que se realiza con la finalidad de determinar la cantidad y tipo de insumos requeridos para la elaboración del producto o para el ofrecimiento del servicio y así poder satisfacer en su totalidad a los usuarios de este bien o servicio. (Reyes, 2009)

¿Cómo se realiza un proceso de inventario o control de insumos?

El control interno sobre los inventarios es importante, ya que estos son el aparato circulatorio de una empresa de comercialización. Las compañías exitosas tienen gran cuidado de proteger sus insumos. Los elementos de un buen control interno sobre los inventarios incluyen:

- ✓ Conteo físico de los insumos por lo menos una vez al año, independientemente del sistema que se utilice.
- ✓ Mantenimiento eficiente de compras, recepción y procedimientos de embarque.
- ✓ Almacenamiento de insumos para protegerlos contra el robo, daño o descomposición.
- ✓ Permitir el acceso a los insumos solamente al personal que no tiene acceso a los registros contables.
- ✓ Mantener registros de insumos perpetuos para las mercancías de alto costo unitario.
- ✓ Comprar los insumos en cantidades económicas.
- ✓ Mantener suficientes insumos disponibles para prevenir situaciones de déficit, lo cual conduce a pérdidas en ventas.
- ✓ No mantener insumos almacenados demasiado tiempo, evitando con eso el gasto de tener dinero restringido en artículos innecesarios. (Colectivo de autores,)

1.2 Estudio de sistemas homólogos

En los años 70 fue introducida la planificación de requerimiento de materiales (*Material Requirements Planning*, MRP por sus siglas en inglés) que inicialmente se suponían disponibilidad ilimitada de recursos de producción, lo que generaba planificaciones no admisibles con los recursos que contaba la empresa. Una década más tarde fueron introducidos los MRPII, que suponían un salto conceptual importante con respecto a los anteriores. En los 90 surgen los ERP que integran no solo los procesos relacionados con la producción, sino todos los procesos de la empresa. En la actualidad dichos sistemas han adicionado características orientadas hacia el interior de la organización, además de sus funciones tradicionales; esta última evolución se conoce como sistemas ERP II. (Colectivo de autores)

Los ERP están diseñados para aprovechar las potencialidades que ofrecen las tecnologías de la información y las comunicaciones en la integración de todos los subsistemas de una empresa. Dichos sistemas implementan, también, mejores prácticas para facilitar la rápida toma de decisiones, las reducciones de costos y el mayor control directivo, logrando con ello el uso eficiente y eficaz de los recursos empresariales. (Colectivo de autores)

Sistema Integral de Gestión Cedrux

En los últimos años se ha incrementado la necesidad de que las empresas puedan gestionar todos los inventarios para así tener control sobre todos los productos y los materiales que almacenan las mismas. En Cuba actualmente existen varios sistemas que ayudan en este sector tan importante de la sociedad. La UCI en conjunto con algunos centros de desarrollo han creado el Sistema Integral de Gestión Cedrux, el cual dentro de los módulos que lo conforman se encuentra el control de inventarios.

Dentro de las herramientas que fueron utilizadas para el desarrollo e implementación de Cedrux se encuentran los lenguajes PHP, XML, HTML, XHTML y JavaScript. Como sistema gestor de bases de datos PostgreSQL, Wamp 5 como servidor web, Mozilla Firefox como navegador web y Zend Studio como IDE para el desarrollo de aplicaciones web. (Oliva Martínez, et al., 2010)

Con este sistema de control de inventarios se logra una aplicación estándar que se adapta a las características específicas de cada tipo de empresa cubana y que permite centralizar toda la información existente en la misma, adaptado a las características del país, siendo pieza fundamental de un software de gestión integral que controla los procesos empresariales, ajustado a las normativas del Ministerio de Finanzas y Precios. (Oliva Martínez, et al., 2010)

El sistema satisface muchas necesidades que tenían las entidades, por ejemplo el sistema es utilizado en almacenes del MINFAR, garantizando la disponibilidad de sus productos y garantizando que se realice la gestión de manera eficiente. (Oliva Martínez, et al., 2010)

Cedrux cuenta con diferentes módulos como Contabilidad, Costos y Procesos, Finanzas, Recursos Humanos, Logística, que agrupan las áreas presentes en todas las entidades a nivel internacional, además de otros que configuran la estructura, proporcionan seguridad de la información contenida en la aplicación, brindan reportes de las acciones que se realizan en la misma y permite la gestión multimoneda de la empresa en caso necesario. (UCI)

Sistema Único de Identificación Nacional (SUIN)

En Cuba dentro de la Dirección de Identificación y Registros del MININT se encuentra el Centro de Personalización de Documentos de Identificación (CPDI), encargado de la confección de documentos de identificación de la República de Cuba. En dicho centro se explota el Sistema Único de Identificación Nacional. El documento de identidad es solo el último eslabón visible del sistema que implementa el Ministerio del Interior desde el 2012. (Dominguez, 2016)

Está conformado por bases de datos inteligentes donde se integran nombres, apellidos, nombre de los padres y lugar de nacimiento, número de identidad permanente, datos asociados al Registro Civil (tomo, folio, fecha de asiento), domicilio legal, firma, fotos e información de las impresiones dactilares. (Dominguez, 2016)

El SUIN posee dentro de sus funcionalidades un módulo para controlar los insumos que se utilizan en la confección de documentos de identificación en nuestro país, este módulo cuenta con varios submódulos, dentro de los submódulos se encuentra gestionar solicitud, gestionar lotes y administración. El submódulo gestionar lotes se encuentra compuesto por seis actividades capaces de satisfacer las operaciones que se pueden realizar con los lotes de los insumos destinados a la confección de documentos de identificación. Dichas actividades se describen a continuación: (García Hernández, et al., 2012)

- Registrar lote: Momento en que se recibe un lote de insumos en un almacén, local o unidad.
- Reclamar lote: Cuando al llegar a su destino correspondiente el lote o alguno de sus componentes se encuentran en mal estado; para este caso se realiza una reclamación del lote en cuestión.
- Realizar movimiento interno: Este tipo de movimiento es el que se realiza dentro de la misma unidad.
- Realizar movimiento externo: Este tipo de movimiento es el que se realiza entre unidades, locales y almacenes.
- Dar baja: Es la actividad de inhabilitar o eliminar la utilización de un determinado lote de insumos.
- Reconfeccionar lote: En el caso de que sea necesario la división de un lote en varios sublotes con el fin de ubicarlos en destinos distintos.

SAP R/3

SAP R/3 ERP es una de las mejores herramientas conocidas en el área ERP para la planeación empresarial. Basada en múltiples módulos con más de una década en desarrollo y mejoramientos, SAP ofrece herramientas administrativas para las PyME como también para las grandes empresas, incluyendo la facilidad de poder utilizar algunas de ellas desde un entorno web. (SAP)

El proceso de implementación de SAP consiste en diversos módulos que se enfocan en áreas de mercadeo, ventas, servicios internos, desarrollo del producto, creación y control de inventario, recursos humanos, contabilidad y finanzas (siendo los últimos 3 los más desarrollados). (SAP)

El SAP R/3 como sistema ERP, es una herramienta que permite optimizar la distribución y planificación del uso de los recursos disponibles de la empresa, también amplía el alcance y la efectividad de los procesos de los negocios al conectar a los empleados con los proveedores y clientes, todo esto en tiempo real. Una de las ventajas de contar con este sistema de planificación de recursos es que la empresa pueda sistematizarse de manera que los procesos funcionen bajo normativas y controles internos alineados con la estrategia corporativa. (Rodríguez Gutiérrez , et al., 2011)

Posee un módulo de gestión de materiales donde abarca entre otras actividades el control de inventarios. La gestión del inventario se convierte en una gran herramienta para la planificación y permite comparar los materiales pedidos con los recibidos. Contiene enlaces directos con los módulos de compras y control de calidad. Con este módulo siempre se tiene registrado el control de *stocks* actualizado, dado que todo movimiento de material es inmediatamente reflejado en el sistema. (Rodríguez Gutiérrez , et al., 2011)

En un nivel superior, cada módulo puede ser personalizado de forma individual en el proceso de integración, para llevar al módulo a una adaptación optimizada con respecto a la empresa. Esta flexibilidad, modularidad, adaptabilidad con la empresa y control mejorado de servicios, garantiza que SAP sea una herramienta necesaria para las empresas que desean mejor productividad, mejor calidad para sus clientes y un mejor control de sus servicios. No es en vano que múltiples empresas en la lista de Fortune 500¹, utilizan SAP como base para sus operaciones.

Los beneficios que SAP nos ofrece como empresarios son:

- Información en tiempo real
- Creación de un ambiente más eficiente (Para servicios, departamentos, etc.)
- Integración más fácil de diversas áreas de la empresa
- Actualizaciones locales o globales
- Comunidad de expertos en SAP y un servicio de respuestas
- Interfaz gráfica (y por terminal) personalizable para las tareas necesarias.

¹ La lista Fortune Global 500 es un escalafón de las primeras 500 empresas de todo el mundo, medidas por los ingresos. La lista es compilada y publicada anualmente por la revista Fortune.

Algunas empresas famosas que utilizan SAP en la actualidad son Kelloggs y Lionsgate. (SAP)

SAP es uno de los grandes proveedores de software para la gestión empresarial, esto hace que sea muy difícil la adquisición de este tipo de sistemas, la licencia es muy costosa y la universidad no puede hacer ese gasto, no es necesario. En Cuba, la compañía de telecomunicaciones ETECSA utiliza una versión de SAP y es la única empresa que lo utiliza en el país.

Comparación de sistemas

Para la comparación de los sistemas se tuvieron en cuenta varias características que poseerá el sistema a desarrollar, ver tabla 1. El sistema se desarrollará en código abierto por la política que tiene nuestro país de desarrollar sistemas que no sean privativos, controlará los inventarios, generará reportes y será web. Con estas características y otras más contará la propuesta de solución.

Tabla 1 Comparación de sistemas

	Características			
Sistemas	Código abierto	Control inventarios	de Reportes	Tipo de aplicación
SUIN	Si	Si	No	Web
CedruX	Si	Si	Si	Web
SAP R/3	No	Si	Si	Multiplataforma

El estudio de estos sistemas homólogos sirvió como guía para la implementación y el diseño de la propuesta de solución, pues, aunque no se adapten a lo que se necesita en la oficina son módulos que de una manera no muy diferente controlan los inventarios en una entidad.

Después de haber analizado en el ámbito nacional e internacional el autor llega a la conclusión de que existen razones por las cuales no pueden ser utilizados en la Oficina de Identificación de la UCI, no cuentan con varias funcionalidades que son indispensables para dar solución al problema planteado. No poseen una funcionalidad que grafique como ha sido el comportamiento del gasto de materiales y por lo tanto no ayudan a la toma de decisiones, no permiten generar reportes con los datos necesarios que necesita la oficina, no tienen un requerimiento que diga para un determinado pedido de productos cuál es la cantidad de materiales que se necesitan para su confección.

Teniendo en cuenta el estudio anterior se concluye que los sistemas ERP son de suma importancia para la gestión de las empresas, se fortalece la necesidad de implementación de un sistema para la planificación y el control de los inventarios en la Oficina de Identificación y con esto alcanzar el objetivo de que todos los inventarios que intervienen en el proceso de personalización de documentos sean utilizados correctamente y con el fin al cual llegaron a la universidad.

1.3 Metodología de desarrollo, herramientas y tecnologías

El proceso de desarrollo de un sistema informático tiene como propósito la producción eficaz y eficiente de un producto de software que satisfaga los requisitos del cliente. Dicho proceso, en términos globales define como entrada una serie de requisitos definidos previamente o modificados y como salida un sistema nuevo o modificado. Este proceso es intensamente intelectual, afectado por la creatividad y juicio de las personas involucradas (Sommerville, 2002). El mismo debe limitar su alcance, en un momento del tiempo dado, a las realidades que permitan las tecnologías, herramientas, personas y patrones de organización. (Quispe-Otazu, 2007)

Metodologías de desarrollo de software

Dentro de las metodologías existen fundamentalmente dos tipos que contribuyen al desarrollo de software, asegurando en un caso y otro la calidad del producto terminado, estas son: las tradicionales y las ágiles.

Independientemente de las especificidades de una u otra, las metodologías de desarrollo pudieran definirse de la siguiente manera:

Son un conjunto de métodos, reglas, que sirven como modelo para desarrollar el software y que guían a la dirección del proyecto y a los componentes de desarrollo a realizar ciertas comprobaciones sistemáticas de modo que el producto o resultado del trabajo final tenga éxito, o sea que se cumplan los objetivos y los

requerimientos, se entregue en tiempo, forma y cumpla con la calidad que requiere o que el cliente exija. (Quispe Carita, et al., 2011)

De las categorías anteriormente mencionadas es válido aclarar que las metodologías ágiles están basadas en heurísticas provenientes de prácticas de producción de código, están especialmente preparadas para sufrir cambios durante el proyecto y son un proceso menos controlado y con pocos principios, por lo que desarrollan software en cortos períodos de tiempo y exigen poca documentación.

Por otro lado, se pueden encontrar las metodologías tradicionales que responden a otros intereses en cuanto a tiempo de desarrollo y control de este. Pudiendo ser definidas de la siguiente manera:

Las tradicionales están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo; ofrecen por lo general cierta resistencia a los cambios que puedan producirse durante el ciclo de desarrollo del producto, y es un proceso muy bien controlado con muchas políticas y normas. (Quispe Carita, et al., 2011)

Dentro de este grupo se encuentra *Rational Unified Process*, que responde a un proceso iterativo e incremental por lo que se espera una nueva versión del sistema en cada iteración respondiendo a la necesidad de mantener la calidad durante todo el ciclo de vida del proyecto y por tanto asegurando de esta forma la calidad final del producto. No obstante, este método de trabajo presenta algunas desventajas que pueden hacerlo inviable para el desarrollo de ciertas aplicaciones, estas son:

La generación de un gran cúmulo de documentación que no genera valor respecto a la calidad del desarrollo y que necesita incluir un mayor número de personas al equipo de desarrollo, tales como especialistas en casos de uso. (Rumbaugh, et al., 2000)

De esto último se decanta que en el desarrollo de aplicaciones con necesidad de una rápida implementación o con poco personal para el desarrollo no es aconsejable la utilización de estas metodologías tradicionales.

Metodologías ágiles

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de

desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. (Letelier, et al., 2003)

SXP

SCRUM-*Xtreme Programming* (SXP) es una metodología ágil compuesta de XP y SCRUM. XP (*eXtreme Programming* - Programación Extrema) es una metodología de desarrollo formulada por Kent Beck, es el más destacado de los procesos ágiles de desarrollo de software, su particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para poder llegar al éxito del proyecto. Por su parte SCRUM, marco de trabajo también ágil para la gestión y desarrollo de software está basado en un proceso iterativo e incremental, fue desarrollada por Hirotaka Takeuchi e Ikujiro Nonaka, es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepan por dónde van. SXP por su parte, desarrollada en el 2007 en la UCI está especialmente indicada para proyectos pequeños, con rápidos cambios de requisitos donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. (Orozco Vaillant, 2013)

Metodología de desarrollo *eXtreme Programming* (XP)

Es una metodología ágil que se centra en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo y fortaleciendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Letelier, et al., 2003)

AUP

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del Proceso Unificado de *Rational* (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas. Se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. Se centra en la funcionalidad que el sistema debe poseer para satisfacer las necesidades de

un usuario (persona, sistema externo, dispositivo) que interactúa con él. Estas son las disciplinas de AUP: (Flores, 2009)

- **Model. (Modelo)** El objetivo de esta disciplina es entender el negocio de la organización.
- **Implementation. (Aplicación)** El objetivo de esta disciplina es transformar su modelo (s) en código ejecutable y realizar un nivel básico de pruebas.
- **Test. (Prueba)** El objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad.
- **Deployment. (Despliegue)** El objetivo de esta disciplina es ejecutar el sistema y que esté a disposición de los usuarios finales.
- **Configuration Management. (Gestión de configuración)** El objetivo de esta disciplina es la gestión de acceso a artefactos de su proyecto.
- **Project Management. (Gestión de proyectos)** El objetivo de esta disciplina es dirigir las actividades que lleva a cabo en el proyecto.
- **Environment. (Entorno)** El objetivo de esta disciplina es apoyar el resto de los esfuerzos para garantizar que el proceso sea el adecuado.

Dentro de las ventajas que posee esta metodología se destacan: (Flores, 2009)

- El personal sabe lo que está haciendo: no obliga a conocer detalles.
- Simplicidad: apuntes concisos.
- Agilidad: procesos simplificados del RUP.
- Centrarse en actividades de alto valor: esenciales para el desarrollo.
- Herramientas independientes: a disposición del usuario.
- Fácil adaptación de este producto: de fácil acomodo (HTML).

De las desventajas: (Flores, 2009)

- El AUP es un producto muy pesado en relación al RUP.

- Como es un proceso simplificado, muchos desarrolladores eligen trabajar con el RUP, por tener a disposición más detalles en el proceso.

O sea que AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. (Flores, 2009)

El proceso AUP establece un modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de las disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP.

Variación de AUP para la UCI.

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. 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)

Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas, para ello esta metodología adaptada a la universidad se apoya en el Modelo CMMI-DEV v1.3 el cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad. (Rodríguez Sánchez , 2015)

De las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes tres fases de AUP en una sola, nombrada Ejecución y se agrega la fase de Cierre. (Rodríguez Sánchez , 2015)

Con la adaptación de AUP que se propone para la actividad productiva de la UCI se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3. Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. Se redujo a uno la cantidad de metodologías que se usaban y de más de veinte roles en total que se definían se redujeron a once. (Rodríguez Sánchez , 2015)

Esta metodología define 4 escenarios para el trabajo con requisitos teniendo en cuenta las características del equipo de proyecto y del negocio. Estos escenarios se muestran a continuación:

Escenario No 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta respectivamente, donde la atención se centra en cómo el usuario va a utilizar el sistema. Es necesario que se tenga claro por el proyecto que los Casos de uso del negocio (CUN) muestran cómo los procesos son llevados a cabo por personas y los activos de la organización.

Escenario No 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

Escenario No 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. Se debe tener presente que este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y las relaciones entre los procesos identificados.

Escenario No 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una historia de usuario (HU) no debe poseer demasiada información. Las disciplinas definidas en la variación de AUP para la UCI son: modelado de negocio, requisitos, análisis y diseño, implementación, pruebas internas, pruebas de liberación y pruebas de aceptación. Estas disciplinas (desde modelado de negocio hasta pruebas de aceptación) se desarrollan en la Fase de Ejecución, de ahí que en la misma se realicen iteraciones y se obtengan resultados incrementales.

En una iteración se repite el flujo de trabajo de las disciplinas; Requisitos, Análisis y diseño, Implementación y Pruebas internas. De esta forma se brinda un resultado más completo para un producto final de manera creciente. Para llegar a lograr esto, cada requisito debe tener un completo desarrollo en una única iteración. (Rodríguez Sánchez , 2015)

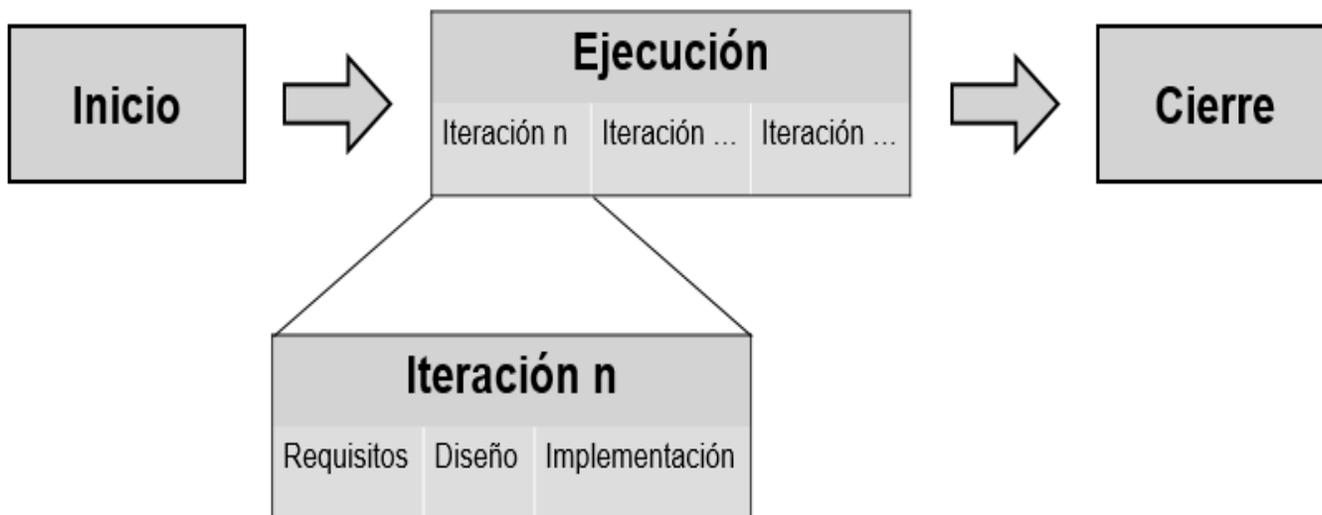


Figura 1: Fases e iteraciones de AUP para la UCI. (Rodríguez Sánchez , 2015)

Metodología escogida

Para guiar la implementación del sistema y para la realización de la documentación del proyecto se definió la metodología variación AUP-UCI debido al enfoque ágil que posee, dirigida a aumentar la eficiencia de las personas involucradas en los proyectos productivos de la universidad. AUP-UCI garantiza que el trabajo sea exitoso, satisfaga las necesidades cambiantes del cliente, así como, la relación estrecha que establece con el mismo por lo que se adapta fácilmente a las circunstancias y, es ideal para equipos de trabajo pequeños. Además, se apoya en las buenas prácticas que propone CMMI-DEV v1.3.

Herramientas y tecnologías

El objetivo principal del proceso de desarrollo de un sistema informático es la producción eficiente de un producto de software capaz de cumplir con todos los requerimientos del cliente, para lograr este cometido se debe definir bien claro como entrada una serie de requisitos, para obtener como salida un producto eficaz y funcional. En este proceso de desarrollo debe quedar plasmado el alcance del producto haciendo uso de los factores de medición como: las tecnologías, herramientas, recursos humanos y patrones de organización.

Las herramientas informáticas son programas, aplicaciones o simplemente instrucciones usadas para efectuar otras tareas de modo más sencillo. A continuación, se describen las herramientas y tecnologías utilizadas para el desarrollo de la propuesta de solución.

Marco de trabajo

Para lograr un desarrollo acorde a las necesidades existentes en cuanto a tiempo y requisitos del sistema se hizo necesario el análisis de varios marcos de trabajos o librerías de desarrollo tanto del lado del cliente como del servidor. En términos generales un marco de trabajo es definido como un conjunto de códigos o utilidades previamente implementadas siguiendo diversos patrones que permiten a los desarrolladores reutilizar código y como consecuencia directa ahorrar ya sea tiempo de trabajo o esfuerzo. Aunque existen marcos de trabajos de desarrollo en todos los lenguajes de programación existentes, en la presente investigación se analizará solamente los basados en el lenguaje Python que es el seleccionado para la implementación de la propuesta de solución.

Marco de trabajo Django 1.10.3

Django es un marco de trabajo de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–Vista–Controlador. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la *World Company* de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD² en julio de 2005; el marco de trabajo fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt. En junio de 2008 fue anunciado que la recién formada *Django Software Foundation* se haría cargo de Django en el futuro. (Lawrence Journal-World, 2015)

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio **no te repitas** (DRY, del inglés *Don't Repeat Yourself*). Python es usado en todas las partes del marco de trabajo, incluso en configuraciones, archivos, y en los modelos de datos. (Lawrence Journal-World, 2015)

Algunas de sus características son: (Holovaty, et al., 2007)

² La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*) público bajo una licencia BSD.

- ✓ Django fue diseñado para ayudar a los desarrolladores que reciben solicitudes desde el concepto hasta su finalización lo antes posible. Resulta tranquilizador, seguro.
- ✓ Django toma en serio la seguridad y ayuda a los desarrolladores a evitar muchos errores comunes.
- ✓ Documentación incorporada accesible a través de la aplicación administrativa (incluyendo documentación generada automáticamente de los modelos y las bibliotecas de plantillas añadidas por las aplicaciones).
- ✓ Un sistema incorporado de "vistas genéricas" que ahorra tener que escribir la lógica de ciertas tareas comunes. Aplicaciones "enchufables" que pueden instalarse en cualquier página gestionada con Django.
- ✓ Un sistema "*middleware*" para desarrollar características adicionales; por ejemplo, la distribución principal de Django incluye componentes middleware que proporcionan compresión de la salida, normalización de URLs, protección CSRF y soporte de sesiones.
- ✓ Django posee una API de base de datos robusta.

Lenguaje de Marcado de Hipertexto (HTML5)

HTML, que significa Lenguaje de Marcado para Hipertextos (*HyperText Markup Language*) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad. Otras tecnologías distintas de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript). (Colectivo de autores, 2017)

HTML le da "valor añadido" a un texto estándar en español. Híper Texto se refiere a enlaces que conectan una página web con otra, ya sea dentro de una página web o entre diferentes sitios web, los vínculos son un aspecto fundamental de la web. Al subir contenido a internet y vincularlo a páginas de otras personas, te haces participante activo de esta red mundial. (Colectivo de autores, 2017)

Esta versión del lenguaje básico de la web proporciona mecanismos para simplificar el trabajo y facilitar la inclusión de elementos multimedia. El principal criterio de diseño de html5 ha sido el de resolver problemas prácticos, y con este objetivo adopta soluciones dirigidas a facilitar el trabajo en situaciones reales.

Xhtml se creó para sustituir la sintaxis en la que se basa HTML: el SGML (*Standard Generalized Markup Language*), cuya finalidad principal era facilitar la creación manual de documentos. HTML era un lenguaje poco riguroso y como consecuencia los documentos resultaban un tanto caóticos y no siempre cumplían con la sintaxis. Esto provocó que los navegadores se hicieran más complejos para poder asumir ambigüedades y equivocaciones, es decir, para poder representar los documentos, aunque tuvieran errores. Como las máquinas tienen ciertas dificultades para leer y manipular contenido etiquetado con la sintaxis de SGML, se propuso una sintaxis nueva basada en XML, un lenguaje derivado de SGML, pero más comprensible para las máquinas. Así, al aplicar a HTML la sintaxis de XML, los ordenadores son capaces de manipular HTML con facilidad y precisión, y en particular los dispositivos móviles pueden procesar la información de manera más eficiente. (Franganillo, 2011)

Las facilidades que brinda y la preferencia que existe actualmente a utilizar esta versión de HTML, hizo que el autor para el desarrollo de la aplicación web utilizara dicho lenguaje; también se apoyó en las múltiples ventajas que este proporciona a los desarrolladores.

JavaScript

JavaScript es el lenguaje interpretado orientado a objetos desarrollado por Netscape que se utiliza en millones de páginas web y aplicaciones de servidor en todo el mundo. JavaScript de Netscape es un superconjunto del lenguaje de scripts estándar de la edición de ECMA-262 3 (ECMAScript) que presenta sólo leves diferencias respecto a la norma publicada.

JavaScript puede funcionar como lenguaje procedimental y como lenguaje orientado a objetos. Los objetos se crean programáticamente añadiendo métodos y propiedades a lo que de otra forma serían objetos vacíos en tiempo de ejecución, en contraposición a las definiciones sintácticas de clases comunes en los lenguajes compilados como C++ y Java. Una vez que se ha construido un objeto, puede usarse como modelo (o prototipo) para crear objetos similares. (developer.mozilla, 2015)

Hojas de Estilo en Cascada (CSS 3)

Hojas de Estilo en Cascada (*Cascading Style Sheets*) es el lenguaje utilizado para describir la presentación de documentos HTML o XML, esto incluye varios lenguajes basados en XML como son XHTML o SVG. CSS describe como debe ser renderizado el elemento estructurado en pantalla, en papel, hablado o en otros medios (Mozilla, 2016), o sea, es el lenguaje para describir la presentación de páginas web, incluidos los

colores, el diseño y las fuentes. Le permite a uno adaptar la presentación a diferentes tipos de dispositivos, como pantallas grandes, pantallas pequeñas o impresoras. CSS es independiente de HTML y se puede usar con cualquier lenguaje de marcado basado en XML. La separación de HTML de CSS facilita el mantenimiento de sitios, el intercambio de hojas de estilo entre páginas y la personalización de páginas en diferentes entornos. Esto se conoce como la separación de la estructura o el contenido de la presentación. (Colectivo de autores, 2016)

Por las características que tendrá y las herramientas con las cuales se implementará la propuesta de solución se decidió utilizar CSS, este se usa para darle estilo y posicionarlo visualmente. CSS se puede usar, por ejemplo, para cambiar la fuente, el color, el tamaño y el espaciado del contenido, para formar múltiples columnas, añadir animaciones y otros elementos decorativos.

Sistemas Gestores de Bases de Datos

Los sistemas gestores de bases de datos (SGBD) son como una capa intermedia entre los programas que el usuario final utiliza y el sistema operativo. Los SGBD son, por tanto, el sistema encargado de establecer la comunicación entre estos dos sistemas. Se conoce que antes de que aparecieran las bases de datos y los SGBD, las aplicaciones eran las encargadas de definir su propio sistema de ficheros para almacenar datos de forma que, cada vez que cambiara la aplicación, era necesario cambiar la estructura de todos los ficheros y, al contrario. Con estos sistemas se evitan todos estos inconvenientes. (Zambrano Ramírez, 2008)

Los lenguajes se clasifican en: lenguaje de definición de datos (LDD o DDL), que se utiliza para especificar el esquema de la base de datos, las vistas de los usuarios y las estructuras de almacenamiento. Este es el lenguaje que utilizan los diseñadores y los administradores de bases de datos (Zambrano Ramírez, 2008) y el lenguaje de manipulación de datos (LMD o DML) que se utiliza para realizar operaciones de consulta y/o modificación de la base de datos. Es el lenguaje utilizado por los usuarios de las bases de datos. (Zambrano Ramírez, 2008)

Los SGBD prestan varios servicios como la creación de las bases de datos, la manipulación de los datos realizando consultas, intersecciones y actualizaciones, también garantizan la integridad de los datos, entre otros. El SGBD que se utilizará para la administración de la base de datos del sistema es PostgreSQL en su versión 9.4.4.

PostgreSQL 9.4.4

PostgreSQL es un potente sistema de base de datos, de código abierto objeto-relacional. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación por su fiabilidad, integridad de datos y corrección. Se ejecuta en todos los sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. Incluye, además: 2008 tipos de datos, incluyendo enteros, numéricos, booleanos, CHAR, VARCHAR, DATE, INTERVALO y TIMESTAMP. También es compatible con el almacenamiento de grandes objetos binarios, como imágenes, sonidos, o de video. Cuenta con las interfaces de programación nativas para C / C ++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre otros, además de poseer una amplia documentación.

PostgreSQL cuenta con características sofisticadas como *Multi-Version Concurrency Control* (MVCC), punto de recuperación en el tiempo, *tables-paces*, replicación asíncrona, transacciones anidadas (puntos de retorno), en línea copias de seguridad / calor, un sofisticado planificador de consulta / optimizador, y escribir por delante el registro para la tolerancia a fallos. Es compatible con los juegos de caracteres internacionales, codificación de caracteres *multibyte*, *unicode*, y es consciente de la configuración regional de clasificar mayúsculas y minúsculas y el formato. Es altamente escalable tanto en la enorme cantidad de datos que puede manejar como en el número de usuarios simultáneos que puede acomodar. Hay sistemas activos de PostgreSQL en entornos de producción que manejan más de cuatro terabytes de datos. (PostgreSQL)

Herramienta CASE

CASE es un acrónimo para *Computer-Aided Software Engineering*. Esencialmente, un CASE es una herramienta que ayuda al ingeniero de software a desarrollar y mantener el software. (Departamento de Sistemas Informáticos y Computación) Se ha seleccionado la herramienta Visual Paradigm por sus múltiples facilidades a la hora de trabajar y de modelar los procesos.

Visual Paradigm para UML 8

Visual Paradigm para UML es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación.

Entre sus principales ventajas está que permite modelar todos los diagramas de clases.

De estas características se pueden destacar:

- Disponibilidad en varias plataformas (Windows, Linux).
- Capacidades de ingeniería directa e inversa.
- Puede ser utilizado en el diseño de aplicaciones web.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento y Flujo de Datos.

Debido a las ventajas claras que ofrece esta potente herramienta y a que la Universidad de las Ciencias Informáticas posee una licencia de tipo comercial de este software, se opta por utilizarlo en aras de lograr disponer de su alta efectividad en el modelado.

Se integra con diversos IDE´s como: NetBeans (de Sun), Eclipse (de WebShere), JDeveloper, entre otros. Está disponible en varias ediciones: *Enterprise*, *Professional*, *Community*, *Standard*, *Modeler* y *Personal*. Genera código y realiza ingeniería inversa para diferentes lenguajes de programación como: PHP, Java, Python, entre otros. (Colectivo de autores)

IDE

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de *Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). (Alonzo Velázquez, 2010)

Pycharm 2017.2.2

PyCharm incluye un editor de código inteligente que entiende las capacidades específicas de Python y ofrece destacados impulsores de productividad: formateo de código automático, finalización del código, navegación de código de un solo clic, y más. Respaldadas por rutinas de análisis de código avanzadas, estas funciones hacen de PyCharm una eficaz herramienta en manos de los desarrolladores profesionales de Python y los que acaban de iniciarse en la tecnología. PyCharm es un IDE que se utiliza para la programación en Python. Proporciona análisis de código, un depurador gráfico, un probador de unidad integrada, integración con sistemas de control de versiones (VCSes), y apoya el desarrollo web con Django. PyCharm es desarrollado por la empresa checa JetBrains. (Jetbrains, 2017)

Funciona en Windows, Mac OS o Linux con una única clave de licencia. Posee un conjunto de herramientas: un depurador integrado y corredor de prueba gráfica; incorporado un terminal; integración con Git (*Concurrent Versioning System*), SVN, Mercurial y herramientas integradas de base de datos. (Jetbrains, 2017)

Permite desarrollar, depurar, probar e implementar aplicaciones en los servidores remotos o máquinas virtuales, con intérpretes remotos, terminal SSH integrada, y la integración Vagrant³. Ofrece un gran apoyo específico para los marcos de desarrollo web como Django, Frasco, Google App Engine, Pirámide, y web2py.

Presenta una asistencia inteligente de codificación con la finalización de código inteligente, inspecciones de código, resaltado de error sobre la marcha, auto-correcciones, junto con refactorizaciones de código automatizados y capacidades de navegación ricos para las principales implementaciones de Python. (Jetbrains, 2017)

Lenguaje de programación

Un lenguaje de programación consiste en un conjunto de órdenes o comandos que describen el proceso deseado. Cada lenguaje tiene instrucciones y enunciados verbales propios, que se combinan para formar los programas de cómputo. Los lenguajes de programación no son aplicaciones, sino herramientas que permiten construir y adecuar aplicaciones. (Gallardo López, et al., 2008)

Python 3.5

Para el desarrollo de las funcionalidades del sistema se ha decidido utilizar el lenguaje de programación Python el cual es un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos. Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete. La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo los lenguajes interpretados son más flexibles y más portables. (González Duque, 2008) Python tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. En Python, como en Java y muchos otros lenguajes, el código fuente se traduce a un pseudo código llamado *bytecode*, generando archivos .pyc o .pyo (*bytecode* optimizado),

³ Herramienta que nos permite crear cualquier entorno de desarrollo basado en máquinas virtuales.

que son los que se ejecutarán en sucesivas ocasiones.

Algunas de las características del lenguaje son:

✓ Tipado dinámico:

La característica de tipado dinámico se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo del valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo.

✓ Fuertemente tipado:

No se permite tratar a una variable como si fuera de un tipo distinto al que tiene, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente. Por ejemplo, si se tiene una variable que contiene un texto (variable de tipo cadena o string) no se puede tratar como un número (sumar la cadena "9" y el número 8). En otros lenguajes el tipo de la variable cambiaría para adaptarse al comportamiento esperado, aunque esto es más propenso a errores.

✓ Multiplataforma:

El intérprete de Python está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc.) por lo que si no se utilizan librerías específicas de cada plataforma el programa podrá correr en todos estos sistemas sin grandes cambios.

✓ Orientado a objetos:

La orientación a objetos es un paradigma de programación en el que los conceptos del mundo real relevantes para el problema se trasladan a clases y objetos en el programa. La ejecución del programa consiste en una serie de interacciones entre los objetos.

Python también permite la programación imperativa, programación funcional y programación orientada a aspectos.

Python es un lenguaje que todo el mundo debería conocer. Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo, muy rápido y, lo que es más importante, divertido.

Existen algunos casos de éxito en el uso de Python como lo son Google, Yahoo, la NASA, Industrias Light & Magic, y todas las distribuciones Linux, en las que Python cada vez representa un tanto por ciento mayor de los programas disponibles. (González Duque, 2008)

Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado fue creado con el objetivo de llenar la brecha existente entre la idea de un proyecto y su implementación, permitiendo modelar uno o varios sistemas de manera que no sean necesarios procesos engorrosos para expresar de manera clara y concisa lo que deberían ser. Basado en tecnología orientada a objetos este es sin duda alguna, una herramienta de uso obligatorio al menos en los proyectos de gran alcance, aunque los pequeños proyectos también podrían obtener grandes ventajas de este lenguaje. Es válido decir que involucra todo el ciclo de vida del proyecto y está pensado para varios lenguajes y plataformas. (Popkin Software and System)

1.4 Conclusiones parciales del capítulo

En el presente capítulo se determinan las siguientes conclusiones parciales:

- El estudio de las diferentes herramientas, tecnologías y tendencias actuales para la gestión de inventarios permitió evidenciar las características que constituyen las premisas para el planteamiento de las funcionalidades que se definirán en la propuesta de solución.
- El estudio sobre las metodologías, herramientas y lenguajes permitió definir los componentes base para el desarrollo de la solución, donde se define a AUP-UCI como metodología de desarrollo, UML como lenguaje de modelado, Visual Paradigm como herramienta CASE, Python como lenguaje de programación, Django como marco de trabajo, IDE Pycharm, PostgreSQL como Sistema Gestor de Bases de Datos y para la maquetación web HTML y CSS.

Capítulo 2: Caracterización del sistema para la planificación y control de inventarios

En el presente capítulo se presenta la propuesta de solución: Sistema para la planificación y control de inventarios en el proceso de personalización de documentos de identificación; se describe su funcionamiento, así como la arquitectura y los patrones de diseños aplicables al sistema. Además, se definen los Requisitos Funcionales (RF) y Requisitos No Funcionales (RNF), las Historias de Usuarios (HU) y los artefactos generados que guiarán la implementación de la propuesta de solución.

Actualmente existen numerosos sistemas web o de escritorio que se encuentran destinados a la gestión de inventarios, sin embargo, como fue planteado en el capítulo anterior, poseen ciertas características que no los hacen idóneos para su explotación en la Oficina de Identificación. Del estudio de algunos de estos sistemas, debidamente mencionados anteriormente, se obtuvo experiencia para realizar la propuesta de solución que este capítulo tiene la responsabilidad de introducir.

2.1 Descripción de la propuesta de solución

El sistema que se pretende implementar fundamentalmente deberá gestionar todos los usuarios, los roles y los permisos que sean asignados. También va a gestionar todos los materiales que se utilizan en la confección de la credencial, donde se les dará entrada a los materiales, todo esto se realizará por el tipo de material y teniendo en cuenta la fecha en que se realizan las acciones en el sistema. Al sistema se le debe solicitar una cantidad de productos y este debe ser capaz de calcular y mostrar que cantidad de materiales por tipo se deben utilizar para su realización, se tendrán en cuenta los que por una causa u otra no fueron impresos por error de impresión o por algún error de sistema. Se generará reportes y aquí se visualizará todo lo relacionado con la producción, mostrará cuánto se realizó, cuánto se gastó y el costo de esa producción. Será capaz de exportar a Excel el reporte y el gasto de materiales se podrá visualizar en un gráfico para después poder saber cómo ha sido su comportamiento. Se gestionará el presupuesto anual de la oficina para tener en cuenta cuánto se ha gastado monetariamente. El sistema apoyará la toma de decisiones porque a la vez que se tenga un comportamiento histórico de cuánto material se ha gastado y cuánto se ha producido será posible predecir cuánto se necesitará para un período determinado.

Teniendo en cuenta las características de la investigación se ha determinado usar el escenario 4 de la metodología AUP-UCI, ya que se tiene un negocio muy bien definido, el cliente ha sido partícipe de la confección de los requisitos a implementar, se adapta a este proyecto porque no es muy extenso y por lo tanto las historias de usuarios no poseerán mucha información.

2.2 Descripción del proceso actual

Los inventarios que se utilizan para la confección de los documentos de identificación se encuentran guardados en la oficina con acceso restringido. Al inicio de cada día laboral el Jefe de Oficina o en caso de este no encontrarse, el Encargado de Confección, extrae la cantidad necesaria de inventarios según la demanda de documento de identificación. Los inventarios, ya sea pre-impreso, micas, tonel de impresión, son entregados a los funcionarios encargados de la confección de los documentos de identificación, los cuales tendrán que contar con los inventarios que le son entregados, y también deberán firmar un acta haciendo constancia de que recibieron la cantidad solicitada de inventarios.

En el caso de que durante el proceso de confección del documento existiera algún error, automáticamente el pre-impreso es anulado, se le corta el número de serie que contiene y se guarda para que quede constancia de la cantidad que no pudieron confeccionarse por algún tipo de error, el cual contiene además los datos del titular del documento de identificación.

Sin embargo, si el proceso de confección del documento resultara satisfactorio, se pasa a anotar los datos del titular en el documento de seriado y el titular debe firmar el acta de entrega del documento.

Cuando finaliza la jornada laboral si existiera algún sobrante de inventarios, el funcionario que confecciona los documentos retorna los excedentes a donde se guardan dichos inventarios (en la oficina), implicando que se firme un acta donde se recoge la cantidad de inventarios entregados, los que se consumieron y los que se quedan en la oficina donde se confeccionan.

Descripción del Proceso de Planificación y Control de Inventarios:

Tabla 2 Descripción del proceso actual

Nombre:	Control de inventarios
Evento(s) que lo generan:	Llevar un control de todos los inventarios utilizados en el proceso de personalización de documentos de identificación.
Precondiciones:	Debe existir el inventario en la oficina.

Poscondiciones:	-	
Reglas de Negocio:	<ul style="list-style-type: none"> • Se debe firmar el acta de entrega cuando reciben los inventarios. • En caso de anulación de un documento, recortar el número de serie y guardarlo para la constancia. • Los inventarios sobrantes al final del día se deben quedar en la oficina. 	
Responsable(s):	Jefe de Oficina, Funcionario de Confección .	
Clientes		
Rol(es):	Jefe de Oficina.	Entregar los inventarios que serán procesados en el día.
	Funcionario de confección.	Utilizar los inventarios correspondientes para confeccionar y personalizar el documento.
Entradas:	Pre-impreso a procesar, Micas, Tonel de impresión.	
Salidas:	Documento de identificación, documento de control de seriados.	

2.3 Requisitos de la propuesta de solución

Un requerimiento es una característica que el sistema tiene o es una restricción que debe satisfacer para ser aceptada por el cliente. El levantamiento de requerimientos es la especificación del sistema en términos que el cliente entienda, de forma que se especifique todo en el contrato entre el cliente y los desarrolladores.

Requisitos funcionales

Describen la interacción entre el sistema y su ambiente independientemente de su implementación. El ambiente incluye al usuario y cualquier otro sistema externo que interactúa con el sistema.

Tabla 3 Requisitos Funcionales

No	Requisitos Funcionales
01	Gestionar usuarios
02	Gestionar unidad de medida
03	Registrar tipo material
04	Gestionar entrada de materiales
05	Gestionar salida de materiales
06	Gestionar producto
07	Gestionar norma de consumo
08	Gestionar solicitud de productos
09	Generar cantidad de materiales por productos
10	Gestionar presupuesto
11	Generar reporte
12	Graficar comportamiento de los gastos de materiales

Requisitos no funcionales

Los requisitos no funcionales no son más que las características o cualidades con las que el sistema debe contar, estas cualidades son las que hacen que el producto sea usable, rápido y confiable. Estos requerimientos son de vital importancia para el cliente, permitiéndole valorar las características no funcionales del producto. (Massó Verrier, et al., 2010)

Usabilidad

RNF1: El sistema deberá ser usado por usuarios con conocimientos básicos en el uso de computadoras, además de poseer conocimientos básicos relacionados con el control de los inventarios usados en la personalización de documentos de identificación.

RNF2: El sistema será distribuido en soporte de idioma español.

RNF3: Los términos utilizados en el sistema estarán acorde al negocio de la organización, con el fin de facilitar la comprensión de los procesos y una mayor integración con los usuarios.

RNF4: El sistema contendrá menús desplegables a los laterales permitiendo el acceso rápido a la navegación en el sistema.

RNF5: El sistema estará disponible durante toda la semana en el horario laboral de la oficina.

Apariencia o interfaz externa

RNF6: Navegador web Mozilla Firefox v19.0 o superior, Google Chrome v20.0 o superior, o versiones actuales de Opera, Internet Explorer y Safari. Google Chrome v20.0 o superior.

Seguridad

RNF7: El sistema será utilizado solo por los usuarios registrados y usuario administrador.

RNF8: Protección de la integridad de los datos ante operaciones no autorizadas.

Hardware

RNF9: PC Pentium 4 a 2 GHz o superior, mínimo 2 GB de RAM, 40 GB o superior de disco duro.

2.4 Historias de usuario (HU)

Según el escenario 4 de la metodología seleccionada aquí se muestran las historias de usuario HU las cuales definen mediante su redacción qué es lo que desea obtener el cliente según se ha planteado en

capítulos anteriores. En este paso es determinado el alcance del sistema y permite a los desarrolladores familiarizarse con las herramientas y tecnologías usadas en la solución. A continuación, se muestra una HU del sistema, las otras pueden ser revisadas en los anexos:

Tabla 4 HU 04 Gestionar Entrada de Materiales

Número: 04	Nombre del requisito: Gestionar Entrada de Materiales
Programador: Eduardo Baratutes González	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: -
Riesgo en Desarrollo: -	Tiempo Real: -
<p>Descripción:</p> <p>Deben recogerse los siguientes datos:</p> <ul style="list-style-type: none"> • Identificador del material (Valor numérico de carácter obligatorio) • Cantidad de material (Valor numérico de carácter obligatorio) • Fecha de entrada (Valor que se genera automáticamente) 	
<p>Observaciones:</p> <p>Esta funcionalidad tiene como objetivo la implementación de la parte de la aplicación</p>	

encargada de la gestión de la entrada de los materiales a la oficina.

Prototipo de interfaz:

El prototipo de interfaz muestra un formulario con el título "Nueva Entrada de Material". El formulario contiene tres campos de entrada: "Nombre del Material *" con un menú desplegable que muestra "Seleccione una opción:", "Cantidad de Material *" con un campo de texto vacío, y "Fecha de Expiración *" con un campo de texto vacío. En la parte inferior derecha del formulario hay dos botones: "ACEPTAR" y "CANCELAR".

2.5 Descripción de la arquitectura de software y los patrones de diseño

Aquí se analizarán todos los elementos imprescindibles para la realización del Análisis y el Diseño del Sistema para la planificación y control de inventarios en el proceso de personalización de documentos de identificación, este modelo sirve como guía para la implementación de la propuesta de solución.

Estilo de arquitectura: Modelo Vista Plantilla (MTV, Model Template View)

En el desarrollo de software la arquitectura a seguir es un aspecto de vital importancia, debido a que esta es la encargada del diseño de componentes de una aplicación generalmente utilizando patrones de arquitectura. Un diseño arquitectónico describe en general el cómo se construirá una aplicación de software. (Holovaty, et al., 2007)

Django sigue el patrón MVC tan al pie de la letra que puede ser llamado un marco de trabajo MVC. Someramente, la M, V y C se separan en Django de la siguiente manera:

M, la porción de acceso a la base de datos, es manejada por la capa de la base de datos de Django, la cual se describirá en este capítulo. V, la porción que selecciona qué datos mostrar y cómo mostrarlos, es manejada por la vista y las plantillas. C, la porción que delega a la vista dependiendo de la entrada del

usuario, es manejada por el mismo siguiendo la URLconf y llamando a la función apropiada de Python para la URL obtenida.

La arquitectura utilizada en la presente propuesta de solución puede ser apreciada en la siguiente figura:

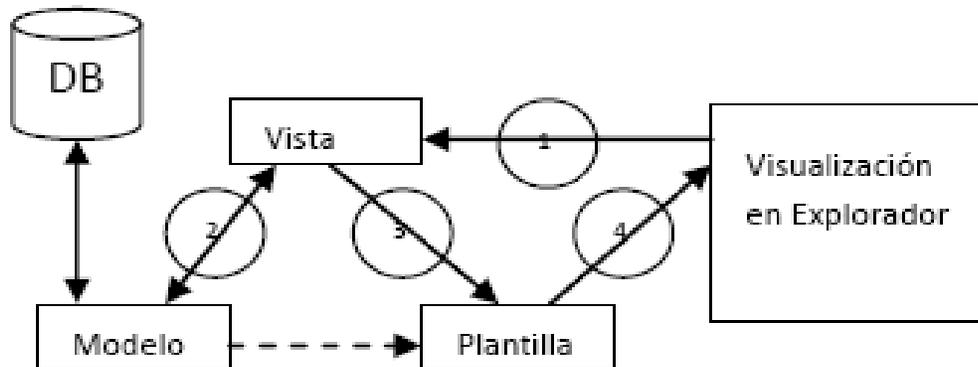


Figura 2 Modelo-Vista-Plantilla

Debido a que la "C" es manejada por el mismo marco de trabajo y la parte más emocionante se produce en los modelos, las plantillas y las vistas, Django es conocido como un marco de trabajo MTV. En el patrón de diseño MTV: (Holovaty, et al., 2007)

- M significa "*Model*" (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.
- T significa "*Template*" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: cómo algunas cosas son mostradas sobre una página web u otro tipo de documento.
- V significa "*View*" (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: se interpreta como un puente entre el modelo y la plantilla. (Holovaty, et al., 2007)

Patrones de diseño

En el desarrollo de una aplicación informática la solución de problemas de características similares, en distintos módulos, secciones, etc., es una realidad a la que se enfrentan los programadores comúnmente. En aras de mejorar la capacidad de darle solución a estos, muchas veces pequeños, pero muy recurrentes problemas, surgen los patrones de diseño a partir de soluciones ya probadas. De esta forma cada patrón tiene una descripción de la solución según el problema a tratar, permitiendo emplear la misma solución sin implementarla nuevamente. A continuación, se exponen los principales patrones presentes en la solución implementada.

Los **Patrones GRASP** describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades (Giraldo G., et al., 2011), constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objetos esenciales, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Este enfoque para la comprensión y utilización de los principios de diseño se basa en los patrones de asignación de responsabilidades. (Giraldo G., et al., 2011)

Experto

El patrón Experto en Información se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos. El patrón Experto no pretende ser una idea oscura o extravagante; expresa la "intuición" común de que los objetos hacen las cosas relacionadas con la información que tienen. (Larman, 2003) Se evidencia en la clase Material, la misma contiene toda la información que comprende a un material.

```
class Material(models.Model):
    nombre = models.TextField(max_length=145, unique=True, validators=[validate_only_letters_may])
    unidad_medida = models.ForeignKey(Unidad, unique=False, null=False, on_delete=models.CASCADE)
    precio = models.FloatField(validators=[MinValueValidator(0)])
    total = models.FloatField(default=0)

    def __str__(self):
        return self.unidad_medida.unidad
    def __str__(self):
        return self.nombre
```

Figura 3 Patrón Experto presente en la clase Material

Creador

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común. La intención básica del patrón Creador es encontrar un creador que necesite conectarse

al objeto creado en alguna situación. Eligiéndolo como el creador se favorece el bajo acoplamiento. (Larman, 2003) Se refleja en la clase registrarMaterial ya que es la encargada de registrar un material en el sistema.

```
class registrarMaterial(RequiredSecurityMixin, SuccessMessageMixin, CreateView):
    need_login = True
    permission = RequiredSecurityMixin.CREATE
    model = Material
    template_name = 'crear.html'
    form_class = FormularioRegistrarMaterial
    success_url = '/listar_material'
    success_message = "Material : %(nombre)s Insertado satisfactoriamente."

    def form_invalid(self, form):
        messages.Error(self.request, 'Por favor corrija los errores.')
        return super(registrarMaterial, self).form_invalid(form)
```

Figura 4 Patrón Creador presente en la clase registrarMaterial

Bajo acoplamiento

Describe que debe existir una alta reutilización entre las funcionalidades de las clases con una mínima dependencia, contribuyendo así al mantenimiento de las mismas. Django implementa este patrón por defecto evitando altas dependencias entre los componentes. (Larman, 2003) Se refleja en la clase Producto donde la dependencia es baja entre las funcionalidades.

```
class Producto(models.Model):
    tipo_producto = models.CharField(max_length=145, unique=True, validators=[validate_only_letters_may])
    material = models.ManyToManyField(Material, through='Norma_Consumo')

    def __str__(self):
        return self.tipo_producto
```

Figura 5 Patrón Bajo acoplamiento presente en la clase Producto

Alta cohesión

Una clase con alta cohesión es ventajosa porque es relativamente fácil de mantener, entender y reutilizar. El alto grado de funcionalidad relacionada, combinada con un número pequeño de operaciones, también simplifica el mantenimiento y las mejoras. El grano fino de la funcionalidad altamente relacionada también aumenta el potencial de reutilización. (Larman, 2003) Se refleja en el método eliminarMaterial el cuál es instanciado desde otra funcionalidad del sistema.

```

def eliminarMaterial(request, pk):
    art = getMaterialById(pk)
    if art:
        art.delete()
        success(request, 'Material eliminado satisfactoriamente.')
        return HttpResponseRedirect(reverse_lazy('listarmaterial'))
    return HttpResponseRedirect(reverse_lazy('listarmaterial'))

```

Figura 6 Patrón Alta cohesión presente en método eliminarMaterial

Gang-of-Four (GoF, en español Banda de los Cuatro): describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos. (Larman, 2003)

Decorador

Es un patrón estructural que extiende la funcionalidad de un objeto dinámicamente de tal modo que es transparente a sus clientes, utilizando una instancia de una subclase de la clase original que delega las operaciones al objeto original. Provee una alternativa muy flexible para agregar funcionalidad a una clase. (Larman, 2003) Este patrón Django lo incluye por defecto, ejemplo de ello lo constituye la función: **@login_required.**

```

@login_required
class insertarSolicitud(RequiredSecurityMixin, SuccessMessageMixin, CreateView):
    need_login = True
    permission = RequiredSecurityMixin.CREATE
    model = Solicitud
    template_name = 'crear.html'
    form_class = FormularioInsertarSolicitud
    success_url = '/listar_solicitud'
    success_message = "Solicitud realizada satisfactoriamente."

    def get_context_data(self, **kwargs):...

    def form_invalid(self, form):...

    def form_valid(self, form):...

```

Figura 7 Patrón Decorador presente en la clase insertarSolicitud

2.6 Diagrama de clases del diseño

Un Diagrama de Clases de Diseño (DCD) representa las especificaciones de las clases e interfaces de software (por ejemplo, las interfaces de Java) en una aplicación. (Sommerville, 2002)

A diferencia de las clases conceptuales del Modelo del Dominio, las clases de diseño de los DCD muestran las definiciones de las clases software en lugar de los conceptos del mundo real. (Sommerville, 2002) A continuación se muestra el funcionamiento de la solicitud, ver figura 8:

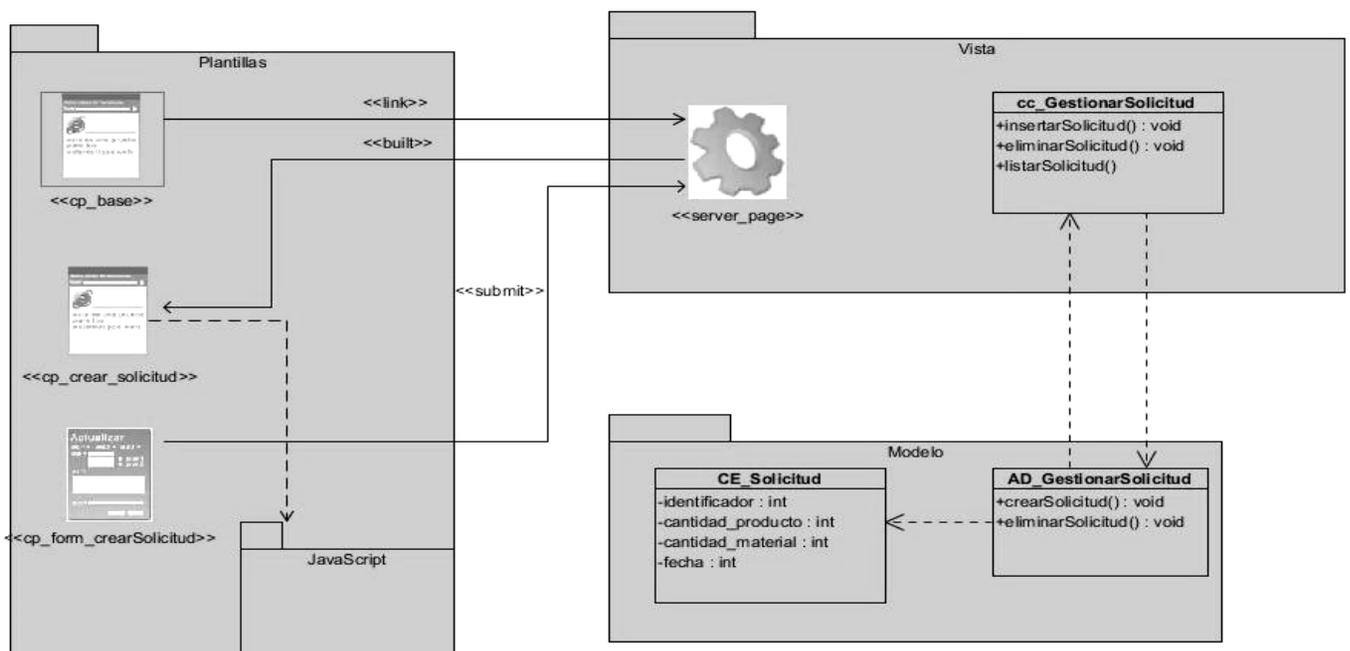


Figura 8 Diagrama de Clases del Diseño HU 10 Gestionar Solicitud

2.7 Diagrama de componentes

El diagrama de componentes ayuda a modelar el aspecto físico de un sistema de software orientado a objetos. Ilustra las arquitecturas de los componentes de software y las dependencias entre ellos. Aquellos componentes de software que incluyen componentes en tiempo de ejecución, componentes ejecutables y, los componentes de código fuente. (Colectivo de autores) A continuación de muestra todos los componentes utilizados para la realización del software, ver figura 9:

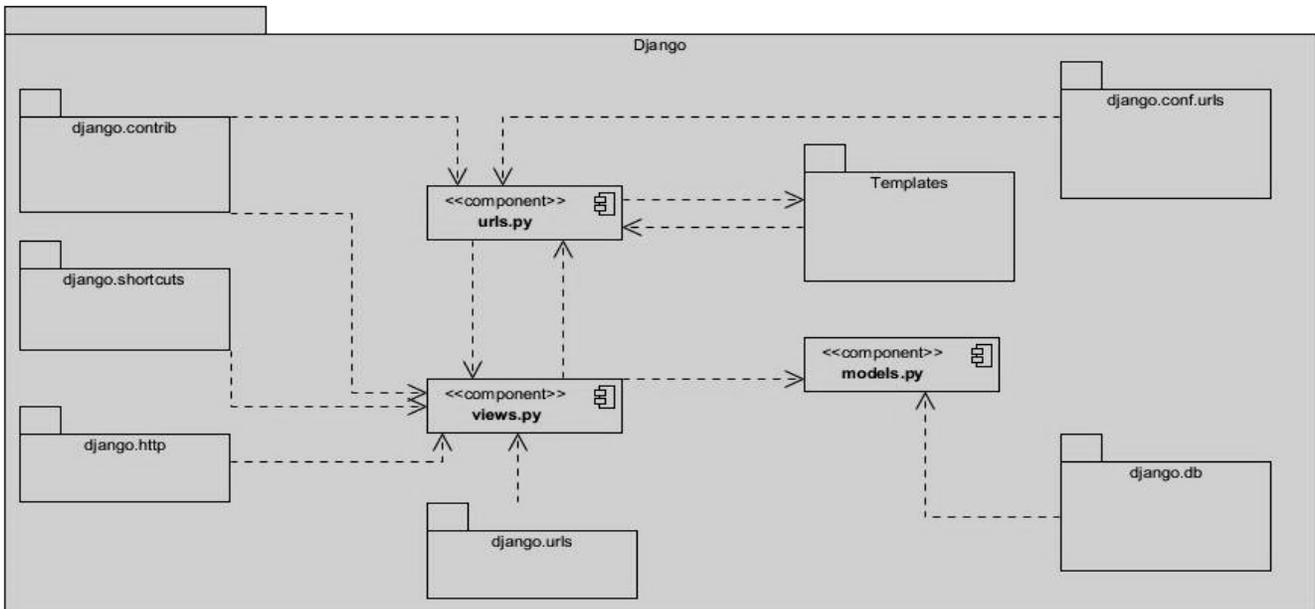


Figura 9 Diagrama de Componentes

Django.conf.urls: es un paquete que gestiona las configuraciones relacionadas con las urls en Django.

Django.contrib: se suministra con una variedad de herramientas adicionales opcionales que resuelven problemas comunes de desarrollo web.

Django.shorcuts: este paquete recopila funciones de ayuda y clases que "abarcen" múltiples niveles de MVC.

Django.http: este paquete recopila una serie de parámetros Http.

Django.contrib.auth: este sistema maneja tanto la autenticación, que verifica si un usuario tiene permiso para acceder a un sitio determinado, como la autorización, que define lo que un usuario puede hacer o no en el sitio.

Django.db: va a manejar los datos referentes al trabajo con la base de datos.

Django.contrib.auth.models: implementa una serie de módulos asociados a la autenticación.

urls.py: archivo que contiene todas las direcciones que va a contener la aplicación permitiendo encontrar la vista correcta.

models.py: archivo que contiene todas las clases modelo que posibilitan la interacción con la base de datos.

views.py: archivo que contiene todas las funciones de nuestra aplicación y que mediante las direcciones interactúa con la plantilla.

Templates: paquete que contiene todas las plantillas relacionadas con la aplicación.

2.8 Modelo de datos

El modelo de datos es una colección de herramientas conceptuales que se emplean para especificar datos, las relaciones entre ellos, su semántica asociada y las restricciones de integridad. Entre sus clasificaciones está el modelo de datos lógico que se usa para describir datos a nivel conceptual y externo caracterizándose por tener una amplia capacidad expresiva, son muy flexibles y permiten especificar ciertas restricciones de integridad. (CARRILLO, 2005) Para el desarrollo del sistema fue necesario crear el modelo de datos, ver figura 10:

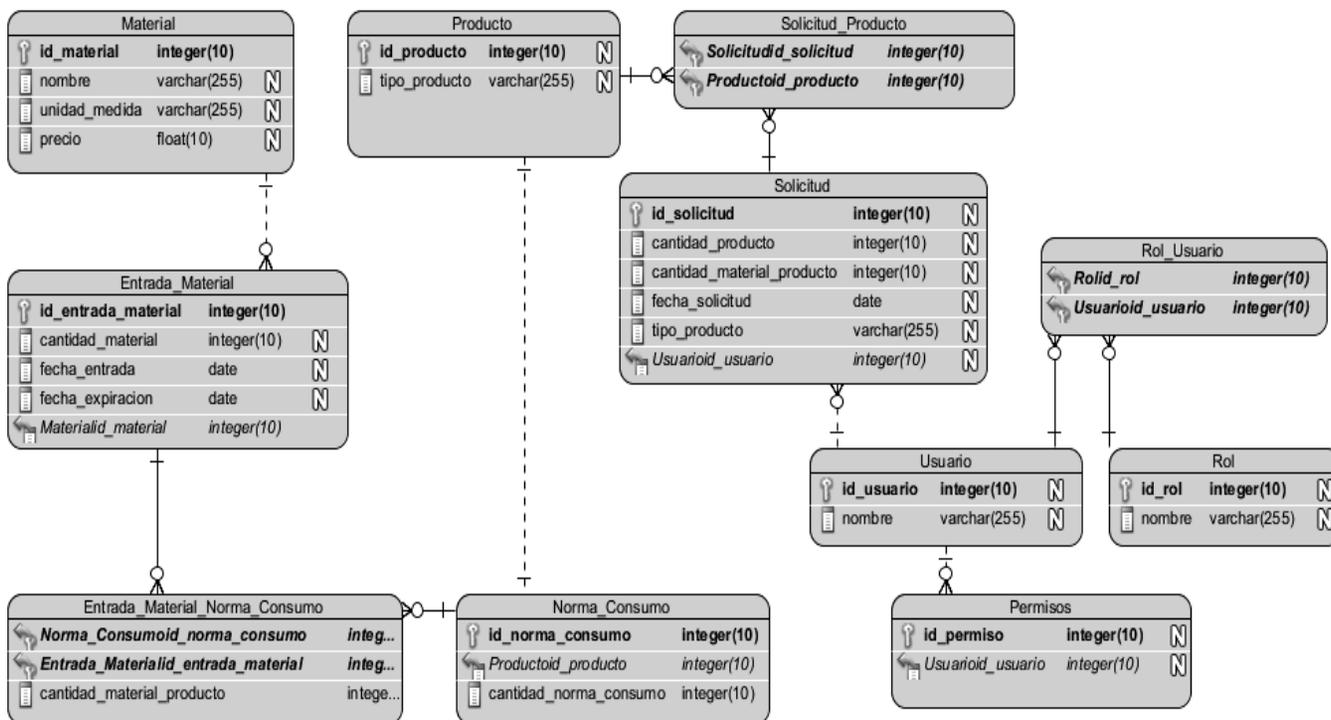


Figura 10 Modelo de Datos

2.9 Diagrama de despliegue

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema, la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP. Son útiles para facilitar la comunicación entre los ingenieros de hardware y los de software. (Ferré Grau, et al., 2004)

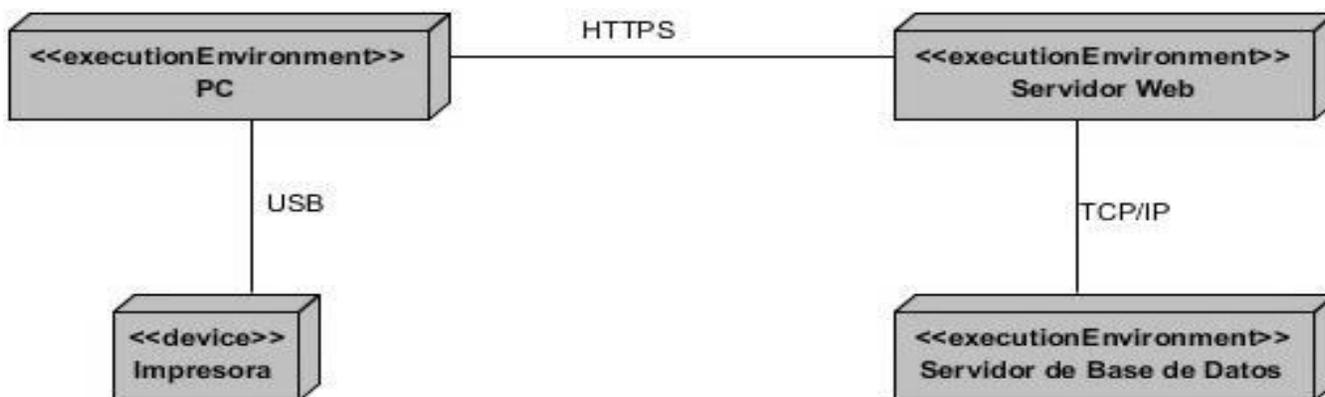


Figura 11 Diagrama de Despliegue

El **Dispositivo PC** representa el dispositivo desde el cual el usuario, mediante un navegador web, podrá ejecutar alguna gestión. El **Servidor web** es el encargado de atender las solicitudes del cliente el **Servidor de Base de Datos** provee servicios de base de datos. La comunicación entre el **Dispositivo PC** y el **Servidor web** se realiza mediante el Protocolo Seguro de Transferencia de Hipertexto (HTTPS) y con el **Servidor de Base de Datos** mediante TCP/IP⁴ por el puerto 5432.

2.10 Conclusiones parciales del capítulo

- En el presente capítulo se realizó una descripción detallada de las características con las que debe contar el Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.

⁴ Provee conectividad de extremo a extremo especificando cómo los datos deberían ser formateados, direccionados, transmitidos, enrutados y recibidos por el destinatario.

- La elaboración de los requisitos funcionales y no funcionales, historias de usuario y demás artefactos generados permitió conocer las funcionalidades del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI y sus comportamientos específicos y, la aplicación de patrones GRASP y GoF para un adecuado diseño de la propuesta de solución.

Capítulo 3: Implementación y pruebas del sistema de planificación y control de inventarios

En este capítulo se definen los artefactos correspondientes a las etapas de implementación y pruebas, así como los estándares de programación que debe seguir el equipo de desarrollo, las tareas de programación derivadas de cada Historia de Usuario (HU), y las diferentes pruebas seleccionadas para la aprobación de la implementación del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.

3.1 Estándar de codificación.

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Una faceta del desarrollo de software en la que todos los programadores influyen especialmente es en la técnica de codificación. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación. (Colectivo de autores, 2013)

Tabla 5 Estándar de codificación

Tipo de estándar	Descripción
Sangría	<ul style="list-style-type: none">Las líneas de continuación deben alinearse verticalmente con el carácter que se ha utilizado (paréntesis, llaves, corchetes).Utilizar una indentación de una tabulación para cada línea con excepción de la primera.La indentación se realizará solamente con tabulaciones, no debe utilizarse nunca los cuatro (4) espacios.
Máxima longitud de líneas	<ul style="list-style-type: none">Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas.

	<ul style="list-style-type: none"> • En cualquier circunstancia se puede utilizar el carácter “\” para cortar las líneas largas.
Líneas en blanco	<ul style="list-style-type: none"> • Separar las funciones de alto nivel y definiciones de clases con dos (2) líneas en blanco. • Las definiciones de métodos dentro de una clase deben separarse por una (1) línea en blanco. • Se pueden utilizar líneas en blanco escasamente para separar secciones lógicas.
Codificación	<ul style="list-style-type: none"> • Utilizar la codificación UTF-8. • Se pueden incluir cadenas que no correspondan a esta codificación utilizando “\x”, “\u” o “\U”.
Importaciones	<ul style="list-style-type: none"> • Las importaciones deben estar en líneas separadas. • Siempre deben colocarse al comienzo del archivo. • Deben quedar agrupadas de la siguiente forma: <ol style="list-style-type: none"> 1. Importaciones de la librería estándar. 2. Importaciones terceras relacionadas. 3. Importaciones locales de la aplicación / librerías. • Cada grupo de importaciones debe estar separado por una línea en blanco.

<p>Espacios en blanco en expresiones y declaraciones</p>	<ul style="list-style-type: none"> • Evitar utilizar espacios en blanco en las siguientes situaciones: • Inmediatamente dentro de paréntesis, corchetes y llaves. • Inmediatamente antes de una coma, un punto y coma o dos puntos. • Inmediatamente antes del paréntesis que comienza la lista de argumentos en la llamada a una función. • Inmediatamente antes de un corchete que empieza una indexación. • Más de un espacio alrededor de un operador de asignación (u otro) para alinearlos con otro. • Deben rodearse con exactamente un espacio los siguientes operadores binarios: <ol style="list-style-type: none"> 1. Asignación (=). 2. Asignación de aumentación (+=, -=, etc.). 3. Comparación (==, <, >, >=, <=, !=, <>, in, not in, is, is not). 4. Expresiones lógicas (and, or, not). • Si se utilizan operadores con prioridad diferente se aconseja rodear con espacios a los operadores de menor prioridad. • No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto.
<p>Comentarios</p>	<ul style="list-style-type: none"> • Los comentarios deben ser oraciones completas.

	<ul style="list-style-type: none"> • Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula. • Nunca cambiar las minúsculas y mayúsculas en los identificadores de clases, objetos, funciones, etc. • Si un comentario es corto el punto final puede omitirse.
Comentarios en bloque	<ul style="list-style-type: none"> • Deben estar indentados al mismo nivel que el código a comentar. • Cada línea de un comentario en bloque comienza con un numeral (#) y un espacio en blanco.
Comentarios en la línea	<ul style="list-style-type: none"> • Se recomienda utilizarlos moderadamente. • Se debe definir comenzando por un numeral (#) seguido de un espacio en blanco. • Deben ubicarse en la misma línea que se desea comentar.
Cadenas de documentación	<ul style="list-style-type: none"> • Deben quedar documentados todos los módulos, funciones, clases y métodos públicos. • Para definir una cadena de documentación debe quedar encerrada dentro de (""). • Los ("" que finalizan una cadena de documentación deben quedar en una línea a no ser que la cadena sea de una sola línea.

<p>Convenciones de nomenclatura</p>	<ul style="list-style-type: none"> • Nunca se deben utilizar como simples caracteres para nombres de variables los caracteres de minúscula “l”, o mayúscula “O”, o mayúscula “L” ya que en algunas fuentes son indistinguibles de los números uno (1) y cero (0). • Los módulos deben tener un nombre corto y en minúscula. • Los nombres de clases deben utilizar la convención “<i>CapWords</i>” (palabras que comienzan con mayúsculas). • Los nombres de las excepciones deben estar escritos también en la convención “<i>CapWords</i>” utilizando el sufijo “Error”. • Los nombres de las funciones deben estar escrito en minúscula separando las palabras con un guion bajo “_”. • Las constantes deben quedar escritas con letras mayúsculas separando las palabras por un guion bajo (_).
-------------------------------------	---

3.2 Estrategia de pruebas de software

Una estrategia de prueba de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados, además, debe ser suficientemente flexible para promover un uso personalizado de la prueba. Al mismo tiempo, debe ser suficientemente rígida para alentar la planificación razonable y el seguimiento de la gestión conforme avanza el proyecto. (Pressman, 2010)

3.2.1 Pruebas funcionales al sistema

Un aspecto crucial en el control de calidad del desarrollo de software son las pruebas y, dentro de estas, las pruebas funcionales, en las cuales se hace una verificación dinámica del comportamiento de un sistema, basada en la observación de un conjunto seleccionado de ejecuciones controladas o casos de prueba. (Gutierrez, et al., 2005)

Las pruebas funcionales son aquellas que se aplican al producto final, y permiten detectar en qué puntos el producto no cumple sus especificaciones, es decir, comprobar su funcionalidad. Para realizarlas se debe hacer una planificación que consiste en definir los aspectos a examinar y la forma de verificar su correcto funcionamiento, punto en el cual adquieren sentido los casos de prueba. (Lewis, 2005)

Diseño de casos de prueba:

Tabla 6 Variables para el caso de prueba funcional Solicitud de productos

	Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
1	Código	Campo numérico	No	Se inserta el identificador de la solicitud de productos	IS
2	Cantidad	Campo numérico	No	Se inserta la cantidad de productos que se desea confeccionar	CP
3	Producto	Campo seleccionable	No	Se selecciona el tipo de producto a confeccionar	TP
4	Fecha de solicitud	Campo automático	No	El valor por defecto es la fecha en que se realiza la solicitud	FS
5	Cantidad de materiales	Campo automático	No	Se genera automáticamente la cantidad necesaria de materiales para la confección de los productos solicitados	CM

Tabla 7 Caso de prueba funcional Insertar Solicitud

Descripción General							
Permitirá realizar una solicitud de productos							
Condiciones de ejecución							
Para realizar una nueva solicitud de productos el usuario debe estar logueado en el sistema y poseer los permisos necesarios, además, debe estar definido previamente un tipo de producto en el sistema.							
Escenario	Descripción	IS	CP	TP	CM	R/ del sistema	Flujo central
ESC 1.1 Insertar solicitud de productos	El usuario debe seleccionar el campo del producto e insertar la cantidad demandada	30	20	Solapín	20 pre.impresos. 40 micas. 0.004 Tonel de Impresión	El sistema debe insertar una solicitud satisfactoriamente y muestra notificación de éxito	En el menú a la izquierda se sigue la siguiente ruta: "Demandas de Productos –Realizar Demandas". Aparecerá un formulario con un campo seleccionable en los cuales se encontrará el producto a solicitar, un campo para insertar el identificador de la solicitud, un campo para insertar la cantidad de productos que desea solicitar diario. Una vez realizada la solicitud debe mostrarse una lista con todas las solicitudes y la

							fecha en la cual se realizaron, incluyendo la insertada
--	--	--	--	--	--	--	---

Tabla 8 Caso de prueba funcional Eliminar Solicitud

Descripción General							
Permitirá eliminar una solicitud de productos							
Condiciones de ejecución							
Para eliminar una solicitud de productos el usuario debe estar logueado en el sistema y poseer los permisos necesarios, además, debe ser insertada previamente una solicitud en el sistema por el usuario.							
Escenario	Descripción	IS	CP	TP	CM	R/ del sistema	Flujo central
ESC 1.2 Eliminar una solicitud de productos.	El usuario debe seleccionar botón de eliminar.	No	No	No	No	El sistema debe eliminar una solicitud.	En el menú a la izquierda se sigue la siguiente ruta: "Demandas de Productos –Registro de Demandas". Aparecerá un listado con todas las solicitudes creadas por el usuario. El usuario selecciona la opción eliminar que se encuentra en la columna acciones a la derecha de la lista.

Resultados de las pruebas funcionales:

Al sistema para la aprobación de los requisitos funcionales se le realizaron tres iteraciones donde se encontraron un total de 19 no conformidades, 13 errores ortográficos y 6 funcionalidades incorrectas, en la primera iteración se identificaron 9 errores ortográficos y 4 funcionalidades incorrectas, en la segunda iteración 4 errores ortográficos y 2 funcionalidades incorrectas, en la tercera iteración se eliminaron las no conformidades y con esto se obtuvo un sistema funcional, ver figura 7:

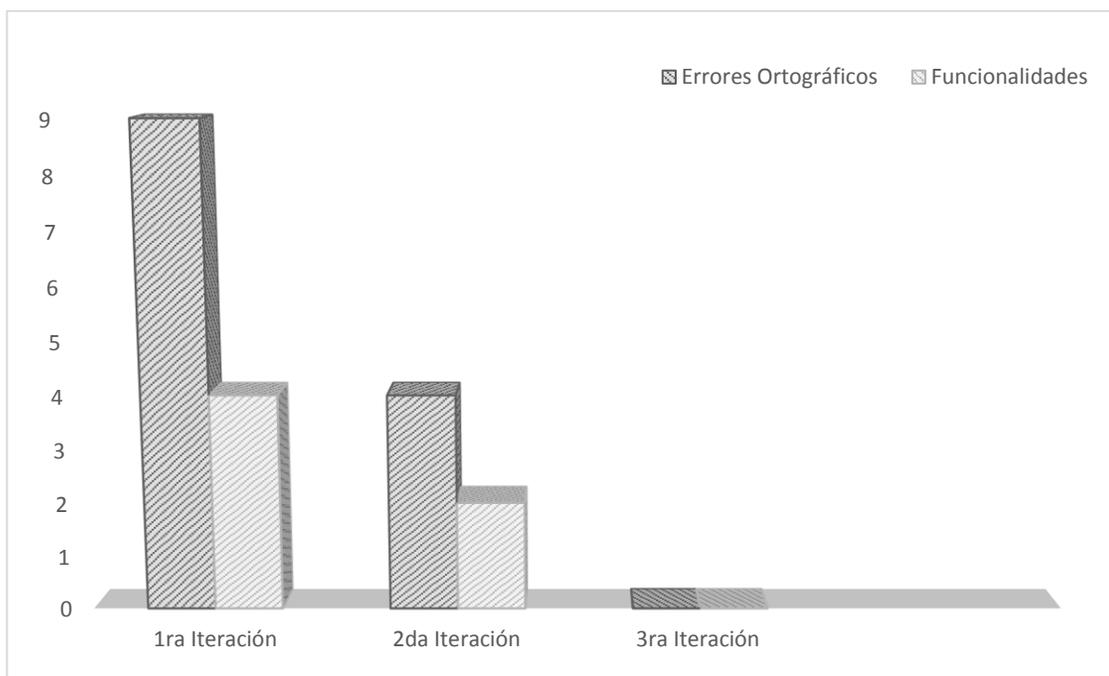


Figura 12 Resultados de las Pruebas Funcionales

Entre las no conformidades que se detectaron durante el proceso de pruebas se pueden encontrar las siguientes:

- Errores ortográficos en los nombres de los campos.
- Los mensajes de error no corresponden con los errores que ocurren.
- Errores por la ausencia de validación en los formularios y el modelo.
- Escaso tratamiento de excepciones.

3.2.2 Pruebas de seguridad

Las pruebas de seguridad buscan medir la Confidencialidad, Integridad y Disponibilidad de los datos, desde la perspectiva del aplicativo, es decir partiendo a identificar amenazas y riesgos desde el uso o interface de usuario final. Una vez ejecutadas las pruebas de seguridad es posible medir y cuantificar los riesgos a los cuales se ven expuestos los aplicativos tanto en la infraestructura interna como externa, valiéndose de la filosofía del Hacking ético. (Quality V V, 2015)

Al sistema desarrollado se le realizaron una serie de pruebas de seguridad mediante el software Acunetix v9.5, las cuales se presentan a continuación:

- Ataques de inyección SQL.
- *Cross-Site Scripting (XSS)*.
- Falsificación de petición (CSRF).

Resultados de las pruebas de seguridad:

La prueba realizada permitió detectar en la primera iteración varios errores de seguridad que se muestran en la figura 10, y quedan resumidos en credenciales en texto plano, ataques por fuerza bruta, directorios sensibles sin protección. A continuación, se muestran los resultados:

1ra Iteración:

Fueron detectadas un total de 18 vulnerabilidades, ver la figura 13.

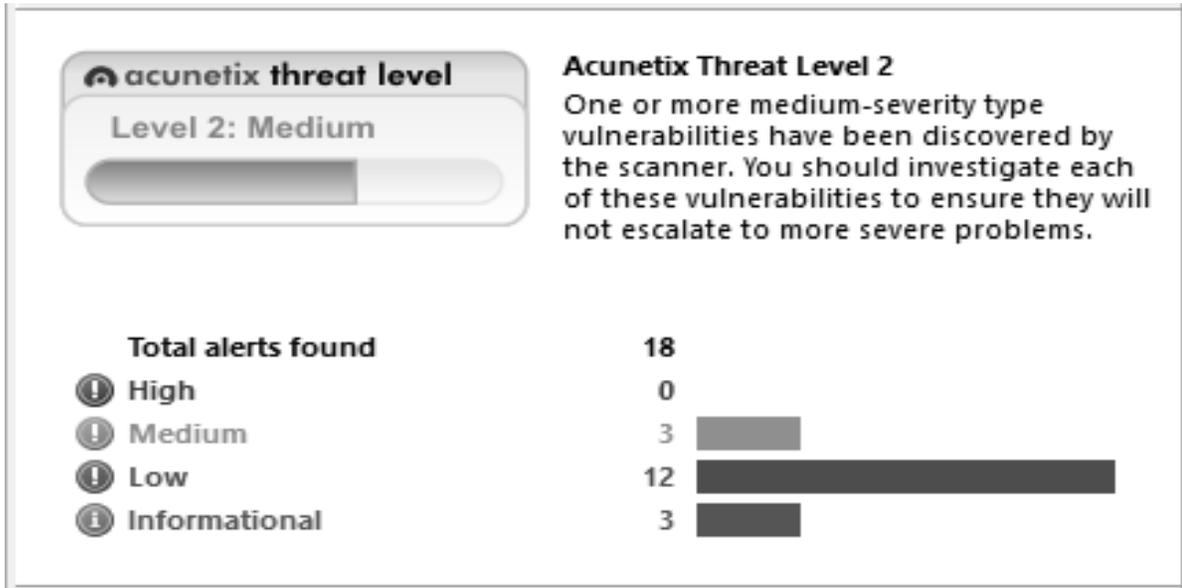


Figura 13 Resultado de la prueba de seguridad 1ra iteración

2da Iteración:

Fueron detectadas un total de 6 vulnerabilidades, ver figura 14.

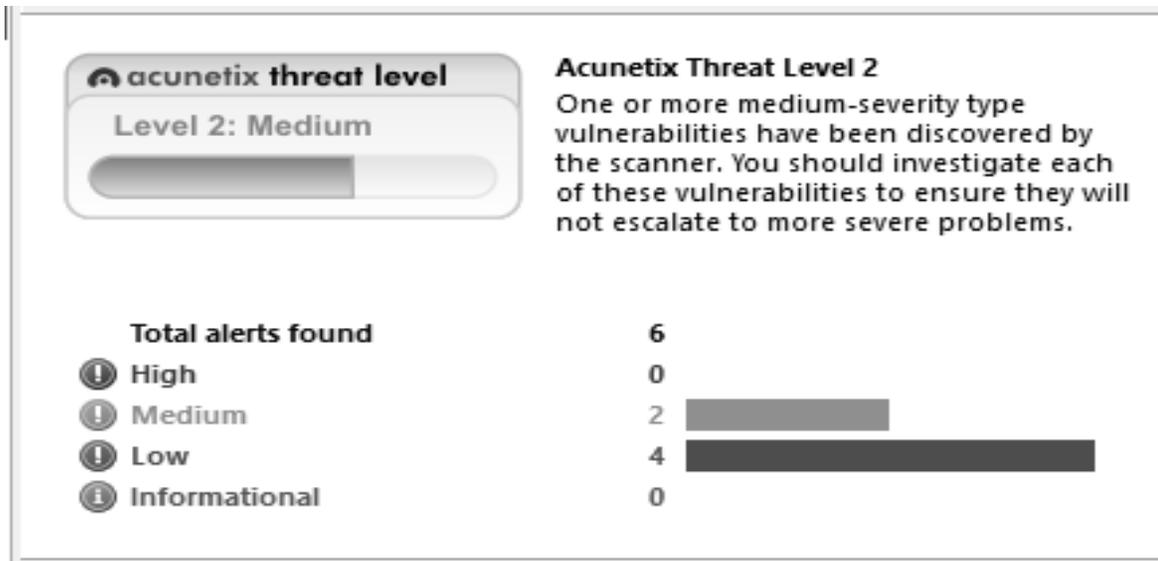


Figura 14 Resultado de la prueba de seguridad 2da iteración

3ra Iteración:

No fue detectada ninguna vulnerabilidad por lo que se obtuvo un sistema seguro.

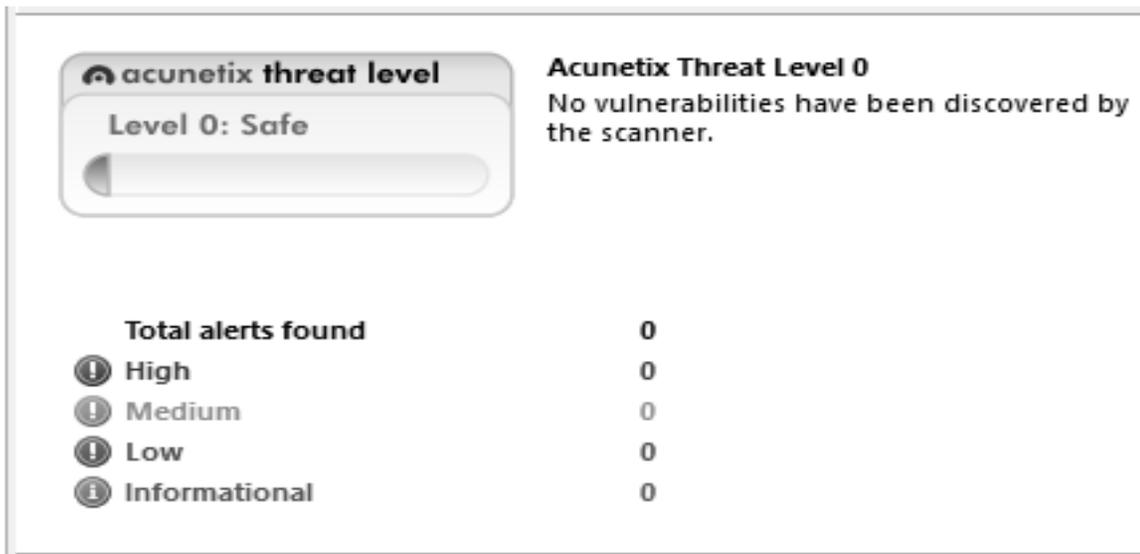


Figura 15 Resultado de la prueba de seguridad 3ra iteración

3.3 Validación de la propuesta

Para la validación de la propuesta de solución se empleó el método Diagnóstico, con el fin de evaluar cómo se comportaba el proceso antes del desarrollo del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI y cómo se comporta con la asistencia del mismo. Se tomaron como muestra los meses de enero a febrero sin el apoyo del sistema, luego se tomaron los meses de marzo a abril con el sistema funcionando y se aplicó el método (ver figura 9), arrojándose los siguientes resultados:

Muestra:

Como muestra se tomó en un período de dos meses cómo fue el comportamiento de los gastos, los consumos de inventarios y la producción de solapines en la oficina antes del funcionamiento del sistema, y para el empleo del método se tomó otra muestra, pero después de la puesta en marcha del sistema.

Antes del funcionamiento del sistema:

Cantidad de solapines planificados: 250

Cantidad de solapines confeccionados: 180

Cantidad de micas utilizadas: 420

Cantidad de pre-impresos utilizados: 200

Cantidad de tonel de impresión utilizados: 4

Después del funcionamiento del sistema:

Cantidad de solapines planificados: 200

Cantidad de solapines confeccionados: 190

Cantidad de micas utilizadas: 380

Cantidad de pre-impresos utilizados: 195

Cantidad de tonel de impresión utilizados: 4

Indicadores de comparación (Mejía Cañas, 2013):

Norma de consumo:

Para este indicador se toma la muestra de cómo ha sido el consumo de cada inventario entre la norma de consumo de una determinada cantidad de solapines y se evalúa con el fin de saber cuál es el ajuste más óptimo.

$$\frac{\text{consumo}}{\text{norma de consumo} * S} > 1$$

Antes:

S = 180

micas = 420 / 360

micas = 1.11

pre-impreso = 200 / 180

pre-impreso = 1.11

tonel = 4 / 4

Después:

S = 190

micas = 380 / 380

micas = 1

pre-impreso = 195 / 190

pre-impreso = 1.02

tonel = 4 / 4

tonel = 1

tonel = 1

Se suman los valores obtenidos y se divide entre la cantidad para obtener la media:

Antes:

1.11+1.11+1

= 1.073

Después:

1+1.02+1

= 1.006

Después de la puesta en funcionamiento del sistema se puede observar que se ajusta mejor a la norma de consumo, por lo tanto, mejora en comparación a cuando no se utilizó el sistema.

consumo: consumo de los inventarios en un período.

normadeconsumo: cantidad de inventarios necesarios para la confección de una determinada cantidad de solapines.

S: cantidad de solapines.

Planificación:

Para este indicador se toma la muestra de cuál ha sido la planificación de los inventarios contra la demanda real ejecutada del servicio, con el fin de obtener una mejor planificación de los inventarios.

$\frac{\text{planificacion}}{\text{demandareal}} > 1$

Antes:

250 / 180

= 1.38

Después:

200 / 190

= 1.05

Con el uso del sistema funciona mejor la planificación en comparación a cómo lo realizaban anteriormente.

planificacion: planificación de los inventarios para la confección de los solapines.

demandareal: demanda real de los inventarios para la confección de los solapines.

Gastos:

Para este indicador se toma la muestra de cuál ha sido el gasto del servicio de personalización.

$$\text{gasto} = \frac{\Sigma(\text{consumo} * \text{material})}{s}$$

Antes:

$$(420+200+4) / 180$$

$$= 620.02$$

Después:

$$(380+195+4) / 190$$

$$= 575.02$$

$$\text{antes} - \text{después} = 45$$

Como es evidente el gasto disminuye con el sistema funcionando y se ahorra un total de 45 unidades por solapines.

gasto: gasto del servicio de personalización.

material: tipo de inventario.

Tabla 9 Diagnóstico para validar la propuesta

Indicadores	Diagnóstico		
	Antes del sistema	Con el sistema	Descripción
Norma de consumo	1.073	1.006	Mejor ajuste a la norma con el sistema.
Planificación	1.38	1.05	Mejor planificación con el sistema.
Gastos	620.02	575.02	Con el sistema el gasto actual es menor que antes de desplegarlo. La diferencia es el ahorro de los gastos(45).

Como resultado final se evidenció que el sistema mejora la planificación y el control de los inventarios utilizados en el proceso de personalización de documentos de identificación.

Conclusiones parciales del capítulo

- El uso de los estándares de codificación permitió homogeneizar la estructuración del código asegurando la calidad, evitando errores y permitiendo un fácil mantenimiento del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.
- La aplicación de las pruebas funcionales y de seguridad ayudó a demostrar el cumplimiento de las funcionalidades del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI e identificar las principales deficiencias, así como, la estrategia para solucionar los errores detectados y obtener un producto con un alto valor y utilidad para una adecuada planificación y control de inventarios.
- Con la validación de la propuesta de solución se comprobó que el sistema mejora la planificación y el control de los inventarios utilizados en la oficina.

Conclusiones

Con la investigación realizada se llegaron a las siguientes conclusiones:

- Los referentes teóricos asociados a la planificación y el control de los inventarios, facilitaron la comprensión del problema de investigación y contribuyó en la elaboración de su diseño teórico-metodológico.
- A partir de la valoración del estado del arte se identificaron soluciones, tecnologías y métodos necesarios, que pudieran reutilizarse para desarrollar el Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.
- La descripción y diagnóstico del proceso de planificación y control de los inventarios llevado a cabo en la Oficina de Identificación de la UCI permitió obtener los requisitos del sistema para diseñar la propuesta de solución.
- A partir de los requerimientos identificados se obtuvieron los modelos de análisis y diseño del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.
- Mediante los modelos de análisis y diseño se elaboró el modelo de datos persistentes que permitió la implementación del Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI.
- Mediante las pruebas de software realizadas se erradicaron las insuficiencias detectadas en el Sistema para la planificación y el control de inventarios en la Oficina de Identificación de la UCI, proporcionándole mayor seguridad y facilidad de uso a las funcionalidades presentes.
- La validación de la propuesta mediante el método Diagnóstico demostró la mejora de los procesos de planificación y control de los inventarios en la Oficina de Identificación de la UCI.

Recomendaciones

Luego de concluir con la investigación y alcanzar los resultados deseados, el paso siguiente es definir algunas recomendaciones a la UCI para que el trabajo sea aún más beneficioso:

- ✓ A la UCI, en particular al centro CISED continuar la investigación con vistas a incorporar un algoritmo de predicción, donde el sistema pueda determinar la solicitud de materiales en función de la demanda de solapines en determinados niveles de tiempo.
- ✓ A la UCI, en particular al centro CISED continuar la investigación con vistas a incorporar el cálculo del gasto monetario por materiales utilizados.
- ✓ A la UCI, en particular al centro CISED y a la Oficina de Identificación desplegar el sistema hacia cualquier entidad productiva para planificar y controlar los inventarios de la misma.

Referencias bibliográficas

Lawrence Journal-World. 2015. *Django, The Web framework for perfectionists with deadline.* 2015.

Zuzuarregui Ibarbia, Ane and Huerta Mendizabal, Amaia. 2015.

Análisis de las características de los ERPs para pymes: Una guía preliminar de cara a la elección de las soluciones más eficientes. 2015.

¿Qué es Programación Informática? 2017. conceptodefinicion.de. [Online] 2017.

<http://conceptodefinicion.de/programacion-informatica/>.

Alonso Velázquez, José Luis . 2010. *Lenguaje de Programación: Introducción a C/C++(IDE).*

Universidad de Guanajuato. Ciudad de Guanajuato : s.n., 2010.

Bahit, Eugenia. 2012. *Curso: Python para Principiantes.* Buenos Aires : s.n., 2012.

Cabrera, Elibeth. 2003. *Control como función administrativa en la empresa.* 2003.

CARRILLO, SILVIA ACID y Otros. 2005. *Introducción a las bases de datos. El modelo relacional.*

Madrid : Thomsom Paraninfo, 2005.

Colectivo de autores. *Tool, Full-Featured UML Software Design.*

—. **2017.** Concepto de Seguridad. [Online] 2017. [Cited: diciembre 2017, 5.]

<http://www.mitecnologico.com/Main/ConceptoDeSeguridad..>

—. **2017.** developer.mozilla.org. [Online] 2017. <https://developer.mozilla.org/es/docs/Web/HTML>.

—. **2013.** Guía de estilo para el código Python – PEP 8 en Español. 2013.

—. Ministerio de comunicación. [Online] [Cited: diciembre 2017, 7.]

<http://www.mincom.gob.cu/?q=node/550>.

Colectivo de Autores. 2017. *ReportLab PDF Library User Guide.* 2017.

Colectivo de autores. visual-paradigm.com. [Online] [Cited: marzo 2, 2018.] <https://www.visual-paradigm.com/VPGallery/diagrams/Component.html>.

—. **2016.** W3C. [Online] 2016. <http://www.w3.org/standards/webdesign/htmlcss#whatcss>.

Colectivo de autores,. Monografías. *Inventarios.* [Online] [Cited: octubre 3, 2017.]

<http://www.monografias.com/trabajos11/conin/conin2.shtml/#contr>.

Departamento de Sistemas Informáticos y Computación. *Introducción a Herramientas CASE y System Architect.* Universidad Politécnica de Valencia.

- developer.mozilla. 2015.** developer.mozilla.org Acerca de JavaScript. [Online] 2015.
https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript.
- Dominguez, Jessica . 2016.** postdata.club : ¿Es usted único en Cuba? [Online] 2016.
<http://www.postdata.club/issues/201609/es-usted-unico-en-cuba.html>.
- Ferré Grau, Xavier and Sánchez Segura, María Isabel. 2004.** Desarrollo Orientado a Objetos con UML. Madrid : s.n., 2004.
- Flores, Ervin . 2009.** *METODOLOGÍAS ÁGILES: PROCESO UNIFICADO ÁGIL (AUP)*. Caracas : s.n., 2009.
- Franganillo, Jorge. 2011.** Html5: El nuevo estándar básico de la Web. s.l. : ThinkEPI, 2011, pp. 261–265.
- Gallardo López, D and Pomares Puig, C . 2008.** *Lenguajes de Programación y Paradigmas de Programación*. 2008.
- García Hernández, Teymor David and Torres Diéguez, Adrián . 2012.** *Módulo para la gestión de los insumos para el proceso de personalización de documentos de identificación de la República de Cuba*. La Habana : s.n., 2012.
- Gavilema, B. 2015.** CONTROL DE INVENTARIO EN LA FARMACIA Y BODEGA DE LA CLINICA MATERNIDAD MOSQUERA. [Online] 2015. [Cited: octubre 3, 2017.]
<https://www.emprendepyme.net/inventario>.
- Giraldo G., Gloria L and Acevedo, Juan F. 2011.** *Una ontología para la representación de conceptos de diseño de software*. Facultad de Minas, Universidad Nacional de Colombia. Medellin : s.n., 2011. pp. 103-110.
- Gomez and Suarez. 2003.** 2003.
- Gomez, Suárez. 2003.** 2003.
- González Duque, Raúl. 2008.** *Python para todos*. s.l. : Creative Commons, 2008.
- Grupo ISSI (Ingeniería del Software y Sistemas de Información). 2003.** *Metodologías Ágiles en el Desarrollo de Software*. Alicante : s.n., 2003.
- Gutierrez, Javier, et al. 2005.** *Analysis of Proposals to Generate System Test Cases From System Requirements*. Porto : s.n., 2005.
- Holovaty, Adrian and Kaplan Moss, Jacob. 2007.** *The Definitive Guide to Django*. 2007.

- Holovaty, Adrian and Kaplan-Moss, Jacob. 2007.** *The Definitive Guide to Django: Web Development Done Right*. s.l. : Apress, 2007.
- Jetbrains. 2017.** jetbrains.com. [Online] 2017. [Cited: noviembre 7, 2017.] <https://www.jetbrains.com/pycharm/>.
- Larman, Craig. 2003.** *UML y Patrones*. Madrid : Prentice Hall, 2003.
- Letelier, Patricio, Canó, José H and Penadés, María Carmen. 2003.** *Metodologías Ágiles en el Desarrollo de Software*. Alicante : s.n., 2003.
- Lewis, William . 2005.** *Software Testing and Continuous Quality Improvement*. Segunda. Florida : CRC Press LLC , 2005.
- Massó Verrier, Eliska Francy and Pachecho González, Alberto . 2010.** *Módulo de control de insumos para el CPDI*. 2010.
- Mejía Cañas, Carlos Alberto. 2013.** *Planning Consultores Gerenciales*. Columbia : s.n., 2013. 1302.
- Mozilla. 2016.** developer.mozilla.org. [Online] MDN web docs, 2016. <https://developer.mozilla.org/es/docs/Web/CSS>.
- Oliva Martínez, Saily , et al. 2010.** Sistema para el control de inventarios del ERP cubano. La Habana : s.n., 2010.
- Orozco Vaillant, María E. . 2013.** *Informe de investigación de la Metodología SXP*. Bayamo : s.n., 2013.
- Popkin Software and System.** *Modelado de Sistemas con UML*.
- PostgreSQL.** postgresql.org. [Online] [Cited: noviembre 8, 2017.] <http://www.postgresql.org/about/>.
- Pressman, R. 2010.** *Ingeniería de Software. Un Enfoque Práctico*. Séptima. 2010.
- Quality V V. 2015.** Pruebas de seguridad. *V&V Quality Performance, security & Process Solutions*. 2015.
- Quality, V V. 2015.** Pruebas de rendimiento. *V&V Quality Performance, security & Process Soluti*. 2015.
- Quispe Carita , Vilma, Huamantuco Solorzano, Dante Harry and Vargas Yupanqui, Jose Luis. 2011.** *METODOLOGIA RUP (RATIONAL UNIFIED PROCESS)*. Lima : s.n., 2011.
- Quispe Carita, Vilma, Huamantuco Solorzano, Dante Harry and Vargas Yupanqui, José Luis. 2011.** *Metodología RUP (RATIONAL UNIFIED PROCESS)*. Lima : s.n., 2011.
- Quispe-Otazu, Rodolfo. 2007.** *Computación e Informática*. 2007.

Reyes, Alfonso. 2001. Secretaría de Economía. [Online] 2001. [Cited: octubre 3, 2017.] <http://www.contactopyme.gob.mx/guiasempresariales/guias.asp?s=9&g=7..>

Reyes, Primitivo. 2009. *Administración de Inventarios en almacenes: Logística y Operación.* México : s.n., 2009.

Robbins, Stephen P. and Coulter , Mary . 2005. *Administración.* Octava. s.l. : Pearson Education, Inc, 2005.

Rodríguez Gutiérrez , Diana Paola and Refulio Sánchez , Ronald Percy . 2011. *MODELO DE CONTROL Y GESTIÓN DE EXISTENCIAS EN UNA EMPRESA DE TELECOMUNICACIONES USANDO COMO HERRAMIENTA DE SOPORTE LOS MÓDULOS MM - WM DEL SISTEMA SAP R/3.* Lima : s.n., 2011.

Rodríguez Sánchez , Tamara . 2015. *Metodología de desarrollo para la Actividad productiva de la UCI.* La Habana : s.n., 2015.

Rosemann. 1999. *Sistemas ERP.* 1999.

Rumbaugh, James, Booch, Grady and Jacobson, Ivan. 2000. *El Proceso Unificado de Desarrollo.* 2000.

SAP. sap.com. [Online] [Cited: diciembre 2017, 7.] <https://www.sap.com/index.html>.

Sommerville, I. 2002. *Ingeniería de Software.* s.l. : Pearson Educación, 2002.

UCI. Universidad de las Ciencias Informáticas. [Online] [Cited: diciembre 5, 2017.] <http://www.uci.cu/investigacion-y-desarrollo/productos/xedro/cedrux>.

Zambrano Ramírez, Raquel. 2008. *Sistemas Gestores de Base de Datos.* Córdoba : s.n., 2008. ISSN 1988-6047.

Anexos

Historias de Usuarios

Tabla 10 HU 01 Gestionar Usuarios

Número: 01	Nombre del requisito: Gestionar Usuarios
Programador: Eduardo Baratutes González	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: -
Riesgo en Desarrollo: -	Tiempo Real: -
Descripción: Deben recogerse los siguientes datos: <ul style="list-style-type: none">• Nombre (Valor de solo letras y de carácter obligatorio)• Apellidos (Valor de solo letras y de carácter obligatorio)• Correo electrónico (Valor alfanumérico y de carácter obligatorio)• Usuario (Valor de solo letras y de carácter obligatorio)• Contraseña (Valor alfanumérico, de carácter obligatorio)• Permisos (Valor de solo letras y de carácter obligatorio)	

Observaciones:

Esta funcionalidad tiene como objetivo la implementación de la parte de la aplicación encargada del registro de usuarios.

Prototipo de interfaz:

The form is titled "Nuevo" and contains the following fields and controls:

- Nombre de usuario:
- Dirección de correo electrónico:
- Es superusuario:
- Es staff:
- Nombre:
- Apellidos:
- Activo:
- Contraseña (confirmación):
- Contraseña:

Buttons: ACEPTAR, CANCELAR

Tabla 11 HU 02 Gestionar Unidad de Medida

Número: 02	Nombre del requisito: Gestionar Unidad de Medida
Programador: Eduardo Baratutes González	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: -

Riesgo en Desarrollo: -	Tiempo Real: -
<p>Descripción:</p> <p>Deben recogerse los siguientes datos:</p> <ul style="list-style-type: none"> • Unidad de medida (Valor alfanumérico, de carácter obligatorio) 	
<p>Observaciones:</p> <p>Esta funcionalidad tiene como objetivo la implementación de la parte de la aplicación encargada de la definición de las unidades de medidas de los materiales.</p>	
<p>Prototipo de interfaz:</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>Nuevo Unidad</p> <p>Unidad de Medida *</p> <input style="width: 100%; height: 20px;" type="text"/> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="ACEPTAR"/> <input type="button" value="CANCELAR"/> </div> </div>	

Tabla 12 HU 03 Registrar Tipo de Material

Número: 03	Nombre del requisito: Registrar Tipo de Material
Programador: Eduardo Baratutes González	Iteración Asignada: 1

Prioridad: Alta	Tiempo Estimado: -
Riesgo en Desarrollo: -	Tiempo Real: -
<p>Descripción:</p> <p>Deben recogerse los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre (Valor de solo letras y de carácter obligatorio) • Unidad de Medida (Campo seleccionable de carácter obligatorio) • Precio (Valor numérico real) • Total (Valor numérico inicializado en 0) 	
<p>Observaciones:</p> <p>Esta funcionalidad tiene como objetivo la implementación de la parte de la aplicación encargada del registro de los materiales por tipo.</p>	
<p>Prototipo de interfaz:</p>	

Nuevo Material

Nombre del Material *

Unidad de Medida *

Precio *

Tabla 13 HU 05 Gestionar Salida de Materiales

Número: 05	Nombre del requisito: Gestionar Salida de Materiales
Programador: Eduardo Baratutes González	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: -
Riesgo en Desarrollo: -	Tiempo Real: -
<p>Descripción:</p> <p>Deben recogerse los siguientes datos:</p> <ul style="list-style-type: none"> Identificador del material (Campo seleccionable de carácter obligatorio) Cantidad gastada en reparación (Valor numérico) 	

- Cantidad gastada en confección (Valor numérico)

Observaciones:

Esta funcionalidad tiene como objetivo la implementación de la parte de la aplicación encargada de la salida de todos los materiales existentes en la oficina.

Prototipo de interfaz:

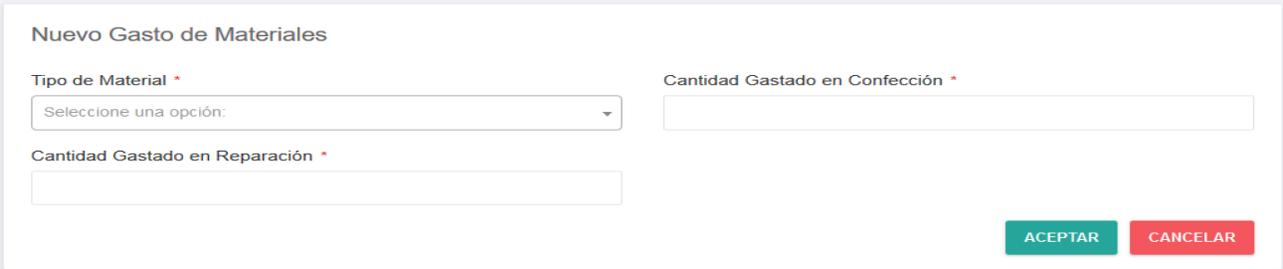


Tabla 14 HU 06 Gestionar Producto

Número: 06	Nombre del requisito: Gestionar Producto
Programador: Eduardo Baratutes González	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: -
Riesgo en Desarrollo: -	Tiempo Real: -

Descripción:

Deben recogerse los siguientes datos:

- Tipo de Producto (Valor de solo letras y de carácter obligatorio)

Observaciones:

Esta funcionalidad tiene como objetivo la implementación de la parte de la aplicación encargada de la gestión de los productos por su tipo.

Prototipo de interfaz:

Prototipo de interfaz para 'Nuevo Producto'. El formulario muestra el título 'Nuevo Producto', un campo de entrada etiquetado 'Nombre del Producto *' y dos botones: 'ACEPTAR' (verde) y 'CANCELAR' (rojo).

Tabla 15 HU 07 Gestionar Norma de Consumo

Número: 07	Nombre del requisito: Gestionar Norma de Consumo
Programador: Eduardo Baratutes González	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: -

Riesgo en Desarrollo: -	Tiempo Real: -
<p>Descripción:</p> <p>Deben recogerse los siguientes datos:</p> <ul style="list-style-type: none"> • Cantidad de material (valor numérico de carácter obligatorio) • Identificador del material (Campo seleccionable de carácter obligatorio) • Producto (Campo seleccionable de carácter obligatorio) 	
<p>Observaciones:</p> <p>Esta funcionalidad tiene como objetivo la implementación de la parte de la aplicación encargada de la gestión, la norma de consumo de los productos y los materiales que se utilizan para su confección.</p>	
<p>Prototipo de interfaz:</p> <div data-bbox="167 1182 1474 1503" style="border: 1px solid #ccc; padding: 10px;"> <p>Norma de Consumo para 1 Producto</p> <p>Material * Cantidad de Material *</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <input type="text" value="Seleccione una opción:"/> </div> <div style="width: 45%; border: 1px solid #ccc; height: 20px;"></div> </div> <p>Producto *</p> <input type="text" value="Seleccione una opción:"/> <div style="text-align: right; margin-top: 10px;"> ACEPTAR CANCELAR </div> </div>	

Tabla 16 HU 08 Solicitud de Producto



Número: 08	Nombre del requisito: Solicitud de Producto
Programador: Eduardo Baratutes González	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: -
Riesgo en Desarrollo: -	Tiempo Real: -
<p>Descripción:</p> <p>Deben recogerse los siguientes datos:</p> <ul style="list-style-type: none"> • Identificador (Valor numérico de carácter obligatorio) • Cantidad de Producto (valor numérico de carácter obligatorio) • Cantidad de Material por cada Producto (valor numérico que se genera solo) 	
<p>Observaciones:</p> <p>Esta funcionalidad tiene como objetivo la implementación de la parte de la aplicación encargada de la solicitud de los productos en la oficina para realizar el servicio, de aquí se genera la cantidad de materiales necesarios para la confección de una cantidad de solapines.</p>	
<p>Prototipo de interfaz:</p>	

Nueva Solicitud

Código *

Producto *

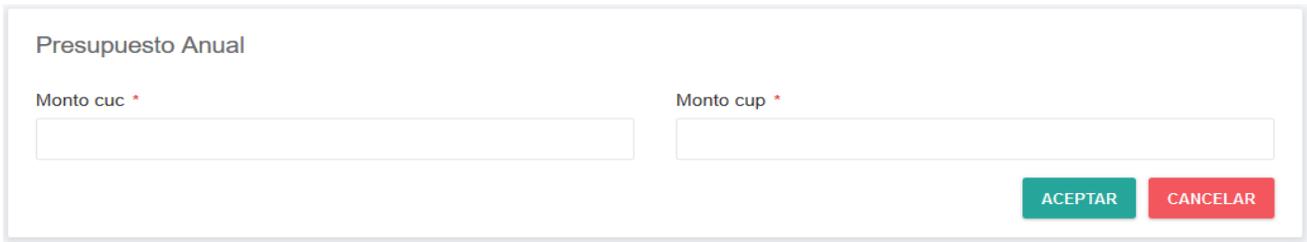
Cantidad *

Tabla 17 HU 10 Gestionar Presupuesto

Número: 10	Nombre del requisito: Gestionar Presupuesto
Programador: Eduardo Baratutes González	Iteración Asignada: 2
Prioridad: Media	Tiempo Estimado: -
Riesgo en Desarrollo: -	Tiempo Real: -
<p>Descripción:</p> <p>Deben recogerse los siguientes datos:</p> <ul style="list-style-type: none"> • Presupuesto anual en CUC (Valor numérico de carácter obligatorio) • Presupuesto anual en CUP (Valor numérico de carácter obligatorio) 	
Observaciones:	

Esta funcionalidad tiene como objetivo la implementación de la parte de la aplicación encargada del registro del presupuesto asignado a la oficina para poder hacer las compras de los materiales para la confección de las credenciales.

Prototipo de interfaz:



Presupuesto Anual

Monto cuc *

Monto cup *

ACEPTAR CANCELAR

Tabla 18 HU 11 Generar Reporte

Número: 11	Nombre del requisito: Generar Reporte.
Programador: Eduardo Baratutes González	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: -
Riesgo en Desarrollo: -	Tiempo Real: -
<p>Descripción: Permitirá tener un historial de todo lo que se ha realizado y debe ser capaz de exportar en Excel las demandas realizadas, los gastos de materiales utilizados en la confección de los credenciales.</p>	

<p>Observaciones:</p> <p>Tiene como objetivo la implementación de los reportes que se deben generar en la aplicación con los datos recogidos en el servicio de confección de credenciales.</p>
<p>Prototipo de interfaz:</p>

Tabla 19 HU 12 Graficar comportamiento de los gastos de materiales

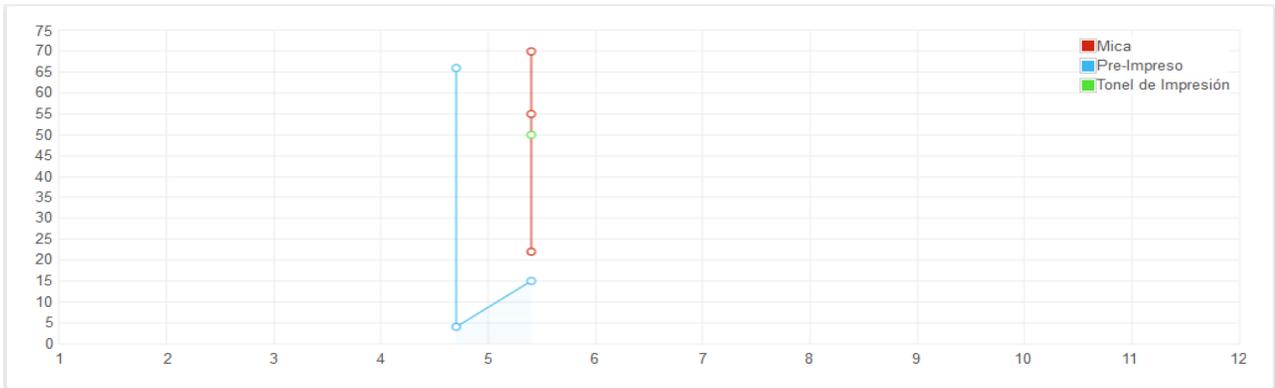
Número: 12	Nombre del requisito: Graficar comportamiento de los gastos de materiales.
Programador: Eduardo Baratutes González	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: -
Riesgo en Desarrollo: -	Tiempo Real: -

Descripción: El modelo tendrá los siguientes datos:

- Datos de los gastos de materiales utilizados en reparación y confección extra
- Los datos se observarán por meses

Observaciones: Se podrá visualizar en un gráfico los gastos de materiales para después poder saber cómo ha sido su comportamiento histórico.

Prototipo de interfaz:



Pruebas Funcionales

Tabla 20 Variables para los casos de prueba funcional Registrar Material

Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
------------------	---------------	------------	-------------	------------------------------

1	Nombre del material	Campo de solo letras	No	Se inserta el Nombre del material	NM
2	Unidad de medida	Campo seleccionable	No	Se selecciona el tipo de unidad de medida	UM
3	Precio	Campo de solo números	No	Se inserta el precio del material	PM
4	Total	Campo automático	No	El valor por defecto es 0.	TM

Tabla 21 Caso de prueba funcional Registrar Material

Descripción General							
Permitirá registrar un material.							
Condiciones de ejecución							
Para registrar un nuevo material el usuario debe estar logueado en el sistema y poseer los permisos necesarios, además, debe estar definido previamente una unidad de medida en el sistema.							
Escenario	Descripción	NM	UM	PM	TM	R/ del sistema	Flujo central
ESC 1.1 Registrar material.	El usuario debe insertar un material.	Micas	u	10.0	No	El sistema debe registrar un material. Una vez registrado el material el sistema debe mostrar un mensaje de éxito así como el listado de los materiales incluyendo el insertado.	En el menú a la izquierda se sigue la siguiente ruta: "Almacén –Materiales de la Oficina". Aparecerá un listado con todos los materiales registrados por el usuario. El usuario selecciona la opción insertar que se encuentra en la parte superior derecha y le aparecerá un formulario con un campo Nombre del material

							<p>donde debe insertar el nombre del material, un campo seleccionable con las unidades de medida y un campo precio. Cuando se inserte el material debe aparecer una lista con todos los materiales que tiene la oficina y el total que se tiene de cada uno en la misma.</p>
--	--	--	--	--	--	--	--

Tabla 22 Caso de prueba funcional Modificar Material

Descripción general							
Permitirá modificar un material							
Condiciones de ejecución							
Para modificar un material el usuario debe estar logueado en el sistema y poseer los permisos necesarios, además, debe ser insertado previamente un material en el sistema por el usuario.							
Escenario	Descripción	NM	UM	PM	TM	R/ del sistema	Flujo central

ESC 1.2 Modificar Material.	El usuario debe seleccionar el botón de editar.	No	No	10.0	200.0	El sistema debe mostrar un formulario con el precio. Una vez modificado el material el sistema debe mostrar un mensaje de éxito así como el listado de los materiales incluyendo la modificación.	"Almacén –Materiales de la Oficina". Aparecerá un listado con todos los materiales registrados por el usuario y una columna acciones en la parte derecha de la lista, son las acciones a ejecutar por el mismo. El usuario selecciona la opción editar y saldrá un formulario con un campo precio y otro campo total.
-----------------------------------	---	----	----	------	-------	---	---

Tabla 23 Variables para los casos de prueba funcional Entrada Material

	Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
1	Tipo de material	Campo seleccionable	No	Se selecciona el tipo de material	TM
2	Cantidad	Campo solo números	No	Se inserta la cantidad de material	CM
3	Fecha de entrada	Campo automático	No	La fecha se genera sola por la de la PC	FE
4	Fecha de expiración	Campo con formato fecha	No	Se inserta el la fecha de expiración	FV

Tabla 24 Caso de prueba funcional Insertar Entrada Material

Descripción general							
Permitirá entrar material							
Condiciones de ejecución							
Para entrar material el usuario debe estar logueado en el sistema y poseer los permisos necesarios, además, debe ser insertado previamente un material en el sistema.							
Escenario	Descripción	TM	CM	FE	FV	R/ del sistema	Flujo central
ESC 1.1 Entrar material.	El usuario debe seleccionar el campo del formulario e insertar las cantidades de materiales que se gastaron extra.	Micas	100	8 de mayo de 2018 (Campo automático)	10/10/20	El sistema debe insertar la cantidad de material. Una vez realizada la entrada de material el sistema debe mostrar un mensaje de éxito así como el listado de las entradas incluyendo la nueva.	“Almacén –Registro de Entrada de Materiales”. Aparecerá un listado con todas las entradas registradas por el usuario. El usuario selecciona la opción insertar que se encuentra en la parte superior derecha y le aparecerá un formulario con un campo seleccionable Tipo de Material donde seleccionar el material al cual se le dará entrada, un campo Cantidad para insertar la cantidad de

							<p>materiales, un campo Fecha de Expiración para insertar cual es la fecha en que el material expira, la Fecha de Entrada se genera solo por la fecha que tenga la PC.</p>
--	--	--	--	--	--	--	--

Tabla 25 Caso de prueba funcional Eliminar Entrada Material

Descripción general							
Permitirá eliminar la entrada de material							
Condiciones de ejecución							
Para eliminar la entrada de material el usuario debe estar logueado en el sistema y poseer los permisos necesarios, además, debe ser insertada previamente una entrada en el sistema y debe existir definido anteriormente el material.							
Escenario	Descripción	TM	CM	FE	FV	R/ del sistema	Flujo central
ESC 1.2 Eliminar entrada de material.	El usuario debe seleccionar el botón de eliminar.	No	No	No	No	El sistema debe eliminar un registro de entrada de material.	<p>“Almacén –Registro de Entrada de Materiales”.</p> <p>Aparecerá un listado con todas las entradas registradas por el usuario y una columna acciones en la parte derecha de la lista, son las acciones a ejecutar por el mismo. El usuario selecciona la opción eliminar.</p>

Tabla 26 Variables para los casos de prueba funcional Salida Material

	Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
1	Tipo de material	Campo seleccionable	No	Se selecciona el tipo de material	TM
2	Gasto en Confección	Campo solo números	No	Se inserta la cantidad de material	GC
3	Gasto en Reparación	Campo automático	No	Se inserta la cantidad de material	GR
4	Fecha de Salida	Campo automático	No	La fecha se genera sola por la de la PC	FS

Tabla 27 Caso de prueba funcional Salida Material

Descripción general							
Permitirá darle salida al material							
Condiciones de ejecución							
Para dar salida al material el usuario debe estar logueado en el sistema y poseer los permisos necesarios, además, debe ser insertada previamente una salida en el sistema y debe existir definido anteriormente el material.							
Escenario	Descripción	TM	GC	GR	FS	R/ del sistema	Flujo central
ESC 1.1 Salida de material.	El usuario debe seleccionar el campo del formulario e	Micas	21.0	12.0	8 de mayo de 2018 (Campo automático)	El sistema debe insertar la salida de material definida. Una	“Almacén y Reparación”. Aparecerá un listado con todas las salidas

	insertar los demás datos.					vez realizada la salida de material el sistema debe mostrar un mensaje de éxito así como el listado de las salidas incluyendo la nueva.	registradas por el usuario. El usuario selecciona la opción insertar que se encuentra en la parte superior derecha y le aparecerá un formulario con un campo seleccionable Tipo de Material donde debe seleccionar el material al cual se le dará salida, un campo Cantidad Gastado en Confección para insertar la cantidad de materiales gastados en confección extra, un campo Cantidad Gastado en Reparación para insertar la cantidad de materiales gastados en reparación extra, un campo Fecha de Salida que se genera solo por la fecha que tenga la PC.
--	---------------------------	--	--	--	--	---	---

Tabla 28 Caso de prueba funcional Eliminar Salida Material

Descripción general
Permitirá eliminar la salida de material

Condiciones de ejecución							
Para eliminar la salida de material el usuario debe estar logueado en el sistema y poseer los permisos necesarios, además, debe ser insertada previamente una salida en el sistema y debe existir definido anteriormente el material.							
Escenario	Descripción	TM	CM	FE	FV	R/ del sistema	Flujo central
ESC 1.2 Eliminar salida de material.	El usuario debe seleccionar el botón de eliminar.	No	No	No	No	El sistema debe eliminar un registro de salida de material.	“Almacén –Registro de Entrada de Materiales”. Aparecerá un listado con todas las salidas registradas por el usuario y una columna acciones en la parte derecha de la lista, son las acciones a ejecutar por el mismo. El usuario selecciona la opción eliminar.

Tabla 29 Variables para los casos de prueba funcional Gestionar Presupuesto

	Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
1	Monto en CUC	Campo seleccionable	No	Se selecciona el tipo de material	MC
2	Monto en CUP	Campo solo números	No	Se inserta la cantidad de material	MP

Tabla 30 Caso de prueba funcional Insertar Presupuesto

Descripción general
Permitirá insertar presupuesto
Condiciones de ejecución

Para insertar el presupuesto el usuario debe estar logueado en el sistema y poseer los permisos necesarios.

Escenario	Descripción	MC	MP	R/ del sistema	Flujo central
1.1 Insertar Presupuesto	El usuario debe insertar los montos en los campos del formulario.	100.00	150.00	El sistema debe insertar el presupuesto. Una vez insertado el presupuesto el sistema debe mostrar un mensaje de éxito así como el listado de los presupuestos incluyendo el insertado.	<p>“Presupuesto –Insertar Presupuesto”.</p> <p>En el menú a la izquierda se sigue la siguiente ruta: Aparecerá un formulario con un campo para insertar el monto en cuc y otro para el monto en cup. Una vez realizada la inserción debe mostrarse una lista con los montos insertados.</p>

Tabla 31 Caso de prueba funcional Modificar Presupuesto

Descripción general					
Permitirá modificar presupuesto					
Condiciones de ejecución					
Para modificar el presupuesto el usuario debe estar logueado en el sistema y poseer los permisos necesarios.					
Escenario	Descripción	MC	MP	R/ del sistema	Flujo central
1.2 Modificar Presupuesto	El usuario debe seleccionar el botón de editar.	102.00	152.00	El sistema debe modificar el presupuesto. Una vez modificado el presupuesto el sistema debe mostrar un mensaje de éxito así	<p>“Presupuesto –Visualizar Presupuesto”.</p> <p>En el menú a la izquierda se sigue la siguiente ruta: Aparecerá un listado con el presupuesto creado por el</p>

				como el listado de los presupuestos incluyendo el modificado.	usuario y una columna acciones en la parte derecha de la lista, son las acciones a ejecutar por el mismo. El usuario selecciona la opción editar y le aparecerá un formulario con los campos montos en cuc y monto en cup.
--	--	--	--	---	--

Tabla 32 Caso de prueba funcional Eliminar Presupuesto

Descripción general					
Permitirá eliminar presupuesto					
Condiciones de ejecución					
Para eliminar el presupuesto el usuario debe estar logueado en el sistema y poseer los permisos necesarios.					
Escenario	Descripción	MC	MP	R/ del sistema	Flujo central
1.3 Eliminar Presupuesto	El usuario debe seleccionar botón de eliminar.	No	No	El sistema debe eliminar el presupuesto.	<p>“Presupuesto –Visualizar Presupuesto”.</p> <p>En el menú a la izquierda se sigue la siguiente ruta: Aparecerá un listado con el presupuesto creado por el usuario y una columna acciones en la parte derecha de la lista, son las acciones a ejecutar por el mismo. El usuario selecciona la opción eliminar.</p>

Tabla 33 Variables para los casos de prueba funcional Insertar Producto

Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
------------------	---------------	------------	-------------	------------------------------

1	Tipo de Producto	Campo de solo letras	No	Se inserta el tipo de producto	TP
---	------------------	----------------------	----	--------------------------------	----

Tabla 34 Caso de prueba funcional Insertar Producto

Descripción General				
Permitirá insertar un producto.				
Condiciones de ejecución				
Para insertar un nuevo producto el usuario debe estar logueado en el sistema y poseer los permisos necesarios.				
Escenario	Descripción	TP	R/ del sistema	Flujo central
ESC 1.1 Insertar Producto.	El usuario debe insertar producto.	Solapín	El sistema debe insertar un producto. Una vez insertado el producto el sistema debe mostrar un mensaje de éxito así como el listado de los productos incluyendo el insertado.	En el menú a la izquierda se sigue la siguiente ruta: "Nomencladores –Productos". Aparecerá un listado con todos los productos insertados por el usuario y una columna acciones en la parte derecha de la lista, las acciones a ejecutar. El usuario selecciona la opción insertar que se encuentra en la parte superior derecha y le aparecerá un formulario con un campo Nombre del Producto donde debe insertar el nombre del producto. Cuando se inserte el producto debe aparecer una lista con todos los productos existentes.

Tabla 35 Caso de prueba funcional Modificar Producto

Descripción General

Permitirá modificar un producto.				
Condiciones de ejecución				
Para modificar un producto el usuario debe estar logueado en el sistema y poseer los permisos necesarios.				
Escenario	Descripción	TP	R/ del sistema	Flujo central
ESC 1.2 Modificar Producto.	El usuario debe seleccionar el botón de editar.	Solapínn	El sistema debe mostrar un formulario con el producto. Una vez modificado el producto el sistema debe mostrar un mensaje de éxito así como el listado de los productos incluyendo la modificación.	En el menú a la izquierda se sigue la siguiente ruta: "Nomencladores –Productos". Aparecerá un listado con todos los productos insertados por el usuario y una columna acciones en la parte derecha de la lista, son las acciones a ejecutar por el mismo. El usuario selecciona el botón de editar y le aparecerá un formulario con un campo Nombre del Producto donde debe insertar el nombre del producto. Cuando se modifique el producto debe aparecer en una lista con todos los productos existentes incluyendo el modificado.

Tabla 36 Variables para los casos de prueba funcional Insertar Unidad de Medidas

	Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
1	Unidad de Medida	Campo de solo letras	No	Se inserta la unidad	UM

Tabla 37 Caso de prueba funcional Insertar Unidad de Medida

Descripción General				
Permitirá insertar una unidad de medida.				
Condiciones de ejecución				
Para insertar una nueva unidad el usuario debe estar logueado en el sistema y poseer los permisos necesarios.				
Escenario	Descripción	TP	R/ del sistema	Flujo central
ESC 1.1 Insertar Unidad de Medida.	El usuario debe insertar la unidad.	u	El sistema debe insertar una unidad. Una vez insertada la unidad el sistema debe mostrar un mensaje de éxito así como el listado de las unidades incluyendo la insertada.	En el menú a la izquierda se sigue la siguiente ruta: "Nomencladores –Unidades de Medidas". Aparecerá un listado con todas las unidades insertadas por el usuario. El usuario selecciona la opción insertar que se encuentra en la parte superior derecha y le aparecerá un formulario con un campo Unidad de Medida donde debe insertar la unidad de un material. Cuando se inserte la unidad debe aparecer una lista con todas las unidades existentes.

Tabla 38 Caso de prueba funcional Modificar Unidad de Medida

Descripción General				
Permitirá modificar una unidad de medida.				
Condiciones de ejecución				
Para modificar una unidad el usuario debe estar logueado en el sistema y poseer los permisos necesarios.				

Escenario	Descripción	TP	R/ del sistema	Flujo central
ESC 1.2 Modificar Unidad de Medida.	El usuario debe seleccionar el botón de editar.	%	El sistema debe mostrar un formulario con la unidad. Una vez modificada la unidad el sistema debe mostrar un mensaje de éxito así como el listado de las unidades incluyendo la modificada.	En el menú a la izquierda se sigue la siguiente ruta: "Nomencladores –Unidades de Medidas". Aparecerá un listado con todas las unidades insertadas por el usuario y una columna acciones en la parte derecha de la lista, son las acciones a ejecutar por el mismo. El usuario selecciona el botón de editar y le aparecerá un formulario con un campo Unidad de Medida donde debe modificar la unidad de un material. Cuando se modifique la unidad debe aparecer una lista con todas las unidades existentes incluyendo la modificada.

Tabla 39 Variables para los casos de prueba funcional Insertar Norma de Consumo

	Nombre del campo	Clasificación	Valor nulo	Descripción	Alias para el caso de prueba
1	Material	Campo seleccionable	No	Se selecciona el tipo de material	MA
2	Producto	Campo seleccionable	No	Se selecciona el producto	PR
3	Cantidad de Material	Campo solo números	No	Se inserta la cantidad de material	CM

Tabla 40 Caso de prueba funcional Insertar Norma de Consumo

Descripción General						
Permitirá insertar una norma de consumo.						
Condiciones de ejecución						
Para insertar una norma el usuario debe estar logueado en el sistema y poseer los permisos necesarios, además, debe estar definido en el sistema un tipo de material y un producto.						
Escenario	Descripción	MA	PR	CM	R/ del sistema	Flujo central
ESC 1.1 Insertar Norma de Consumo.	El usuario debe seleccionar los campos de tipo de material y producto e insertar la cantidad de materiales que se necesitan para hacer un producto.	Micas	Solapín	2	El sistema debe insertar la norma. Una vez insertada la norma el sistema debe mostrar un mensaje de éxito así como el listado de las normas incluyendo la insertada.	En el menú a la izquierda se sigue la siguiente ruta: "Nomencladores –Norma de Consumo". Aparecerá un listado con todas las normas insertadas por el usuario y una columna acciones en la parte derecha de la lista, las acciones a ejecutar. El usuario selecciona la opción insertar que se encuentra en la parte superior derecha y le aparecerá un formulario con los campos seleccionables de tipo de material y producto y debe insertar la cantidad de material. Cuando se inserte la norma debe aparecer una lista con todas las normas existentes.

Tabla 41 Caso de prueba funcional Modificar Norma de Consumo

Descripción General						
Permitirá modificar una norma de consumo.						
Condiciones de ejecución						
Para modificar una norma el usuario debe estar logueado en el sistema y poseer los permisos necesarios, además, debe estar definido en el sistema un tipo de material y un producto.						
Escenario	Descripción	MA	PR	CM	R/ del sistema	Flujo central
ESC 1.2 Modificar Norma de Consumo.	El usuario debe seleccionar el botón de editar.	No	No	2.0	El sistema debe mostrar un formulario con la cantidad de materiales. Una vez modificada la norma el sistema debe mostrar un mensaje de éxito así como el listado de las normas incluyendo la modificada.	En el menú a la izquierda se sigue la siguiente ruta: "Nomencladores –Norma de Consumo". Aparecerá un listado con todas las normas insertadas por el usuario y una columna acciones en la parte derecha de la lista, son las acciones a ejecutar por el mismo. El usuario selecciona el botón de editar y le aparecerá un formulario con un campo Cantidad de Material donde debe modificar la cantidad de material. Cuando se modifique la norma debe aparecer una lista con todas las normas existentes incluyendo la modificada.