



**Facultad de Ciencias Técnicas
Departamento de Informática**

Aplicación web para la gestión de solicitudes de almacén en la Dirección Provincial de Bufetes Colectivo Matanzas

Trabajo de Diploma en Opción al Título de Ingeniero Informático

Autor: John de Jesús Lara Sobrino

Tutora: Msc. Liz Pérez Martínez

Matanzas, 2019

*“No hace falta alas para hacer un sueño,
hasta con las manos, hasta con el pecho,
hasta con las piernas y con el empeño”*

Silvio Rodríguez

Este trabajo es dedicado a mi madre, hermano y abuela. Gracias por todo.

Agradecimientos

No alcanzan las palabras para poder demostrar cuanto agradecimiento alberga en mi. Solo se que sin el apoyo incondicional de mi familia, mi tutora Liz Pérez, mis amigos, mis compañeros de trabajo no hubiera sido posible este resultado.

Gracias a todos por acompañarme en esta travecia y a esos profesores que me acompañaron con sus conocimientos, gracias.

Declaración de Autoría

Yo, John de Jesús Lara Sobrino, declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas, a la Dirección Provincial de Bufetes Colectivos de Matanzas, a que hagan el uso que estimen pertinente de él.

Y para que así conste, firmo la presente a los 18 días del mes de junio del 2019.



Firma del Autor

Firma del Tutores

Opinión del Tutor

DATOS PERSONALES DEL TUTOR

Nombre y apellidos: Liz Pérez Martínez.

Centro de trabajo: Universidad de Matanzas.

Organismo a que pertenece: Ministerio de Educación Superior – MES.

Cargo que ocupa: Vicedecana Facultad de Ciencias Técnicas.

Especialidad de la que es graduado: Ingeniería Informática. Universidad de Matanzas, 2012.

Categoría docente o investigativa: Asistente.

Grado científico: Master en Ciencias, Universidad de Matanzas, 2015.

DATOS DE LA TESIS Y EL DIPLOMANTE

Nombre y apellidos: John de Jesús Lara Sobrino.

Centro de estudio: Universidad de Matanzas sede “Camilo Cienfuegos”.

Título de la Tesis: Aplicación web para la gestión de solicitudes de almacén en la Dirección Provincial de Bufetes Colectivo Matanzas.

OPINION SOBRE EL TRABAJO

La tesis presentada posee gran actualidad, pues intenta resolver un problema real presente en las empresas de nuestro país, y además contribuye a la informatización de nuestra sociedad.

El tutor de este trabajo de diploma considera que, durante su ejecución, la estudiante mostró las cualidades que a continuación se detallan:

Independencia y capacidad de investigación. Supo adecuar en su trabajo tecnologías actuales y de demostrada valía en el campo del desarrollo web a nivel mundial. Fueron jornadas adentrándose en temas complejos y nuevos pues se apartan en gran medida de los contenidos recibidos durante su formación, y logró captar con rapidez y profesionalidad el conocimiento necesario para enfrentar el problema planteado. Aportó soluciones importantes que demuestran su madurez como investigador. Fue consecuente con los aspectos tanto metodológicos como de la investigación científica propiamente. Esto le permitió una feliz culminación de la solución desarrollada, de la documentación y de las pruebas realizadas.

En el trabajo se aprecia rigor, manifestado desde el tratamiento de los conceptos estudiados y referenciados en la bibliografía, hasta las conclusiones, lo que ha contribuido a la correcta solución de los problemas encontrados.

Una gran cantidad de clases y métodos, un producto bien concebido y validaciones correctamente realizadas para culminar su investigación, unido a una excelente planificación de tiempo y recursos, dieron una gran calidad al trabajo obtenido.

También fueron horas de revisión, discusión y consenso en las que demostró notables cualidades para la investigación. El trabajo que hoy presenta y que sintetiza un periodo de aprendizaje no solo académico. Un reto asumido y vencido.

Es válido señalar que la investigación se realizó en el contexto de la integración con la Dirección Provincial de Bufetes Colectivo en Matanzas. Que representó horas de esfuerzo y adaptación al sistema de trabajo de la entidad por parte del estudiante, y que derivó en la obtención de un producto de software al nivel de las exigencias y expectativas de la entidad y de la Universidad respectivamente.

Por todo lo anteriormente señalado, considero que el estudiante John de Jesús Lara Sobrino reúne los requisitos para el título de Ingeniero Informático y espero le sea otorgada la mejor calificación de este Tribunal. Espero, más temprano que tarde saber que en algún momento estará defendiendo su título de Máster y que su labor investigativa no se detenga hasta alcanzar la cima.



MSc. Liz Pérez Martínez

Dpto. Informática
Universidad de Matanzas
Junio/2019

Opinión del Cliente



DIRECCIÓN PROVINCIAL MATANZAS

Matanzas, 17 de junio del 2019

"Año 61 de la Revolución"

A quien pueda interesar:

Ref: Desarrollo de la Aplicación web para la gestión de solicitudes de almacén en la Dirección Provincial de Bufetes Colectivo Matanzas (DPBC)

Considero que sistema de gestión de solicitudes de almacén en la Dirección Provincial de Bufetes Colectivo Matanzas deviene como una poderosa herramienta que satisface las necesidades implantadas por la organización. Esta herramienta simplifica las operaciones necesaria para la conformación de las solicitudes de almacén y de igual forma las distribuciones correctas de los materiales necesarios para nuestro quehacer diario. Ahorra el tiempo en la entrega de materiales que de forma manual se dilataba dicho proceso. Esta herramienta es la antesala del proceso de informatización del área de Aseguramiento de la Dirección Provincial de Bufetes Colectivos Matanzas.

Atentamente,

MSc. Sergio Jorge Pagés Valdés
Director Provincial de Bufetes
Colectivos Matanzas



Dirección Provincial Matanzas
Calle Contreras N. 26008, cv Jovellanos y Matanzas, Ciudad de Matanzas
Teléf. (+53) 243890 correos: john.lara@mtz.onbc.cu

Resumen

La aplicación Web fue realizada en la Dirección Provincial de Bufetes Colectivos de Matanzas. La propuesta se fundamenta en diseñar un sistema de solicitudes de almacén, que permita la flexibilidad y agilidad del proceso de solicitud de material, teniendo en cuenta los parámetros normados de la organización. Entre las facilidades que brinda el sistema se encuentran la generación de reportes y estadísticas útiles para tomar decisiones posteriores a la hora de desarrollar el presupuesto por partidas; así como la gestión del estado de las solicitudes y el período de distribución. Se automatizó el proceso de solicitudes de almacén que permite, conocer la existencia actual del almacén mediante conexión al Sistema Versat Sarasola y el cálculo de asignación automática. El sistema posibilita agilizar el proceso de entrega que se realizan actualmente de forma directa y manual en la entidad. Se empleó la metodología de desarrollo de software Programación Extrema (XP), el lenguaje de programación C# y como gestor de base de datos MySQL. Se obtuvo un sistema funcional, el cual se le realizaron una serie de pruebas que permitieron constatar su funcionalidad y verificar que cumpliera con las necesidades y exigencias del cliente.

Abstract

The Web application was made in the Provincial Office of Collective Law Offices of Matanzas. The proposal is based on designing a system of warehouse requests, which allows the flexibility and agility of the material request process, taking into account the norm parameters of the organization. Among the facilities provided by the system are the generation of reports and useful statistics to make subsequent decisions when developing the budget by items; as well as the management of the status of the requests and the distribution period. The process of warehouse requests was automated, allowing to know the current existence of the warehouse through connection to the Versat Sarasola System and the calculation of automatic allocation. The system makes it possible to streamline the delivery process that is currently performed directly and manually in the entity. The software development methodology Extreme Programming (XP), the programming language C # and MySql database manager were used. A functional system was obtained, which was made a series of tests that allowed to verify its functionality and verify that it met the needs and requirements of the client

Índice de Contenido

Contenido 2

| | |
|---|----|
| Introducción | 1 |
| Capitulo I Marco Teórico y Tendencias Tecnológicas | 5 |
| 1.1. Introducción | 5 |
| 1.2. Antecedentes del problema de investigación..... | 5 |
| 1.2.3 Análisis comparativo de los sistemas existentes con la propuesta de trabajo | 8 |
| 1.3. Descripción del proceso..... | 8 |
| 1.4. Metodología de Desarrollo de Software: XP | 9 |
| 1.5. Tendencias Tecnológicas | 11 |
| 1.5.1 Lenguaje del lado del cliente | 12 |
| 1.5.2 Lenguaje del lado del servidor | 14 |
| 1.5.3 Frameworks | 15 |
| 1.5.4 Entity Framework y Object Relational Mapping | 16 |
| 1.5.5 Sistemas de Gestión de Bases de Datos (SGBD) | 17 |
| 1.6. Conclusiones del Capítulo | 18 |
| Capitulo II Descripción de la Solución Propuesta | 19 |
| 2.1. Introducción | 19 |
| 2.2 Solución propuesta | 19 |
| 2.2.1. Equipo de trabajo o Roles..... | 19 |
| 2.2.2. Análisis de Requerimientos | 22 |
| 2.2.3. Requisitos no funcionales | 23 |
| 2.2.4. Plan de Iteración..... | 24 |
| 2.2.5 Historias de usuarios iniciales | 25 |
| 2.2.6 Descripción de las historias de usuario y tareas de programación | 28 |
| HU -1 Diseño y Creación de la Base de Datos | 28 |
| HU -2 Diseño interfaz de usuario | 29 |
| HU -4 Gestionar usuarios | 29 |
| HU-7 Gestionar Producto..... | 31 |
| 2.2.7 Estudio de factibilidad | 33 |

| | |
|---|----|
| 2.3 Conclusión del Capítulo..... | 35 |
| Capítulo 3. Resultados obtenidos y pruebas realizadas al software | 36 |
| 3.1 Introducción | 36 |
| 3.2.1 Pruebas de Aceptación..... | 36 |
| 3.2.2 Validación de la solución propuesta..... | 41 |
| 3.2.3 Pruebas de carga y estrés | 43 |
| 3.2.4 Pruebas de Seguridad | 45 |
| 3.3 Conclusiones del capítulo. | 46 |
| Conclusiones generales..... | 47 |
| Recomendaciones | 48 |
| Referencias Bibliográficas | 49 |
| Acrónimos..... | 51 |

Índice de Ilustraciones

| | |
|--|----|
| Ilustración 1.1 Proceso de Programación Extrema (Pressman, 2010) | 11 |
| Ilustración 3.2 Pruebas de carga y estrés realizadas | 44 |
| Ilustración 3.3 Pruebas de Seguridad Realizadas | 46 |

Índice de Tablas

| | |
|--|----|
| Tabla No.2.1: Integrantes del equipo de trabajo. Fuente: Elaboración propio | 20 |
| Tabla No.2.2: Usuarios y permisos del sistema. Fuente: Elaboración propia..... | 22 |
| Tabla No.2.3: Requisitos de Hardware. Fuente: Elaboración propia..... | 24 |
| Tabla No.2.4: Requisitos del Software. Fuente: Elaboración propia | 24 |
| Tabla No.2.5: Plan de Entrega. Fuente: Elaboración propia | 27 |
| Tabla No.2.6: HU -1 Diseño y Creación de la Base de Datos. Fuente: Elaboración propia | 28 |
| Tabla No.2.7: Tarea 1.1 Diseño de la BD. Fuente: Elaboración propia | 28 |
| Tabla No.2.8: Tarea 1.2 Creación de la BD.. Fuente: Elaboración propia | 29 |
| Tabla No.2.9: HU-2 Diseño interfaz de usuario Fuente: Elaboración propia | 29 |
| Tabla No.2.10: Tarea 2.1 Crear Interfaz Principal Fuente: Elaboración propia | 29 |
| Tabla No.2.11: HU-4 Gestionar Usuario Fuente: Elaboración propia | 30 |
| Tabla No.2.12: Tarea 4.1 Autenticación de usuario Fuente: Elaboración propia | 30 |
| Tabla No.2.13: Tarea 4.2 Agregar usuario Fuente: Elaboración propia | 30 |
| Tabla No.2.14: Tarea 4.3 Editar usuario Fuente: Elaboración propia | 31 |
| Tabla No.2.15: Tarea 4.4 Desactivar usuario Fuente: Elaboración propia..... | 31 |
| Tabla No.2.16: HU- Gestionar Usuarios Fuente: Elaboración propia | 32 |
| Tabla No.2.17: Tare 7.1- Agregar Producto Fuente: Elaboración propia..... | 32 |
| Tabla No.2.18: Tare 7.2- Editar Producto Fuente: Elaboración propia..... | 32 |
| Tabla No.2.19: Tare 7.3- Eliminar Producto Fuente: Elaboración propia | 33 |
| Tabla No.2.20: Tare 7.4- Asociar Producto Fuente: Elaboración propia | 33 |
| Tabla 3.1: PA1 Seguridad y Autenticación de usuario Fuente: Elaboración Propia..... | 38 |
| Tabla 3.2: PA2 Gestionar Producto Fuente: Elaboración Propia | 39 |
| Tabla 3.3: PA3 Gestionar Clasificador Fuente: Elaboración Propia | 40 |
| Tabla 3.4: PA4 Gestionar Unidad de Medida Fuente: Elaboración Propia | 41 |
| Tabla 3.4: Tiempo de obtención y procesamiento de información. Fuente: Elaboración Propia .. | 43 |

Introducción

El proceso de desarrollo de las Tecnologías de la Información y las Comunicaciones (TICs) ha transformado de forma objetiva nuestra sociedad. Cada día, la Informática adquiere más relevancia en la vida de las personas y en las empresas. El desarrollo de las TICs ha revolucionado el mundo dando paso a grandes adelantos en los distintos sectores de la actividad humana.

De igual forma la Organización Nacional de Bufetes Colectivo con su sede nombrada Dirección Provincial de Bufetes Colectivos de Matanzas (DPBC), es una organización autónoma, de carácter y patrimonio propio vinculada al sector jurídico de nuestro país, su objetivo fundamental es la prestación de servicios jurídicos tanto a personas naturales como jurídicas, para ello requiere de un aseguramiento material distribuido racionalmente según las normas reglamentarias establecida por dicha institución.

De ahí la importancia de contar con herramientas que ayuden en el análisis de información de negocios, con el propósito de marcar tendencias, señalar problemas y contribuir a la toma de decisiones. Un sistema informático constituye una herramienta que, destinada a un negocio específico, puede englobar las características anteriormente descritas.

En la DPBC, existe el proceso de entrega de materiales para el aseguramiento de cada bufete colectivo de la provincia, para ello se debe gestionar una solicitud de materiales al almacén teniendo en cuenta lo establecido en el Manual de Normas y Procedimientos, este proceso no se encuentra automatizado, lo que propicia que la gestión de la información no se haga de manera eficiente, ya que se genera gran cantidad de documentación y sea ineficiente a la hora de consultar datos históricos o estadísticos. Actualmente dicho proceso se realiza de forma manual, se reciben las solicitudes vía electrónica y en muchos de los casos deben ser rechazadas por no estar correctamente elaboradas.

Lo anteriormente planteado, describe la **situación problémica** de esta investigación, a raíz de la cual se tiene el siguiente problema **científico**: ¿Cómo agilizar el flujo de información referente al proceso de solicitudes de almacén en la Dirección Provincial de Bufetes Colectivos Matanzas?

Por lo que se formula la siguiente **hipótesis**: si se implementa una aplicación web entonces se agilizará el flujo de información referente al proceso de solicitudes de almacén en la Dirección Provincial de Bufetes Colectivos Matanzas.

El proceso de solicitud de almacén en la Dirección Provincial de Bufetes Colectivo constituye el **objeto de estudio** de esta investigación. Se define como **campo de acción** la automatización del proceso de solicitud de almacén en la Dirección Provincial de Bufetes Colectivos Matanzas.

Expuesto lo anterior se define como **objetivo general**: desarrollar una aplicación web para la gestión del flujo de información que agilice el proceso de solicitud de almacén en la Dirección Provincial de Bufetes Colectivos Matanzas. Para complementar el objetivo general se trazan los siguientes **objetivos específicos**:

1. Realizar un estudio de los antecedentes y tendencias actuales del proceso de solicitud de almacén en la Dirección Provincial de Bufetes Colectivos Matanzas.
2. Diseñar una aplicación web para la gestión del proceso de solicitud de almacén en la Dirección Provincial de Bufetes Colectivos Matanzas utilizando la metodología de desarrollo de software XP.
3. Implementar una aplicación web para la gestión solicitud de almacén en la Dirección Provincial de Bufetes Colectivos Matanzas.
4. Validar la aplicación web para la gestión de solicitud de almacén en la Dirección Provincial de Bufetes Colectivos Matanzas.

El **resultado esperado** de esta investigación es contar con una aplicación web que contribuya a la informatización de la organización, y permita a los trabajadores agilizar la solicitud de almacén, así como ahorrar tiempo, más confiabilidad en la interactividad entre los usuarios y generar reportes con registros específicos. El sistema será multiplataforma, optando por alternativas de código abierto al obtener una alternativa basada en la web, adaptada al contexto de la organización.

Durante la investigación se utilizaron diversos **métodos y técnicas de metodología de la investigación**, tales como:

-) Dentro de los métodos teóricos:

-) Método de análisis histórico – lógico: permitió estudiar la trayectoria y desarrollo del proceso de gestión de solicitud de almacén existente en la entidad objeto de estudio.
-) Método de análisis y síntesis: este se precisó durante la revisión bibliográfica y el análisis de los resultados, permitiendo descomponer lo complejo en sus partes y cualidades, la división del todo en sus múltiples relaciones para luego unir las partes analizadas, descubrir las relaciones y características generales entre ellas.
-) Método inductivo - deductivo: su uso fue necesario tanto en la revisión bibliográfica, como en el análisis de los resultados, permitiendo arribar a conclusiones que se infirieron a partir de propiedades y relaciones existentes entre los elementos que conforman el fenómeno objeto de estudio.
-) Como métodos empíricos, utilizados por medio de las siguientes técnicas:
 -) Observación: permitió entender el proceso de gestión de solicitudes en la entidad objeto de estudio.
 -) Entrevistas: fue útil en distintos momentos de la investigación; fundamentalmente al inicio, cuando se realizó el levantamiento de requisitos para efectuar una exploración preliminar del problema a investigar. Se entrevistaron a los funcionarios y responsables del proceso para dar cumplimiento a las tareas planteadas como vía de obtención de datos.
 -) Análisis documental: Se realiza una revisión de la documentación llevada en el proceso de gestión y del reglamento de trabajo por el cual se rige, así como las resoluciones más importantes acerca de este proceso.

Entre los **aportes** de la investigación se destacan:

-) el teórico-investigativo, al integrar los procedimientos tradicionales más utilizados por autores relacionados con el tema, a través de los diferentes artefactos de la metodología de desarrollo de software que permitió orientar metodológicamente la secuencia de acciones lógicas a desarrollar; y los elementos a tener en cuenta para la continuidad de la investigación,
-) el práctico, al desarrollar una herramienta automatizada que asista a la manipulación de la información referente a la gestión de solicitud de almacén.

-) el económico, al obtener una herramienta informática que permita disminuir los gastos que se incurren por conceptos de consumo de recursos, dígame suministros de oficina, tiempo, efectivos, entre otros.

Atendiendo a lo planteado anteriormente, la tesis queda estructurada en introducción, tres capítulos, conclusiones, recomendaciones y referencias bibliográficas, según sigue:

-) Una Introducción, donde se caracteriza la situación problemática y se fundamenta el problema científico a resolver.
-) Un primer capítulo donde se recoge el marco teórico referencial del tema y los principales conceptos que constituyen la base teórica de la investigación, así como el análisis de las principales tendencias tecnológicas y el estudio de los antecedentes que enmarcan la problemática planteada.
-) Un segundo capítulo donde se caracteriza el objeto de estudio, se describe el desarrollo e implementación del software a través de la metodología XP.
-) Un tercer capítulo donde se muestran las principales interfaces del prototipo inicial y se detallan los resultados de las pruebas aplicadas al software.
-) Un apartado de conclusiones donde se verifica el cumplimiento de los objetivos trazados al inicio de la investigación.
-) Las recomendaciones en la cual se plasman una serie de propuestas encaminadas a la continuidad de esta investigación.
-) Y las referencias de la bibliografía citada.

Capítulo I Marco Teórico y Tendencias Tecnológicas

1.1. Introducción

En este capítulo se aborda el marco teórico conceptual asociado al objeto de la investigación. Se analiza el flujo actual de los procesos a automatizar, con el objetivo de permitir una mejor orientación para la solución. Se realiza un estudio del estado del arte de la temática, con la intención de determinar los posibles antecedentes de la misma. Se describe la metodología de desarrollo de software, así como las herramientas y tecnologías que serán empleadas durante el desarrollo de la investigación.

1.2. Antecedentes del problema de investigación

Con el objetivo de investigar la existencia de aplicaciones informáticas que guardaran semejanza con la idea que se propone desarrollar en la investigación, se consultaron diversas fuentes de información del ámbito nacional e internacional.

1.2.1 Ámbito Internacional

DvSGA es un sistema de gestión de almacén (SGA) sobre la plataforma Microsoft Dynamics. Según (Dávila, 2008) este sistema permite realizar una gestión completa de los almacenes y cadena de suministros. Este SGA cubre la totalidad de las necesidades de gestión logística de una compañía, por ello está diseñado de forma modular, se puede incorporar diferentes módulos en función de las necesidades de la empresa, y es escalable. Gestiona desde un simple almacén hasta múltiples almacenes con diversidad de complejidades y todos los casos posibles de almacén:

-) Almacén como un saco: Se controla la cantidad almacenada dentro de ese almacén.
-) Almacén con ubicaciones manuales: Se controla la cantidad almacenada de forma manual.
-) Almacén con zonas y/o ubicaciones: Se controla la cantidad almacenada y su lugar de almacenamiento.

Easy WMS® Según (Smena, 2010) es un software de gestión de almacenes dirigido a empresas de cualquier envergadura y dedicadas a los más diversos sectores de actividad ya

que encaja en cualquier compañía: desde una pequeña empresa con un almacén convencional hasta el control logístico de un aeropuerto. Tiene como principal objetivo controlar, coordinar y gestionar todos los procesos que se desarrollan en un almacén. Los diferentes niveles de funcionalidad de Easy WMS® permiten adaptar el software a las necesidades concretas de cada instalación en función del grado de complejidad y automatización del almacén y puede adaptarse fácilmente al crecimiento de un negocio o a nuevos procesos y volúmenes gracias a su gran flexibilidad. El software ha sido creado y desarrollado íntegramente por Mecalux, una de las compañías punteras en el mercado de sistemas de almacenaje

Software de gestión de almacenes de SAP es una solución de la firma alemana SAP para la gestión de almacenes según (SAP, 2013), puede optimizar las actividades de almacén, incluido el procesamiento de entrada y de salida, la gestión de instalaciones y el almacenamiento, el inventario físico y el *cross-docking*¹ planificado y oportuno. Puede beneficiarse de las tecnologías de recopilación de datos como, por ejemplo, las de identificación por radiofrecuencia (RFID) y las de voz, así como de las nuevas herramientas de equilibrio de la carga de trabajo. Las soluciones SAP para la gestión de almacenes ofrecen soporte para el procesamiento de entrada y la confirmación de entradas mediante tecnología RFID, de modo la empresa puede recibir y procesar mercancías aprovisionadas externamente en sus almacenes con un solo escaneado de radiofrecuencia. Puede capturar información detallada y de resumen, incluidos los detalles de procesos de entregas de entrada. El procesamiento de salida puede aprovechar la tecnología RFID para gestionar la distribución y las actividades de comprobante de entrega.

1.2.2 Ámbito Nacional

El Sistema de Gestión de Almacenes (distraSGA) es una solución desarrollada por el Centro de Gestión Logística de la Empresa de Tecnologías de la Información para la Defensa (XETiD) que propone una solución adaptable y configurable a cualquier entorno y está desarrollada bajo los principios de soberanía tecnológica. Esta solución optimiza, controla el almacén, ahorrando tiempo y dinero, ya que disminuye el costo de las operaciones que se ejecutan en

¹ En Logística el *cross-docking* corresponde a un tipo de preparación de pedido (una de las funciones del almacén logístico) sin colocación de mercancía en stock (inventario), ni operación de picking (recolección).

los almacenes según (Rodríguez, 2018), además permite incrementar el desempeño del almacén. Para ello se dividió la información en diferentes áreas, denominadas módulos, con el fin de organizar los procesos del almacén, los cuales son:

-) Configuración y seguridad
-) Ubicación
-) Gestión de tareas
-) Etiquetado
-) Carga y paletizado

Algunos de los sistemas contables más utilizados en Cuba y cuenta con una gestión de almacén es:

Versat-Sarasola, abreviadamente VERSAT, es un paquete de gestión económico-financiera, que permite registrar, controlar y analizar los hechos económicos, que acontecen en una entidad, a partir de la información contenida en los documentos primarios. VERSAT cuenta con los módulos de Configuración del sistema, Contabilidad General y Costos, como elementos básicos e imprescindibles del paquete. Además, contiene los módulos adicionales de Finanzas, Inventario, Activos Fijos y Facturación. La interrelación entre estos subsistemas económicos, se logra a partir del mecanismo de contabilización implementado, donde todos los subsistemas, tributan comprobantes para la contabilidad, la cual los valida y los asienta en los libros (Casa Consultora DISAIC, 2010a). VERSAT es una valiosa herramienta para el control económico, contable y financiero, es un software integrado, orientado a la gestión empresarial y presupuestaria, soportado sobre una plataforma de trabajo “Cliente-Servidor” (Colectivo de autores del CNFRI, 2012).

Aplicación web para la Solicitud de Productos en la División EMAE Matanzas. (Torres Ojea, Ernesto 2008). Es una aplicación web que contribuye a la informatización de la empresa, y permite a los trabajadores agilizar la solicitud de productos de manera eficiente y fácil, así como ahorrar tiempo, más confiabilidad en la interactividad entre los usuarios y generar reportes con registros específicos de las características de dicha institución. El sistema es

multiplataforma, optando por alternativas de código abierto al obtener una alternativa basada en la web, adaptada al contexto de la empresa.

1.2.3 Análisis comparativo de los sistemas existentes con la propuesta de trabajo

Los sistemas hallados fueron sistemas para gestión de almacenes (SGA) o sistemas contables que tienen módulos de desktop de almacén para el tratamiento de los inventarios, lo cual permite hacer búsquedas de productos, entrar y rebajar productos del inventario, pero ninguno de los sistemas permite la gestión de solicitudes de productos al almacén. Los mismo son de carácter propietario, solo existe mayor semejanza la aplicación web para la solicitud de productos en la División EMAE, pero la misma no cumple con los resultados esperados de la solución, la cual se debe adaptar a la necesidad del cliente y procedimientos con lo que labora la Dirección Provincial de Bufetes Colectivo de Matanzas.

1.3. Descripción del proceso

La Organización Nacional de Bufetes Colectivos (ONBC) surge como una organización autónoma para que todo ciudadano natural o no tuvieran una persona capacitada para representarlos en todos los ámbitos jurídicos y encontraran confianza en quien los representaba. A raíz que fue creciendo la organización se situó direcciones en los municipios cabeceras, así como bufetes en resto de la provincia por la demanda creciente de la población. Está organización surge como idea del Comandante en Jefe de la Revolución Cubana con sus normas y procedimientos los cuales la hacen única en el país. La dirección provincial de Matanzas se encuentra ubicada en Contreras entre Jovellanos y Matanzas, municipio Matanzas, provincia Matanzas, su actividad fundamental es la contratación de personas jurídicas para representar en procesos litigiosos o no.

Para el aseguramiento de estos servicios según lo establecido en el Acuerdo 1313 del año 2013, se aprueba la norma de consumo material en la cual se toma como regla a la hora de elaborar la solicitud de almacén de cada una de las dependencias de la institución, la misma establece que los periodos de distribución serán trimestralmente y se despachara según lo

establecido según norma y teniendo en cuenta la disponibilidad financiera o existencia física en almacén de cada producto.

El proceso se inicia cuando el personal autorizado en cada bufete elabora una solicitud genérica. La misma cuenta con las necesidades requeridas y la cantidad correspondiente según lo establecido en la norma. Posteriormente es recibida en el departamento de aseguramiento por el balancista distribuidor, el cual es el personal encargado de conformar las solicitudes, los códigos de los productos oficiales en el sistema contable. De igual forma le corresponde aprobar o modificar la cantidad según la existencia y las demandas que puedan existir de cada producto. Todo con el objetivo de lograr una equidad en la distribución en cada una de las entidades.

Después de ser validada y aprobada, por el personal encargado el proceso continúa hasta la intervención del almacenero personal encargado de colocar el despacho y posterior entrega al solicitante.

Entre los principales procesos que serán objeto de automatización tenemos:

-) Administrar la seguridad y el control de usuarios y su acceso a datos.
-) Gestión de las solicitudes para su funcionamiento en tiempo real de la información.
-) Generar diferentes reportes basados en el nivel de consumo y gasto por elementos, para la estimación de las partidas de gastos correspondiente a cada uno de los materiales despachados.

1.4. Metodología de Desarrollo de Software: XP

Según (Borja López, 2017) las metodologías de desarrollo describen los pasos a seguir para desarrollar un producto de software, definen qué es lo que se debe hacer, cómo y quién debe hacerlo. Precisan el orden de las tareas, los artefactos que van a ser generados, quiénes son los responsables y cómo deben hacerse cada una de estas tareas durante el ciclo de desarrollo del proyecto.

Para la elección de la metodología a utilizar se deben tener en cuenta las características que reúnen el equipo, las necesidades específicas de la situación y las prioridades demandadas

por el cliente. Deben analizarse además dos factores fundamentales: el tipo de proyecto que se desea desarrollar y el tiempo que se dispone para desarrollar el mismo.

Podemos decir que se selecciona dicha metodología para el aplicación, pues XP es el enfoque más utilizado de desarrollo de software ágil según (Pressman, 2010). Es una metodología ágil utilizada para proyectos de corto plazo y propensos a cambios, equipos reducidos, cuya particularidad es tener como parte de éste al usuario final.

En todo trabajo realizado como parte de XP debe existir (Beck K., 2004):

-) **Comunicación:** Entre los propios desarrolladores y entre estos y los clientes.
-) **Simplicidad:** Crear un diseño sencillo que se implemente con facilidad en forma de código.
-) **Retroalimentación:** Se obtiene del software implementado, del cliente y otros miembros del equipo de software.
-) **Disciplina:** Por parte del equipo de desarrollo al ser muy probable que los requerimientos cambien, lo que traerá repeticiones sustanciales del diseño y del código implementado.
-) El proceso de XP consta de cuatro actividades estructurales: **Planeación, Diseño, Codificación y Pruebas**, como muestra la Figura 1.

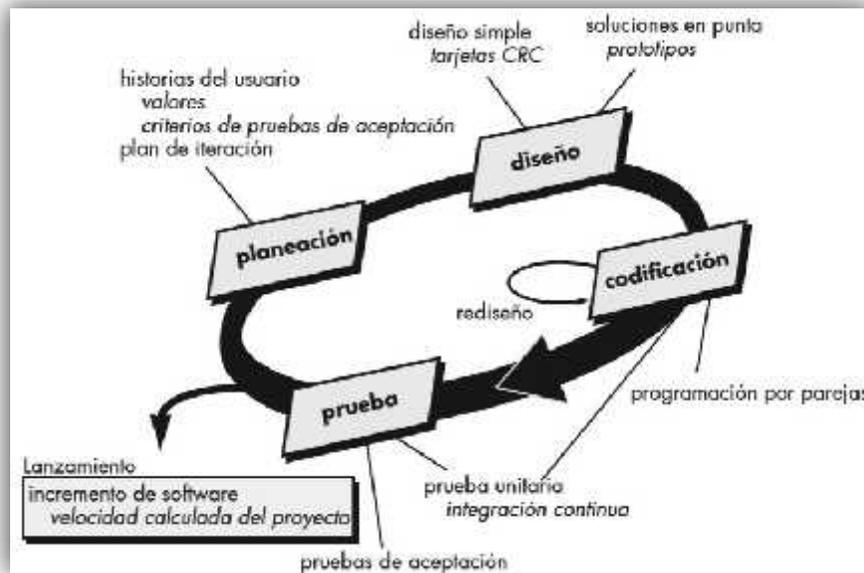


Ilustración 1.1 Proceso de Programación Extrema (Pressman, 2010)

-) **Planeación:** Entender el contexto del negocio y obtener los requerimientos necesarios. Se crean las primeras Historias de Usuario (HU) y el plan de entrega.
-) **Diseño:** Es sencillo, lo cual se prefiere sobre una representación más compleja. Ocurre tanto antes como después de que comienza la codificación.
-) **Codificación:** Se realizan pruebas unitarias a cada HU antes de iniciar la codificación, después de estas, el desarrollador está mejor capacitado para centrarse en lo que debe implementarse para pasar la prueba.
-) **Pruebas:** Además de las pruebas unitarias, en XP se desarrollan las pruebas de aceptación, son especificadas por el cliente y se centran en las funcionalidades generales del sistema.

1.5. Tendencias Tecnológicas

Para el desarrollo de una solución eficiente es necesario realizar un estudio previo de las tecnologías existentes. A su vez el cliente proporcionó normas y procedimientos que rigen la organización objeto de estudio, para así seleccionar las más adecuadas para la tarea en

cuestión. Sobre esta base se seleccionaron las herramientas tecnológicas que a continuación se detallan.

1.5.1 Lenguaje del lado del cliente

Los lenguajes de programación del lado cliente se usan para su integración en páginas web. Un código escrito en un lenguaje de script se incorpora directamente dentro de un código HTML y se ejecuta interpretado. Con la programación del lado del cliente se pueden validar algunos de los datos en la máquina cliente antes de enviarlos al servidor. Esto proporciona a los usuarios informes de error inmediatos, mientras siguen en esa página de formulario y sin necesidad de volver atrás tras recibir un mensaje de error. De forma general se trata de un lenguaje de programación porque es el navegador el que soporta la carga de procesamiento, en el proyecto se emplearon algunas tecnologías de este tipo como:

HTML 5

Del lado cliente se desea utilizar HTML (Hyper Text Markup Language), lenguaje con el que se escriben las páginas web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento. Un documento hipertexto no sólo se compone de texto, puede contener imagen, sonido, video, etc., por lo que el resultado puede considerarse como un documento multimedia. Los documentos HTML deben tener la extensión html o htm (HyperText Markup), para que puedan ser visualizados en los navegadores que se encargan de interpretar el código y mostrar a los usuarios las páginas web resultantes del código interpretado. (Mora, 2014).

HTML5 promete cambiar la forma en que las aplicaciones web se desarrollan a través de su abanico de funcionalidades de lado del cliente incluyendo hojas de estilos de código (CSS, por sus siglas en inglés) y otras características compatibles con los navegadores actuales. (Collins, 2012). Este lenguaje como su antecesor HTML 4 hace uso también de las librerías de JavaScript y la combinación de estos lenguajes hacen del diseño y maquetación de la web una experiencia más fácil y amena, de aquí la justificación para su uso en este trabajo.

CSS 3

CSS son las siglas de *Cascade Style Sheet* que traducido al español significa hojas de estilo en cascada. Las hojas de estilo es una tecnología que permite controlar la apariencia de una página web. En un principio, los sitios web se concentraban más en su contenido que en su presentación, tal como sucede con HTML que no pone atención en la apariencia del documento, sino en la estructura. CSS describe cómo los elementos dispuestos en la página son presentados al usuario, siendo es un gran avance que complementa el HTML y la Web en general. Con las hojas estilo se pueden especificar el tamaño, fuentes, color, espaciado entre textos y recuadros, así como el lugar donde disponer texto e imágenes en la página. (Gauchat, 2012).

La elección de CSS3 para desarrollar este proyecto viene dada debido a la gran variedad de propiedades que incorpora con respecto a su antecesor. En esta nueva versión según (Gauchat, 2012) se añaden propiedades como *border-radius* (genera esquinas redondeadas). *Box-shadow* y *text-shadow* se encargan de proporcionar sombra a la caja de texto y al propio texto. *Border-image* (crea un borde con una imagen personalizada), estas entre otras propiedades dotan al diseño de un resultado elegante y atractivo.

JavaScript

Java Script es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Java Script se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. Java Script permite ejecutar instrucciones como respuesta a las acciones del usuario (eventos), con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo. (Gauchat, 2012). Además, Java Script pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente. Se hará uso de este lenguaje en el

proyecto debido a que es un lenguaje pensado para mejorar la interfaz de usuario y proveer dinamismo a la web, se complementa además con otro de los lenguajes que serán utilizados en este trabajo (HTML5).

1.5.2 Lenguaje del lado del servidor

Los Lenguajes de programación del lado del servidor son especialmente útiles en trabajos que se tiene que acceder a información centralizada, situada en una base de datos en el servidor, y cuando por razones de seguridad los cálculos no se pueden realizar en la computadora del usuario.

Es importante destacar que los lenguajes de programación del lado del servidor son necesarios porque para hacer la mayoría de las aplicaciones web se debe tener acceso a muchos recursos externos a la computadora del cliente, principalmente bases de datos alojadas en servidores de Internet. Alguna de las herramientas utilizadas fueron:

C#

Este lenguaje se basa en una fuerte herencia de C++.y como se convierte en inmediatamente familiar para los programadores de C++ y Java. “C# es un lenguaje de programación moderno e intuitivo orientado a objetos que ofrece importantes mejoras, incluidos un sistema de tipos unificados, código no seguro para un control máximo del programador y nuevas estructuras de lenguaje muy eficaces y fáciles de entender por la mayoría de los programadores”. (Microsoft, 2014) En varios sitios y bibliografías se define como un lenguaje moderno, potente, flexible. Para este trabajo la autora optará por el empleo de este lenguaje por su sintaxis fácil y rápida de asimilar, unido a que la curva de aprendizaje de este lenguaje ha sido más elevada con respecto a otros. C# es además desarrollado por Microsoft como parte de su plataforma .NET, por tanto, se integra perfectamente al Entorno de Desarrollo Integrado (IDE) Visual Studio que se utilizará en este proyecto.

ASP.Net Core

ASP.NET Core es un marco web gratuito y de código abierto , y un rendimiento más alto que *ASP.NET*, desarrollado por Microsoft y la comunidad. Es un marco modular que se ejecuta tanto en el completo *.NET Framework* , en Windows , como en el multiplataforma *.NET* . El marco es una reescritura completa que une *ASP.NET MVC* y la API web *ASP.NET* previamente separadas en un solo modelo de programación . (community., 2017)

A pesar de ser un nuevo marco, construido sobre una nueva pila web, tiene un alto grado de compatibilidad de concepto con *ASP.NET MVC*. Las aplicaciones de *ASP.NET Core* son compatibles con las versiones en paralelo en las que diferentes aplicaciones, que se ejecutan en la misma máquina, pueden apuntar a diferentes versiones de *ASP.NET Core*. Esto no es posible con versiones anteriores de *ASP.NET*. (Microsoft, 2017)

Características de ASP.NET Core

-) Experiencia de desarrollador sin compilación (es decir, la compilación es continua, de modo que el desarrollador no tiene que invocar el comando de compilación)
-) Marco modular distribuido como paquetes NuGet
-) Tiempo de ejecución optimizado para la nube (optimizado para Internet)
-) Una historia unificada para crear UI web y API web (es decir, ambas iguales)
-) Un sistema de configuración basado en el entorno listo para la nube.
-) Un canal de solicitud HTTP ligero y modular.
-) Cree y ejecute aplicaciones de varias plataformas *ASP.NET Core* en Windows, Mac y Linux
-) Código abierto y enfocado en la comunidad.
-) Aplicación de versiones lado a lado cuando se dirige a *.NET Core*.

1.5.4 Entity Framework y Object Relational Mapping

Entity Framework (EF) es un *Object Relational Mapping (ORM)* que permite a los desarrolladores de *.NET* trabajar con datos relacionales usando objetos específicos del dominio (Lerman & Rowan, 2011)

Es una técnica de programación para convertir las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador. (Lerman, 2011).

-) **Persistencia transparente:** los objetos del dominio no saben nada acerca de la base de datos donde son persistidos, el framework lo resuelve en forma automática utilizando archivos de *mapping* expresados en XML.
-) **Soporte de polimorfismo:** puede cargar jerarquías de objetos en forma polimórfica. Soporte de los 3 niveles de mapeo de herencia: mapear toda una jerarquía de clases a una sola tabla, crear una tabla por cada clase concreta o crear una tabla por cada escalón de la jerarquía.
-) **Soporte completo de asociaciones:** los *frameworks* de ORM soportan el mapeo de todos los tipos de relaciones que pueden existir en un modelo de objetos del dominio (asociaciones 1...1, 1...N, N...M, unidireccionales y bidireccionales).
-) **Soporte de *caching*:** en el contexto de una transacción, se pueden disminuir la cantidad de veces en que se recupera información a la base de datos cacheando en memoria los objetos que son accedidos varias veces.
-) Soporte de múltiples dialectos SQL: se puede independizar completamente del tipo de base de datos utilizada. La aplicación puede persistir sus datos en SQL Server, en Oracle, en MySQL, etc. simplemente cambiando la configuración correspondiente.

Existen tres formas diferentes de trabajo del ORM en EF: *Database First* (permite primero crear la base de datos con sus tablas y luego incorporarlas a la aplicación), *Model First* (define las clases en la aplicación y a partir de ello crear la base de datos asociada) y *Code First* (utilizada para generar una nueva base de datos a partir de un modelo o mapear su modelo para una base de datos existente). En este trabajo se utilizó *Model First*.

1.5.5 Sistemas de Gestión de Bases de Datos (SGBD)

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por *Oracle Corporation* y está

considerada como la base datos de código abierto más popular del mundo,¹² y una de las más populares en general junto a Oracle y *Microsoft SQL Server*, sobre todo para entornos de desarrollo web.

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL AB fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL. (MySQL, 1995)

1.6. Conclusiones del Capítulo

Luego de estudiar los antecedentes del proceso objeto de estudio y los aspectos generales de la entidad, así como las herramientas y tecnologías y la metodología de desarrollo de software, se concluye que:

1. El análisis de los fundamentos teóricos acerca de las normas de consumo material y la forma de gestionar las solicitudes de almacén fue esencial para entender el funcionamiento del proceso.
2. Los sistemas existentes no resuelven la situación polémica planteada.
3. La combinación de las herramientas de desarrollo antes mencionadas y la metodología XP, es la apropiada para desarrollar la aplicación que dará solución al problema de esta investigación.
4. Para el desarrollo de las etapas de la Ingeniería de Software es factible utilizar XP.

En sentido general se ha contribuido a la mejor comprensión del objeto de estudio y se han establecido las bases para las siguientes fases de la investigación.

Capítulo II Descripción de la Solución Propuesta

2.1. Introducción

La elaboración de una propuesta informática requiere tiempo y dedicación. El fruto es obtener una herramienta que facilite y agilice un proceso determinado. Con la organización y constancia requerida el producto a alcanzar tendrá un mejor acabado y logrará satisfacer a sus usuarios. En este capítulo se realiza una descripción detallada del sistema informático propuesto como solución a la problemática explicada en el capítulo anterior. Se realiza una descripción partiendo del inicio del proceso hasta el alcance abarcado en esta investigación. Se explica aplicando la metodología XP la construcción del sistema, en cada etapa de planificación del mismo en función de las necesidades del proceso.

2.2 Solución propuesta

La aplicación web GESSAB (Sistema de Gestión de Solicitudes de Almacén en Bufete) tiene como objetivo fundamental automatizar la gestión de solicitudes de materiales insumos en la Dirección Provincial de Bufetes Colectivos. De esta forma, permite reducir el tiempo de gestión, manipular datos confiables y obtener la información requerida en el momento oportuno. El mismo cuenta con un sistema de autenticación basado en roles (**Administrador, Solicitante, Almacenero, Jefe de Almacén**)

A continuación, se especifican los roles y se crean los artefactos de la metodología, con el fin de dar seguimiento al proceso de desarrollo y lograr un producto de calidad.

2.2.1. Equipo de trabajo o Roles

La metodología XP está diseñada para ser aplicada en un equipo de trabajo formado por el cliente y más de un desarrollador según los diferentes roles; en éste caso el equipo de trabajo está formado por el cliente, un entrenador y un desarrollador. Entre las características fundamentales del equipo de trabajo de XP tenemos:

-) El cliente forma parte del equipo, escribe las Historias de Usuario (HU), las pruebas funcionales para validar su implementación, asigna la prioridad de la HU y propone cuál se implementará en cada iteración.
-) Se trabaja cuarenta horas semanales.
-) Se trabaja desarrollando pequeñas versiones funcionales que se entregan al final de cada iteración.

El equipo de trabajo se encuentra formado por:

1. John de Jesús Lara Sobrino
2. Sergio Jorge Pagés Valdés
3. Manuel Miguel Gongora Vera
4. Oslaida Paéz Elías
5. Liz Pérez Martínez

| Miembros | Roles que desempeñan |
|-----------------------------------|--|
| John de Jesús Lara Sobrino | Programador, Encargado de pruebas, Encargado de seguimiento. |
| Sergio Jorge Pagés Valdés. | Clientes, Encargado de pruebas. |
| Manuel Miguel Gongora Vera | |
| Oslaida Paéz Elías | |
| MsC. Liz Pérez Martínez | Entrenador, Gestor, Consultor. |

Tabla No.2.1: Integrantes del equipo de trabajo. Fuente: Elaboración propio

En la Tabla 2.2 se visualizan los cuatros usuarios principales y los permisos que le corresponden a ambos, en la aplicación.

| Usuarios | Permisos |
|------------------------|--|
| Administrador | <p>Administra el sistema y tiene acceso a la gestión de usuarios, unidades de medidas y centro de costo. Inserta, modifica y desactiva usuarios (solicitante, jefe de almacén y almacenero).</p> <p>Inserta, modifica y elimina municipios, unidades de medidas.</p> |
| Solicitante | <p>Gestiona todo lo relacionado con las solicitudes de almacén y visualiza todos los reportes correspondientes a este proceso.</p> <p>Inserta, modifica y elimina solicitudes.</p> |
| Jefe de Almacén | <p>Gestiona todo lo relacionado con las solicitudes. Se encarga de la gestión del proceso de solicitud.</p> <p>Inserta, aprueba y cancela solicitudes. Consulta toda la información que se gestiona en el sistema.</p> |

| | |
|-------------------|--|
| Almacenero | Gestiona todo el flujo informativo acerca de las solicitudes. Se ocupa de la gestión del estado de la fase de las solicitudes. |
|-------------------|--|

Tabla No.2.2: Usuarios y permisos del sistema. Fuente: Elaboración propia

2.2.2. Análisis de Requerimientos

El levantamiento de los requerimientos constituye el paso primordial en el proceso de construcción de software, pues estos darán origen a las funcionalidades principales a implementar. Dichos requerimientos se especifican en las HU, las cuales, pueden diferir de los requerimientos iniciales, ya que un mismo requerimiento puede dividirse en varias HU, o pueden agruparse varios en una misma HU. Los requerimientos funcionales iniciales son los siguientes los cuales pueden ser modificados de ser necesarios:

1. Crear y almacenar la información en Bases de Datos (BD)
2. Seguridad y autenticación de usuarios
3. Gestionar usuarios
4. Gestionar roles
5. Gestionar solicitudes
6. Gestionar solicitudes normadas
7. Gestionar producto
8. Gestionar unidades de medidas
9. Gestionar parámetros
10. Gestionar conexión con la base de datos de Versat
11. Gestionar normas

- 12. Gestionar clasificadores
- 13. Generar reportes informativos
- 14. Trazas de Seguridad

2.2.3. Requisitos no funcionales

Son requerimientos que imponen restricciones en el diseño o implementación de un sistema como estándares de calidad. Son propiedades o cualidades que el producto debe tener. Los siguientes requisitos no funcionales, están orientados a la seguridad, estandarización, y centralización de datos, para asegurar así, el rendimiento y calidad del software.

-) Los tiempos de respuesta del software deben ser mínimos.
-) El software deberá permitir el ingreso solo al personal autorizado.
-) El software deberá mostrarle a cada usuario solo sus datos.
-) El software debe poder ser usado por personas con conocimientos básicos en el manejo de computadoras.
-) El software debe contar con una ayuda dinámica del sistema, que guíe al usuario en su interacción con él.
-) El software debe presentar una interfaz amigable y sencilla.

Además de los requisitos no funcionales anteriormente descritos, en las Tablas No.2.3 y 2.4 relacionan los requerimientos de hardware y software que debe cumplir el sistema.

| | | | |
|--|------------|-------------------------------------|------------|
| | Procesador | Ram (<i>Random Access Memory</i>) | Disco Duro |
|--|------------|-------------------------------------|------------|

| | | | |
|-----------------|-----------------|------|--------------|
| Servidor | Core i3 4 GHz | 4 GB | 80 GB libres |
| Cliente | Core Duo 1+ GHz | 1 GB | 40 Mb libres |

Tabla No.2.3: Requisitos de Hardware. Fuente: Elaboración propia

| Servidor | Cliente |
|--|---|
| <p>Sistemas Operativos:</p> <ul style="list-style-type: none"> ➤ Windows Server 2008 R2 SP1 (64 bits) ➤ Windows Server 2008 SP2 (32 y 64 bits) <p>Sistema Gestor de Base de Datos: SQL Server 2008</p> | <p><i>Navegadores:</i></p> <ul style="list-style-type: none"> ➤ Mozilla Firefox 15.0 o superior ➤ Google Chrome 30.0 o superior |

Tabla No.2.4: Requisitos del Software. Fuente: Elaboración propia

2.2.4. Plan de Iteración.

El proyecto fue dividido en cuatro iteraciones, por lo que se obtuvo un total de cuatro entregas para las cuales se desarrollaron partes de la aplicación completamente funcionales. Para la determinación de cada una de las iteraciones se tuvo en cuenta la opinión del cliente a través de las entrevistas que se le realizaron continuamente antes de comenzar a desarrollar cada iteración donde se tomaron todos los acuerdos necesarios. Una vez concluida la iteración que va a estar conformada por un conjunto de historias de usuarios, teniendo en cuenta los requisitos exigidos por el cliente, la aplicación poseerá mayor número de funcionalidades.

2.2.5 Historias de usuarios iniciales

Las historias de usuarios son definidas por el cliente y aunque a partir de ellas se planifica todo el proceso de desarrollo son de vital importancia en la comunicación del cliente y los desarrolladores. Además, los desarrolladores le asignan a estas historias un riesgo en desarrollo. Las iteraciones se definen en función de las HU teniendo en cuenta, la prioridad en el negocio, el riesgo de desarrollo y la estimación de tiempo en desarrollo que debe estar entre uno y tres semanas.

A continuación se muestra una tabla con las HU definidas por el cliente y el desarrollador:

| No | Historias de Usuarios | Prioridad | Riesgo | Esfuerzo | Iteración | Entregas |
|----|---------------------------------------|-----------|--------|----------|-----------|--------------------------------|
| 1 | Diseño y Creación de La Base de Datos | Alta | Alta | 2,5 | 1 | 11/12/2018 al 28/12/2018 |
| 2 | Diseño interfaz de Usuario | Alta | Alta | 2 | 2 | 28/12/2018 al 11/01/2019 |
| 3 | Autenticarse | Alta | Medio | 0.5 | 3 | 11/01/2019 al 1/02/ 2019 |
| 4 | Gestionar Usuarios | Medio | Medio | 2 | 3 | 11/01/2019 al 1/02/ 2019 |

| | | | | | | |
|--|--|--------------|--------------|------------|----------|---|
| 5 | Gestionar Roles y Permisos | Medio | Medio | 0.5 | 3 | 11/01/2019 al 1/02/ 2019 |
| 1 semana de Pruebas de Aceptación | | | | | | |
| 6 | Gestionar conexión a la base de datos de Versat | Alta | Alta | 2.5 | 4 | 8/02/2019 al 6/03/2019 |
| 7 | Gestionar Producto | Alta | Medio | 1.5 | 4 | 8/02/2019 al 6/03/2019 |
| 8 | Gestionar Clasificadores | Medio | Medio | 1 | 4 | 8/02/2019 al 6/03/2019 |
| 9 | Gestionar unidad de medida | Medio | Medio | 1 | 4 | 8/02/2019 al 6/03/2019 |
| 1 semana de Pruebas de Aceptación | | | | | | |
| 10 | Gestionar existencia de almacén | Alta | Medio | 1 | 5 | 13/3/2019 al 17/04/2019 |

| | | | | | | |
|--|---|--------------|--------------|-------------|----------|---|
| 11 | Gestionar Solicitudes | Alta | Alta | 1.5 | 5 | 13/3/2019 al 17/04/2019 |
| 12 | Gestionar solicitudes normadas | Alta | Alta | 1.5 | 5 | 13/3/2019 al 17/04/2019 |
| 1 semana de Pruebas de Aceptación | | | | | | |
| 13 | Gestionar período de solicitud normada | Medio | Alta | 1 | 6 | 24/04/2019 al 15/05/2019 |
| 14 | Gestionar norma | Alta | Alta | 2 | 6 | 24/04/2019 al 15/05/2019 |
| 15 | Gestionar parámetros | Alta | Medio | 0.5 | 6 | 24/04/2019 al 15/05/2019 |
| 16 | Gestionar Reportes | Medio | Medio | 0.5 | 6 | 24/04/2019 al 15/05/2019 |
| 1 semana de Pruebas de Aceptación | | | | | | |
| Total | | | | 21.5 | 6 | 4 |

Tabla No.2.5: Plan de Entrega. Fuente: Elaboración propia

2.2.6 Descripción de las historias de usuario y tareas de programación

HU -1 Diseño y Creación de la Base de Datos

| | |
|---|-----------------------------------|
| Historia de Usuario | |
| Número: 1 | Usuario: Administrador |
| Nombre historia: Diseño y Creación de la Base de Datos | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 2,5 | Iteración asignada: 1 |
| Programador responsable: John de Jesús Lara Sobrino | |
| Descripción: Diseñar e implementar la base de datos. | |
| Observaciones: Verificar la integridad de los datos. | |

Tabla No.2.6: HU -1 Diseño y Creación de la Base de Datos. Fuente: Elaboración propia

Tarea 1.1 Diseño de la BD.

| | |
|--|------------------------------|
| Tarea | |
| Número de tarea: 1.1 | Número de historia: 1 |
| Nombre de tarea: Diseño de la BD | |
| Tipo de tarea: Desarrollo | Puntos estimados: 2,5 |
| Responsable: John de Jesús Lara Sobrino. | |
| Descripción: Diseñar la base de datos con el mapeo ORM. | |

Tabla No.2.7: Tarea 1.1 Diseño de la BD. Fuente: Elaboración propia

Tarea 1.2 Creación de la BD.

| | |
|---|------------------------------|
| Tarea | |
| Número de tarea: 1.2 | Número de historia: 1 |
| Nombre de tarea: Creación de la BD | |
| Tipo de tarea: Desarrollo | Puntos estimados: 2,5 |
| Responsable: John de Jesús Lara Sobrino. | |

| |
|---|
| Descripción: Se crea la base de datos con el mapeo de ORM y el gestor de base de datos MySql |
|---|

Tabla No.2.8: Tarea 1.2 Creación de la BD.. Fuente: Elaboración propia

HU -2 Diseño interfaz de usuario

| | |
|--|-----------------------------------|
| Historia de Usuario | |
| Número: 2 | Usuario: Administrador |
| Nombre historia: Diseño de la interfaz de usuario | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alta |
| Puntos estimados: 2 | Iteración asignada: 2 |
| Programador responsable: John de Jesús Lara Sobrino. | |
| Descripción: Crear las interfaces de usuarios de la aplicación Web. | |
| Observaciones: Deben ser creadas correctamente. | |

Tabla No.2.9: HU-2 Diseño interfaz de usuario Fuente: Elaboración propia

Tarea 2.1 Crear Interfaz Principal

| | |
|---|------------------------------|
| Tarea | |
| Número de tarea: 2.1 | Número de historia: 2 |
| Nombre de tarea: Crear Interfaz Principal | |
| Tipo de tarea: Desarrollo | Puntos estimados: 2 |
| Responsable: John de Jesús Lara Sobrino. | |
| Descripción: Crear la interfaz principal de la aplicación. | |

Tabla No.2.10: Tarea 2.1 Crear Interfaz Principal Fuente: Elaboración propia

HU -4 Gestionar usuarios

| | |
|--|---|
| Historia de Usuario | |
| Número: 4 | Usuario: Administrador y cliente |
| Nombre historia: Gestionar usuarios | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Medio |

| | |
|--|------------------------------|
| Puntos estimados: 2 | Iteración asignada: 3 |
| Programador responsable: Celia Fuentes Gómez. | |
| Descripción: Mostrar, desactivar, editar y agregar usuarios que puedan acceder al sitio. | |
| Observaciones: Solo puede acceder a esta página un usuario con rol administrativo, debidamente autenticado. | |

Tabla No.2.11: HU-4 Gestionar Usuario Fuente: Elaboración propia

Tarea 4.1 Autenticación de usuario

| | |
|--|------------------------------|
| Tarea | |
| Número de tarea: 4.1 | Número de historia: 3 |
| Nombre de tarea: Autenticación de usuario | |
| Tipo de tarea: Desarrollo | Puntos estimados: |
| Responsable: John de Jesús Lara Sobrino. | |
| Descripción: Implementar una función que permita al usuario acceder al sistema dependiendo de su rol. | |

Tabla No.2.12: Tarea 4.1 Autenticación de usuario Fuente: Elaboración propia

Tarea 4.2 Agregar usuario

| | |
|---|------------------------------|
| Tarea | |
| Número de tarea: 4.2 | Número de historia: 4 |
| Nombre de tarea: Agregar usuario | |
| Tipo de tarea: Desarrollo | Puntos estimados: |
| Responsable: John de Jesús Lara Sobrino. | |
| Descripción: Implementar una función que permita agregar usuarios llenando los campos correspondientes a sus datos personales. | |

Tabla No.2.13: Tarea 4.2 Agregar usuario Fuente: Elaboración propia

Tarea 4.3 Editar usuario

| | |
|---|------------------------------|
| Tarea | |
| Número de tarea: 4.3 | Número de historia: 4 |
| Nombre de tarea: Editar usuario | |
| Tipo de tarea: Desarrollo | Puntos estimados: |
| Responsable: John de Jesús Lara Sobrino. | |
| Descripción: Se implementará una función que permita editar el usuario seleccionado. | |

Tabla No.2.14: Tarea 4.3 Editar usuario Fuente: Elaboración propia

Tarea 4.4 Desactivar usuario

| | |
|---|------------------------------|
| Tarea | |
| Número de tarea: 4.3 | Número de historia: 4 |
| Nombre de tarea: Desactivar usuario | |
| Tipo de tarea: Desarrollo | Puntos estimados: |
| Responsable: John de Jesús Lara Sobrino. | |
| Descripción: Se implementará una función que permita desactivar el usuario seleccionado. | |

Tabla No.2.15: Tarea 4.4 Desactivar usuario Fuente: Elaboración propia

HU-7 Gestionar Producto

| | |
|--|------------------------------------|
| Historia de Usuario | |
| Número: 7 | Usuario: Jefe de Almacén |
| Nombre historia: Gestionar Producto | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Media |
| Puntos estimados: 1.5 | Iteración asignada: 4 |
| Programador responsable: John de Jesús Lara Sobrino | |

| |
|--|
| Descripción: Se inicia cuando el usuario necesita asociar los productos de la base de datos con los códigos de versat.. |
| Observaciones: Es de vital importancia verificar la integridad de los datos. |

Tabla No.2.16: HU- Gestionar Usuarios Fuente: Elaboración propia

Tarea 7.1 Agregar Producto

| | |
|--|------------------------------|
| Tarea | |
| Número de tarea: 7.1 | Número de historia: 7 |
| Nombre de tarea: Agregar Producto | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Responsable: John de Jesús Lara Sobrino | |
| Descripción: Implementar una función que permita agregar un producto. | |

Tabla No.2.17: Tare 7.1- Agregar Producto Fuente: Elaboración propia

Tarea 7.2 Editar Producto

| | |
|--|------------------------------|
| Tarea | |
| Número de tarea: 7.2 | Número de historia: 7 |
| Nombre de tarea: Editar Producto | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Responsable: John de Jesús Lara Sobrino | |
| Descripción: Implementar una función que permita agregar un producto. | |

Tabla No.2.18: Tare 7.2- Editar Producto Fuente: Elaboración propia

Tarea 7.3 Eliminar Producto

| | |
|---|------------------------------|
| Tarea | |
| Número de tarea: 7.3 | Número de historia: 7 |
| Nombre de tarea: Eliminar Producto | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |

| |
|---|
| Responsable: John de Jesús Lara Sobrino |
| Descripción: Implementar una función que permita eliminar un producto. |

Tabla No.2.19: Tare 7.3- Eliminar Producto Fuente: Elaboración propia

Tarea 7.4 Asociar Producto

| | |
|--|------------------------------|
| Tarea | |
| Número de tarea: 7.4 | Número de historia: 7 |
| Nombre de tarea: Asociar Producto | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Responsable: John de Jesús Lara Sobrino | |
| Descripción: Implementar una función que permita asociar un producto de la base datos del sistema con la base de datos del sistema contable (Versat). | |

Tabla No.2.20: Tare 7.4- Asociar Producto Fuente: Elaboración propia

2.2.7 Estudio de factibilidad

Una de las cuestiones a tener en cuenta en el momento de desarrollar un software, es validar los beneficios y ventajas de éste con respecto a su costo, por lo que es necesaria una estimación del costo del software, además de un análisis de los beneficios tangibles e intangibles que reportará el proyecto.

Estimación de costo.

En el momento de la planificación del proyecto, es casi imposible determinar el número de líneas de código que poseerá la aplicación, siendo el método tradicional para calcular los costos COCOMO (Modelo Constructivo de Costos, por su acrónimo del inglés *CO*nstructive*CO*st*MO*del) poco idóneo en este caso, ya que está orientado a la magnitud del producto final, midiendo el tamaño del proyecto, en líneas de código principalmente. Dentro de sus principales inconvenientes podemos citar:

-) Los resultados no son proporcionales a las tareas de gestión ya que no tiene en cuenta los recursos necesarios para realizarlas.

-) Se puede desviar de la realidad si se indica mal el porcentaje de líneas de comentarios en el código fuente.
-) Es un tanto subjetivo, puesto que está basado en estimaciones y parámetros que pueden ser "vistos" de distinta manera por distintos analistas que usen el método.
-) Se miden los costes del producto, de acuerdo a su tamaño y otras características, pero no la productividad.

Sin embargo, ya que estamos utilizando XP como metodología para el desarrollo del software se tiene estimado el tiempo de desarrollo del mismo, aprovechando que la fórmula de *Bohem* comprende este parámetro ($\text{Costo} = \text{Cantidad de Hombres} * \text{Salario Medio} * \text{Tiempo de Desarrollo}$), es posible obtener una estimación del salario del autor. Haciendo los cálculos pertinentes tenemos, considerando como valor promedio para el salario mensual por hombres \$ 365.00, un desarrollador y un tiempo de desarrollo estimado de 5.375 meses, al sustituir y calcular se obtiene un costo de \$ 1961.87.

Beneficios tangibles e intangibles

El sistema permite la dinamización del manejo de la documentación relativa a las solicitudes de almacén en la Dirección Provincial de Bufetes Colectivos. La cual permite la confección de las solicitudes normadas y no normadas, correspondientes a los periodos de distribución establecidos, minimizando el tiempo de cálculo y la toma de decisiones por parte del personal encargado de dicho proceso. Además, implementa los reportes necesarios para llevar el estimado de gasto por cada solicitud.

Análisis de costos y beneficios.

Con estos beneficios y la estimación de costos realizada anteriormente, se evidencia que la solución propuesta presenta una buena relación entre costo y beneficios. Constatándose la oportunidad que representa para la Dirección Provincial de Bufetes Colectivo la implantación de un sistema de esta índole.

2.3 Conclusión del Capítulo.

Al describir la propuesta de solución al problema científico de esta investigación se concluye que:

1. Se dispone de un equipo de trabajo capaz de darle solución a la problemática existente.
2. El cliente conformó las HU de acuerdo a sus necesidades.
3. La planificación inicial es vital para tener una visión general del problema que se enfrenta y pensar desde el comienzo en soluciones eficaces.
4. La planificación de las iteraciones se realiza según las características de las HU, teniendo en cuenta la prioridad y el riesgo de desarrollo.
5. Las HU iniciales permitieron declarar los requisitos funcionales de la aplicación.
6. El estudio de factibilidad realizado demostró la pertinencia de desarrollar la aplicación, ya que los costos están por debajo de los beneficios que reporta la misma.

CAPÍTULO 3. Resultados obtenidos y pruebas realizadas al software

3.1 Introducción

El desarrollo de todo sistema informático consta de varias etapas como se ha explicado anteriormente. En este capítulo se muestran los resultados de la última etapa de desarrollo en la cual se presentan los resultados de las pruebas aplicadas al software para evaluar su funcionamiento. El proceso de validación que en este capítulo se describe está elaborado con el objetivo de encontrar la mayor cantidad de errores posibles en el sistema para de esta forma erradicarlos. A partir de los errores y su posterior solución, la calidad de la herramienta a obtener es mucho mayor, y su despliegue definitivo más sólido y robusto ante errores de los usuarios.

3.2. Pruebas

En este acápite se reflejan las pruebas realizadas al software que permitieron validar y verificar su funcionamiento.

3.2.1 Pruebas de Aceptación

Al finalizar cada iteración del software se realizaron sus correspondientes pruebas, las cuales permitieron evaluar el cumplimiento o no de las funcionalidades requeridas por el cliente y marcar el camino a seguir en la próxima iteración. En la siguiente figura se observa una representación de las pruebas de aceptación realizadas al finalizar cada iteración.

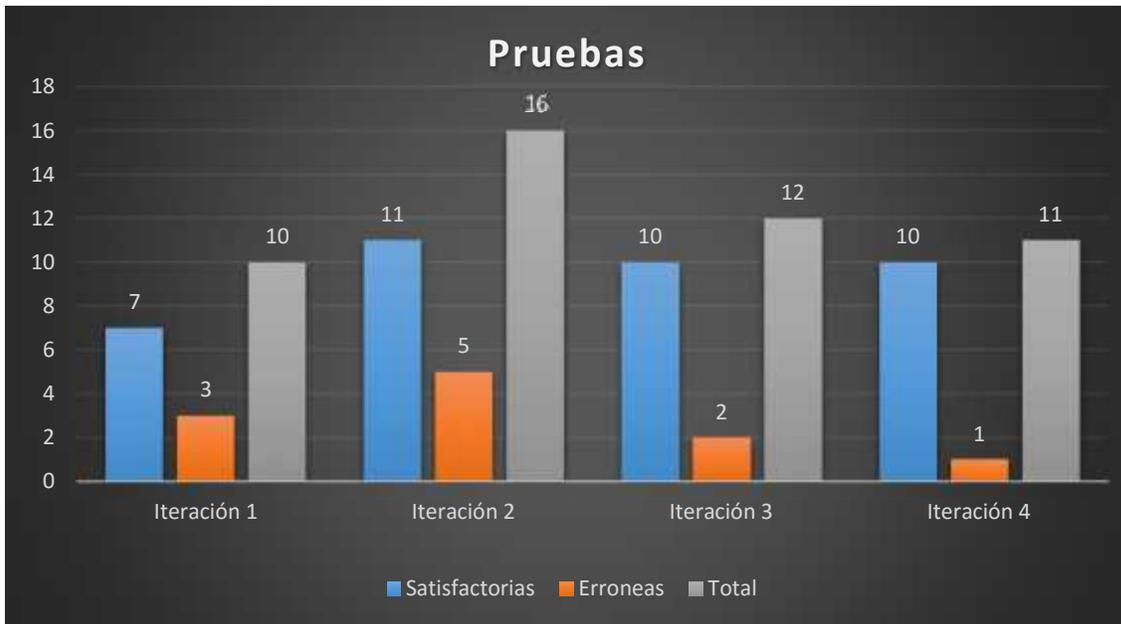


Ilustración 3.1 Pruebas de aceptación realizadas

A continuación, se especifican algunas pruebas de aceptación elaboradas.

| Pruebas de Aceptación | |
|---|----------------------------------|
| Número de caso de prueba: 1 | Número de Historia de Usuario: 2 |
| Nombre del caso de prueba: Seguridad y autenticación de usuario | |
| Descripción: Se inserta el nombre de usuario y la contraseña para entrar al software. Se insertarán de forma incorrecta, dejando campos en blanco para verificar la validación, se tratará de entrar al software poniendo una ruta destino para verificar que entra solo cuando se autentifica correctamente. Luego se insertaran los datos correctos para comprobar esta funcionalidad. | |
| Condiciones de ejecución: El usuario tendrá acceso a las funcionalidades en las que tenga los permisos correspondientes | |

| |
|--|
| <p>Entrada/Pasos de ejecución:</p> <p>Presionar el botón Entrar con los campos en blanco</p> <p>Fijar una ruta para entrar sin haberse autenticado</p> <p>Introducir los datos correctamente y presionar el botón Entrar</p> |
| <p>Resultado esperado: El software debe alertar al usuario cuando se insertan datos erróneos o no se ha autenticado. Cuando se insertan los datos correctamente el software debe entrar y mostrar las funcionalidades a las cuales el usuario tiene permiso</p> |
| <p>Evaluación de la prueba: Satisfactoria</p> |

Tabla 3.1: PA1 Seguridad y Autenticación de usuario Fuente: Elaboración Propia

| Pruebas de Aceptación | |
|---|---|
| Número de caso de prueba: 2 | Número de Historia de Usuario: 7 |
| Nombre del caso de prueba: Gestionar Producto | |
| <p>Descripción: Se inserta el nombre y demás campos para crear un nuevo producto, se insertan de forma incorrecta dejando campos en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados.</p> <p>Se modifican los datos del producto, se modifican incorrectamente para verificar la validación, se modifican correctamente para verificar que los datos sean almacenados y cargados</p> <p>Se elimina el producto aceptando el mensaje de confirmación.</p> | |
| Condiciones de ejecución: Los usuarios que pueden realizar esta acción son los que tienen el rol de jefe de almacén. Deben haberse insertado previamente el codificador de asunto a gestionar. | |

| |
|--|
| <p>Entrada/Pasos de ejecución:</p> <p>Dejar campos en blanco.</p> <p>Insertar datos de manera incorrecta.</p> <p>Insertar los datos correctamente.</p> <p>Modificar los datos dejando campos en blanco.</p> <p>Modificar los datos de manera incorrecta.</p> <p>Modificar los datos de forma correcta.</p> |
| <p>Resultado esperado: El software debe alertar al usuario cuando se insertan datos erróneos. Cuando se insertan los datos correctamente el software debe almacenarlos en la base de datos y mostrarlos</p> |
| <p>Evaluación de la prueba: Satisfactoria</p> |

Tabla 3.2: PA2 Gestionar Producto Fuente: Elaboración Propia

| Pruebas de Aceptación | |
|---|---|
| Número de caso de prueba: 2 | Número de Historia de Usuario: 7 |
| Nombre del caso de prueba: Gestionar Clasificador | |
| <p>Descripción: Se inserta el nombre y demás campos para crear un nuevo clasificador, se insertan de forma incorrecta dejando campos en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados.</p> <p>Se modifican los datos del clasificador, se modifican incorrectamente para verificar la validación, se modifican correctamente para verificar que los datos sean almacenados y cargados</p> <p>Se elimina el producto aceptando el mensaje de confirmación.</p> | |

| |
|---|
| Condiciones de ejecución: Los usuarios que pueden realizar esta acción son los que tienen el rol de jefe de almacén. Deben haberse insertado previamente el codificador de asunto a gestionar. |
| Entrada/Pasos de ejecución: Dejar campos en blanco. Insertar datos de manera incorrecta. Insertar los datos correctamente. Modificar los datos dejando campos en blanco. Modificar los datos de manera incorrecta. Modificar los datos de forma correcta. |
| Resultado esperado: El software debe alertar al usuario cuando se insertan datos erróneos. Cuando se insertan los datos correctamente el software debe almacenarlos en la base de datos y mostrarlos |
| Evaluación de la prueba: Satisfactoria |

Tabla 3.3: PA3 Gestionar Clasificador Fuente: Elaboración Propia

| Pruebas de Aceptación | |
|---|---|
| Número de caso de prueba: 2 | Número de Historia de Usuario: 9 |
| Nombre del caso de prueba: Gestionar Unidad de Medida | |
| Descripción: Se inserta el nombre y demás campos para crear un nueva unidad de medida, se insertan de forma incorrecta dejando campos en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados. | |

| |
|--|
| <p>Se modifican los datos de la unidad de medida, se modifican incorrectamente para verificar la validación, se modifican correctamente para verificar que los datos sean almacenados y cargados</p> <p>Se elimina el producto aceptando el mensaje de confirmación.</p> |
| <p>Condiciones de ejecución: Los usuarios que pueden realizar esta acción son los que tienen el rol de jefe de almacén. Deben haberse insertado previamente el codificador de asunto a gestionar.</p> |
| <p>Entrada/Pasos de ejecución:</p> <p>Dejar campos en blanco.</p> <p>Insertar datos de manera incorrecta.</p> <p>Insertar los datos correctamente.</p> <p>Modificar los datos dejando campos en blanco.</p> <p>Modificar los datos de manera incorrecta.</p> <p>Modificar los datos de forma correcta.</p> |
| <p>Resultado esperado: El software debe alertar al usuario cuando se insertan datos erróneos. Cuando se insertan los datos correctamente el software debe almacenarlos en la base de datos y mostrarlos</p> |
| <p>Evaluación de la prueba: Satisfactoria</p> |

Tabla 3.4: PA4 Gestionar Unidad de Medida Fuente: Elaboración Propia

3.2.2 Validación de la solución propuesta

El software se desarrolló con el objetivo de facilitar la manera de obtener y gestionar la información referente a los servicios que brinda la dirección provincial de Bufetes Colectivos de Matanzas en cada uno de sus municipios. Para demostrar el cumplimiento de este objetivo, en este acápite se hace un análisis de los resultados

obtenidos con la implementación de la solución propuesta, mediante una comparación con respecto a la situación inicial.

La siguiente tabla muestra en detalle el nivel de aceptación de la solución propuesta basado en las HU implementadas, las cuales obtuvieron el puntaje correspondiente por el cliente luego de realizar las pruebas de aceptación, al finalizar cada iteración de software.

La Tabla 3.5 muestra el tiempo de obtención y procesamiento de información con el uso de la solución propuesta y sin la utilización de la misma, es decir, como se hacía antes de esta investigación.

| Descripción | Sin la solución propuesta (Valor aproximado) | Con la solución propuesta |
|---|--|--|
| Gestionar solicitudes de todos los municipio | De 20 minutos a 48 horas | Menos de 1 minuto |
| Consultar situación el estado de las solicitudes | De 25 minutos a 30 días | Menos de 1 minuto |
| Consultar existencia de almacén | 45 minutos (se solicita al departamento económico y se remite vía correo) | Menos de 1 minuto |
| Consultar estado de las solicitudes | 45 minutos (requiere solicitar almacenero si todas las solicitudes fueron despachadas) | Menos de 1 minutos |

| | | |
|--------------------------------------|--|--------------------------------|
| | | |
| Reporte el gasto por elemento | 5 minutos (se solicita al departamento económico y se remite vía correo) | Menos de 1 minutos por reporte |
| Generar reportes informativos | 30 minutos | Menos de 1 minutos por reporte |

Tabla 3.4: Tiempo de obtención y procesamiento de información. Fuente: Elaboración Propia

Puede constatar que la búsqueda de información mejoró considerablemente, en especial los Reportes, así como la elaboración rápida de las solicitudes, ya que debían reportar una inmensa cantidad de datos de todos los municipios de la provincia. Con la ayuda de la solución propuesta, los reportes generados son más confiables y precisos, además disminuyen el tiempo de su elaboración.

De lo expuesto anteriormente se puede afirmar que la utilización de la solución propuesta influye de manera positiva en la gestión de solicitudes de almacén en la Dirección Provincial de Bufetes Colectivos Matanzas.

3.2.3 Pruebas de carga y estrés

Las pruebas de carga se realizan generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. La carga está determinada por el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de acciones durante el tiempo que dura la carga. Esta prueba muestra los tiempos de respuesta de todas las acciones importantes de la aplicación.

Por otra parte las pruebas de estrés se utilizan normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación

y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

Las pruebas fueron realizadas a la aplicación empleando la herramienta Webserver Stres, para una simulación de 200 usuarios conectados simultáneamente, cifra que supera la real esperada para la aplicación, y con una frecuencia de dos click por segundo por cada usuario. El resultado arrojado fue satisfactorio, lo que evidencia que la aplicación soporta condiciones de carga y estrés por encima de lo concebido para su funcionamiento, tal y como se muestra en la siguiente figura.

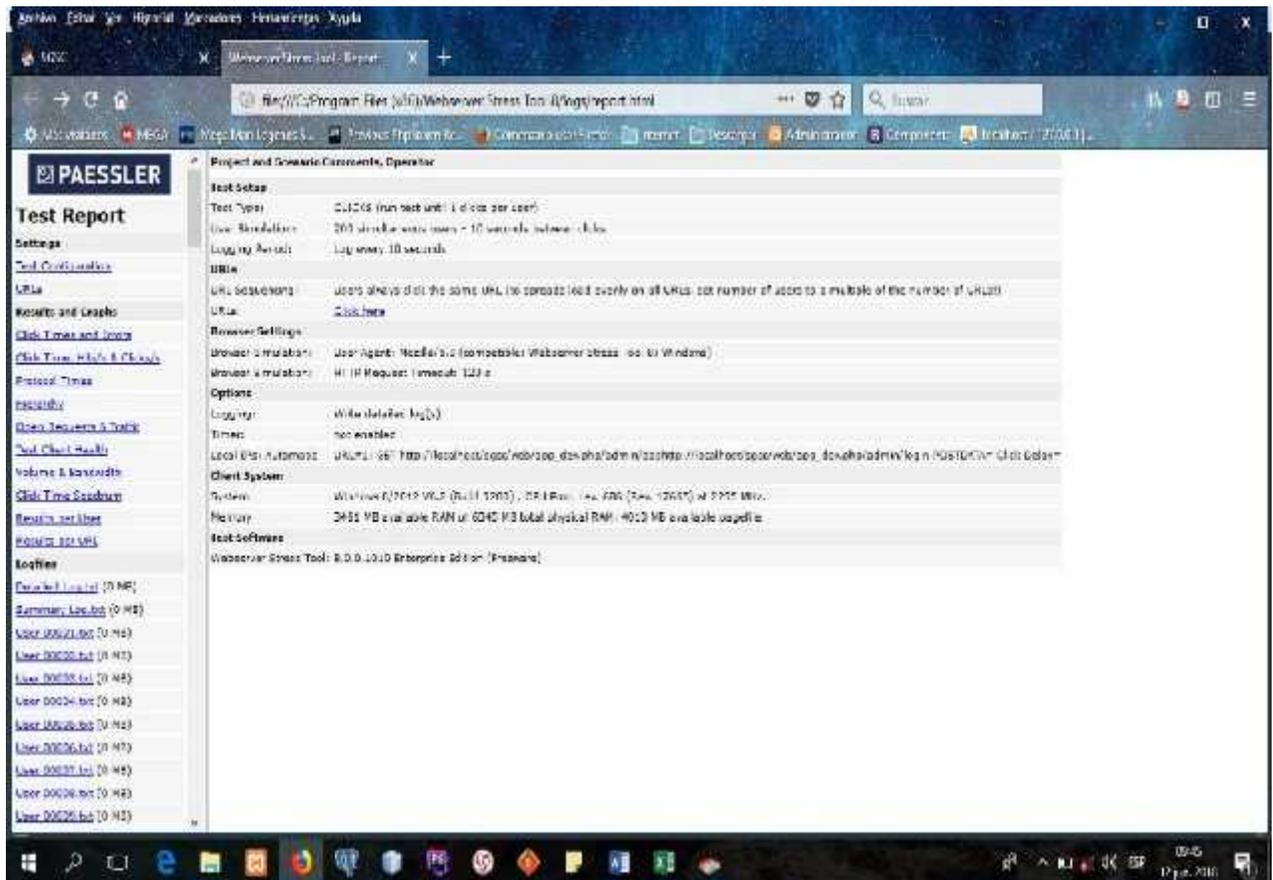


Ilustración 3.2 Pruebas de carga y estrés realizadas

3.2.4 Pruebas de Seguridad

Las pruebas de seguridad se realizan en función de evaluar las siguientes características en los sistemas:

-) Capacidad de realización de Análisis de Vulnerabilidades y Crawler automático.
-) Ejecución de pruebas de Accesibilidad UI.
-) Capacidad de realizar un Crawler (descargar copia) del website al completo.
-) Función de manipulación manual de paquetes HTTP mediante interceptación mediante Proxy (similar a Paros Proxy, ZAP, entre otras).
-) Ataques MiTM con SSL (similar a sslstrip).
-) Análisis del contenido
-) Mensajes de notificación personalizables
-) Modelo de datos propio
-) API en Javascript personalizable para el desarrollo de complementos y extensiones personalizadas.

Se realizaron las pruebas de seguridad con él con objetivo de detectar vulnerabilidades, teniendo en cuenta las técnicas de ataque más comunes en los entornos web, por ejemplo: *SQL Injection*, *XSS*, entre otros.

Los resultados obtenidos fueron satisfactorios, no detectándose errores peligrosos en la seguridad del sistema, tal y como se evidencia en la siguiente imagen que muestra el resultado de la prueba empleando VEGA.



Scan Alert Summary

| | | |
|---|---|--------------|
| High | | (None found) |
| Medium | | (2 found) |
| Local filesystem Paths Found | 2 | |
| Low | | (2 found) |
| Directory Listing Detected | 1 | |
| Form Password Field with Autocomplete Enabled | 1 | |
| Info | | (1 found) |
| Interesting Meta Tags Detected | 1 | |

Ilustración 3.3 Pruebas de Seguridad Realizadas

3.3 Conclusiones del capítulo.

Una vez construida la propuesta de solución al problema científico de esta investigación a partir de la metodología XP, y analizado los resultados de la misma a partir de las pruebas funcionales, se concluye que:

1. La planificación fue acertada, pues permitió el desarrollo del sistema según el cronograma y cumpliendo con los objetivos del cliente al iniciar esta investigación.
2. El uso de pruebas funcionales durante el desarrollo del sistema permitió tanto al cliente como al desarrollador encontrar defectos y errores en el sistema y su rápida corrección.
3. A partir del diseño de la propuesta y con el uso de las tecnologías adecuadas para su implementación se obtuvo un sistema que cumple a cabalidad con las expectativas del cliente.

CONCLUSIONES GENERALES

Como resultado de esta investigación quedaron satisfechos los objetivos trazados arribando a las siguientes conclusiones:

1. El estudio realizado sobre los antecedentes, el estado actual de la temática, la bibliografía y documentos relacionados con el objeto de estudio, permitió contar con los elementos necesarios para dar solución a la problemática planteada.
2. Los sistemas automatizados encontrados, vinculados al tema no le dan solución al problema planteado por lo que no es factible su utilización.
3. Se utilizaron las herramientas de software más factibles para la construcción de la solución.
4. Se implementó el sistema web para la gestión de las solicitudes de almacén en la Dirección Provincial de Bufetes Colectivos, cumpliendo con el cronograma de desarrollo planteado al cliente.
5. Se realizó la estimación de costo de implementación del sistema y el estudio de factibilidad, arrojando como resultado la factibilidad de la realización del sistema informático.
6. La realización de las pruebas funcionales a este sistema informático permitió detectar errores en el sistema y la rápida corrección de los mismos.
7. La implementación del sistema y la aplicación de las pruebas de validación con resultados satisfactorios demostraron que el software elaborado cumple con los requerimientos especificados constatándose, mediante avales, su aporte práctico a la organización.

Para finalizar, tras haber verificado los objetivos específicos del trabajo de diploma se puede concluir que el objetivo general se cumplió al obtener la aplicación web GESSAB. Facilitando de esta forma, la rapidez y efectividad de las solicitudes de almacén normadas, dándole un nivel de aceptación de un 97,6 %y a través del criterio de los expertos, valorándolo de forma adecuada, con se afirma que la hipótesis planteada en los inicios de la investigación es verdadera.

Recomendaciones

Desde el punto de vista del alcance del presente trabajo y teniendo en cuenta el momento de desarrollo del mismo, se proponen las siguientes recomendaciones:

1. Ampliar la aplicación, incorporándole aspectos para facilitar el trabajo del cliente.
2. Generar reportes en forma de gráficas para una mejor comprensión de los mismos.
3. Incorporar una ayuda dinámica al sistema.
4. Extender la herramienta web a toda la nación para facilitar la gestión de los procesos en el resto del país.

Referencias Bibliográficas

- . **2014.** AJAX un nuevo acercamiento a Aplicaciones Web. *AJAX un nuevo acercamiento a Aplicaciones Web*. [En línea] 2014. [Citado el: 3 de Diciembre de 2014.] <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>.
- . **2012.** Especificación JSON. *Especificación JSON*. [En línea] __, 2012. [Citado el: 10 de Diciembre de 2014.] <http://www.json.org/json-es.html>.
- . **2010.** Patrón de arquitectura Modelo Vista Controlador (MVC). *Patrón de arquitectura Modelo Vista Controlador (MVC)*. [En línea] __, 2010. [Citado el: 12 de Diciembre de 2014.] <http://www.inf.uc3m.es/~a0080802/RAI/mvc.html>.
- Apache. 2010.** Javascript a fondo. *Javascript a fondo*. [En línea] Apache, 2010. [Citado el: 9 de Diciembre de 2014.] <http://www.desarrolloweb.com/javascript>.
- Beck, Kent. 1999.** *Extreme Programming Explained: Embrace Change*. __ : __, 1999.
- CHAFFER, JONATHAN / SWEDBERG, KARL. 2010.** *APRENDE JQUERY*. s.l. : ANAYA MULTIMEDIA-ANAYA INTERACTIVA , 2010.
- Cochran, David. 2012.** *Twitter Bootstrap Web Development*. __ : Packt Publishing, 2012. Vol. 1st Edition.
- . **2012.** *Twitter Bootstrap Web Development*. 2012.
- community., Microsoft and the. 2017.** https://en.wikipedia.org/wiki/ASP.NET_Core. [En línea] 2017.
- Consortio W3C. 2007.** HTML5: ¿qué es HTML5? *HTML5: ¿qué es HTML5?* [En línea] Consortio W3C, 2007. [Citado el: 9 de Diciembre de 2014.] <http://desarrolloweb.dlsi.ua.es/html5-que-es-html5>.
- Corporation, Microsoft. 2013.** ADO.Net Entity Framework. [En línea] Microsoft Software Developers Network, 2013. [Citado el: 1 de Diciembre de 2015.] <http://msdn.microsoft.com/en-us/data/ef.aspx>.
- Corporation, Microsoft. 2014.** <http://www.asp.net/mvc>. [En línea] 2014.
- J. McCabe, T. 1976.** *A Complexity Measure*. s.l. : IEEE Transactions on Software Engineering, 1976.
- Lerman, Julia, Miller, Rowan. 2011.** *Programming Entity Framework: Code First*. 2011.
- Microsoft Corporation. 2014.** Asp.Net Mvc. [En línea] Corporation, Microsoft, 2014. [Citado el: 2014 de Diciembre de 2.] <http://www.asp.net/mvc>.
- . **2012.** C# Web Programación. *C# Web Programación*. [En línea] Microsoft Corporation, 2012. [Citado el: 8 de Diciembre de 2014.] <http://www.webprogramacion.com/csharp.aspx>.
- . **2012.** Página Web de Microsoft Sql Server. *Página Web de Microsoft Sql Server*. [En línea] Microsoft Corporation, 2012. [Citado el: 2014 de Diciembre de 4.] <http://www.microsoft.com/spain/sql/default.aspx>.

—. 2013. Sitio Oficial del Visual Studio. *Sitio Oficial del Visual Studio*. [En línea] Microsoft Corporation, 2013. [Citado el: 11 de Diciembre de 2014.] <http://www.microsoft.com/spain/visualstudio>.

Microsoft. <https://www.nuget.org/packages/Microsoft.AspNetCore.All>. [En línea]

—. 2017. <https://www.nuget.org/packages/Microsoft.AspNetCore.All>. [En línea] 2017.

Microsoft, C# Web Programación. - 2014. <http://www.webprogramacion.com/csharp.aspx>. . [En línea] 8 de 12 de 2014.

Mora, Sergio Luján. 2014. *HTML & CSS*. 2014.

MySQL. 1995. <https://es.wikipedia.org/wiki/MySQL>. [En línea] 23 de 5 de 1995.

Otto, Mark y Thornton, Jacob. 2014. Tweeter Bootstrap. [En línea] 2014. [Citado el: 7 de mayo de 2014.] <http://twitter.github.com/bootstrap>.

Pressman, R. S. 2005. *Ingeniería de Software. Un enfoque práctico*. 2005.

Resig, John. 2014. Sitio oficial del jQuery. *Sitio oficial del jQuery*. [En línea] jQuery Development Team, 2014. [Citado el: 5 de Diciembre de 2014.] <http://jquery.com/>.

Colectivo de Autores, Normas y procedimientos de la ONBC, así como sus directivas, 2 de febrero del 2018

Acrónimos

| | |
|------|---|
| MVC | Modelo Vista Controlador |
| HTML | <i>HyperText Markup Language</i> (Lenguaje de Marcado de Hipertexto) |
| CSS | <i>Cascading Style Sheets</i> (Hoja de Estilo en Cascada) |
| XP | <i>eXtreme Programming</i> (XP) |
| HU | Historia de Usuario |
| BD | Base de Datos |
| EF | <i>Entity Framework</i> |
| DDL | <i>Data Definition Language</i> (Lenguaje de Definición de Datos) |
| URLs | <i>Uniform Resource Locator</i> (Localizador de Recursos Uniforme) |
| GUI | <i>Graphical User Interface</i> (Interfaz Gráfica de Usuario) |
| DLLs | <i>Dynamic Link Library</i> (Biblioteca de Enlace Dinámico) |
| IDE | <i>Integrated Development Environment</i> (Entorno de Desarrollo Integrado) |
| UML | <i>Unified Modeling Language</i> (Lenguaje Unificado de Modelado) |
| CRUD | <i>Create Read Update Delete</i> (Crear Leer Actualizar Eliminar) |
| DPBC | <i>Dirección Provincial de Bufetes Colectivos Matanzas</i> |