



Facultad de Ciencias y Tecnologías Computacionales.

Sistema de inventario en la Consultoría Nacional Económica para la Cultura

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

Autor(es): Chabely Bressler Martínez
Yiliany Isabel Pacheco Gutiérrez

Tutor(es):
Msc. Yerania Sardá
Ing. Angélica Vázquez

La Habana, abril de 2023

Año 65 de la Revolución

DECLARACIÓN DE AUTORÍA

El (Los) autor(es) del trabajo de diploma con título “***Sistema de Inventario para la empresa Consultoría Nacional de Economía para la Cultura***”, conceden a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declaran como únicos autores de su contenido. Para que así conste firman la presente a los ____ días del mes de ____ del año _____.

Chabely Bressler Martínez

Firma del Autor

Angelica Vázquez

Firma del Tutor

Yiliany Isabel Pacheco Gutiérrez

Firma del Autor

Yerania Sardá

Firma del Tutor

AGRADECIMIENTOS

Chabely:

Quiero agradecer primeramente a mi familia, por siempre apoyarme y ayudarme en todo momento, a mi mamá Any, mi papa Irais, mis hermanos Ale, Cecy y Nathy, a Messi y Coco, a Yiyi por acompañarme en este largo viaje; a mi inquebrantable compañero Ariel por aparecer en mi camino, a mi latinoamericano preferido Sandor por aguantar mis angustias cada noche debido a mi procrastinación, a Frank Junior porque pobrecito, se va quedar calvo pronto; a Robertico por ser prácticamente mi instructor desde niña. A todos los profesores que a lo largo de mi carrera me han ayudado, así como a las tutoras Angelica y Yerania por su dedicación, y la profesora Graciela por sus consejos. A todas mis amistades que me llevo en el corazón desde el día uno: Malú, Laura y Mel, por ser las primeras y mejores personas que la carrera me ha podido brindar, a Rey Alexis por ser la genial persona que fue, a mi vecina UCI María que es mi "tanke", a Roberto el manicero por razones más que obvias y conocidas (gracias de corazón), a Alfredo, y a Marco por chismoso, pero de manera general a todos mis compañeros de clases, por hacer los días y los turnos más divertidos, y por toda la ayuda brindada a través de estos 5 años en la universidad.

Yiliany

Quisiera agradecer primeramente a mi mamá, que gracias a ella he logrado construir todo lo que tengo hasta hoy, ha sido mi motor impulsor a lo largo de mi carrera y mi vida, a mis compañeros de aula

por ser un pilar fundamental desde que ingrese en la universidad , destacando a mi amiga Anahomi que lleva un recorrido extendido a mi lado , a lo largo de los años siempre ha sido un gran apoyo y aquí no fue diferente , a mi amigo eric , que juntos perdíamos la paciencia y pensábamos que muchas cosas eran imposibles y aquí estamos cumpliendo una de ellas , a Roberto , el cual nunca dio un no como respuesta para sentarse y repetirme las clases de los profesores cuando no entendía algo y tenía exámenes , alumno estrella y siempre me tuvo como su alumna ayudante aventajada , también el papel de Denis Duranza el cual bauticé como gruñón pero con su poca paciencia me dedicaba sus minutos , mis compañeras del 106 , donde solo bastó integrarnos en un padrino para que todo se descontrolara , al ingeniero Leonel Perdomo , que a pesar de sus exigencias , siempre lo veré como un amigo , me ayudó mucho cuando lo necesité , a mis tutoras Angelica Vázquez y Yerania , por la disposición y ayuda en este proceso , a Graciela por siempre tenerme como un diamante además de ser esencial por el tiempo que nos regaló , y por último y no menos importante a mi compañera Chabely que sin ella no estaría hoy cumpliendo este sueño , no hubiese sido posible , fue un bonito recorrido a su lado.

DEDICATORIA

Chabely:

Quiero dedicar este trabajo principalmente a mi abuelo, para que se sienta orgulloso de todo lo que hemos logrado como familia, a mi mama porque se lo merece, a Ale "porque los majás si vuelan", a mi tío Hernán, por ser el genio de la familia, y con cariño, a Ariel y Kenia.

Yiliany

Quiero dedicar este trabajo a mi mamá, es especialmente para ella, para que se sienta orgullosa de su niña.

RESUMEN

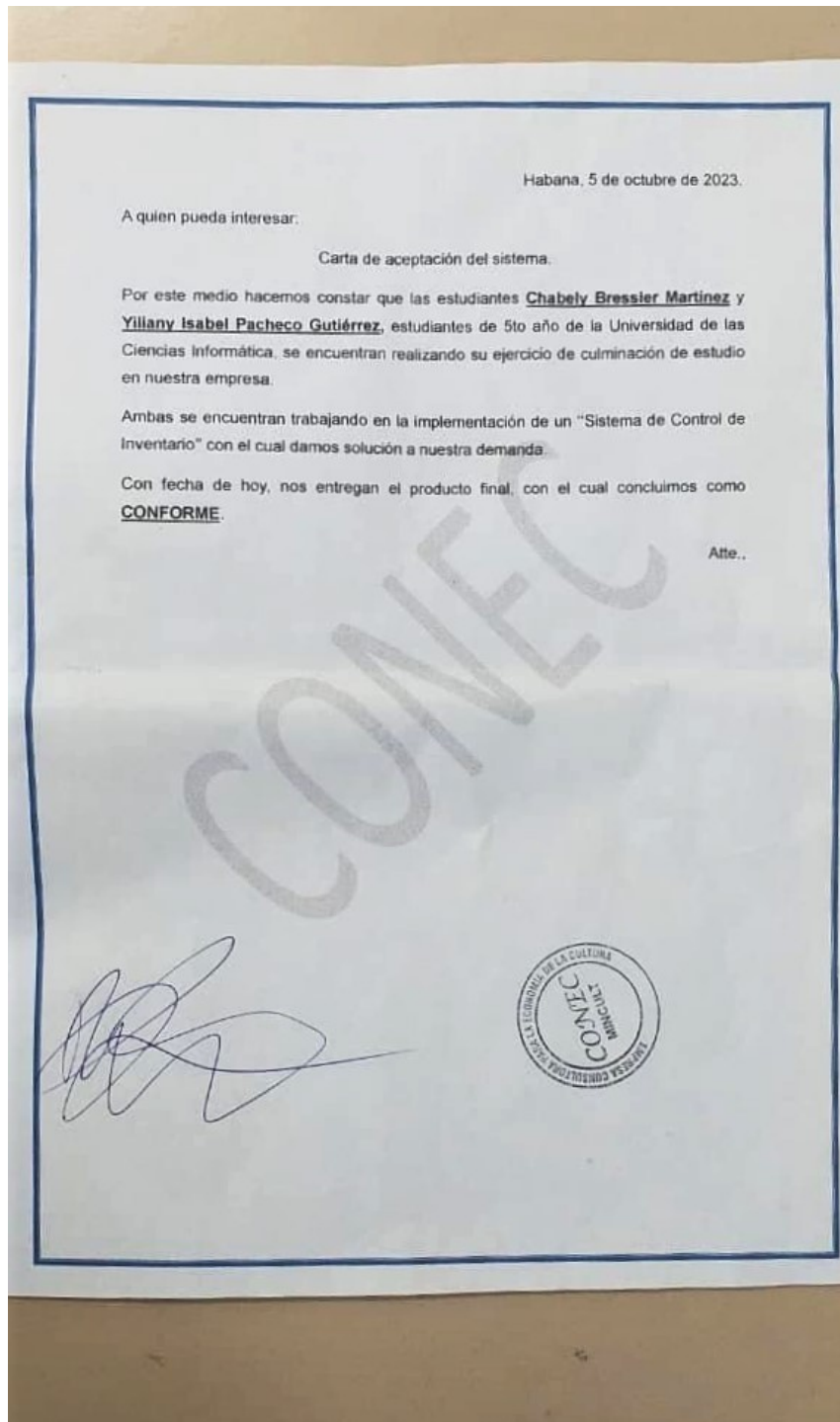
Los Sistemas de Inventario, cuyo desarrollo está marcado por el avance de las Tecnologías de la Información y las Comunicaciones (TIC), han propiciado la optimización de los procesos en las organizaciones, lo que contribuye al control efectivo de su información. En la empresa Consultoría Nacional Económica para la Cultura, se lleva a cabo un registro manual del inventario en relación a los medios básicos que se almacenan, con el fin de abastecer a las distintas Casas de Culturas. En la presente investigación se desarrolla un Sistema de Inventario para tener controlados y organizados dichos artículos. Con el fin de alcanzar los objetivos trazados se analizan los principales conceptos asociados a la obtención de inventarios, se realiza una investigación a los sistemas similares existentes, tanto en el ámbito nacional como internacional. Se utilizaron como lenguaje de desarrollo PHP, se seleccionó la metodología Scrum y como herramientas de base de datos SQL Server 2017 y SQLite 3. Se realizan un conjunto de pruebas para garantizar el correcto funcionamiento de las funcionalidades propuestas, teniendo en cuenta también que la aplicación cumple con los requerimientos del cliente. Como resultado se obtuvo un sistema que cumple todas estas necesidades.

Palabras Claves: artículos, control, gestión, información, inventario.

Inventory Systems, whose development is marked by the advancement of Information and Communications Technologies (ICT), have led to the optimization of processes in organizations, which contributes to the effective control of their information. In the National Economic Consultancy for Culture company, a manual inventory record is carried out in relation to the basic means that are stored, in order to supply the different Houses of Cultures. In this research, an Inventory System is developed to control and organize these items. In order to achieve the established objectives, the main concepts associated with obtaining inventories are analyzed, and research is carried out on existing similar systems, both nationally and internationally. PHP was used as the development language, the Scrum methodology was selected, and SQL Server 2017 and SQLite 3 were used as database tools. A set of tests were carried out to guarantee the correct functioning of the proposed functionalities, also taking into account that the application meets customer requirements. As a result, a system was obtained that meets all these needs.

Keywords: articles, control, management, information, inventory.

AVAL DEL CLIENTE



Índice

INTRODUCCIÓN.....	10
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL	
1.1 Conceptos básicos asociados al dominio de la investigación.....	14
1.2 Investigación de los principales sistemas homólogos.....	14
1.3 Descripción de los lenguajes, herramientas y tecnologías utilizadas.....	17
1.3.1 Metodología de desarrollo del Software.....	17
Conclusiones del capítulo.....	22
CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO.....	23
2.1 Modelado del dominio.....	23
2.2 Descripción de conceptos asociados al Modelado del dominio.....	24
2.3 Product Backlog.....	25
2.4 Modelado de los requisitos.....	28
2.5 Diseño de la interfaz de usuario.....	30
2.5.1 Mapa de Navegabilidad.....	31
2.6 Arquitectura de Software.....	33
2.6 Diagrama de Clases del Diseño y Patrones.....	34
2.7 Modelo Entidad-Relación.....	37
Conclusiones del capítulo.....	38
CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE INVENTARIO PARA LA EMPRESA CONEC	
.....	39
3.1 Implementación del sistema.....	39
3.1.2 Spring Backlog.....	39
3.1.2 Resultados de la revisión del Sprint.....	40
3.2. Estándares de codificación.....	41
3.3 Pruebas del software.....	42
3.3.1. Estrategia de pruebas.....	42
3.4 Aplicación de Pruebas.....	43
3.5 Resultados de las pruebas utilizando el método de caja negra.....	46
CONCLUSIONES GENERALES.....	48
RECOMENDACIONES.....	49
REFERENCIAS BIBLIOGRÁFICAS.....	50
ANEXOS.....	53

INTRODUCCIÓN

En términos generales, las Tecnologías de Informatización y Comunicación, (TIC en lo adelante) son el conjunto de herramientas y soluciones tecnológicas que permiten ordenar y procesar la información y las comunicaciones de las empresas y organizaciones en pro de la eficiencia y la agilidad. Su principal objetivo es mejorar la calidad de vida de sus usuarios por medio de tecnologías que les permitan llevar a cabo tareas y procesos más ágiles y oportunos. Como parte de dichas herramientas, podemos citar los Sistemas de inventarios, ya que estos ayudan a las organizaciones a mejorar su desempeño, y contribuyendo a crear una cultura empresarial con una dinámica de autocrítica, corrección y mejora de procesos. (GONZÁLEZ SOTO, 2019).

La evolución de los Sistemas de inventarios en Cuba tiene un gran impacto económico positivo por adoptar un enfoque orientado a facilitar el logro de objetivos, encontrándose aún en pleno proceso de ejecución. En dicho contexto y como base para lograr competitividad, continúa siendo un importante reto para el sistema de dirección y organización en los sistemas empresariales; es por ello que estas empresas deben enfrentar este desafío adoptando una visión de la competitividad como requisito sostenible, tanto hacia su interior como hacia su entorno, haciendo valer la responsabilidad social que asume.

Actualmente, la empresa de Consultoría Nacional Económica para la Cultura (CONEC en lo adelante) es un centro de trabajo donde se almacenan artículos (como mesas, sillas, ventanas, escaleras, grandes proporciones de cartones, maderas etc.) con el fin de abastecer a las distintas Casas de Culturas. En esta empresa, se lleva un registro manual del inventario en relación a los medios básicos que se almacenan, en el que el trabajador se encarga de: contar todos los artículos, puntualizar la fecha en que se dan de baja, clasificar según el tipo de artículo (ya que sus clasificaciones se dividen en: Medio Básico, Activo Fijo Tangible o Útiles y Herramientas), y verificar el estado en el que se encuentran; luego se encarga de revisar las peticiones de las Casas de Cultura y abastecerlas según la necesidad y disponibilidad en el almacén, por lo que se ha decidido crear un Sistema de inventario encargado de llevar a cabo el funcionamiento, que permitirá a los trabajadores tener control de todos los artículos que se guardan en el almacén.

A partir de lo planteado anteriormente se define como el **problema de la investigación** ¿Cómo controlar y clasificar los artículos de la empresa? Dicho proceso se ha tornado engorroso para el centro, pues ocupa largas jornadas de tiempo a los trabajadores para su realización, incluyendo errores humanos y problemas en la gestión y logística de dichos artículos. Se determinó, con la finalidad de este sistema de control de inventario, facilitar el trabajo del negocio, reduciendo procesos y mejorando los resultados.

En consecuencia, se definió como **Objeto de Estudio** los sistemas para el control de inventario para en el sector empresarial, se enmarcó como **Campo de Acción** los sistemas para el control de inventario en el sector empresarial para la Cultura. Se plantea como **Objetivo General**, desarrollar un sistema de inventario para el control de los artículos que se utilizan en la prestación de servicios en la Consultoría Nacional Económica para la Cultura.

Para guiar el desarrollo de la presente investigación surgen las siguientes **Preguntas Científicas:**

- ¿Cómo se realiza actualmente la gestión de inventario en la empresa CONEC?
- ¿Qué metodología, tecnologías y herramientas conforman el ambiente de desarrollo del sistema para el control de inventario de los artículos del centro de trabajo?
- ¿Cómo desarrollar un sistema de inventario basado en la web para el control de los artículos que se ajuste a las características de la empresa CONEC?
- ¿Cómo validar la propuesta de solución a partir de los resultados obtenidos?

Para dar solución al objetivo general de la investigación, se proponen las siguientes

Tareas de Investigación:

- Elaboración de un marco teórico metodológico sobre los sistemas de inventario empresariales.
- Selección de la metodología, herramientas y tecnologías para el desarrollo del sistema.
- Obtención del diseño del sistema de inventario para el control de artículos de la empresa CONEC.
- Implementación de un sistema de inventario para el control de artículos de la empresa CONEC.

- Validación de la propuesta de solución a partir de los resultados obtenidos.

Para apoyar el desarrollo de la investigación se emplean los siguientes **Métodos Científicos**:

Métodos Teóricos:

- Analítico sintético: Se analizaron e identificaron los sistemas homólogos internacionales y nacionales Holded, SAP y Versat Sarasola, respectivamente, referentes a los sistemas de inventario, facilitando de esta forma la extracción de los elementos más importantes, y permitiendo generar una propuesta adecuada a la situación planteada.
- Histórico Lógico: Este método permitió estudiar los sistemas inventarios, para así obtener un conocimiento de su desarrollo y comportamiento tanto a nivel internacional como nacional.

Métodos Empíricos:

- Observación: permitió analizar cada fase del proceso e ir observando las tareas realizadas y tomar experiencia de estas para aplicarlas en todas las demás.
- Entrevista: Esta técnica se aplicó a los trabajadores de la empresa CONEC, para conocer los recursos con los que disponen para el desarrollo de un sistema basado en la web que permita satisfacer sus necesidades

El trabajo de diploma está compuesto por tres capítulos, los cuales se dividen de la siguiente forma:

- **Capítulo1. “Fundamentos y referentes teórico-metodológicos sobre el objeto de estudio”:** La fundamentación teórica de una tesis, es el tratamiento conceptual del tema, se define el asunto o problema de estudio desde determinado punto de vista, con base en datos acumulados en el transcurso de la investigación bibliográfica. Contiene un análisis a partir de la investigación realizada sobre las tecnologías inmersas en la solución. Además, se fundamentan los conceptos, metodologías, lenguajes, y herramientas utilizadas.
- **Capítulo 2. “Diseño de la solución propuesta al problema científico”:** en este capítulo se obtienen los productos de trabajo resultantes del desarrollo de las actividades ejecutadas en las disciplinas de modelado del negocio,

requisitos, análisis y diseño que se definen en la metodología.

- **Capítulo 3: Implementación y Validación del Sistema:** Se analizan los procesos referentes al desarrollo de la solución y las pruebas realizadas para validar el correcto funcionamiento del sistema de inventario.

Finalmente, se presentan las Conclusiones, Recomendaciones, Referencias Bibliográficas y Anexos derivados de la investigación.

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO.

En el presente capítulo se brinda información acerca de lo referente a los sistemas de inventarios, concepto y funciones del mismo. Se realizan comparaciones entre los sistemas de gestión de inventarios que existen a nivel nacional e internacional. También se describen las herramientas, tecnologías y la metodología que se utilizará para el proceso de desarrollo de software.

1.1 Conceptos básicos asociados al dominio de la investigación.

A continuación, se exponen los principales conceptos relacionados con el marco teórico de la investigación, con el objetivo de profundizar en los distintos puntos de vista y asumir una posición al respecto.

Inventario:

Documento donde se registran todos los bienes tangibles y en existencia de una empresa, que pueden utilizarse para su alquiler, uso, transformación, consumo o venta. Debe brindar una relación detallada en la que se incluyan, además de los tangibles, los derechos y deudas de una empresa. (GONZÁLEZ, 2023)

Sistema de inventario:

Conjunto de métodos y procedimientos que se utilizan para planificar y controlar los artículos que se utilizan una empresa (GONZÁLEZ, 2023)

Artículos:

Serie de objetos y herramientas que se utilizan para la realización de tareas, dentro de las empresas. (GONZÁLEZ, 2023)

1.2 Investigación de los principales sistemas homólogos.

Los inventarios, hasta principios del siglo pasado, fueron analizados con criterio estrictamente contable, es decir, únicamente utilizando registros de entradas y salidas, cuyo objetivo principal era estar informado sobre fugas o pérdidas por malos manejos. Al aplicar nuevas técnicas de las finanzas, la administración de este rubro fue cobrando importancia en el análisis económico de las empresas, debido a que se experimentó una acelerada expansión de la industria norteamericana como consecuencia de la baja en la producción en la industria europea en el período de 1914 a 1918. (BAUS, 2019)

1990 - La creación del primer software de gestión de inventarios

Con la llegada de la era digital, se crearon los primeros sistemas de gestión de inventarios basados en software. Estos sistemas permiten una gestión más eficiente y automatizada de los inventarios en las empresas.

Actualidad - La gestión de inventarios en la era digital.

En la era digital, la gestión de inventarios ha evolucionado significativamente gracias a la tecnología. Actualmente, existen sistemas de gestión de inventarios en la nube, aplicaciones móviles y herramientas de análisis de datos que permiten una gestión mucho más eficiente de los inventarios en las empresas. (AYALA BARRERA, 2021)

Debido a la importancia que se le confiere a la información, hoy existen varias herramientas creadas con el objetivo de apoyar el proceso de gestión de la misma. Es por ello que se decide hacer un análisis de varios sistemas desarrollados a nivel nacional e internacional con el objetivo de determinar las principales características y funcionalidades presentes en estos sistemas similares. Así como sus ventajas y desventajas que puedan favorecer el desarrollo de la investigación.

A continuación, se muestran algunas de ellas:

1.2.1 Sistemas de gestión en el ámbito internacional.

SAP: es uno de los principales fabricantes de software para la gestión de procesos empresariales a nivel mundial, gracias al desarrollo de soluciones que facilitan el procesamiento eficaz de los datos y el flujo de información en las organizaciones. La empresa fundada en 1972, desarrolla soluciones de software que son utilizadas por pequeñas, medianas y grandes empresas. Con las aplicaciones estándar, las soluciones sectoriales, las plataformas y las tecnologías, es posible mapear y diseñar cada proceso empresarial. El software recoge y procesa datos en una misma plataforma, que van desde la compra de materias primas hasta la producción y la satisfacción del cliente.

Las autoras determinan que SAP no es viable por el alto costo de comprar e implementar el programa, ya que la empresa debe comprar el software y hardware necesarios para ejecutar los programas en toda la empresa. Otra desventaja de SAP es la complejidad con la que está construido el programa.

Holded: Es un ERP en la nube (software para la gestión de los recursos empresariales, por sus siglas en inglés). Fue diseñado por desarrolladores españoles en 2015 y desde entonces no ha dejado de crecer. En la actualidad más

de 20.000 empresas lo usan como la solución a su gestión empresarial. Es una solución completa para la gestión empresarial, ya sea para autónomos o pymes. Permite llevar la contabilidad, facturación, la gestión de clientes y de proyectos, además de la gestión de recursos humanos e inventario. Permite sincronizarse con las aplicaciones, entre ellas: Amazon, Google Drive, PayPal, Dropbox, Shopify, entre otras.

Se determinó que su adopción no es factible ya que cuenta con una versión de prueba gratuita de 14 días, suficientes para poder probar sus funcionalidades desde el principio, para luego adquirir el plan que mejor se adapte a nuestro negocio.

1.2.2 Sistema de gestión en el ámbito nacional

Versat-Sarasola: Es el primer sistema integral de gestión de contabilidad certificado, desarrollado en 2001, para la gestión económica. Actualmente es utilizado en Cuba en alrededor de 200 entidades de varias provincias. Este sistema integrado cuenta con un conjunto de 12 módulos entre los que se encuentran: configuración y seguridad, contabilidad general y de gastos, costos y procesos, análisis económico empresarial, control de activos fijos, finanzas y caja, planificación y presupuestos, control de inventarios, pago de salario, paquete de gestión, contratación, facturación. (ROMERO, 2019).

1.2.3 Resultados de investigación de los sistemas homólogos.

Ambos softwares internacionales, cumplen con los requisitos y expectativas del cliente, pero su adopción no es factible ya que presentan altos grados de tecnologías y necesitan de altos costos de soporte y mantenimiento, ya que utilizan como base tecnológica el uso de la nube y herramientas con licencias privativas teniendo que efectuar pagos mensuales de altos costos; por lo tanto, no presentan una completa solución para la empresa CONEC.

El sistema nacional necesita de grandes prestaciones y de integraciones entre módulos lo cual no es del todo aplicable, pues es una empresa que no cuenta con la tecnología y los equipamientos necesarios para su funcionamiento y se maneja información mucho más complicada de la que en realidad se necesita.

Por todo lo antes expuesto se concluye, que se hace necesario crear un sistema de inventario que permita el control de los artículos en los almacenes internos en la empresa, que sea capaz de buscar los artículos que se encuentran almacenados, además el sistema debe ser consecuente con la filosofía de conseguir herramientas

y tecnologías libres.

1.3 Descripción de los lenguajes, herramientas y tecnologías utilizadas

A continuación, se describen la metodología de software que guía el proceso de desarrollo de software, los lenguajes de desarrollo que se utilizaron en esta investigación, y las herramientas a utilizar.

1.3.1 Metodología de desarrollo del Software

La metodología no es más que un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software. Compuesta por tareas (actividades en que se dividen los procesos), procedimientos (define la forma de ejecutar la tarea), técnica (herramienta utilizada para aplicar un procedimiento), herramientas (herramientas de software que automatizan la aplicación) y el producto (resultado de cada etapa). (LOPEZ,2021).

1.3.2 Metodología a elegir

SCRUM es una de las más famosas dentro de las metodologías ágiles. Aparece alrededor del año 1986, y fue creado por Ikujiro Nonaka e Hirotaka Takeuchi, a partir de un estudio que realizaron a distintas empresas que estaban viendo un nuevo enfoque de trabajo. Más tarde, en 1995, crearon un conjunto de reglas, o conjunto de buenas prácticas, enfocadas al desarrollo de software y la bautizaron con el nombre de Scrum.

Centra todo en el desarrollo de un conjunto tareas y procedimientos dirigidos a la gestión de proyectos. El conjunto de buenas prácticas de Scrum se aplica, esencialmente, a la gestión del proyecto. El objetivo es obtener resultados rápidos con adaptación a los cambios de las necesidades de los clientes. (REQUENA MESA,2018).

El desarrollo de software mediante iteraciones incrementales y las reuniones a lo largo del proyecto, es lo que la convierte en una metodología, ya que el cliente tiene gran participación a lo largo del proyecto; razón por la cual se decide aplicar esta metodología en la empresa CONEC.

A continuación, se muestra una figura que describe las fases de la metodología SCRUM.



Ilustración 1- Fases de SCRUM. Fuente: creación propia.

Ver **anexo 1**, para fundamentación de la metodología elegida.

Artefactos generados

Se conoce como “Artefacto” o “Entregable” todos aquellos elementos que garantizan la transparencia y el registro de la información clave de un proceso. Es decir, son los recursos que sientan las bases para la calidad y la productividad de cualquier proyecto. (BARÁ, 2018)

Los principales artefactos del scrum ágil son:

- Producto Backlog: Es el listado de requisitos funcionales y no funcionales, y la descripción de los mismos.
- Sprint: Corresponde a una ventana de tiempo donde se crea una versión utilizable del producto (incremento). Cada Sprint, como en el rugby¹, es considerado como un proyecto independiente. Su duración máxima es de un mes.
- Sprint Backlog: Es la planificación que se realiza para saber qué requisitos se implementan en cada sprint.

1.3.3 Fundamentación de la Metodología elegida

A través del estudio de esta metodología en el transcurso escolar, y estudiada la

¹

el melé o scrum, es una de las jugadas más relevantes en el rugby. De ahí proviene el nombre de dicha metodología.

bibliografía referenciada, se logró arribar a las siguientes ventajas:

- Permite afrontar mejor los proyectos en relación con el tiempo y recursos a emplear.
- Permite fijar una fecha de entrega más precisa y realista al disponer de mayor control sobre todo el proceso.
- Se reducen los tiempos de entrega y se incrementa la calidad del producto o servicio ofrecido.
- Se mejora la satisfacción del cliente, que pasa de un agente externo estático, a participar en el proyecto.
- Crea un mejor clima de trabajo donde se abordan los problemas e incidencias desde una perspectiva positiva, buscando soluciones en lugar de culpables.
- Aumenta la productividad y competitividad de la empresa.

1.3.5 Lenguaje Unificado de Modelado (versión 2.5.1)

El lenguaje de Modelamiento Unificado o UML por sus siglas en inglés, es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Permite, según el sitio oficial, especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos. Estimula el crecimiento de herramientas orientadas a objetos. Integra las mejores prácticas para la modelación y puede soportar todos los lenguajes de programación.

De forma general las principales características son:

- Lenguaje unificado para la modelación de sistemas
- Tecnología orientada a objetos
- El cliente participa en todas las etapas del proyecto
- Corrección de errores viables en todas las etapas

1.3.6 Herramientas de Modelado

Visual Paradigm (versión 16.3): Es una herramienta CASE, utilizada en un ambiente de software libre, debido a la posibilidad de ejecutarse sobre cualquier sistema operativo, por lo que se convierte en una herramienta multiplataforma, permite crear diferentes tipos de diagramas en un ambiente totalmente visual. Soporta además el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite graficar

todos los tipos de diagramas de clases, generar documentación, generar código desde diagramas y código inverso.

Beneficios:

- Navegación intuitiva entre el modelo visual y el código
- Poderosa herramienta de generación de PDF/HTML a partir de diagramas UML
- Sincronización entre el código fuente y el modelo en tiempo real o bajo demanda
- Entorno visual de modelado superior
- Soporte para toda la notación UML
- Análisis de texto sofisticados y automáticos diagramas de capas
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación
- Capacidades de ingeniería directa (versión profesional) e inversa
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
- Disponibilidad en múltiples plataformas

1.3.7 Sistema Gestor de Base de Datos

El propósito general de un sistema gestor de bases de datos es manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para un buen manejo de datos.

Un sistema de gestión de bases de datos relacionales (RDBMS) es una colección de programas y capacidades que permiten a los equipos de TI y a otros crear, actualizar, administrar e interactuar con una base de datos relacional. (VELAZCO,2021)

SQLite3 nació como un motor de bases de datos muy ligero, de código abierto y escrito en C, donde podremos guardar todo tipo de información relacionada con un programa o una app. Su principal ventaja es que, a diferencia de lo que ocurre con otros motores de base de datos, SQLite funciona como un servidor propio e independiente, evitando tener que realizar consultas externas en procesos separados. O lo que es lo mismo, se incluye la base de datos y el motor dentro del programa y los datos se consultan (o guardan) desde él mismo, eliminando la

necesidad de tener otros servicios abiertos en segundo plano.

SQLiteStudio (versión 3.4.1): es un interfaz gráfico para bases de datos SQLite, potente, ligero, rápido e intuitivo. Se trata de una aplicación multiplataforma, disponible para los principales sistemas operativos *Windows*, *MacOSX* y *Linux*.

Entre las diferentes características que podemos destacar de SQLiteStudio se encuentran las siguientes:

- No hay que instalarlo, tan solo desempaquetar y ejecutar.
- Podemos exportar a diferentes formatos. Desde instrucciones SQL a CSV, HTML, XML, PDF o JSON.
- Está soportado Unicode.
- Se puede configurar el aspecto, colores, fuentes y atajos.
- Es de código abierto y gratuito.
- Permite el uso de complementos.

Existen diferentes complementos, algunos que vienen por defecto, y otros que se pueden instalar. De la misma forma, existen tanto complementos oficiales, como desarrollados por terceros.

1.3.8 Lenguaje de Programación

PHP (versión 7.4) acrónimo de Hypertext Preprocessor, es un lenguaje de secuencias de comandos del lado del servidor creado en 1995 por Rasmus Lerdorf. Según su página oficial, es un lenguaje de programación muy usado en el desarrollo de sitios web, es un lenguaje de código abierto que ofrece las siguientes ventajas:

1. Lenguaje totalmente libre y abierto.
2. Posee una curva de aprendizaje muy baja.
3. Los entornos de desarrollo son de rápida y fácil configuración.
4. Fácil acceso e integración con las bases de datos.
5. Posee una comunidad muy grande.
6. Es el lenguaje con mayor usabilidad en el mundo.
7. Es un lenguaje multiplataforma.
8. Completamente orientado al desarrollo de aplicaciones web dinámicas y/o páginas web con acceso a una Base de Datos.
9. El código escrito en PHP es invisible al navegador ya que se ejecuta al lado del servidor y los resultados en el navegador es HTML.

10. Posee una versatilidad para la conexión con la mayoría de base de datos que existen en la actualidad.

Conclusiones del capítulo

Después de realizado el estudio de la bibliografía relacionada con los sistemas de inventario, se identificó que las soluciones nacionales e internacionales, no eran factibles ya que no satisfacen las necesidades de la empresa CONEC. En consecuencia, las autoras proponen la implementación de un sistema que sea capaz de ajustarse a los requerimientos de la empresa

Analizadas las tendencias actuales en el desarrollo del software, se seleccionó la metodología de desarrollo una vez aplicado el método de Boehm-Turner, y, por consiguiente, se decide utilizar una metodología ágil según las características marcadas en la investigación. Estudiado un grupo de tecnologías y herramientas, se definió: como metodología de desarrollo, la metodología ágil SCRUM, como sistemas de gestión de base de datos SQL Server 2017 y SQLite 3 para el almacenamiento de la información, como herramienta para el modelado se seleccionó Visual Paradigm for UML 8.0, PHP 7.4 como lenguaje de programación.

CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO.

En el presente capítulo se aborda una descripción a la situación problemática planteada anteriormente. En lo adelante, se presenta el modelado del dominio, se especifican y se describen los requisitos funcionales y no funcionales incluyendo una muestra del prototipo de interfaz de usuario, se presenta el diagrama de clases. Finalmente, se realiza el modelo de datos de la aplicación para manejar los datos de cada una de las entidades y sus relaciones entre ellas.

2.1 Modelado del dominio.

Un modelo conceptual representa la estructura y dinámica de la organización, en este caso se representarán los principales conceptos, los tipos más importantes de objetos en el contexto del negocio, los cuales representan lo que existe y los eventos que suceden en el entorno de trabajo del sistema. (Pressman, 2010).

El principal objetivo del Modelo Conceptual es explicar a los clientes, los conceptos significativos en un dominio del problema. Según (Larman,2004), es el artefacto más importante a crear durante el análisis orientado a objetos, concepto por el cual se consideró representar el Modelo Conceptual.

A continuación, se muestra el Modelo Conceptual para el sistema de control de inventario para CONEC:

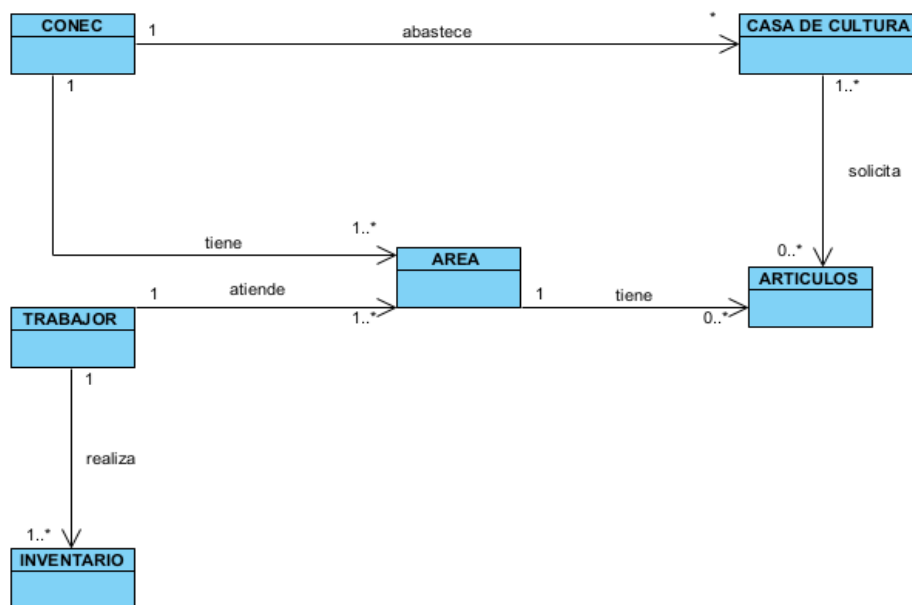


Ilustración 2- Mapa Conceptual. Fuente: Elaboración Propia.

2.2 Descripción de conceptos asociados al Modelado del dominio.

A continuación, se presenta la descripción de los conceptos modelados en el modelo conceptual.

CONEC: Consultoría Nacional Económica para la Cultura, es el centro de trabajo para el cual se realiza el sistema de inventario.

Casa de Cultura: Centro comunitario destinado a mantener actividades que promueven la cultura, es quién solicita las materias primas que pueda necesitar al centro.

Área: es el lugar físico donde se almacenan los artículos que existen en el centro.

Trabajador: es el empleado que trabaja en el centro, el cual está al frente del almacén y tiene la tarea de realizar el inventario.

Artículos: son todos los elementos que existen en el centro.

Inventario: es el archivo donde se registran todos los artículos que entran y se guardan en el almacén.

2.3 Product Backlog

Requisitos Funcionales

Un requisito funcional es una declaración de cómo debe comportarse un sistema. Define lo que el sistema debe hacer para satisfacer las necesidades o expectativas del usuario.

El objetivo del análisis orientado a objetos es desarrollar una serie de modelos que describan el software de la computadora que satisface los requisitos exigidos por el cliente. (Pressman, 2010)

A partir de las entrevistas realizadas a trabajadores y directivos del centro, se pudo definir los requisitos funcionales que se muestran en la tabla a continuación.

Tabla 1- Requisitos Funcionales. Fuente: Creación Propia

RF.1 Gestionar Trabajador		
No.	Nombre	Descripción
RF 1.1	Autenticar trabajador.	El sistema debe permitir autenticar a un trabajador.
RF 1.2	Desconectar Trabajador.	El sistema debe permitir desconectar a un trabajador.
RF.2 Gestionar Artículo		

RF 2.1	Insertar artículo	El sistema debe permitir insertar un nuevo artículo.
RF 2.2	Editar artículo	El sistema debe permitir editar un artículo.
RF 2.3	Eliminar artículo	El sistema debe permitir eliminar un artículo.
RF 2.4	Visualizar artículo	El sistema debe permitir visualizar la información de un artículo.
RF 2.5	Buscar artículo	El sistema debe permitir buscar un artículo.
RF 2.6	Insertar código de artículo	El sistema debe permitir insertar el código de un artículo.
RF.3 Gestionar Inventario		
RF.3.1	Insertar Inventario	El sistema debe permitir insertar un nuevo inventario.
RF.3.2	Editar Inventario	El sistema debe permitir editar un nuevo inventario.
RF.3.3	Eliminar Inventario	El sistema debe permitir eliminar un inventario.
RF.3.4	Visualizar Inventario	El sistema debe permitir visualizar la información de un inventario.
RF.3.5	Buscar en Inventario	El sistema debe permitir buscar un inventario.
RF.3.6	Visualizar Fecha de Inventario	El sistema debe permitir visualizar la fecha de un inventario.
RF.3.7	Insertar Nota de un Inventario	El sistema debe permitir insertar una nota referente al inventario
RF.4 Gestionar Área		
RF.4.1	Adicionar área	El sistema debe permitir insertar una nueva área.
RF.4.2	Actualizar área	El sistema debe permitir actualizar un área.
RF.4.3	Visualizar Área	El sistema debe permitir visualizar la información que contiene un área.
RF.4.4	Eliminar área	El sistema debe permitir elimi-

		nar un área.
RF.4.5	Buscar área	El sistema debe permitir buscar un área.
RF.4.6	Insertar Nota de área	El sistema debe permitir insertar una nota referente a un área.
RF.5 Gestionar Bajas		
FR.5.1	Insertar nueva baja	El sistema debe permitir insertar una nueva baja.
RF.5.2	Visualizar baja	El sistema debe permitir visualizar la información de una baja.

Requisitos No Funcionales

Los requisitos no funcionales explican las limitaciones y restricciones del sistema a diseñar. Estos requisitos no tienen ningún impacto en la funcionalidad de la aplicación. Si los requisitos funcionales especifican lo que debe hacer un sistema, los requisitos no funcionales describen cómo lo hará. En otras palabras, se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema (características de usuario).

Los requisitos no funcionales (RNF), son requisitos que imponen restricciones en el diseño o la implementación como restricciones en el diseño o estándares de calidad. Son propiedades o cualidades que el producto debe tener (PRESSMAN,2010).

A continuación, mencionamos los principales Requisitos No Funcionales obtenidos:

- Apariencia e interfaz externas: El sistema contará con un diseño que muestra en una sola interfaz, el contenido de la sección que el usuario se encuentra utilizando en el momento, evitando las barras de desplazamiento, tanto verticales como horizontales y el uso de un menú claro en todo momento para facilitar el acceso a las distintas funcionalidades del sistema.
- Soporte: Se requiere que el sistema cuente con la documentación necesaria para permitir agilizar la gestión y administración de los procesos que automatizarán el sistema en las auditorías informáticas. Se brindarán cursos de capacitación al personal que utilizará el software en caso de ser necesario.

- Usabilidad: La navegación por la aplicación no debe tener una profundidad de más de 4 niveles, lo cual garantizará que toda la información este accesible para el usuario de forma más rápida.
- Seguridad: La seguridad del sistema se debe garantizar en todo momento, la cual debe velar por tres aspectos fundamentales:
 1. **Confidencialidad**: La información manejada en el sistema estará protegida de acceso no autorizado y cada usuario con acceso al sistema sólo debe acceder a los datos que le correspondan por su rol y cuenta registrada
 2. **Integridad**: El acceso a las funcionalidades estará determinado por los permisos establecidos por el administrador según el tipo de usuario
 3. **Disponibilidad**: Al personal autorizado se le garantizará el acceso al sistema mediante una cuenta de usuario con los privilegios necesarios para poder operar y consultar los datos deseados en el momento establecido.

2.4 Modelado de los requisitos

Según la metodología SCRUM, el modelado de requisitos en las metodologías ágiles, generalmente se representan mediante historias de usuario. Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo proporcionará una función de software valor al cliente. (VERGARA,2021)

Una de las principales ventajas de usar esta forma de modelado de requisitos es centran la atención en el usuario, ya que lista de tareas pendientes mantiene al equipo centrado en tareas que deben completarse, pero un conjunto de historias lo mantiene centrado en solucionar problemas para usuarios reales. (VERGARA,2021) por lo cual, se considera fundamental su representación para la empresa CONEC. A continuación, en las tablas 2, 3 y 4 se muestran una parte de las historias de usuarios para Gestionar Artículo.

Tabla 2- Historia de Usuario Insertar Artículo. Fuente: creación Propia.

Historia de Usuario	
No: 3	Usuario: Trabajador
Nombre de la historia: Insertar Artículo	
Prioridad: alta	Riesgo en desarrollo: alto

Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Chabely Bressler y Yiliany Pacheco	
Descripción: El trabajador desea insertar un artículo.	
Validación: El trabajador puede insertar el artículo.	

Tabla 3- Historia de Usuario Editar Artículo. Fuente: creación Propia.

Historia de Usuario	
No: 4	Usuario: Trabajador
Nombre de la historia: Editar Artículo	
Prioridad: alta	Riesgo en desarrollo: alto
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Chabely Bressler y Yiliany	
Descripción: El sistema permite, luego de insertado un artículo, que el trabajador edite dicho artículo.	
Validación: El sistema permite editar un artículo.	

Tabla 4-Historia de Usuario Buscar Artículo. Fuente: creación Propia.

Historia de Usuario	
No: 5	Usuario: Trabajador.
Nombre de la historia: Buscar Artículo.	
Prioridad: alta.	Riesgo en desarrollo: alto.
Puntos estimados: 2.	Iteración asignada: 1.
Programador responsable: Chabely Bressler y Yiliany Pacheco.	
Descripción: El trabajador desea buscar un artículo en existencia en el almacén.	
Validación: El sistema permite buscar un artículo.	

2.5 Diseño de la interfaz de usuario.

El diseño de la interfaz de usuario crea un medio eficaz de comunicación entre los seres humanos y la computadora. Siguiendo un conjunto de principios de diseño de la interfaz, el diseño identifica los objetos y acciones de ésta y luego crea una plantilla de pantalla que constituye la base del prototipo de la interfaz de usuario (PRESSMAN, 2010).

A continuación, se muestra el prototipo de interfaz de usuario:

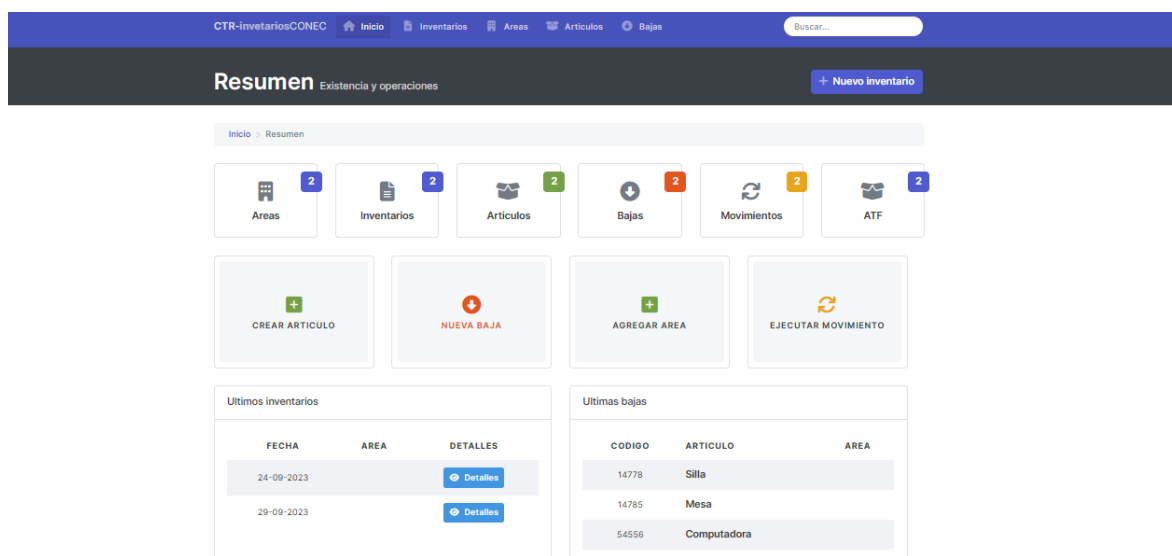


Ilustración 3-Prototipo de interfaz de usuario (Página de Inicio). Fuente: Elaboración propia.

El prototipo de interfaz de usuario para 'Nuevo artículo' es una ventana con un título 'Nueva artículo' y un botón de cerrar 'X'. Contiene los siguientes campos de entrada:

- Area:
- Artículo:
- Código:
- Precio:
- Estado:
- Tipo:
- Descripción:

En la parte inferior derecha hay dos botones: 'Cancelar' (gris) y 'Aceptar' (azul).

Ilustración 4- Prototipo de interfaz de usuario (Nuevo Artículo). Fuente: Elaboración propia.

2.5.1 Mapa de Navegabilidad.

En el presente epígrafe, se modela el mapa de navegabilidad del sistema y se describen sus elementos. Proporcionan una representación esquemática de la estructura del hipertexto, indicando los principales conceptos incluidos en el espacio de la información y las interrelaciones que existen entre ellos. La organización jerárquica de los vínculos ofrece buena orientación a los usuarios pues la jerarquía es una forma de organización elemental.

En la siguiente figura, se muestra el mapa de navegabilidad para dicha interfaz de usuario:

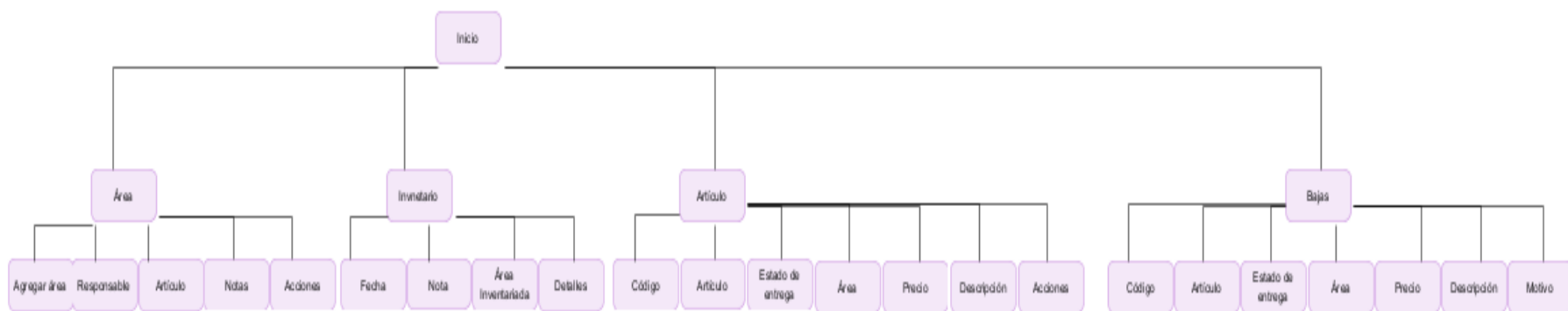


Ilustración 5- Mapa de navegabilidad. Fuente: Elaboración propia.

2.6 Arquitectura de Software

Para llevar a cabo el desarrollo de este sistema, se ha utilizado una arquitectura Modelo-Vista-Controlador (MVC) ya que es un patrón en el diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Esta modelo consta de una separación de la lógica de negocios y la visualización que tiene esta con el usuario.

Esta arquitectura se divide en tres partes, como se describe en la siguiente manera:

- Modelo (Maneja datos y lógica de negocios): El modelo define qué datos debe contener la aplicación. Si el estado de estos datos cambia, el modelo generalmente notificará a la vista (para que la pantalla pueda cambiar según sea necesario) y, a veces, el controlador (si se necesita una lógica diferente para controlar la vista actualizada)
- Vista (Se encarga del diseño y presentación): La vista define cómo se deben mostrar los datos de la aplicación.
- Controlador (Enruta comandos a los modelos y vistas): El controlador contiene una lógica que actualiza el modelo y / o vista en respuesta a las entradas de los usuarios de la aplicación.

La figura a continuación describe el comportamiento de la arquitectura seleccionada.

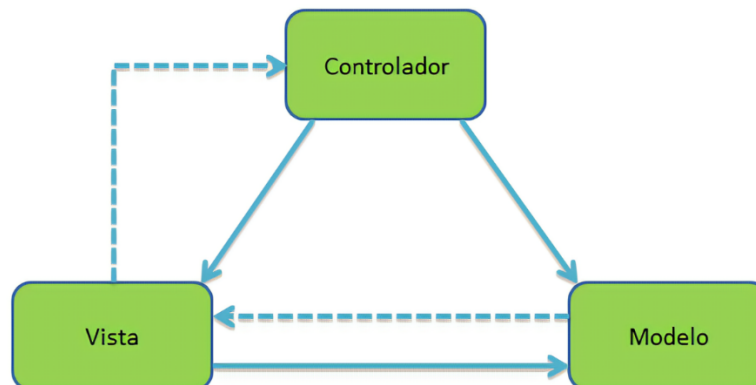


Ilustración 6- Arquitectura MVC. Fuente: creación Propia.

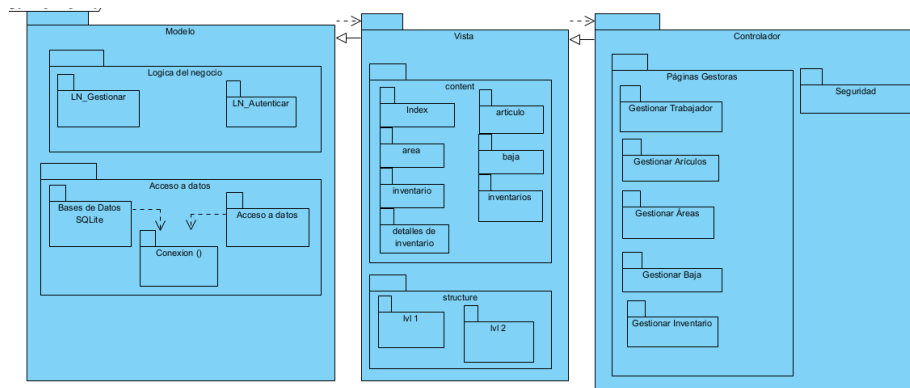


Ilustración 7- Representación Arquitectónica MVC. Fuente: creación Propia

2.6 Diagrama de Clases del Diseño y Patrones

2.6.1 Diagrama de Clases de Diseño

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. El lenguaje utilizado para especificar una clase del diseño es el mismo que el lenguaje de programación utilizado, los métodos tienen correspondencia directa con el debido método de la implementación de clases (Larman, 2004).

A continuación, se muestra el diagrama de clases del diseño para Gestionar Artículo

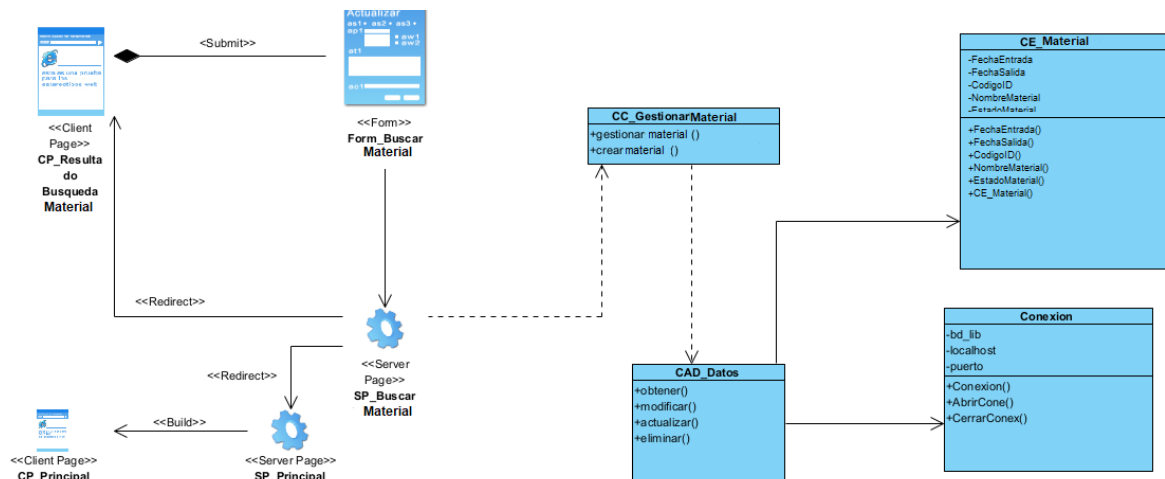


Ilustración 8- Diagrama de clases de diseño Gestionar Artículo. Fuente: creación propia.

2.6.2 Patrones de Diseño

En este epígrafe, se presentan los patrones de diseño utilizados: GRASP. Los patrones de diseño constituyen una descripción de un problema y la solución, la

misma recibe un nombre y se puede aplicar a varios contextos. Muchos patrones proporcionan guías en el modo en que deberían asignarse las responsabilidades a los objetos, dada una categoría específica del problema (VARÓN, 2018)

Patrones GRASP.

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Controlador: Este patrón queda evidenciado en el sistema haciendo uso de las clases controladoras para atender las peticiones realizadas por el usuario. Cada una de ellas se encarga de procesar la petición de su entidad correspondiente de forma independiente.

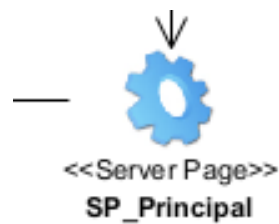


Ilustración 9- Patrón Controlador. Fuente: Elaboración propia.

Bajo acoplamiento: Determina el nivel de dependencia de una clase con respecto a otras. Su uso potencia la reutilización, el mantenimiento y la mitigación de efectos a producirse en una clase al hacer cambios en otra.

En la presente solución se pone de ejemplo la existencia de una baja dependencia en las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o la plantilla.

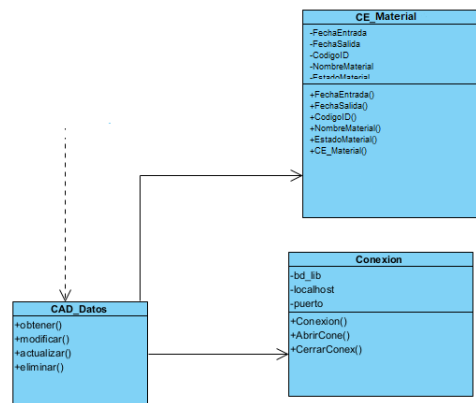


Ilustración 10- Patrón Bajo Acoplamiento. Fuente: Elaboración propia.

Alta cohesión: Una alta cohesión caracteriza a las clases con responsabilidades estrecha-mente relacionadas que no realicen un trabajo enorme. Significa que las clases del sistema tienen asignadas solo las responsabilidades que les corresponde y mantienen una estrecha relación con el resto de las clases.

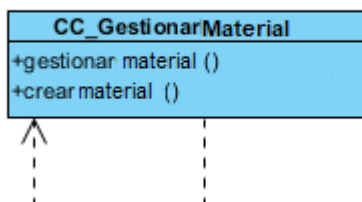


Ilustración 11- Patrón Alta Cohesión. Fuente: Elaboración propia.

2.7 Diagrama de Despliegue

El diagrama de despliegue es un diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista del despliegue (distribución) de los artefactos del software en los destinos de despliegue, donde se encuentra generalmente representado por un nodo que es o bien de los dispositivos de hardware o bien algún entorno de ejecución de software. Los nodos pueden ser conectados a través de vías de comunicación para crear sistemas en red de complejidad arbitraria. (Sarmiento, 2013).

A continuación, se muestra el diagrama de despliegue para el Sistema de inventario del centro CONEC

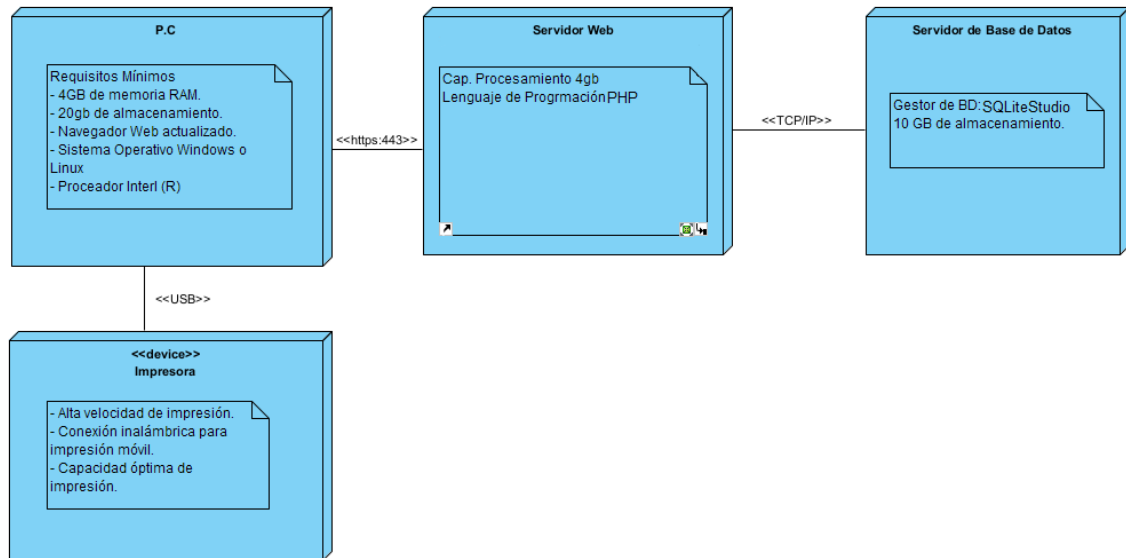


Ilustración 12- Diagrama de Despliegue. Fuente: creación Propia.

2.7 Modelo Entidad-Relación

El modelado de datos entidad-relación usa un diagrama formal para representar las relaciones entre entidades en una base de datos. El propósito fundamental es representar objetos de datos y sus relaciones. (Pressman, 2010).

A continuación, se presenta el Modelo Entidad-Relación para el centro CONEC.

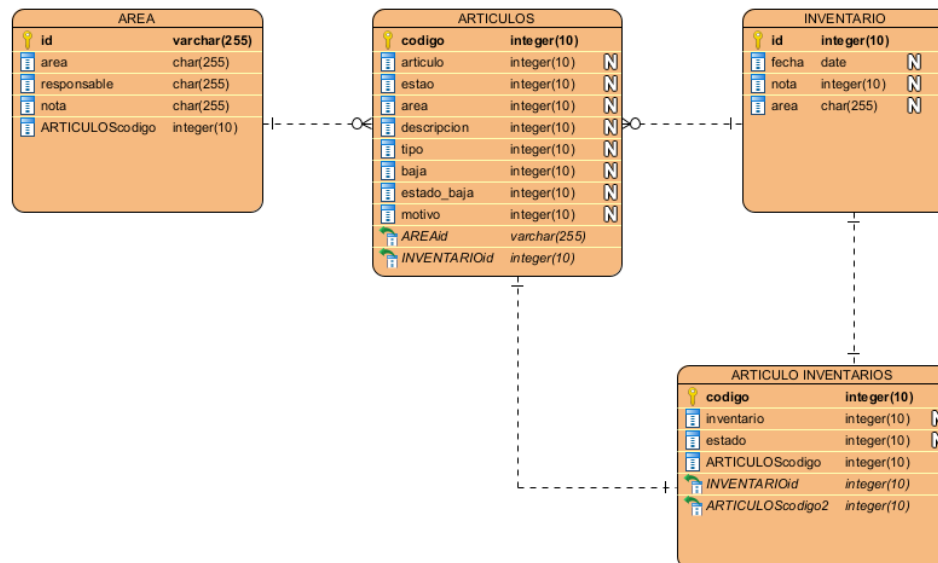


Ilustración 13- Modelo Entidad-Relación Base de Datos CONEC. Fuente: Creación propia.

Conclusiones del capítulo

La utilización de un modelo de negocio bien definido y la generación de los artefactos del mismo, permitieron conocer el funcionamiento de todos los procesos que se llevan a cabo en la empresa CONEC.

La realización del análisis y diseño del sistema a implementar permitió obtener una representación visual de las principales relaciones entre los conceptos asociados a la problemática a través del modelo de dominio.

Se identificaron 23 requisitos funcionales y 4 requisitos no funcionales, que le aportan cualidades significativas al producto en cuanto a rendimiento, disponibilidad, seguridad, usabilidad, interfaz. Para estructurar el sistema se utilizó el patrón arquitectónico MVC.

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE INVENTARIO PARA LA EMPRESA CONEC

En el presente capítulo se describen las actividades relacionadas con los procesos de implementación (la planificación del sprint, como dicta la metodología SCRUM). Se realizan las pruebas al sistema web para validar su correcto funcionamiento, seguridad, así como también la satisfacción del cliente con cada incremento del producto de software que este reciba.

3.1 Implementación del sistema

La fase de implementación constituye una parte transcendental en el proceso de desarrollo de un software, es en este momento donde se define y organiza el código de la propuesta de solución. Durante esta etapa se materializan, en forma de código, todos los artefactos de implementación, descripciones y arquitectura propuestos en la fase de análisis y diseño; que permite conformar el producto final requerido por el cliente.

3.1.2 Spring Backlog

De acuerdo con los artefactos generados por la metodología seleccionada, mencionados anteriormente en el epígrafe 1.3.2, un Spring Backlog es la lista de tareas que elabora el equipo durante la planificación de un sprint. Se asignan las tareas a cada persona y el tiempo que debe realizar dicha tarea. De esta forma el proyecto queda descompuesto en unidades más pequeñas pudiendo determinarse en que tareas se ha avanzado. Pueden existir dependencias entre tareas, es importante diferenciarlas. Todas las tareas deben tener un coste semejante de hasta 15 días.

A continuación, se muestran las tablas correspondientes a los Sprint 0 y 1 respectivamente:

Tabla 5- Planificación del Sprint 0. Fuente: Elaboración propia.

ID	Tarea	Responsable	Estado	Fecha inicial	Fecha final
-----------	--------------	--------------------	---------------	----------------------	--------------------

T1	Montaje del ambiente de desarrollo con los marcos de trabajo seleccionados para el desarrollo.	Chabely Bressler	Resuelta	17/09/2023	18/09/2023
RF2.1	Implementar funcionalidad: Insertar Artículo.	Chabely Bressler	Resuelta	17/09/2023	21/10/2023
RF2.5	Implementar funcionalidad: Buscar artículo.	Yiliany Pacheco	Resuelta	17/09/2023	22/10/2023
RF2.2	Implementar funcionalidad: Editar Artículo.	Chabely Bressler	Resuelta	17/09/2023	22/10/2023
RF2.3.	Implementar funcionalidad: Eliminar artículo.	Yiliany Pacheco	Resuelta	17/09/2023	20/10/2023

Tabla 6- Planificación del Sprint 1. Fuente: Elaboración propia.

ID	Tarea	Responsable	Estado	Fecha inicial	Fecha final
RF4.3	Implementar funcionalidad: Visualizar área en existencia	Chabely Bressler	Resuelta	17/09/2023	23/10/2023
RF4.4	Implementar funcionalidad: eliminar área.	Yiliany Pacheco	Resuelta	19/09/2023	24/10/2023
RF4.6	Implementar funcionalidad: Insertar nota de área.	Chabely Bressler	Resuelta	20/09/2023	25/10/2023

3.1.2 Resultados de la revisión del Sprint.

Los resultados obtenidos con cada iteración del Sprint, arrojaron los siguientes resultados:

- Se obtuvieron un total de 11 no conformidades, las cuales se distribuyen en 6 en la primera tapa, 3 en la segunda, en la tercera iteración se obtuvieron 2 no conformidades.
- En la primera iteración se resolvieron 4/6 no conformidades. En la segunda iteración se resolvieron las (2) no conformidades pendientes y 2/3 no conformidades correspondientes a esta iteración. Por último, se logran resolver (1) no conformidad pendiente, y las (2) correspondientes a la iteración en curso.

El siguiente gráfico de barras representa los resultados obtenidos:

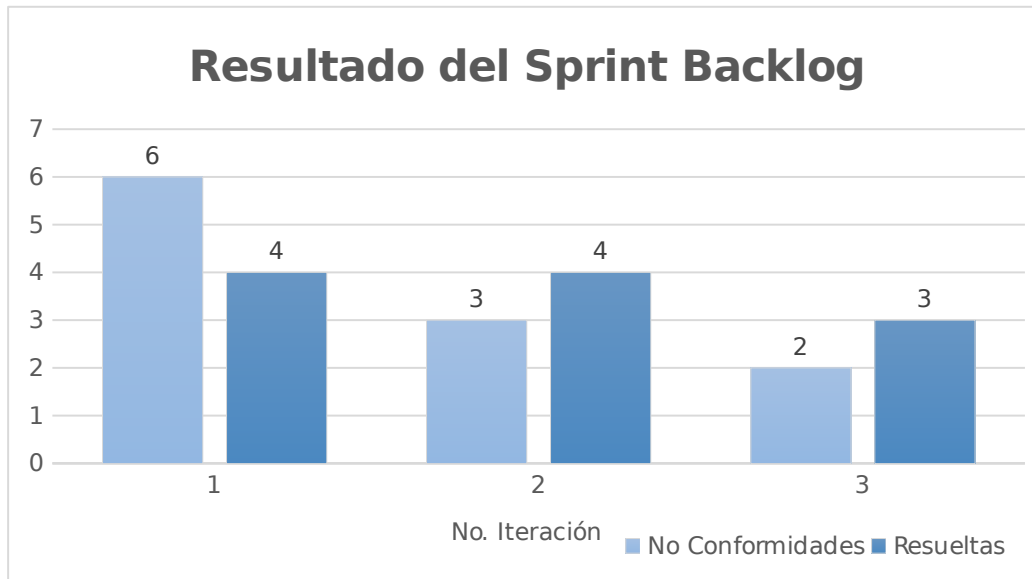


Ilustración 14- Resultados del Sprint Backlog. Fuente: Creación Propia.

3.2. Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación del código, 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.

PHP sigue pocas reglas y mantiene su estilo de codificación. Como hay muchos desarrolladores en todo el mundo, si cada uno de ellos sigue diferentes estilos y estándares de codificación, esto generará una gran confusión y dificultad para que un desarrollador entienda el código de otro desarrollador. Será muy difícil administrar y almacenar el código para futuras referencias. (GREYRAT,2022).

- Comentarios: Se recomienda el uso del estilo de comentarios estándar de C y C++, es decir, (*//*) para una sola línea y (*/* */*) para líneas múltiples, y el uso de comentarios de estilo Python o Perl, es decir, (*#*), es desanimado.

```
//Delete area
if (!empty($_POST["elimina
```

Ilustración 15- Fragmento de código utilizado. Fuente: Creación propia

- Longitud de línea y sangría: es una recomendación estándar no exceder más de 75-85 caracteres por línea de código. No se deben usar tabulaciones para la sangría, sino espacios, ya que es el método de sangría estándar en la

mayoría de los lenguajes de programación. Una declaración por línea o líneas divididas para largas múltiples.

- Estructuras de control: el flujo de control o las declaraciones condicionales deben escribirse de tal manera que puedan diferenciarse de las declaraciones de llamada de función. Al escribir 'if, for, else, foreach, do, while, switch' y otras declaraciones de flujo de controles, debe haber un espacio entre la palabra clave y el paréntesis de apertura. Las definiciones de función no tienen ningún espacio entre el nombre de la función y el paréntesis de apertura.
- Llamadas a funciones: no debe haber espacios entre parámetros, coma, último parámetro y punto y coma. Utilice siempre etiquetas PHP completas y evite las etiquetas abreviadas. Además, dé un espacio entre ambos lados del operador '='.

```
function articulos_area_ids($id_area){  
    $articulos = [];  
    $db = new SQLite3(DATA);  
    $rs = $db->query("select codigo from articulos where area =  
    '$id_area' ");  
    while($row = $rs->fetchArray()){ $articulos[] =  
    $row["codigo"] ; }  
    return $articulos;  
}
```

Ilustración 16- Fragmento del código utilizado. Fuente: Creación Propia

3.3 Pruebas del software

Las pruebas de software son el proceso de ejercitar el software con la intención de encontrar y corregir errores. Estas verifican que el software implemente una función específica de forma correcta y validan que las acciones que realiza respondan a los requisitos del cliente (PRESSMAN, 2005).

3.3.1. Estrategia de pruebas

Una estrategia de pruebas 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 requieren. 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 (PRESSMAN, 2010).

Para la validación del sistema se propone realizar la siguiente estrategia de pruebas:

- **Pruebas funcionales:** Están basadas en ejecución, revisión y retroalimentación de las funcionalidades implementadas para el software y las especificaciones definidas por el usuario. Busca evaluar el sistema mediante modelos de pruebas. (PRESSMAN, 2010).

3.4 Aplicación de Pruebas

Pruebas Funcionales:

En este tipo de pruebas se ejecutan los distintos servicios prestados con datos correctos e incorrectos. En caso de que los datos sean incorrectos se verifica que los mensajes de error sean los deseados y en el caso opuesto que los resultados sean los esperados (PRESS-MAN, 2010).

En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro

Método de Prueba de Caja Negra:

El método de caja negra se aplica a la interfaz del software, examina algún aspecto funcional de un sistema que tiene poca relación con la estructura lógica interna del software (PRESSMAN, 2010).

Se concentran en los requisitos funcionales del software, tratando de encontrar los siguientes errores (PRESSMAN, 2010):

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores de estructuras de datos o en acceso a bases de datos externas.
- Errores de comportamiento o desempeño.
- Errores de inicialización y término.

Para las pruebas del método de Caja Negra se utilizó la Técnica de Partición Equivalente, la consiste en dividir el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. La partición equivalente se esfuerza por definir un caso de prueba que descubra ciertas clases

de errores, reduciendo así el número total de casos de prueba que deben desarrollarse (PRESSMAN, 2010).

A continuación, se presenta el diseño de caso de prueba que se utilizará para comprobar el funcionamiento del sistema:

Tabla 7- Diseño del Caso de prueba "Insertar Área". Fuente: Elaboración propia

No.	Nombre de Campo	Clasificación	Valor Nulo	Descripción
1.	Área	Campo de Texto	No	Contiene el nombre del área.
2.	Responsable	Numérico	No	Contiene el responsable del área.
3.	Nota	Campo de Texto	No	Contiene una breve información sobre el área.

Tabla 8- Diseño del Caso de prueba "Buscar artículo por Código". Fuente: Elaboración propia.

No	Nombre de Campo	Clasificación	Valor Nulo	Descripción
1.	Buscar...	Campo de Texto	No	Contiene el criterio de búsqueda

Descripción de Casos de Prueba

Se evaluó y probó la validez de cada una de las entradas al sistema, utilizando datos válidos (V), inválidos (I).

Tabla 9- Técnica de Partición de Equivalencia del Caso de prueba "Insertar área". Fuente: Elaboración propia.

Es-cena-rio	Descripción	Área	Res-ponsa-ble	Nota	Respuesta del sistema	Flujo central
No.1	El usuario introduce los datos correctamente	V	V	V	El sistema permite que el usuario inserte una nueva área.	1/Se introducen el nombre del área en el campo de texto. 2/Se selecciona el botón aceptar
No.2	El usuario no introduce el nombre del área nueva	I	V	V	El sistema no muestra el área en que se desea almacenar el artículo.	1/El usuario no introduce el nombre del área. 2/El sistema no muestra en el cuadro de texto el área deseada por el usuario.

Tabla 10- Técnica de partición de equivalencia caso de prueba "Buscar artículo código". Fuente: Elaboración propia.

Escenari o	Descripción	Descripción del artículo	Respuesta del sistema	Flujo central
No.1	El usuario introduce la el código del artículo correctamente	V	El sistema muestra el artículo correspondiente al código	1/Se introduce el código del artículo en el campo de texto. 2/El sistema muestra el artículo correspondiente al código.
No.2	El usuario introduce la el código del artículo incorrectamente	I	El sistema no muestra ningún producto con esa descripción	1/Se selecciona la opción de buscar el artículo 2/Se introduce un código del producto 3/El sistema no muestra ningún artículo para ese código.

3.5 Resultados de las pruebas utilizando el método de caja negra

En la iteración 3 se encontró una no conformidad asociada a la validación de “Insertar área”. Una vez realizadas las pruebas utilizando el método de caja negra mediante la técnica de partición equivalente usando el caso de prueba “Insertar nueva área”, se comprobaron las funcionalidades del sistema y la correcta validación de sus campos.

En el último sprint en el cual se entregó el producto completo al cliente para su validación y aceptación se solucionaron todas las no conformidades encontradas anteriormente en el plan de entregas.

Conclusiones del capítulo

En este capítulo, se ha llevado a cabo la planificación efectiva del Sprint Backlog, lo que ha permitido determinar las funcionalidades a implementar en cada sprint y asignarles el tiempo adecuado. Además, la implementación de estándares de codificación ha facilitado la comprensión del código y asegurado un mantenimiento más sencillo del sistema en el futuro.

El objetivo planteado en el diseño teórico se ha cumplido satisfactoriamente, logrando desarrollar un producto que cumple con los requerimientos del cliente. Esto demuestra la utilidad del proceso de desarrollo y la capacidad del equipo para entregar una solución de calidad.

Las pruebas realizadas han sido fundamentales para elevar la calidad de la solución obtenida. Estas pruebas han permitido verificar el correcto funcionamiento del sistema y asegurar que cumple con los estándares de calidad establecidos, lo cual brinda confianza tanto al equipo de desarrollo como al cliente, garantizando que el producto final sea confiable y cumpla con las expectativas.

CONCLUSIONES GENERALES

Con la culminación del presente trabajo se ha dado cumplimiento a los objetivos trazados en la investigación. A continuación, se exponen las conclusiones generales a las que fueron arribadas luego del desarrollo de la propuesta de solución:

- El estudio de los referentes teóricos-metodológico y el análisis de las diferentes herramientas y tendencias para la gestión de inventarios permitió determinar la no existencia de un sistema informático que responda a las necesidades requeridas por el cliente, demostrando que las soluciones similares no cumplen con los requerimientos de la solución deseada, por lo que quedaron descartados para ser utilizados.
- El diseño de la propuesta de solución permitió generar los artefactos más significativos de acuerdo con la metodología de desarrollo de software Scrum, tomándose como referencia el product backlog.
- La implementación del sistema permitió obtener un sistema web capaz de manejar datos referentes a los artículos que se manejan en los almacenes de la empresa CONEC.
- Las pruebas realizadas, permitieron garantizar la calidad del software y comprobar que los requerimientos definidos en el sistema son los que realmente el cliente desea.
- Se obtuvo la versión funcional del sistema de control de inventario, permitiendo así, llevar un control de los artículos, resolviendo así la situación problemática planteada.

RECOMENDACIONES

- Se recomienda evolucionar el sistema implementado para la empresa CONEC a una aplicación móvil, para así visualizar fácilmente desde cualquier parte el inventario de artículos.
- Capacitar al personal involucrado para utilizar el Sistema de Inventario para la empresa.

REFERENCIAS BIBLIOGRÁFICAS

- ¿En qué consiste Scrum? - Explicación sobre la metodología Scrum - AWS. *Amazon Web Services, Inc.* [en línea], [sin fecha]. [consulta: 15 septiembre 2023]. Disponible en: <https://aws.amazon.com/es/what-is/scrum/>.
- GARCÍA-HOLGADO, A., VÁZQUEZ-INGELMO, A. y GARCÍA-PEÑALVO, F.J., 2021. Scrum. [en línea], [consulta: 15 septiembre 2023]. DOI 10.5281/zenodo.5781344. Disponible en: <https://repositorio.grial.eu/handle/grial/2469>.
- GARCÍA-PEÑALVO, F.J., GARCÍA-HOLGADO, A. y VÁZQUEZ-INGELMO, A., 2021. Scrum. [en línea]. S.I. [consulta: 15 septiembre 2023]. Disponible en: <https://zenodo.org/record/5781344>.
- Gestión - Definiciones. *Definición. De* [en línea], [sin fecha]. [consulta: 14 septiembre 2023]. Disponible en: <https://definicion.de/gestion/>.
- Modelo Conceptual - Ingeniería de Software. [en línea], [sin fecha]. [consulta: 14 septiembre 2023]. Disponible en: <https://1library.co/article/modelo-conceptual-ingenier%C3%ADa-de-software.yeeww6ry>.
- requerimientos funcionales para un sistema de inventario | Requisitos Necesarios. [en línea], [sin fecha]. [consulta: 14 septiembre 2023]. Disponible en: <https://www.requisitosnecesarios.com/requerimientos-funcionales-para-un-sistema-de-inventario/>.
- Significado de Gestión. *Significados* [en línea], [sin fecha]. [consulta: 14 septiembre 2023]. Disponible en: <https://www.significados.com/gestion/>.
- Sistema - Definiciones. *Definición. De* [en línea], [sin fecha]. [consulta: 14 septiembre 2023]. Disponible en: <https://definicion.de/sistema/>.
- [Sin fecha]. [consulta: 15 septiembre 2023 a]. Disponible en: <https://www.bing.com/search?pglt=513&q=que+es+SCRUM+en+las+metodologias+de+la+ingenieria&cvid=f0972313f28244a588f82262a471c9f2&aqs=edge.69i57.14020j0j1&FORM=ANNTA1&PC=U531>.
- [Sin fecha]. [consulta: 30 junio 2023 b]. Disponible en: https://www.scielo.org.mx/scielo.php?pid=S2007-36072018000100002&script=sci_abstract&lng=pt.
- Concepto archivos. *SignificadosWeb.com* [en línea], [sin fecha]. [consulta: 5 octubre 2023]. Disponible en: <https://significadosweb.com/concepto/>.

- ¿En qué consiste Scrum? - Explicación sobre la metodología Scrum - AWS. *Amazon Web Services, Inc.* [en línea], [sin fecha]. [consulta: 15 septiembre 2023]. Disponible en: <https://aws.amazon.com/es/what-is/scrum/>.
- GARCÍA-HOLGADO, A., VÁZQUEZ-INGELMO, A. y GARCÍA-PEÑALVO, F.J., 2021. Scrum. [en línea], [consulta: 15 septiembre 2023]. DOI 10.5281/zenodo.5781344. Disponible en: <https://repositorio.grial.eu/handle/grial/2469>.
- GARCÍA-PEÑALVO, F.J., GARCÍA-HOLGADO, A. y VÁZQUEZ-INGELMO, A., 2021. Scrum. [en línea]. S.I. [consulta: 15 septiembre 2023]. Disponible en: <https://zenodo.org/record/5781344>.
- Gestión - Definiciones. *Definición. De* [en línea], [sin fecha]. [consulta: 14 septiembre 2023]. Disponible en: <https://definicion.de/gestion/>.
- MÉTODO, METODOLOGÍA, TÉCNICA, HERRAMIENTA ¿CONOCE LA DIFERENCIA? – Project Management College. [en línea], 2021. [consulta: 5 octubre 2023]. Disponible en: <https://pmcollege.edu.ni/metodo-metodologia-tecnica-herramienta-conoce-la-diferencia/>.
- MILLS, M., 2021. SQLite, para qué sirve y cómo se utilizan sus bases de datos | ITÍGICO. [en línea]. [consulta: 6 octubre 2023]. Disponible en: <https://itigic.com/es/sqlite-what-it-is-for-and-how-its-databases-are-used/>.
- Modelo Conceptual - Ingeniería de Software. [en línea], [sin fecha]. [consulta: 14 septiembre 2023]. Disponible en: <https://1library.co/article/modelo-conceptual-ingenier%C3%ADa-de-software.yeeww6ry>.
- PHP puro frente a PHP framework. *RibosoMatic* [en línea], 2019. [consulta: 6 octubre 2023]. Disponible en: <https://www.ribosomatic.com/articulos/php-puro-frente-a-php-framework/>.
- Qué es Scrum: Origen y características. *OpenWebinars.net* [en línea], 2018. [consulta: 6 octubre 2023]. Disponible en: <https://openwebinars.net/blog/que-es-scrum/>.
- requerimientos funcionales para un sistema de inventario | Requisitos Necesarios. [en línea], [sin fecha]. [consulta: 14 septiembre 2023]. Disponible en: <https://www.requisitosnecesarios.com/requerimientos-funcionales-para-un-sistema-de-inventario/>.

- Roles, Eventos y Artefactos en la metodología Scrum. *OBS Business School* [en línea], 2017. [consulta: 6 octubre 2023]. Disponible en: <https://www.obsbusiness.school/blog/roles-eventos-y-artefactos-en-la-metodologia-scrum>.
- Significado de Gestión. *Significados* [en línea], [sin fecha]. [consulta: 14 septiembre 2023]. Disponible en: <https://www.significados.com/gestion/>.
- Sistema - Definiciones. *Definición. De* [en línea], [sin fecha]. [consulta: 14 septiembre 2023]. Disponible en: <https://definicion.de/sistema/>.
- SQLite, qué es, para qué sirve y cómo se usan sus bases de datos. *SoftZone* [en línea], [sin fecha]. [consulta: 6 octubre 2023]. Disponible en: <https://www.softzone.es/programas/lenguajes/que-es-sqlite/>.

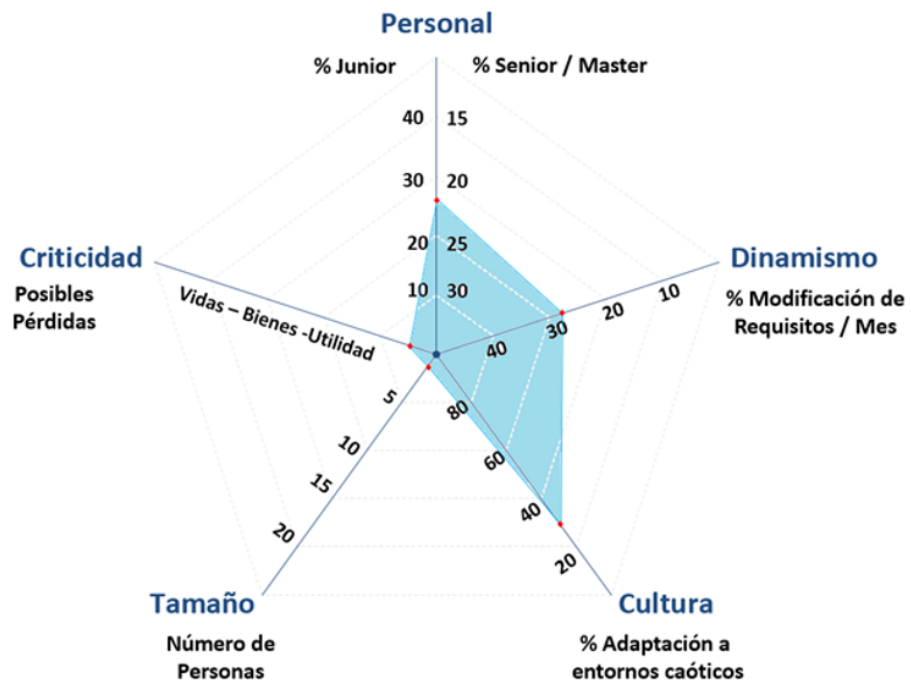
ANEXOS

Anexo 1. Método de Boehm y Turner

El método de Boehm y Turner plantea cinco criterios fundamentales mediante los que se estará valorando el proyecto; estos son:

- **Tamaño:** Este criterio se utiliza para representar el número de personas involucradas en el proyecto. Pueden tenerse en cuenta el nivel de complejidad que pueda presentarse en la comunicación entre los miembros del proyecto y los costos que pueden provocar cambios esperados.
- **Criticidad:** Se utiliza para evaluar la naturaleza del daño ocasionado por defectos que no hayan sido detectados al producto. Su evaluación puede ser cualitativa.
- **Dinamismo:** Representa la rapidez con la que pueden estar cambiando los requerimientos del proyecto. Previendo que el sistema a desarrollar contará con un número importante de requerimientos y por tanto varios cambios durante el mes, se adopta esta necesidad como una de las ventajas que brinda el enfoque ágil. Ubicando de esta forma el punto cercano al centro de la Estrella, considerando que la variación de los mismos estará ubicada entre un 20% y un 28% aproximadamente.
- **Personal:** Representa la proporción del personal con experiencia alta, media y baja. Los métodos orientados al plan no se ven afectados negativamente por este factor pues no interesa el nivel de experiencia con la que cuenten los miembros del equipo.
- **Cultura:** Las organizaciones y las personas que relaciona el proyecto pueden depender de la confianza o de la relación contractual.

El anexo muestra la representación del sistema de gestión para el inventario según los criterios que propone el método de Boehm y Turner. Resultando que la mayoría tiende al centro de la estrella, por tanto, sugiere que el enfoque de metodología de desarrollo de software sea ágil.



Anexo 1 - Estrella de Boehm-Turner. Fuente: Creación propia