



Universidad de las Ciencias
Informáticas

Universidad de las Ciencias Informáticas

Facultad 1

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

Título

**Sistema de gestión para la librería de la Universidad de las
Ciencias Informáticas**

Autor:

Yaisel Valdés Rodríguez

Tutores:

Ing. Mayleidis López Fernández
Ing. Hector Daniel Rodríguez Lorenzo

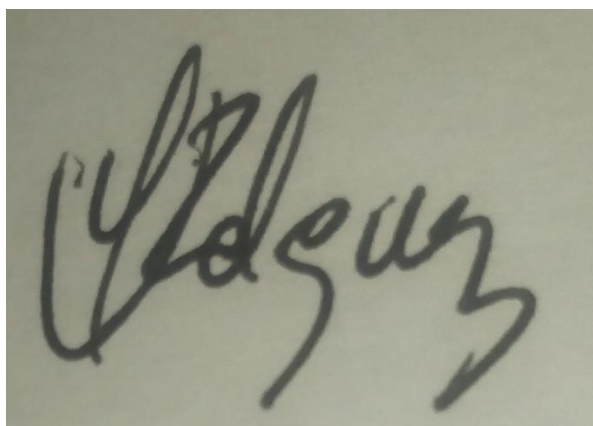
La Habana, _____ de 20__

Declaración de autoría

Declaro ser el único autor del trabajo de diploma “*Sistema de gestión para la librería de la Universidad de las Ciencias Informáticas*”, concedo a la Universidad de las Ciencias Informáticas y en especial al Centro CIDI la autorización a hacer uso del mismo en su beneficio.

Para que conste firmo el presente documento a los ____ días del mes de _____ del año 20__.

Yaisel Valdés Rodríguez



Firma del autor

Ing. Mayleidis López Fernández

Firma del tutor

Ing. Hector Daniel Rodríguez Lorenzo

Firma del tutor

Datos de contacto

Tutor: Mayleidis López Hernández: Graduada en Ingeniería Industrial especializada en Organización de Empresas en el Instituto Superior Politécnico José Antonio Echeverría (CUJAE) en 2003, Profesora Asistente. Ha impartido clases de Contabilidad y Finanzas, Administración de empresas y Práctica profesional en la facultad 1 de la Universidad de las Ciencias Informáticas. Actualmente realizando investigaciones en gestión por procesos y calidad de software.

Dedicatoria

A mi familia y amigos por todos los momentos especiales en los que hemos estado juntos.

Resumen

La Universidad de las Ciencias Informáticas cuenta entre sus múltiples servicios con una librería la cual se presenta como un método de expansión cultural para la comunidad universitaria. En dicha librería el proceso de venta y almacenamiento de los libros se realiza de forma manual, y los subprocesos de control de los inventarios y ventas son los que más tiempo y dedicación requieren, siendo estos auxiliados por documentos Excel lo que ocasiona que el tiempo de realización sea más lento.

La solución propuesta permite llevar a cabo el control de las ventas y almacenamiento de los libros en la librería de la universidad de las Ciencias Informáticas. El proceso de desarrollo de software estuvo guiado por la metodología AUP-UCI, para la implementación de la solución fue utilizado el framework de desarrollo Symfony en su versión 4.2 y los lenguajes PHPv7.2.10 para la programación del lado del servidor, HTML5, CCS3, Java Script para la programación del lado del cliente. Al finalizar la etapa de desarrollo se realizan una serie de pruebas que permiten detectar las no conformidades del sistema garantizando un correcto funcionamiento del mismo.

Como resultado se obtuvo un sistema de gestión para las ventas y el control de libros para la librería de la Universidad de las Ciencias Informáticas que contribuye a informatizar el proceso de ventas y control de libros en la librería.

Palabras clave: almacenamiento, control ,gestión, librería, libros, ventas

Abstract

The University of the Informatics Sciences has among her many services a book store which serves as a method of cultural expansion university's community. In that book store the sale's process and storage of the books makes in manual way and the subprocess of inventory control and sale are which time and dedication need the most, these are being helped by Excel documents causing a slower realization time.

The solution presented allow realize the sale's control and the storage of books in the University of the Informatics Sciences book store. The software development process was guided by the AUP-UCI methodology, for the solution implementation was used the development framework Symfony v4.2, PHP v7.2.10 as programming language in the server side. HTML5, CSS3, Java Script for the programming on the client side. At the end of development stage were made a few test which allow stand out the unconformities of the system ensuring the good working of itself.

As result was obtained a sell's management and book's control system for the University of the Informatics Sciences book store which contributes to computerise the control of books and sale's process in that book store.

Keywords: management, control, sales, storage, book store, book



Contenido

Introducción	1
Capítulo 1. Fundamentación teórica de la investigación	5
1.1 Conceptos Fundamentales:	5
1.2 Estudio de sistemas homólogos:	7
1.3 Metodología para el desarrollo	11
1.4 Herramientas	13
Capítulo 2. Propuesta de solución	17
2.1 Descripción del negocio	17
2.2 Descripción de la arquitectura:	21
2.3 Patrones de diseño:	23
Capítulo 3: Validación de la propuesta de solución	29
3.1 Diagrama de componentes	29
3.3 Validación de la propuesta de solución	35
3.3.1 Pruebas de rendimiento (Carga y Estrés)	35
3.3.2 Pruebas funcionales	36
3.3.3 Pruebas de integración	38
Conclusiones	42
Recomendaciones	43
Referencias bibliográficas	44
Referencias	44
Anexos	47

Índice de ilustraciones

Figura 1	22
Figura 2	24
Figura 3	25
Figura 4	26
Figura 5	26
Figura 6	27
Figura 7	29
Figura 8	30
Figura 9	31
Figura 10	32
Figura 11.....	32
Figura 12	33
Figura 13	34
Figura 14	35
Figura 15	37
Figura 16	40
Figura 17	40
Figura 18	47
Figura 19	47
Figura 20	48

Figura 21	48
Figura 22	49
Figura 23	49
Figura 24	50
Figura 25	50
Figura 26	51
Figura 27	51
Figura 28	52
Figura 29	52
Figura 30	53
Figura 31	53
Figura 32	54
Figura 33	54
Figura 34	55

Índice de tablas

Tabla 1.....	20
Tabla 2.....	36
Tabla 3.....	36
Tabla 4.....	37
Tabla 5.....	38
Tabla 6.....	39

Introducción

En los últimos años las Tecnologías de la Información y Comunicaciones (TIC) han tenido una participación activa en la vida cotidiana del hombre dándole la posibilidad de informatizar los procesos que afronta diariamente. Debido a esto ha surgido la necesidad de mejorar y optimizar los servicios de la manera más sencilla, rápida y menos costosa posible.

La informatización de los procesos contribuye a una mejoría del trabajo dado a que ello permite reducir costos, racionalizar el tiempo y los recursos dedicados a este. Una manera muy recurrente de informatizar procesos es mediante los sistemas de gestión de la información los cuales se han hecho casi imprescindibles en la mayoría de las empresas a tal punto que necesitan contar con estos sistemas para mejorar su desarrollo. (Gestión y Administración, 2016)

La exactitud que otorgan los sistemas de gestión trata de garantizar que la información esté completamente libre de errores. Lo que implica que la misma debe ser clara y a su vez, mediante los sistemas de gestión debe ser adecuadamente dirigida en el sentido de los datos en que se basa. Los sistemas de gestión de información que se manejan en una empresa para poder lograr una correcta administración de los recursos se encuentran completamente al servicio de la empresa y al logro de sus objetivos y por tal motivo no es independiente de los mismos. (Gestión y Administración, 2016)

La Universidad de las Ciencias Informáticas (UCI) que tiene como objetivo principal la informatización de la sociedad cubana, además de producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación, también sirve de soporte a la industria cubana de la informática (UCI, 2002). Como parte fundamental de la formación integral de sus estudiantes, la universidad ofrece entre sus diversos servicios una librería que tiene como objeto social la venta de libros a la comunidad universitaria.

La librería posee una amplia colección de libros que abarcan la mayoría de las literaturas disponibles y tiene entre sus objetivos expandir la cultura de la comunidad universitaria. El proceso de venta y almacenamiento de los libros se realiza de forma manual, y los subprocesos de control de los inventarios y ventas son los que más tiempo y dedicación requieren.

Para el control de los inventarios se almacena la información en tarjetas de estiva y hojas de cálculo en formato xls (Excel) lo que trae riesgos como la duplicidad de la información y la pérdida de registros por el deterioro del activo en el cual se almacena la información. Para la divulgación de los ejemplares que están disponibles cada cierto tiempo se circula un correo con los títulos, pero muchas veces cuando los

clientes ven la información y van a adquirir los ejemplares ya no quedan disponibles, además de que existe en ciertos periodos desconocimiento en la comunidad universitaria de la literatura en oferta.

El proceso de ventas es el proceso clave de la librería y en el cual, es de gran importancia llevar un estricto control de la información dado a que, de forma sistemática se debe realizar y archivar un informe de ventas, costando tiempo en la realización de los mismos debido a la demora a la hora de recolectar la información necesaria. Actualmente, este proceso se realiza con el apoyo de tablas en hojas de cálculo formato xls y se corre el riesgo de pérdida de información debido a corrupciones del documento por fallo energético y posible aparición de virus informáticos.

La situación anteriormente expuesta permite identificar como **problema científico**: ¿Cómo disminuir el tiempo de duración de los procesos de gestión de las ventas y almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas?

Se define como **objeto de estudio**: Procesos de gestión de libros en librerías.

Se define como **objetivo general**: Desarrollar un sistema informático para la librería de la Universidad de las Ciencias Informáticas que elimine la duplicidad de la información, asegure la persistencia de la información y disminuya el tiempo en los procesos de gestión de ventas y almacenamiento de libros.

A raíz del objetivo principal se plantean los siguientes **objetivos específicos**:

- Analizar los referentes teóricos fundamentales que sustentan las investigaciones sobre los procesos de gestión de las ventas y almacenamiento de libros.
- Realizar el diseño del sistema informático para la gestión de ventas y almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas.
- Implementar el sistema informático para la gestión de ventas y almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas
- Validar la calidad del sistema informático para la gestión de ventas y almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas.

La investigación se enmarca en el siguiente **campo de acción**: Procesos de gestión de ventas y almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas.

A partir de la situación de la situación problemática se define la siguiente **hipótesis**: El desarrollo de un sistema informático para la gestión de ventas y almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas eliminará la duplicidad de la información y disminuirá el tiempo de duración

de estos procesos.

Operacionalización de las variables:

Variable independiente: sistema informático para la gestión de ventas y almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas

Variables dependientes: duplicidad de la información y tiempo de duración.

Para dar solución al objetivo propuesto se emplearon métodos científicos de investigación teóricos y empíricos.

Métodos Teóricos:

A continuación, se presentan los métodos teóricos que se emplean en la investigación, así como las ventajas que estos ofrecen a la misma.

- Histórico-Lógico: permitió estudiar la trayectoria histórica real de la gestión de libros en librerías.
- Analítico-Sintético: permitió realizar el estudio teórico de la investigación referente a los sistemas de gestión de la información, facilitando el análisis de documentos y la extracción de los elementos más importantes relacionados con el proceso de desarrollo de sistemas de gestión y el proceso de gestión de libros en librerías.
- Inductivo-Deductivo: permitió llegar al planteamiento del objetivo, además de la extracción de las ideas fundamentales para la elaboración y fundamentación teórica del trabajo de diploma, fue utilizado para el razonamiento de la información consultada, llegando a un grupo de conocimientos particulares y generales sobre los sistemas informáticos de gestión tales como: las características y las ventajas de estos.

Métodos Empíricos:

En esta sección se presentan los métodos empíricos utilizados en la investigación y el aporte de cada uno de ellos a la investigación.

- Observación: permite obtener las características de los procesos que tienen lugar en la librería e identificar algunas de las deficiencias que afectan al mismo.
- Entrevista: permite conocer las necesidades del cliente y obtener los requisitos del sistema informático para la gestión de ventas y almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas.

El documento se estructura en tres capítulos:

Capítulo I: Fundamentación Teórica

Abarca los conceptos fundamentales utilizados en la investigación, además del análisis de las herramientas y tecnologías para el desarrollo del sistema informático para la gestión de ventas y almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas.

Capítulo II: Propuesta de solución

Se realiza una caracterización del negocio y se definen las principales funcionalidades del sistema informático para la gestión de ventas y almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas. Se describe la propuesta de solución teniendo en cuenta los requerimientos mínimos para su configuración y utilización. Además, se describen los principales casos de uso y los artefactos generados.

Capítulo III: Implementación y validación de la solución propuesta

Incluye la implementación del sistema informático para la gestión de ventas y almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas, dándole solución a cada uno de los requerimientos planteados. Una vez concluida la fase de implementación se realizan las pruebas funcionales, de seguridad y de rendimiento, que validan si el sistema cumple con los requerimientos establecidos.

Capítulo 1. Fundamentación teórica de la investigación

El presente capítulo expone el marco teórico sobre el cual se sustenta la investigación. Para ello se realiza un estudio de los sistemas informáticos para la gestión de las ventas y almacenamiento de los libros. Se realiza un análisis de la metodología, plataforma de desarrollo y herramientas necesarias para el desarrollo de la solución, así como como la justificación de su uso en cada caso.

1.1 Conceptos Fundamentales:

Venta: Del latín *vendita*, venta es la acción y efecto de vender (traspasar la propiedad de algo a otra persona tras el pago de un precio convenido). El término se usa tanto para nombrar la operación en si misma como la cantidad de cosas que se venden. La venta también es el contrato a través del cual se transfiere una cosa propia a dominio ajeno por el precio pactado. La venta puede ser algo potencial (un producto que está a la venta pero que aún no ha sido comprado) o una operación ya concretada (en este caso implica necesariamente la compra) (Merino, 2010).

Gestión de la información: “Es un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma.” (Quiroga, 2002)

La gestión de la información se establece como una disciplina transversal que aparece entrelazada en todas las diferentes capas de una organización, en todos los conceptos de administración como recursos humanos, marketing, finanzas, estrategia, operaciones, entre otros. (Quiroga, 2002)

El sistema de gestión ayuda a lograr los objetivos de una organización mediante un conjunto de estrategias que incluyen la optimización de procesos y el enfoque centrado en la gestión de la información brindando **ventajas** como: (Quiroga, 2002)

- Control efectivo de las actividades de la organización.
- Integración de nuevas tecnologías y herramientas.
- Disponibilidad de mayor y mejor información para los usuarios en tiempo real.
- Elimina la barrera de la distancia trabajando con un mismo sistema de puntos distantes.

- Disminuye errores tiempo y recursos.

Además, estos sistemas tienen **desventajas**, como son:

- El tiempo que demore su implementación.
- La resistencia al cambio de los usuarios.
- Posibles problemas técnicos como fallas en el software o el hardware.
- La inadecuada implementación de las funcionalidades para el apoyo de actividades de la organización.

Base de datos:

Es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una organización o empresa. En una base de datos de conjunto con los datos se almacena también la descripción de dichos datos. Esta descripción es a lo que denomina metadatos y es lo que permite que exista independencia de datos lógica-física. (SARRETT, 2006)

Ventajas de las Base de Datos:

- Compactación: No hay necesidad de archivos en papel que voluminosos. (Campbelly, 2007)
- Velocidad: La computadora puede recuperar y actualizar datos más rápido que un humano. (Campbelly, 2007)
- Trabajo más sencillo: Se puede eliminar gran parte del trabajo de llevar los archivos a mano. Las tareas mecánicas se realizan mejor en las máquinas. (Campbelly, 2007)
- Actualidad: La información precisa y actualizada estará a nuestra disposición en el momento en que sea requerida. (Date, 2001)

Lenguaje de programación: Es un sistema estructurado de comunicación, el cual está conformado por conjuntos de símbolos, palabras claves, reglas semánticas y sintácticas que permiten el entendimiento entre un programador y una máquina. (Rock Content, 2019)

Framework de desarrollo: Un framework, es un conjunto de clases, objetos, métodos y propiedades que van a agilizar la forma de desarrollar ya que gran cantidad del código necesario para cubrir las funcionalidades del proyecto encuentra agrupado en un paquete que permite su reutilización, este paquete es el framework en si. Un framework es un paquete que ofrece ciertas funcionalidades las cuales

se encuentran agrupadas en dicho paquete que simplemente se deben importar al proyecto para poder empezar a utilizar. (APPS, 2019)

Desarrollo web: Desarrollo web es un término que define la creación de sitios web para Internet o una intranet. Para conseguirlo se hace uso de tecnologías de software del lado del servidor y del cliente que involucran una combinación de procesos de base de datos con el uso de un navegador web a fin de realizar determinadas tareas o mostrar información. (Vázquez, 2003)

Sistema de gestión: Un sistema de gestión es una herramienta que permitirá optimizar recursos, reducir costes y mejorará la productividad en una empresa. Este instrumento de gestión exportará datos en tiempo real que permitirá tomar decisiones para corregir fallos y prevenir la aparición de gastos innecesarios. Un sistema de gestión está recomendado a cualquier tipo de organización o actividad orientada a la producción de bienes o servicios que necesiten de la gestión de sistemas. (Integra, 2019)

Sistemas de Planificación de Recursos Empresariales(ERP): Los **sistemas de planificación de recursos empresariales (ERP**, por sus siglas en inglés, *enterprise resource planning*) son los sistemas de información gerenciales que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía en la producción de bienes o servicios. (Digital Research, 2019)

1.2 Estudio de sistemas homólogos:

En la actualidad existen a niveles internacionales varios *softwares* que se encargan de la administración y la gestión de las librerías en el mundo. Estos sistemas presentan dos características fundamentales, una es que la mayoría de estos programas se encuentran desarrollados sobre la base de *software* privativo, lo cual limita sus posibilidades de usarlo, modificarlo o redistribuirlo y su código fuente o bien no está disponible o el acceso a él se encuentra restringido.

La otra característica de estos sistemas es que en la actualidad se están siendo desarrollados en tecnologías web lo cual se ha vuelto una tendencia en este siglo por las facilidades y ventajas que estas tecnologías presentan.

Ventajas de los sistemas ERP: Un estudio realizado por la consultora Panorama deja claro que el principal motivo para una empresa para implementar un software ERP es para aumentar la productividad. En realidad, las 342 empresas que participaron en la encuesta dejaron claro que

esperaban 4 beneficios operacionales: (Digital Research, 2019)

- Mejorar el desempeño de la organización.
- Asegurar el cumplimiento con la ley y norma contable.
- Hacer el trabajo de los empleados más fácil.
- Integrar mejor los sistemas entre las diferentes localizaciones de la empresa.

Sin embargo, otro estudio realizado por una empresa internacional dedicada al desarrollo de software de gestión empresarial (SAGE) llevado a cabo en 138 empresas de pequeño y mediano tamaño afirma que: (Digital Research, 2019)

- El 67% de las empresas que han implementado un sistema de planificación de recursos empresariales, lo hacen para servir mejor a sus clientes. (Digital Research, 2019)
- El 56% respondió que los sistemas ERP constituyen de forma significativa a la expansión de su negocio. (Digital Research, 2019)

Dentro de todos los distintos sistemas de gestión de la información y sistemas ERP existe varios que destacan de los cuales se hará una breve presentación a continuación:

- **SAP:** Es el sistema de Planificación de Recursos de Empresa (ERP) más utilizado a nivel mundial y pionero en este mundo. Un 24% de los usuarios de sistemas de ERP apuestan por este programa de gestión integral. Esta herramienta que se suministra bajo demanda, por lo que se paga por lo que se usa, está diseñada para todo tipo de compañías, en especial para pequeñas y medianas empresas. La aplicación integra funciones para la gestión de clientes (CRM), la gestión del capital humano o la gestión financiera, entre otros. Además, dispone de un servicio *cloud* (nube) para que se pueda conectar desde donde esté y a través de cualquier dispositivo. Esta herramienta es una tecnología muy eficaz que permite una buena comunicación y en tiempo real. (JBlogs, 2020)
- **ORACLE:** Ocupa el segundo lugar en los ERP más utilizados, con un 12% de usuarios. Esta herramienta ofrece una gran variedad de módulos sobre gestión financiera, ventas, compras, distribución y logística, planificación, gestión de proyecto o de recursos humanos. Este sistema se diferencia del resto por sus aplicaciones empresariales que mejoran la experiencia de las empresas. (JBlogs, 2020)
- **Odoo:** Es una suite integrada de aplicaciones que incluye módulos de gestión de proyectos,

facturación, contabilidad, gestión de inventarios, fabricación y compras. Estos módulos pueden comunicarse entre sí para intercambiar información de forma eficiente y fluida. Odoos es una herramienta basada en la web. La estructura de precios es diferente a la de otras soluciones. Actualmente se paga una base de 10 euros al mes por usuario y por cada aplicación que se añada se deberá abonar una cuota mensual única (independiente del número de usuarios) que puede oscilar entre los 6 y 36 euros al mes. (JBlogs, 2020)

Versat Sarasola constituye el primer sistema integral de gestión de contabilidad certificado. Desarrollado para la gestión económica eficaz y fiable. Este ofrece al usuario la posibilidad de contar con un instrumento seguro, rápido, eficaz y de fácil manejo para la planificación, control y el análisis de la gestión económica. Ha sido diseñado para ser utilizado en cualquier entidad y permite llevar el control y el registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las mismas y obtener los estados financieros y análisis económicos y financieros en estos niveles. Actualmente es utilizado en Cuba en más de 8 000 entidades de 29 organismos. Este sistema integrado cuenta con un conjunto de 12 módulos entre los que se encuentran control de activos fijos y control de inventarios. En el módulo de Control de Inventarios se definen formatos del clasificador de productos para lograr una uniformidad en el registro y la agregación de información en los reportes de salida, se conceptualizan los movimientos para lograr una información amplia sobre los orígenes y destinos de los recursos. Permite el control de las existencias y movimientos en diferentes monedas. Muestra el cuadro diario de cada uno de los almacenes por las diferentes cuentas. Ofrece la posibilidad de duplicar documentos para agilizar los pases de los mismos y lograr que un mismo documento se convierta en otro con solo adicionar un mínimo de información, realiza un control de las existencias y movimientos por custodios y se emiten diferentes reportes e información de utilidad para la correcta administración de los recursos materiales. El módulo de Control de Activos Fijos recoge las operaciones normales que en esta actividad se realizan (altas, bajas, modificaciones de los medios). Posee un asistente para la configuración del subsistema, la tarea más compleja de este módulo.

El estudio planteado permite determinar que los sistemas ERP no son los adecuados para este tipo de solución dado a que son implementados para empresas más grandes y completas. Además, para la realización de la gestión de ventas en librerías los softwares

empleados son menos complejos y estos son utilizados por empresas pequeñas en su mayoría debido a que estas (las librerías) según la entrevista realizada (Preguntas de la entrevista se encuentran la final de los anexos) no presentan tantos departamentos y subprocesos para requerir la necesidad de un ERP.

A continuación, se presentan varios sistemas de gestión y ventas de libros empleados nacional e internacionalmente.

Fierro: Es un sistema de gestión surgido en 2006 que se ha implementado exitosamente en más de 250 empresas en Argentina, Ecuador, México, Uruguay, Colombia, Chile, Perú y Paraguay. Está desarrollado sobre tecnologías web automatizando así diferentes procesos como son la venta de libros y la contabilidad de la librería. En cuanto a las ventas este sistema permite la facturación mediante múltiples puntos de facturación, establece distintas condiciones de venta permitiendo especificar descuentos a efectuar, formas y plazos de pago permitidos y permite gestionar el proceso de reserva y seña de libros. En cuanto a la contabilidad este sistema genera automáticamente reportes habituales de la contabilidad (libro diario, balance de sumas y saldos). Este sistema está sujeto a un pago regular de licencia de uso. (Fierro, 2018)

Sinlib: Es un software de gestión de librerías que permite la recepción de fichas de artículos, albaranes, facturas, cambios de precio y estado, envío de pedidos y devoluciones, la Creación de las fichas de artículos de forma automática, cambios de precio, gestión de lotes de cursos, cuenta con una forma de notificación al cliente por SMS o email de la recepción del lote completo o parcial. Además, este software realiza exportaciones e importaciones para plataformas de comercio electrónico y tiendas online propias. Este software esta desarrollado sobre la base de tecnologías web y está sujeto a un pago sistemático de licencias de uso. (Arminet Software and Web, 2017)

ManagementPro: Es un sistema de punto de venta diseñado para librerías el cual cuenta entre sus distintos módulos con métodos de facturación electrónica generando automáticamente el archivo de factura para el envío a sus clientes. Presenta un módulo para control de inventario y también emplea módulos de comercio electrónico para sus ventas online permitiendo un mayor alcance de sus ventas. Esta desarrollado con tecnologías web y está sujeto a pago de licencia de uso. (ManagementPro Inc, 2019)

Librería Virtual: Desarrollada por la empresa cubana Citmatel esta aplicación cuenta con

tecnologías de comercio electrónico facilitando la venta de los ejemplares en formato digital a los clientes sin importar su ubicación en el territorio. Está desarrollada con tecnologías web, está libre de pago de licencias de uso pero requiere una constante conexión a internet. (Citmatel, 2020)

Una vez analizados estos sistemas se determina que no son del todo apropiados para realizar el proceso de gestión de ventas y almacenamiento de los libros en la librería de la Universidad de las Ciencias Informáticas debido a que estos fueron desarrollados sobre plataformas de software privativo lo cual trae como consecuencia grandes gastos en licencias y mantenimiento de software, también la mayoría de estos requieren una constante conexión a internet y servicios de la nube.

1.3 Metodología para el desarrollo

Una metodología de desarrollo de software describe un entorno que es usado para organizar, planificar, y dirigir un proceso de desarrollo de un *software*. Existen varias metodologías de desarrollo, todas contienen algunas etapas básicas del ciclo de desarrollo como son la planificación, análisis, diseño, implementación y mantenimiento. (MARTÍNEZ, 2016)

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas (test driven development - TDD en inglés)
- Modelado ágil
- Gestión de Cambios ágil
- Refactorización de Base de Datos para mejorar la productividad (Rodríguez, 20015)

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva.

Fases AUP

- 1. Inicio: El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- 2. Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- 3. Construcción: Durante la fase de construcción el sistema es desarrollado y probado en el ambiente de desarrollo.

- 4. Transición: El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación para finalmente ser desplegado en los sistemas de producción (Rodríguez, 20015)

AUP variante UCI: De las cuatro fases que propone AUP (Inicio, elaboración, construcción, transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de inicio, pero modificando el objetivo de la misma, se unifican las restantes fases de AUP en una sola la cual se llama Ejecución y se agrega la fase de cierre. (Sánchez, 2006)

AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno) para esta variación se decide tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina modelo en la variación para la UCI se consideran a cada una de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de prueba se desagrega en 3 disciplinas: Pruebas Internas, de Liberación y Aceptación. Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI-DEV v1.3 para el nivel 2, serían CM(Gestión de configuración), PP(Planeación de proyecto) y PMC(Monitoreo y control de proyecto). (Sánchez, 2006)

Escenarios para la disciplina Requisitos:

-Características por escenarios

Escenario No 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario) similar a una llamada y respuesta respectivamente, donde la atención se centra en como el usuario va a utilizar el sistema. Es necesario que se tenga claro por el proyecto que los casos de uso del negocio muestran como los procesos son llevados a cabo por personas y los activos de la organización. (Sánchez, 2006)

Escenario No 2: Aplica a proyectos que haya evaluado el negocio a informatizar y como resultado obtenga que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de la información. (Sánchez, 2006)

Escenario No 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas

que los manejan, proporcionando objetividad, solidez y su continuidad. Se debe tener presente que este escenario es muy conveniente si se desea presentar una gran cantidad de niveles de detalles y relaciones entre los procesos identificados. (Sánchez, 2006)

Escenario No 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos ya que una historia de usuario no debe poseer demasiada información. (Sánchez, 2006)

Se decidió utilizar la metodología Variación AUP-UCI, de esta forma se logra estandarizar el proceso de desarrollo de *software* dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3. Se decidió utilizar además el escenario número 2, el cual modela el sistema mediante el modelo conceptual y casos de uso.

1.4 Herramientas

Symfony: Symfony es un completo *framework* diseñado para optimizar el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (Solucionex, 2009)

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de Sistema de Gestión de Bases de Datos (SGBD) en cualquier fase del proyecto.
- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
- Sencillo de usar en la mayoría de casos, está más (Solucionex, 2009) indicado para grandes aplicaciones web que para pequeños proyectos.

Aunque utiliza MVC (Modelo vista controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.

Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.

Sigue la mayoría de mejores prácticas y patrones de diseño para la web. Preparado para aplicación empresarial y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo. Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo. Fácil de extender, lo que permite su integración con las librerías de otros fabricantes (Solucionex, 2009).

PHP: es conocido como un lenguaje basado en servidores. Esto es porque el **PHP** no se ejecuta en la computadora del cliente, sino en la computadora que visita la página. Los resultados son brindados hacia el cliente y mostrados en su navegador (Ángel Cobo, 2005).

Ventajas de php:

PHP es compatible con (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT,ME,2000,XP,hasta las últimas versiones) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al OS.

La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C se incluye Java y Javascript, de hecho mucha de la funcionalidad del PHP se debe a C en funciones como fread() o strlen(), esto provoca que los desarrolladores se sientan más cómodos.

PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código. (Gonzalez, 2016)

Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen yTHTTPD. Otra alternativa es configurarlo como módulo CGI. Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre podrás disponer de ODBC para situaciones que lo requieran.

Una gran variedad de módulos cuando un programador PHP necesite una interface para una librería en particular, fácilmente podrá crear una API para esta. Algunas de las que ya vienen implementadas permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, entre otras.

PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz, así que se ejecuta rápidamente utilizando poca memoria.

PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione. (Gonzalez, 2016)

Además de PHP se encuentran otros lenguajes de programación cliente/servidor como es el caso de Perl que permite su uso tanto en desarrollo web como en otros entornos, posee una buena integración con la mayoría de los sistemas, pero puede ser lento ya que este compila al inicio de su ejecución y al no disponer de control de excepciones se vuelve complicado encontrar algunos errores. También se encuentra Rust que es un lenguaje rápido y seguro (gestión automática de guardado y el manejo de la memoria), está basado en expresiones, pero no se puede utilizar un valor que no haya sido inicializado previamente y es más complejo de aprender que otros lenguajes ya que algunas de sus reglas son diferentes a las habituales. (González, 2018)

MySQL: En cuanto a la definición general, MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional. (Fernandez, 2015).

MySQL presenta varias ventajas para los desarrolladores como son:

- Verificación basada en host.
- Puede usarse aunque no haya ninguna red disponible.
- Sistema flexible de privilegios y contraseñas.
- Cifrado de seguridad de todo el tráfico de contraseñas
- La biblioteca se puede integrar en aplicaciones independientes.
- Ofrece el servidor por separado para entornos cliente/servidor. (Cabrera, 2019)

Existen además de MySQL otros RDBMS como por ejemplo MariaDB que, a pesar de tener una alta escalabilidad, fácil integración y acceso en tiempo real como desventaja presenta la carencia de interfaz en caché de memoria y la carencia de rastreador de optimización. Otro ejemplo sería MongoDB que posee validación de documentos y motor de almacenamiento cifrado este RDBMS no es adecuada para aplicaciones que requieren transacciones complejas, como tampoco es adecuada para la sustitución directa de aplicaciones heredadas y es usado con mayor frecuencia en aplicaciones móviles y catálogos de producto. (Cabrera, 2019)

Como resultado de lo anteriormente planteado se emplea como lenguaje de programación: PHP, como sistema gestor de bases de datos: MySQL, y como framework de desarrollo: Symfony.

Conclusiones del capítulo

- Con el estudio de los referentes teóricos sobre el proceso de ventas, gestión y almacenamiento se evidenció la importancia que este tiene para la entidad, lo que hace necesario una buena gestión de dicho proceso con el objetivo de garantizar que el proceso de ventas y almacenamiento se realice en un menor tiempo.
- El estudio de los sistemas homólogos demostró que las aplicaciones existentes no resuelven el problema, no cumplen las necesidades de la librería de la UCI o no se pueden utilizar para informatizar el proceso en la librería de la UCI. Dada estas condiciones se evidenció la necesidad de desarrollar un sistema que permita la gestión de ventas y almacenamiento con el objetivo de informatizar dicha actividad.
- El estudio de las metodologías, herramientas y lenguajes permitió definir los componentes base para el desarrollo de un sistema de gestión, además permitió definir el ambiente de desarrollo.

Capítulo 2. Propuesta de solución

En el presente capítulo se realiza un análisis del proceso de ventas y almacenamiento de libros en la librería de la UCI y sus características, así como las principales funcionalidades del sistema a desarrollar. Además, se describe la propuesta de solución del problema y los principales casos de uso.

2.1 Descripción del negocio

La Universidad de las Ciencias Informáticas cuenta entre sus servicios con una librería que tiene como objeto social la venta de libros a la comunidad universitaria

La librería cuenta con dos áreas: Almacén y Salón. Cuando un libro llega a la librería proveniente del Centro Provincial del Libro el almacenero realiza un informe de recepción auxiliándose de la factura recibida y se ubica el libro en el almacén. Los libros son clasificados por el almacenero, atendiendo a las categorías: juveniles, cubana, universal, revista, minilibro, etc; y se les realiza una tarjeta estiba. Cuando el libro pasa al Salón se realiza un vale de salida del almacén para el Salón, el cual se tiene digital, en formato xls y se da entrada a las tarjetas de estiba del Salón, enunciando la cantidad. Al realizar la venta de los libros, se registra en un talonario de ventas teniendo en cuenta cantidad, título, precio e importe. Diariamente se realiza un resumen de ventas diario generándose el Modelo de Resumen de Ventas Diarias y se rebaja estas cantidades en el Resumen de Existencias de libros, depositándose posteriormente el dinero en el banco. A su vez, se rebaja de las tarjetas de estiba del Salón, el resumen de ventas. Al final del mes, se realiza un Reporte de Ventas al Centro Provincial del Libro con la cantidad de libros vendidos diarios por título.

El proceso de gestión de los libros en la librería se realiza en su mayoría de forma manual auxiliándose de tablas realizadas en documentos excel, lo que dificulta el control de los libros disponibles, ocasionando pérdida de los registros, manejo de información duplicada y tardanzas en las entregas y elaboración de reportes. Esto trae consigo la pérdida de tiempo en la gestión de los libros de la librería y obstaculizando el proceso de generación de reportes y estadísticas.

Teniendo en cuenta lo anteriormente expuesto se definen los siguientes **requisitos funcionales** del sistema a desarrollar.

RF 1: Clasificar libros.

RF 2: Crear tarjeta de estiba.

RF 3: Modificar tarjeta de estiba

RF 4: Eliminar tarjeta de estiba

RF 5: Mostrar tarjeta de estiba

RF 6: Crear vale de salida del almacén.

RF 7: Mostrar vale de salida del almacén

RF 8: Archivar vale de salida del almacén.

RF 9: Publicar en el sitio de la universidad el listado de libros actualizado.

RF 10: Crear talonario de ventas.

RF 11: Modificar talonario de ventas.

RF 12: Archivar talonario de ventas.

RF 13: Mostrar talonario de ventas.

RF 14: Realizar reporte.

RF 15: Autenticar usuario.

RF 16: Añadir usuario

RF 17: Editar usuario

RF 18: Eliminar usuario

RF 19: Crear rol

Rf 20: Modificar rol

RF 21: Eliminar rol

RF 22: Editar reporte

RF 23: Eliminar reporte

En la tabla 1 se muestran los requisitos funcionales con sus respectivas descripciones.

A continuación, se presentan los principales **casos de uso** del sistema a desarrollar.

Caso de uso 1: Gestionar libros.

Caso de uso 2: Gestionar vale de salida del almacén

Caso de uso 3: Gestionar ventas.

Caso de uso 4: Publicar en el sitio los títulos disponibles

Caso de uso 5: Gestionar usuario.

Caso de uso 6: Gestionar rol

Caso de uso 7: Gestionar reporte

Caso de uso 8: Autenticar usuario.

Además, se presentan a continuación los **requisitos no funcionales** del sistema informático a desarrollar.

Usabilidad:

- Debe ser de fácil manejo incluso para las personas que no sean expertas en el uso de las computadoras.

Interfaz

- La aplicación deberá contar con un diseño orientado a llamar la atención del usuario, o sea, una interfaz amigable y profesional que permita que los usuarios interactúen con facilidad con la misma.

- Permitir un manejo fácil para todo tipo de usuarios, brindando confianza y seguridad en cuanto a su uso.

Rendimiento

-Debe ser eficiente, rápida tanto en los tiempos de respuesta como en la velocidad de procesamiento de la información, que debe ser el mínimo posible en el orden de las fracciones de segundo.

-Estará implementado sobre una tecnología web, facilitando su uso a través de la red

Confiabilidad

-La información debe de estar protegida contra accesos no autorizados, debe ser publicada sólo por quien tiene la autorización para hacerlo.

-Proteger los datos contra acciones no autorizadas o que pueden afectar la integridad de los mismos.

Seguridad

El sistema debe garantizar que la información sea publicada únicamente por quien tiene derecho a publicarla.

-Identificar al usuario antes de que pueda realizar cualquier acción sobre el contenido del sistema.

-Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

Características técnicas

- Requerir un servidor con 1Gb de RAM y un microprocesador Dual-Core como mínimo.
- Requerir un servidor web Apache en su versión 2.4.35 y PHP 7 o superior
- Requerir un servidor de base de datos MySQL para hacer el sistema más ligero.

Tabla 1 Descripción de Requisitos

No	Requisito	Descripción
1	Clasificar libro	El sistema debe permitir clasificar el libro según su literatura
2	Crear tarjeta de estiba	El sistema debe permitir crear el registro del libro en el almacén
3	Modificar tarjeta de estiba	El sistema debe permitir modificar el registro del libro en el almacén
4	Eliminar tarjeta de estiba	El sistema debe permitir eliminar el registro del libro en el almacén
5	Mostrar tarjeta de estiba	El sistema debe mostrar el registro del libro en almacén
6	Crear vale de salida del almacén	El sistema debe e permitir crear el registro del libro que sale del almacén al salón
7	Mostrar vale de salida del almacén	El sistema debe e permitir mostrar el registro del libro que sale del almacén al salón
8	Archivar vale de salida del almacén	El sistema debe permitir guardar el registro del libro que sale del almacén al salón
9	Publicar en la intranet los títulos disponibles	El sistema debe mostrar en un enlace externo la literatura disponible
10	Crear talonario de ventas	El sistema debe permitir crear el talonario donde se ubican las ventas realizadas
11	Modificar talonario de ventas	El sistema debe permitir modificar el talonario donde se ubican las ventas realizadas
12	Archivar talonario de ventas	El sistema debe r guardar el talonario donde se realizan las ventas
13	Mostrar talonario de ventas	El sistema debe permitir mostrar el talonario donde se realizan las ventas
14	Realizar reporte	El sistema debe permitir realizar el reporte de ventas y del estado del almacén

15 Autenticar usuario	El sistema debe permitir al usuario autenticarse con sus credenciales
16 Añadir usuario	El sistema debe permitir añadir un usuario
17 Editar usuario	El sistema debe permitir modificar un usuario existente
18 Eliminar usuario	El sistema debe eliminar un usuario existente
19 Crear rol	El sistema debe crear un rol
20 Modificar rol	El sistema debe modificar un rol existente
21 Eliminar rol	El sistema debe permitir eliminar un rol existente
22 Editar reporte	El sistema debe poder modificar un reporte existente
23 Eliminar reporte	El sistema debe permitir eliminar un reporte

2.2 Descripción de la arquitectura:

El Modelo Vista Controlador (MVC) es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores. (Alvarez, 2014)

Modelo:

Es la capa donde se trabajará con los datos, contiene mecanismos para acceder a la información y también para actualizar su estado. Los datos se tienen en una base de datos por lo que el modelo tendrá todas las funciones que accederán a las tablas (Alvarez, 2014).

Vista:

La vista contiene el código de la aplicación que va a producir la visualización de las interfaces de usuario, el código que permitirá cargar y mostrar los estados de la aplicación en HTML. En la vista generalmente se trabaja con los datos, sin embargo, no se realiza un acceso directo a estos. La vista requerirá los datos al modelo y en ella se generará la salida, tal como la aplicación lo requiera (Alvarez, 2014).

Controlador:

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación. Es una capa que sirve de enlace entre la vista y el modelo, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de la aplicación. Su responsabilidad no es manipular

directamente los datos ni mostrar ningún tipo de salida, sino servir de enlace entre el modelo y la vista para implementar las diversas necesidades del desarrollo (Alvarez, 2014).

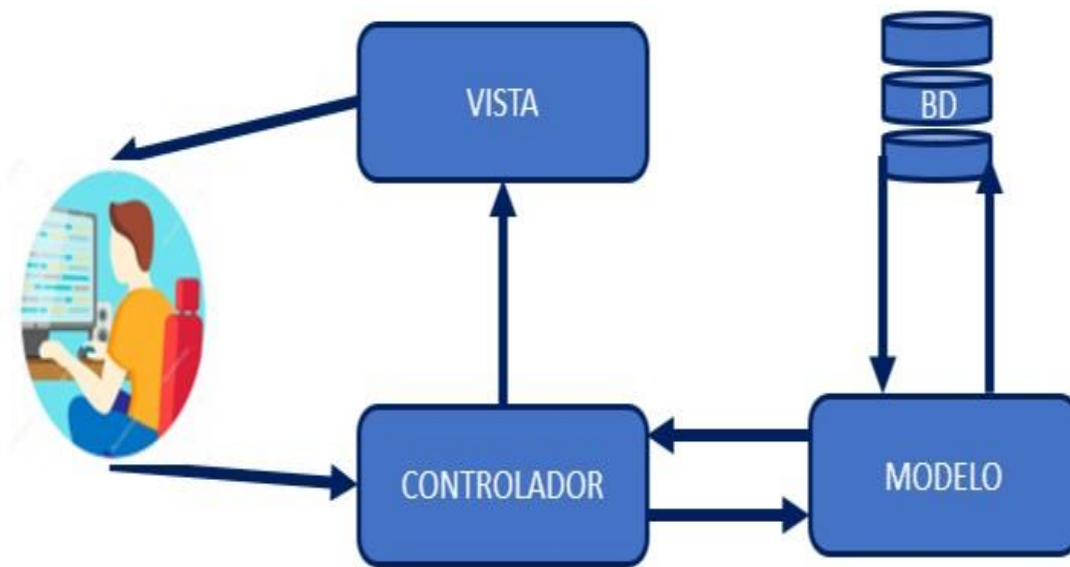


Figura 1 Relación modelo-vista-controlador

2.3 Patrones de diseño:

Los patrones de diseño expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que constituir sistemas de software. Estos permiten diseñar esquemas seguros y que a su vez cumplan con los estándares de diseño establecidos por normas internacionales para el desarrollo de aplicaciones web. Los patrones usados para el desarrollo del sistema se evidencian de la forma siguiente:

Controlador:

Las peticiones web son manipuladas por un controlador frontal que es el punto de entrada de toda la aplicación un entorno determinado. Este patrón se evidencia en clases como “DefaultController” y el `index.php` del ambiente. (Potencier, 2009)

Para el desarrollo de la clase *LibroController* se emplea este patrón, lo cual la convierte en la clase encargada de realizar las tareas asociadas al control de los libros.

Singleton:

Presente en la clase Routing. Esta clase la utiliza el controlador frontal y se encarga de enrutar todas las peticiones que se hagan a la aplicación. (Potencier, 2009)

Registry:

Este patrón es un medio eficiente y sencillo de compartir datos y objetos en la aplicación sin la necesidad de preocuparse por conservar numerosos parámetros o hacer uso de variables globales. Este patrón se aplica en la clase Config que es la encargada de acumular todas las variables de uso global en el sistema. (Potencier, 2009)

Creador:

En la clase Actions se encuentran todas las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades. (Potencier, 2009)

Este patrón está presente en el método *new* de la clase *LibroController* donde este método es el encargado de crear un libro.

Modelo Conceptual:

El diagrama de modelo conceptual es el encargado de modelar cada uno de los conceptos del negocio y sus relaciones entre ellos, así como la dirección en que estas relaciones se ejecutan. En la figura 2 se ejemplifica el modelo conceptual para el sistema de gestión para la librería de la Universidad de las Ciencias Informáticas.

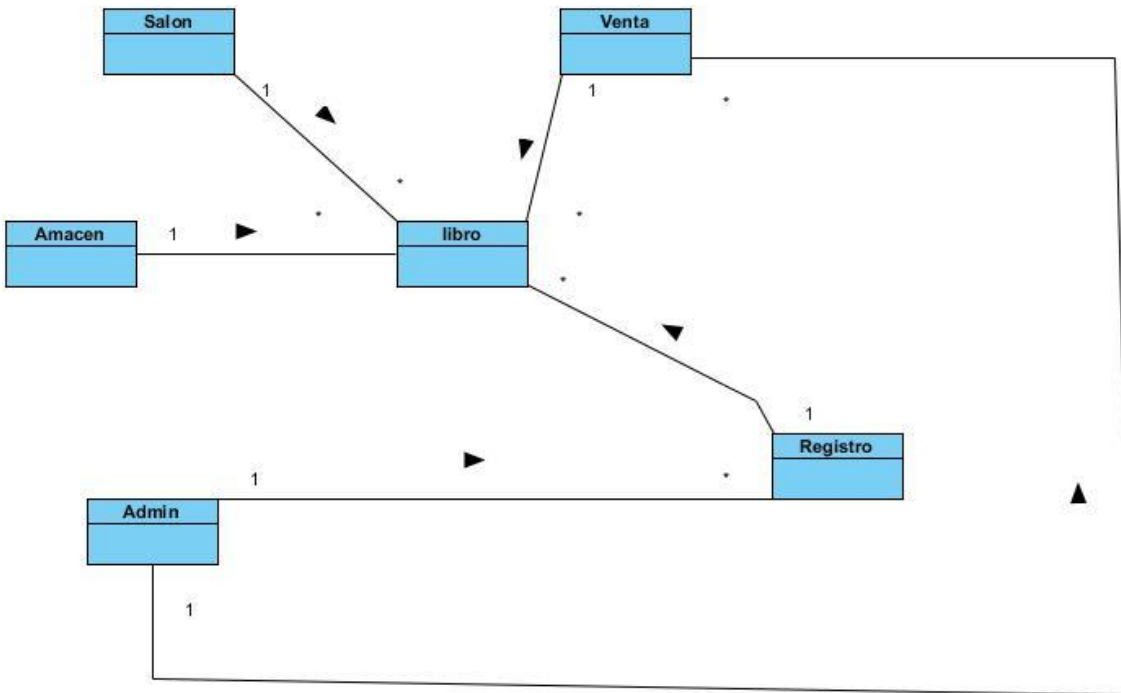


Figura 2 Modelo Conceptual

Diagrama de casos de uso:

El diagrama de casos de uso modela la relación entre los actores del sistema, así como su interacción con los casos de uso determinados anteriormente en la presentación de los requisitos funcionales. (Sommerville, 2005). En la figura 3 se representa el diagrama de casos de uso para el sistema de gestión para la librería de la Universidad de las Ciencias Informáticas.

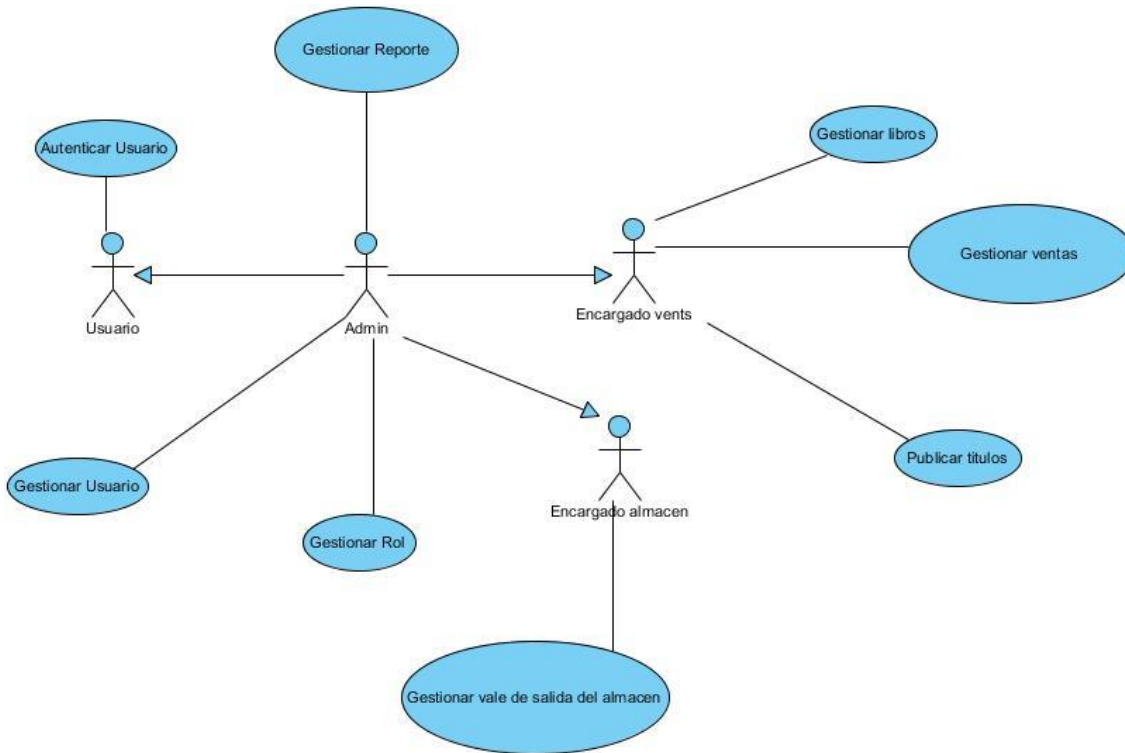


Figura 3 Diagrama de Casos de Uso

Diagrama de clases Crear Libro

El diagrama de clases modela la relación entre las clases identificadas para el sistema (Sommerville, 2005). En la figura 4 se muestra el diagrama de clases correspondiente al requisito funcional de crear libro para el sistema de gestión para la librería de la Universidad de las Ciencias Informáticas.

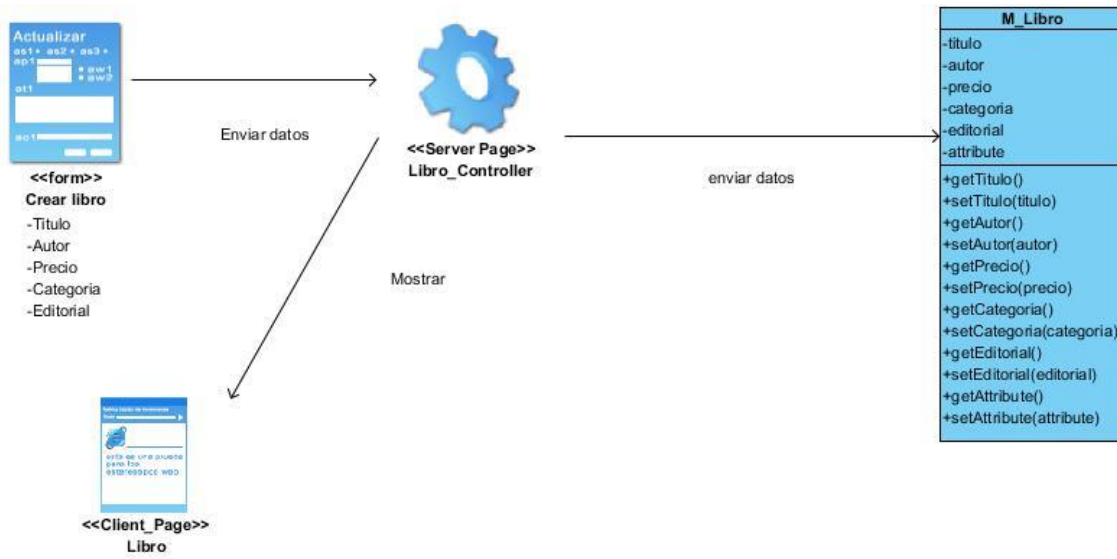


Figura 4 Diagrama de Clases Crear Libro

Modelo de datos:

El modelo de datos realiza la modelación de la relación entre las entidades de la base de datos del sistema (Sommerville, 2005). En la figura 5 se evidencia el modelado de datos para el sistema de gestión para la librería de la Universidad de las Ciencias Informáticas

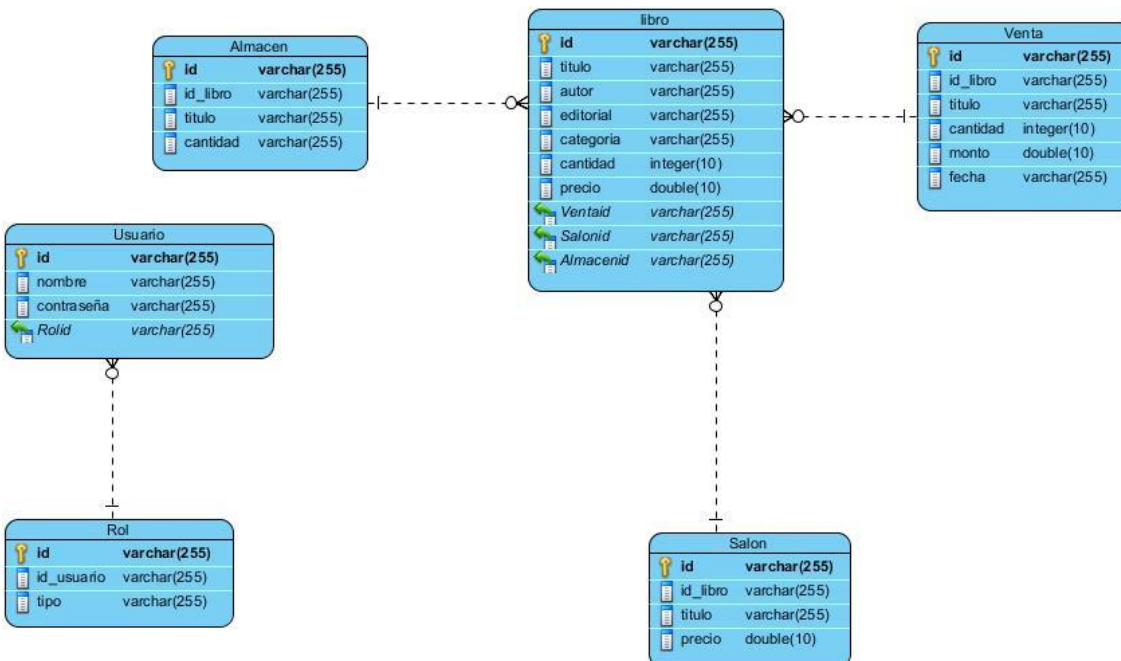


Figura 5 Modelo de datos.

Diagrama de despliegue:

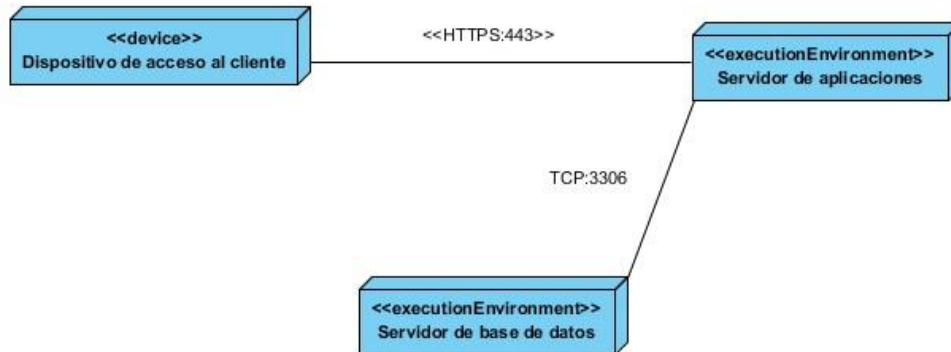


Figura 6 Diagrama de despliegue

Descripción de elementos e interfaces de comunicación:

Dispositivo de acceso al cliente: La estación de trabajo necesitará un navegador web para conectarse con el sistema en el servidor de aplicaciones mediante el protocolo HTTP/HTTPS.

Servidor de aplicaciones: Es la estación de trabajo que contiene el código fuente de la aplicación, le brinda las interfaces al usuario para que este interactúe con la aplicación y este se conecta con el servidor de bases de datos mediante el protocolo TCP.

Servidor de bases de datos: Es el servidor donde se almacenan los datos con los que trabaja la aplicación, este se comunica con el servidor de aplicaciones permitiendo la interacción del usuario con la base de datos del sistema.

HTTPS: Protocolo que establece la conexión segura entre el dispositivo de acceso cliente y el servidor de aplicaciones. La conexión es por cable vía modem, LAN o red inalámbrica con una velocidad de más de 64 Kbps.

TCP: Protocolo que establece la conexión entre el servidor de aplicaciones y el servidor de base de datos. Para el servidor de base de datos de MySQL se define el puerto 3306. La conexión entre el servidor web y el servidor de base de datos permite dar órdenes y obtener información de esta.

Conclusiones del capítulo:

- Los requisitos funcionales y no funcionales permitieron describir las distintas funcionalidades que debe cumplir el sistema de gestión para la librería de la Universidad de las Ciencias Informáticas.
- La utilización de los artefactos que propone la metodología AUP UCI sirvió de guía para el diseño del sistema de gestión para la librería de la Universidad.
- Las actividades contempladas en el análisis y diseño, posibilitaron dar una mayor descripción de los elementos del sistema, permitiendo una mejor comprensión para su implementación.

Capítulo 3: Validación de la propuesta de solución

El presente capítulo presenta los componentes y estándares de codificación que sustentan la implementación del Sistema de gestión para la librería de la Universidad de las Ciencias Informáticas y se describe el proceso de validación de la solución implementada, mediante la utilización de casos de pruebas.

3.1 Diagrama de componentes

El diagrama de componentes muestra la relación entre los diferentes componentes del sistema de gestión para la librería de la UCI. El componente `index.php` es el encargado de relacionar la mayor parte de los componentes y constituye la página principal de la aplicación, las vistas contienen las páginas de la aplicación web, los controladores son los encargados de realizar las operaciones en la aplicación, las rutas se encargan de direccionar las operaciones y estas funcionan a su vez con los controladores y la base de datos es la encargada de manejar la información con la que trabaja el sistema.

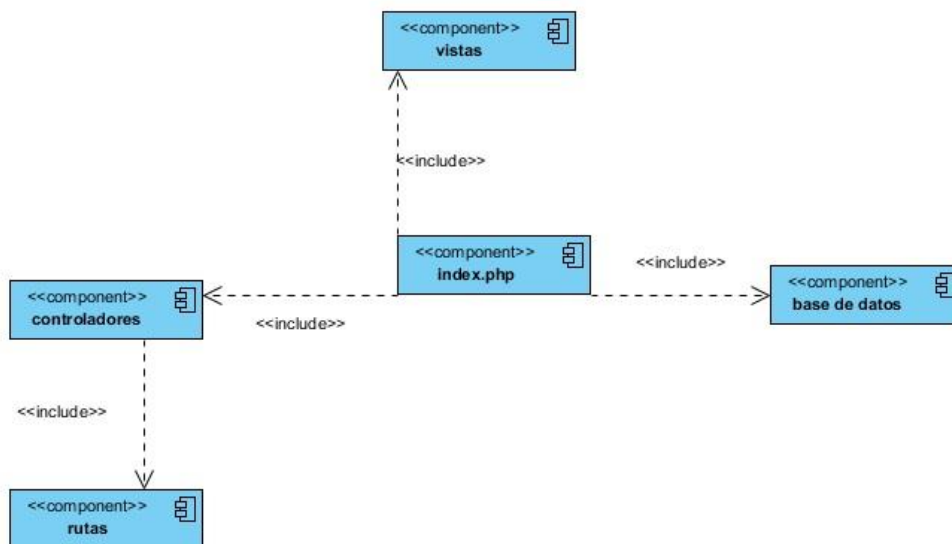


Figura 7 Diagrama de Componentes

3.2 Estándares de codificación:

Los estándares de codificación constituyen un principio esencial en el desarrollo de software. Garantizan que el código obtenido sea fácil de leer, entendido y modificado independientemente de quien haya sido el desarrollador del producto. Son una guía para el equipo de desarrollo, permiten asegurar que el código presente calidad y no contenga errores.

Etiquetas de apertura y cierre PHP

Cuando se escribe en PHP, siempre se deben utilizar las etiquetas `<?php` y `?>`, y en ningún caso la versión corta `<? y ?>`. En general se omite la etiqueta de cierre de PHP (`?>`) al final de los archivos controladores y de entidades. A continuación se puede apreciar el uso de la etiqueta de apertura `<?php` :

```
<?php

namespace App\Entity;

use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity(repositoryClass="App\Repository\AlmacenRepository")
 */
class Almacen
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $id_libro;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $titulo;

    /**
     * @ORM\Column(type="integer")
     */
```

Figura 8 Uso de etiquetas php

Indentación

La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se emplea indentación para anidar elementos. En

Symfony se debe indentar con dos espacios. Además, no se debe dejar espacios en blanco al final de cada línea. A continuación, se muestra un ejemplo de indentación en Symfony:

```
public function new(Request $request): Response
{
    $ventum = new Venta();
    $form = $this->createForm( type: VentaType::class, $ventum);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $entityManager = $this->getDoctrine()->getManager();
        $entityManager->persist($ventum);
        $entityManager->flush();

        return $this->redirectToRoute( route: 'venta_index');
    }

    return $this->render( view: 'venta/new.html.twig', [
        'ventum' => $ventum,
        'form' => $form->createView(),
    ]);
}
```

Figura 9 Uso de la Indentación

Uso de comillas

Se utilizan tanto comillas simples (' ') como las comillas dobles (" ") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables en la cadena de texto. En la siguiente imagen se evidencia el uso de comillas en la propuesta de solución:

```

$form = $this->createForm( type: LibroType::class, $libro);
$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    $this->getDoctrine()->getManager()->flush();

    return $this->redirectToRoute( route: 'libro_index', [
        'id' => $libro->getId(),
    ]);
}

return $this->render( view: 'libro/edit.html.twig', [
    'libro' => $libro,
    'form' => $form->createView(),
]);

```

Figura 10 Empleo de comillas.

Uso de llaves

Se deben utilizar llaves ({ }) para indicar el cuerpo de la estructura de control(funciones, instrucciones condicionales como if, instrucciones cíclicos como while) independientemente del número de declaraciones que contenga. En el siguiente ejemplo se muestra el uso de las llaves en la propuesta de solución:

```

public function new(Request $request): Response
{
    $ventum = new Venta();
    $form = $this->createForm( type: VentaType::class, $ventum);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $entityManager = $this->getDoctrine()->getManager();
        $entityManager->persist($ventum);
        $entityManager->flush();

        return $this->redirectToRoute( route: 'venta_index');
    }

    return $this->render( view: 'venta/new.html.twig', [
        'ventum' => $ventum,
        'form' => $form->createView(),
    ]);
}

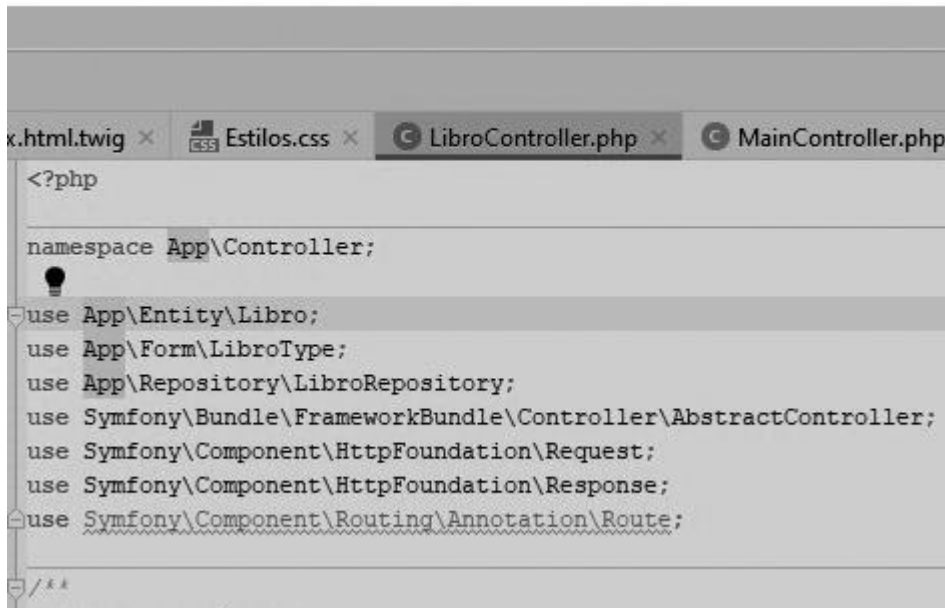
```

Figura 11 Uso de llaves

Use

Se debe emplear una declaración use para cada clase que no sea parte del espacio global de nombres. Se muestra en el siguiente ejemplo el uso de la declaración use en la propuesta de solución:

7.2.1



```
<?php
namespace App\Controller;
use App\Entity\Libro;
use App\Form\LibroType;
use App\Repository\LibroRepository;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
```

Figura 12 Empleo de cláusula use

Funciones

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guion bajo. En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior.

```
public function new(Request $request): Response
{
    $ventum = new Venta();
    $form = $this->createForm( type: VentaType::class, $ventum);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $entityManager = $this->getDoctrine()->getManager();
        $entityManager->persist($ventum);
        $entityManager->flush();

        return $this->redirectToRoute( route: 'venta_index');
    }

    return $this->render( view: 'venta/new.html.twig', [
        'ventum' => $ventum,
        'form' => $form->createView(),
    ]);
}
```

Figura 13 Ejemplo de funciones

Uso de Anotaciones

Dado a la arquitectura que emplea Symfony (modelo vista controlador) para cada vista a mostrar se necesita emplear una ruta y un controlador asociado a esta ruta. Las rutas por lo general se escriben en un archivo llamado *routes.yml* con las anotaciones se ahora la carga sobre ese archivo y en lugar de esto se establece la posibilidad de incluir la cada ruta en su controlador correspondiente como se ejemplifica en la imagen siguiente:

```

/**
 * @Route("/libro")
 */
class LibroController extends AbstractController
{
    /**
     * @Route("/", name="libro_index", methods={"GET"})
     */
    public function index(LibroRepository $libroRepository): Response
    {
        return $this->render( view: 'libro/index.html.twig', [
            'libros' => $libroRepository->findAll(),
        ]);
    }

    /**
     * @Route("/new", name="libro_new", methods={"GET", "POST"})
     */
    public function new(Request $request): Response
    {
        $libro = new Libro();
        $form = $this->createForm( type: LibroType::class, $libro);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $entityManager = $this->getDoctrine()->getManager();
            $entityManager->persist($libro);
            $entityManager->flush();
        }
    }
}

```

Figura 14 Empleo de anotaciones

3.3 Validación de la propuesta de solución

El único instrumento adecuado para determinar el status de la calidad de un producto de *software* es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del *software* o al sistema de *software* en su totalidad, con el objetivo de medir el grado en que cumple con los requerimientos (2005).

3.3.1 Pruebas de rendimiento (Carga y Estrés)

Las pruebas de rendimiento se diseñan para asegurar que el sistema pueda procesar su carga esperada. Éstas se ocupan tanto de demostrar que el sistema satisface sus requerimientos, como de descubrir problemas y defectos en el sistema. (Sommerville, 2005)

Las pruebas de carga consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el sitio esté funcionando, con el fin de detectar si el *software* instalado cumple con los requerimientos de

muchos usuarios simultáneos y también si el *hardware* es capaz de soportar la cantidad de peticiones esperadas (Guía Digital, 2008).

El entorno en el que se realizaron estas pruebas fue una computadora con un microprocesador Intel celeron (2.16Gh) con 4GB de memoria ram.

- **Usuarios:** total de usuarios.
- **# Muestras:** El número de peticiones.
- **Media:** El tiempo medio transcurrido en milisegundos para un conjunto de resultados.
- **Mín:** El mínimo tiempo transcurrido en milisegundos para las muestras de la URL dada.
- **Máx:** El máximo tiempo transcurrido en un milisegundo para las muestras de la URL dada.
- **% Error:** Porcentaje de las peticiones con errores.
- **Rendimiento:** Rendimiento medido en base a peticiones por segundo/minuto/hora.

Tabla 2 Resumen de resultados de pruebas de rendimiento

Usuarios	Muestras	Media	Mínimo	Máximo	Error(%)	Rendimiento(peticiones /segundo)
3	20	1500	1000	2000	0	2.3

3.3.2 Pruebas funcionales

Se denominan pruebas funcionales a las pruebas de *software* que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados (Dias, 2009)

Tabla 3 Caso de prueba Añadir libro

Código de caso de prueba	Nombre del caso de uso: Añadir libro
Nombre del probador: Yaisel Valdés Rodríguez	
Descripción de la prueba: Prueba a la función añadir libro	
Entada: Se introducen los siguientes datos	
Título: título del libro	
Autor: autor del libro	
Precio: precio de venta	
Editorial: Editorial a la que pertenece el libro	

Categoría: Categoría a la que pertenece
Cantidad: cantidad existente del ejemplar
Descripción: El usuario ingresará los datos, si estos están correctos se añadirá el libro en cuestión a la lista de libros; si algún dato es incorrecto se señalará dicho campo con una pequeña notificación de error hasta que se ingrese un valor adecuado.
Resultado de la prueba: Satisfactoria

Resultado de las pruebas funcionales

Para la validación de las funcionalidades del sistema se realizaron 4 iteraciones donde se encuentran un total de 24 no conformidades, 15 en la primera iteración, 6 en la segunda iteración, 3 en la tercera iteración y ninguna en la cuarta iteración. En la figura 15 se muestran los resultados obtenidos en cada una de las iteraciones.

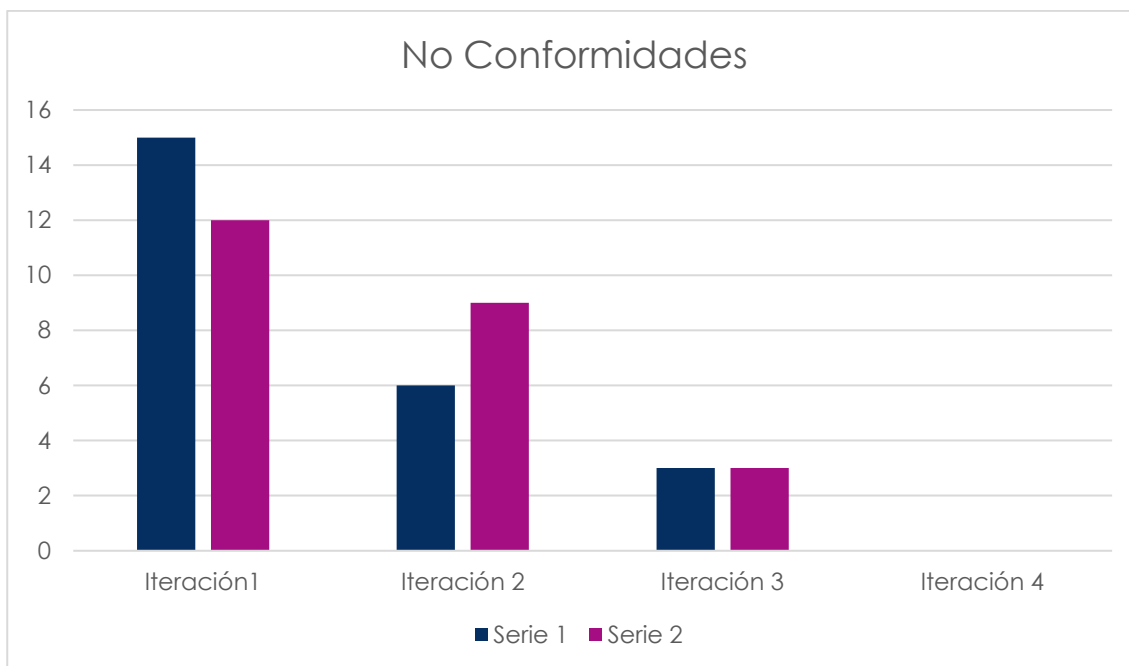


Figura 15 Comportamiento de las no conformidades por iteración

Leyenda

- Serie 1: cantidad de no conformidades detectadas
- Serie 2: cantidad de no conformidades resueltas.

Entre las no conformidades detectadas en el proceso de pruebas se encuentran:

- Los datos introducidos por el usuario de forma incorrecta son guardados en la base de datos sin validación previa.
- Opciones que no funcionan.
- El sistema no muestra mensajes de error claros.
- El sistema mantiene datos en la base de datos después de ser eliminados.

3.3.3 Pruebas de integración

Las pruebas de integración son definidas para verificar el correcto ensamblaje entre los distintos módulos que conforman un sistema informático. Las mismas validan que estos componentes realmente funcionan juntos, son llamados correctamente y transfieren los datos correctos en el tiempo preciso y por las vías de comunicación establecidas (Sommerville, 2005)

Tabla 4 Resultado de las pruebas de integración

Sistema	Funcionalidades	Resultado de la prueba
Servidor de base de datos	La conexión con el servidor de base de datos permite la realización de consultas que permiten revisar, agregar, eliminar y modificar datos en la base de datos	La aplicación se integra satisfactoriamente con el servidor de base de datos posibilitando la manipulación de datos.

3.3.4 Pruebas de seguridad

Intenta verificar que los mecanismos de protección incorporados en el sistema lo protegerán de hechos de acceso impropio. Durante las pruebas de seguridad, el responsable de la prueba desempeña el papel de un individuo que desea entrar en el sistema. Debe intentar conseguir las claves de acceso por cualquier medio, debe bloquear el sistema, negando así el servicio a otras personas. (Pressman, 2005)

Para evaluar la seguridad del sistema implementado se emplea la herramienta Acunetix WVS v8 la cual arrojó los siguientes resultados.

Tabla 5 Resultado de pruebas de seguridad

Categorías de vulnerabilidades	Cantidad de errores
Formularios HTML sin protección CSRF ¹	14
Vínculos rotos	6
Campos de contraseña con auto completamiento activado	1
Ajuste de funcionalidades	10
total	31

En la prueba realizada mediante la herramienta Acunetix WVS v8 de nivel 2, se detectaron 31 vulnerabilidades, de ellas 10 de riesgo alto, 6 de riesgo medio y 14 de riesgo bajo. En la segunda iteración no se encontraron vulnerabilidades por lo que quedaron solucionados los problemas que comprometían la seguridad de la aplicación y la integridad de la misma.

A continuación, se muestran imágenes de la aplicación:

¹ Del inglés: **Cross-site request forgery** . Falsificación de petición en sitios cruzados.

Ingrese sus credenciales

Email

Contraseña

Recordarme

Figura 16 Autenticar Usuario

Crear nuevo Libro

Titulo

Autor

Editorial

Categoría

Cantidad

Precio

Please fill out this field.

Figura 17 Crear un libro

Conclusiones del capítulo:

- La confección del diagrama de componentes permitió observar la integración de los componentes del software.
- Aplicar los estándares de codificación permitió obtener en el sistema un código legible, estándar y de fácil comprensión lo que garantiza calidad y facilita el mantenimiento.
- El proceso de verificación de la solución propuesta a través, de las pruebas realizadas permitió identificar y corregir las no conformidades detectadas para obtener un producto de mayor calidad.

Conclusiones

- Se analizaron los referentes teóricos fundamentales que sustentan las investigaciones sobre los procesos de gestión y las ventas de los libros lo cual permitió una mayor comprensión de estos procesos.
- Se realizó el diseño de un sistema informático para la gestión de las ventas y el almacenamiento de libros en la librería de la Universidad de las Ciencias Informáticas posibilitando una mejoría en el entendimiento del sistema propuesto para la solución.
- Se completó la implementación del sistema informático para la gestión de ventas y almacenamiento de los libros en la librería de la Universidad de Ciencias Informáticas.
- Fue validada la calidad del sistema informático para la gestión de ventas y almacenamiento de los libros de la librería de la Universidad de las Ciencias Informáticas.

Recomendaciones

Se recomienda implementar un módulo para la divulgación de los ejemplares disponibles para así lograr un mejor rendimiento de esta herramienta y contribuir a una mejoría en la librería

Referencias bibliográficas

Referencias

- Alvarez, Miguel Angel. 2014.** Desarrollo web. [En línea] 2 de enero de 2014. [Citado el: 7 de enero de 2020.] www.desarrolloweb.com.
- Ángel Cobo, Patricia Gómez, Daniel Pérez, Docío Rocha. 2005.** *PHP y MySQL tecnologías para el desarrollo web*. Madrid : s.n., 2005. 84-7978-706-6.
- APPS, Programando. 2019.** programandoapps.com. [En línea] 15 de junio de 2019. [Citado el: 3 de febrero de 2020.] <https://programandoapps.com/tutorial/que-son-los-frameworks-de-desarrollo/>.
- Arminet Software and Web. 2017.** [sinlib](http://sinlib.es). [En línea] 17 de enero de 2017. [Citado el: 25 de julio de 2020.] sinlib.es.
- Cabrera, Jennifer Champagne y Marina. 2019.** Capterra. [En línea] 6 de febrero de 2019. [Citado el: 10 de enero de 2020.] www.capterra.es.
- Campbely, Mary. 2007.** *Análisis y diseño de base de datos*. México : Megabyte, 2007.
- Citmatel. 2020.** [portalcuba](http://portalcuba.cu). [En línea] 15 de septiembre de 2020. [Citado el: 15 de septiembre de 2020.] portalcuba.cu.
- Date. 2001.** *Introducción a los sistemas de base de datos*. España : Pearson, 2001.
- Dias, Ernesto. 2009.** Calidad y Software. [En línea] 2009. [Citado el: 27 de mayo de 2020.] http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.
- Digital Research. 2019.** [papeles de inteligencia](http://papelesdeinteligencia.net). [En línea] abril de 2019. [Citado el: 4 de febrero de 2020.] https://papelesdeinteligencia.net/que_es_un_erp.
- . 2019. papelesdeinteligencia.net. [En línea] 23 de febrero de 2019. [Citado el: 7 de abril de 2020.] <http://papelesdeinteligencia.net>.
- Fernandez, Gustavo. 2015.** hostinger.es. [En línea] 2019 de octubre de 2015. [Citado el: 3 de febrero de 2020.] <https://www.hostinger.es/tutoriales/que-es-mysql/>.
- Fierro. 2018.** Fierro. [En línea] 27 de junio de 2018. [Citado el: 2 de septiembre de 2020.] fierro.com.ar.
- Gestión y Administración. 2016.** [gestion y administracion](http://gestionyadministracion.com). [En línea] 20 de marzo de 2016. [Citado el: 1 de noviembre de 2019.] www.gestionyadministracion.com.

- González, Lucía. 2018.** dinahosting. [En línea] 17 de agosto de 2018. [Citado el: 10 de enero de 2020.] www.dinahosting.com.
- Gonzalez, Lucia. 2016.** Programacion.net. [En línea] 12 de junio de 2016. [Citado el: 4 de febrero de 2020.] https://programacion.net/articulo/por_que_elegir_php_143.
- Guia Digital. 2008.** Guia Digital. [En línea] 2008. [Citado el: 25 de mayo de 2020.] <http://www.guiadigital.gob.cl/articulo/pruebas-de-carga>.
- integra. 2019.** integra. [En línea] 12 de mayo de 2019. [Citado el: 2020 de agosto de 15.] consultoresdesistemasdegestion.es.
- JBlogs. 2020.** SoftwarePara. [En línea] enero de 2020. [Citado el: 4 de febrero de 2020.] <https://softwarepara.net>.
- ManagementPro Inc. 2019.** managementpro. [En línea] 11 de julio de 2019. [Citado el: 5 de mayo de 2020.] managementpro.com.
- MARTÍNEZ, A. 2016.** slideshare.net. [En línea] MARTÍNEZ, A., 2016. [Citado el: 13 de enero de 2020.] : <http://es.slideshare.net/yeltsintorres18/metodologias-para-el-desarrollo-del-software>.
- Merino, Julián Pérez Porto y María. 2010.** definicion.de. *definicion.de*. [En línea] 2010. [Citado el: 3 de diciembre de 2019.] <https://definicion.de/venta/>.
- Potencier, Fabien. 2009.** *Symfony la guia definitiva*. 2009.
- Pressman. 2005.** *Software Engineering*. 2005.
- 2005.** Pruebas de Software. [En línea] 2005. [Citado el: 29 de mayo de 2020.] <http://>
- Quiroga, Lourdes Aja. 2002.** *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones*. 2002.
- Rock Content. 2019.** rockcontent.com. [En línea] Rock Content, 20 de abril de 2019. [Citado el: 26 de enero de 2020.] <https://rockcontent.com/>.
- Sánchez, Tamara Rodríguez. 2006.** *Metodología de desarrollo para la actividad productiva en la UCI*. La Habana : s.n., 2006.
- SARRETT, WENDY. 2006.** *An Introduction to Database Systems*. EE.UU : s.n., 2006. ISBN 84-95318-39-3.
- Solucionex. 2009.** solucionex. [En línea] 30 de julio de 2009. [Citado el: 4 de febrero de 2020.] <https://www.solucionex.com/blog/por-que-utilizar-symfony>.
- Sommerville. 2005.** *Ingeniería de Software. Séptima edición*. Madrid : Pearson Educación, SA, 2005. ISBN 84-7829-074-5.
- UCI. 2002.** uci.cu. [En línea] 2002. [Citado el: 15 de octubre de 2019.] www.uci.cu/?q=mision.

Vázquez, José Antonio Gallego. 2003. *Desarrollo Web con PHP y MySQL.* Madrid : Grupo ANAYA S. A, 2003. 84-415-1525-5.

Anexos

Diagrama de clases *adicionar al almacén*

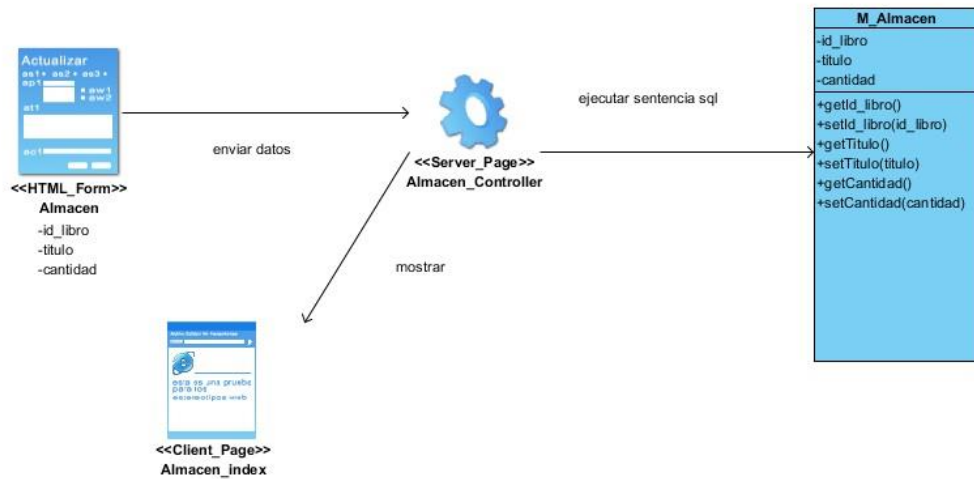


Figura 18 Diagrama de Clases adicionar almacen

Diagrama de clases *adicionar al salón*

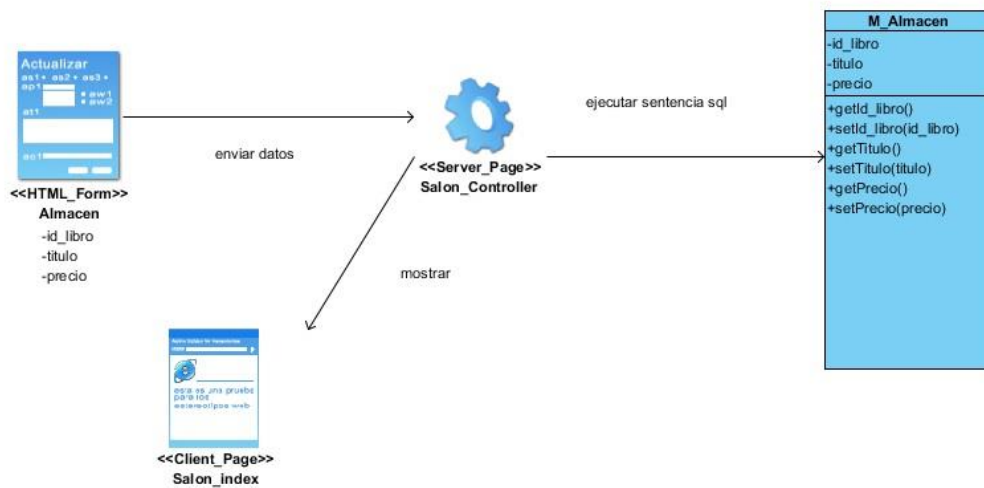


Figura 19 Diagrama de Clases adicionar salón

Diagrama de clases *adicionar usuario*

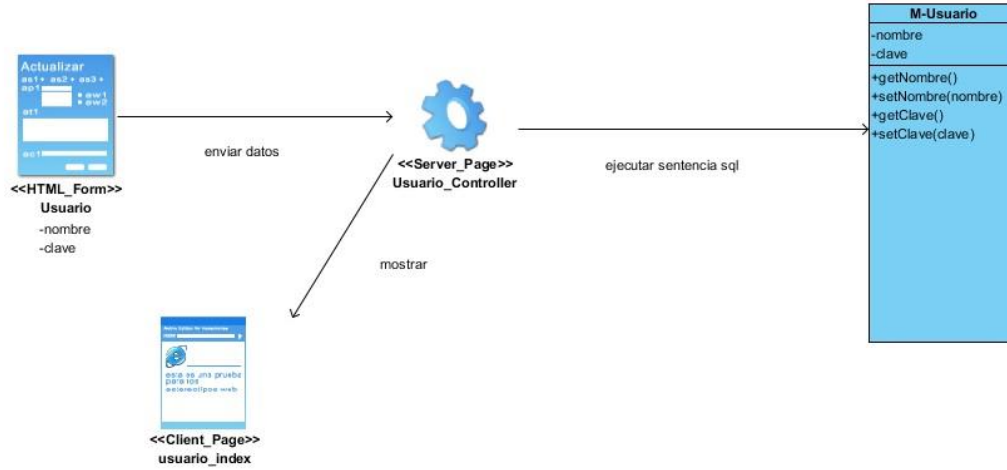


Figura 20 Diagrama de Clases *adicionar usuario*

Diagrama de clases *adicionar ventas*

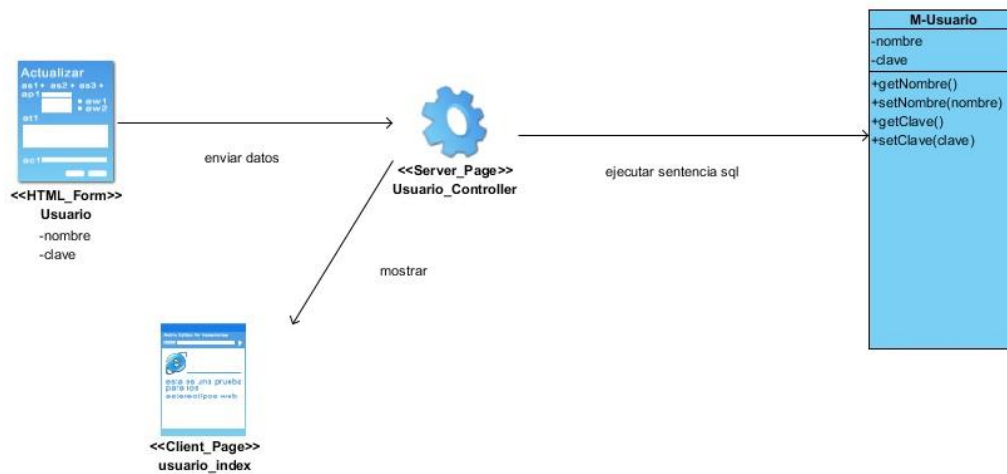


Figura 21 Diagrama de Clases *adicionar ventas*

Diagrama de clases *adicionar rol*

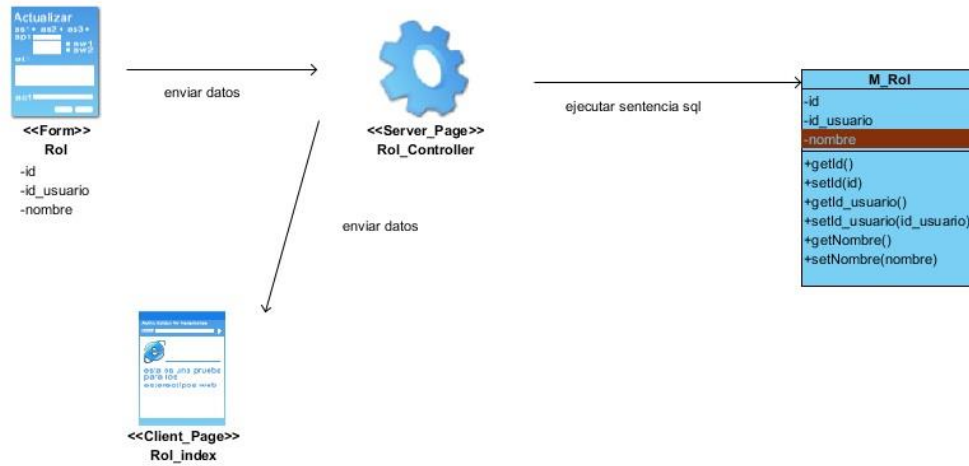


Figura 22 Diagrama de Clases adicionar rol

Diagrama de clases *eliminar almacén*

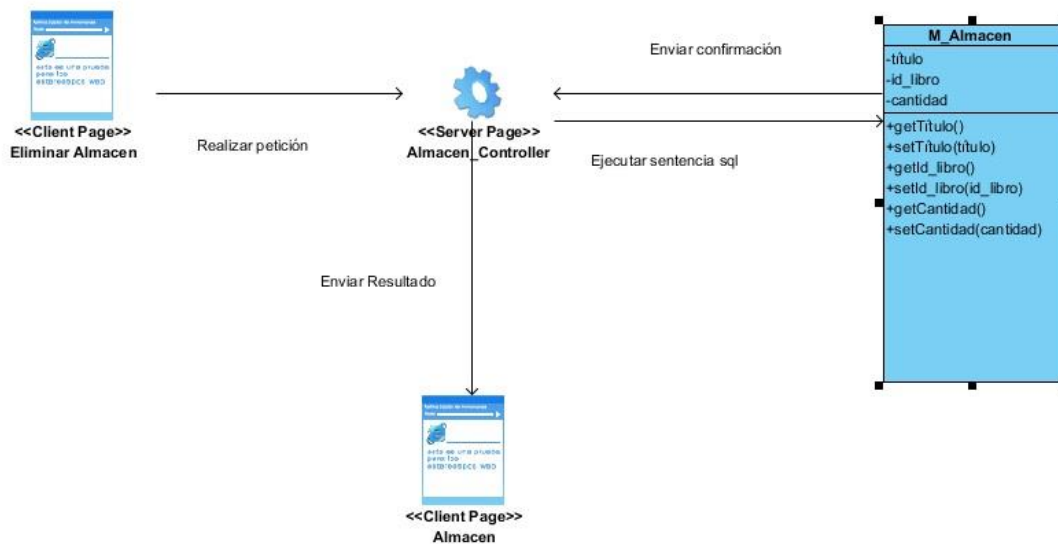


Figura 23 Diagrama de clases eliminar almacén

Diagrama de clases *eliminar salón*

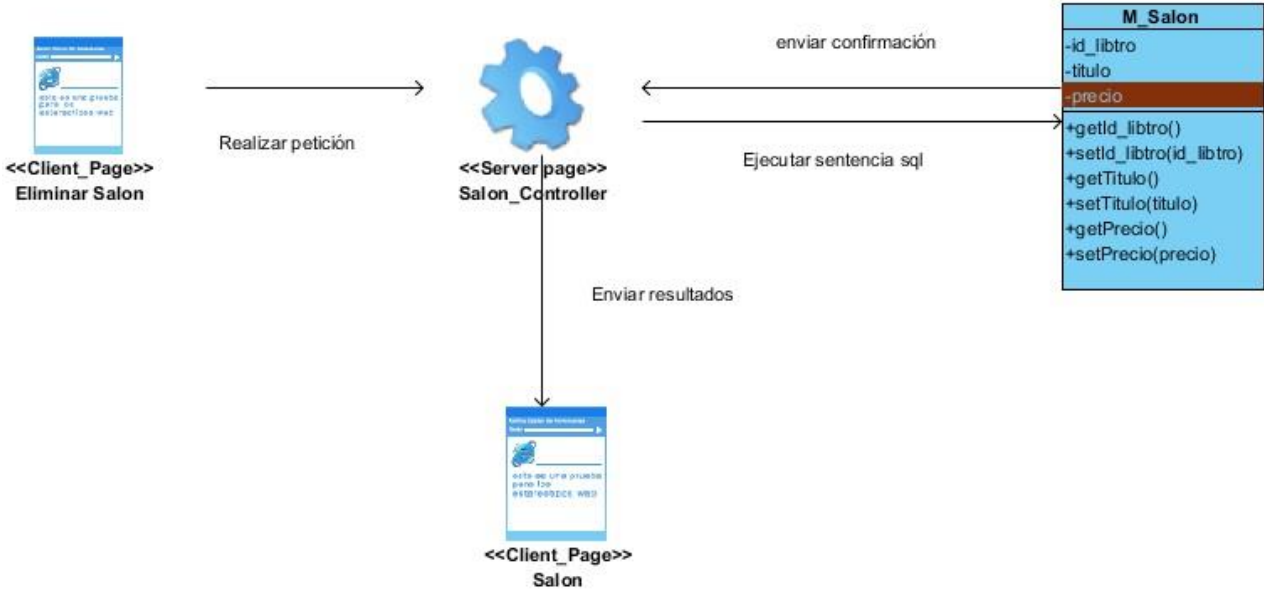


Figura 24 Diagrama de clases eliminar salón

Diagrama de clases *eliminar libro*

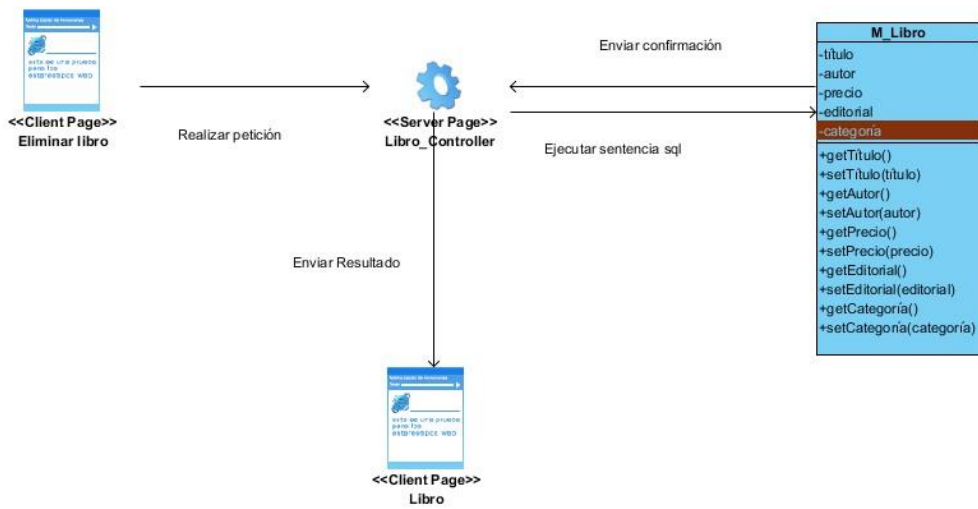


Figura 25 Diagrama de clases eliminar libro

Diagrama de clases *eliminar ventas*

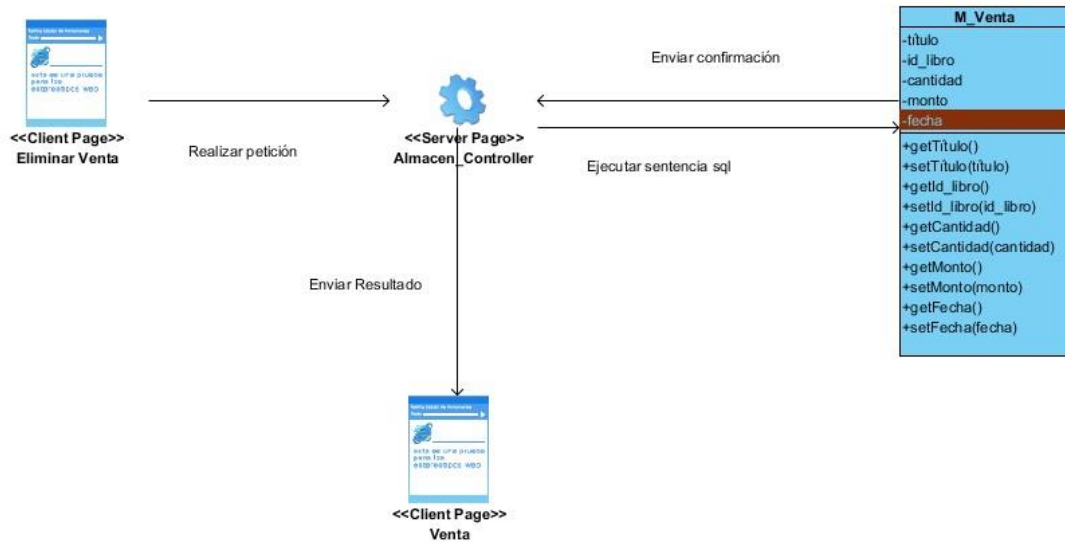


Figura 25 Diagrama de clases eliminar venta

Diagrama de clases *eliminar usuario*

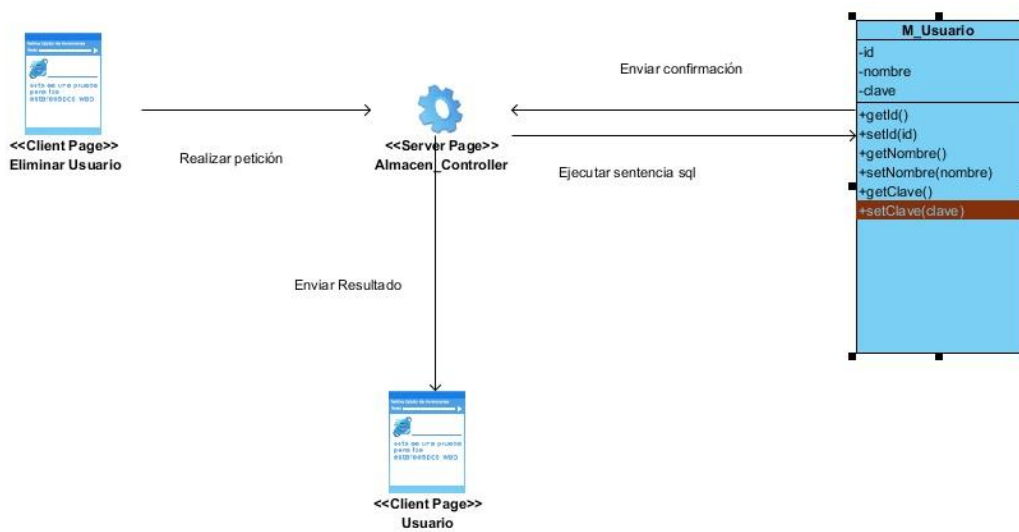


Figura 27 Diagrama de clases eliminar usuario

Diagram de clases *eliminar rol*

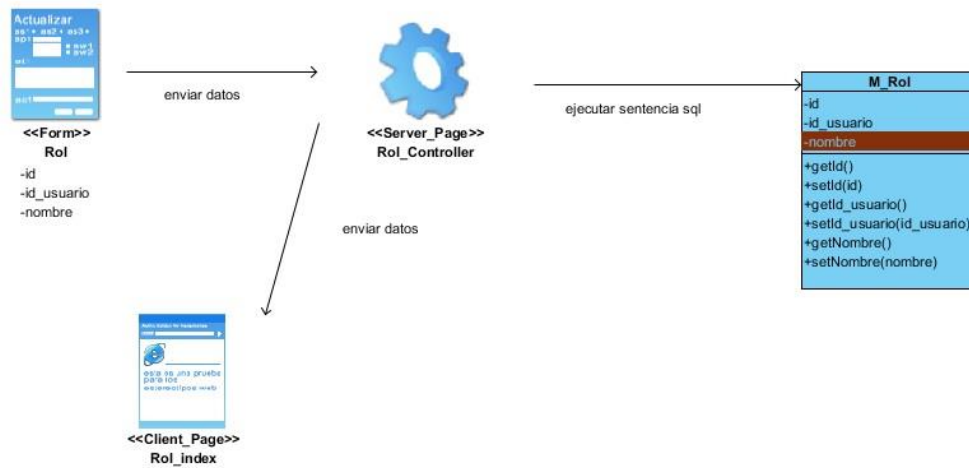


Figura 28 Diagrama de clases eliminar rol

Diagrama de clases *modificar libro*

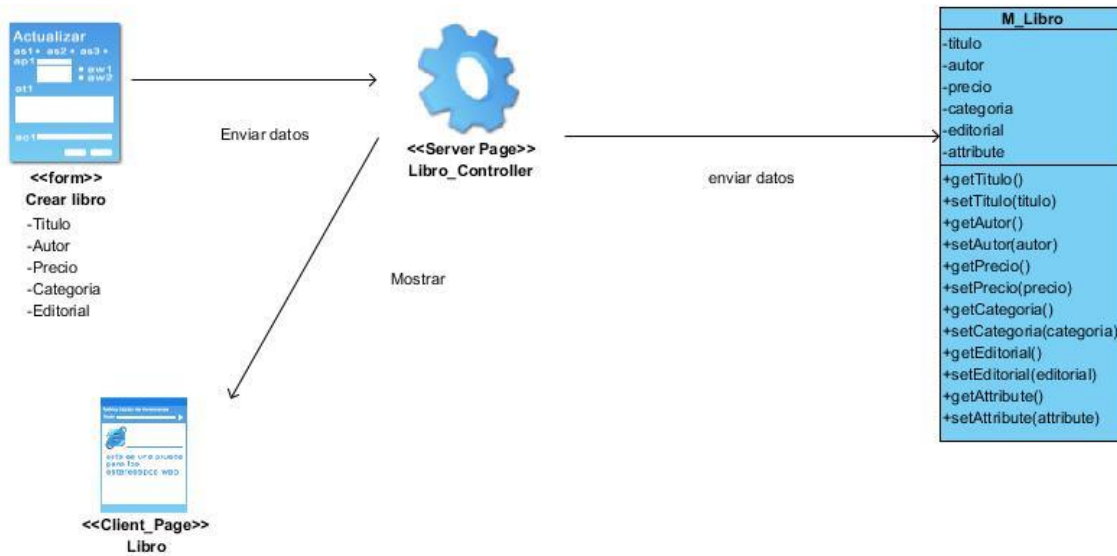


Figura 29 Diagrama de clases modificar libro

Diagrama de clases *modificar almacén*

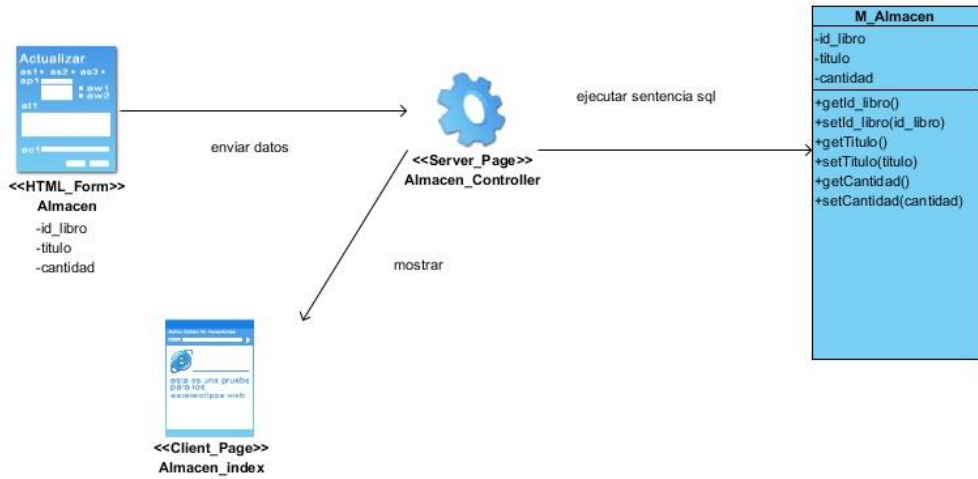


Figura 30 Diagrama de clases modificar almacén

Diagrama de clases *modificar salón*

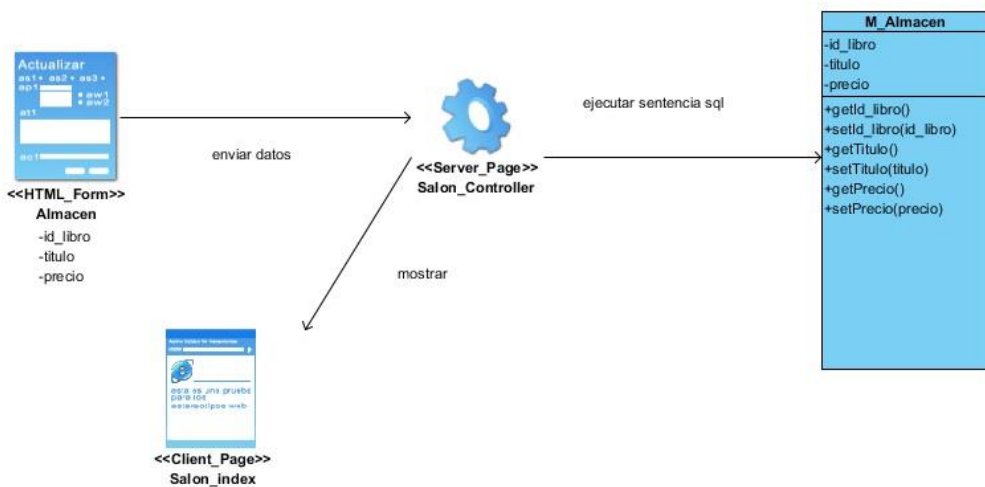


Figura 31 Diagrama de clases modificar salón

Diagrama de clases modificar usuario

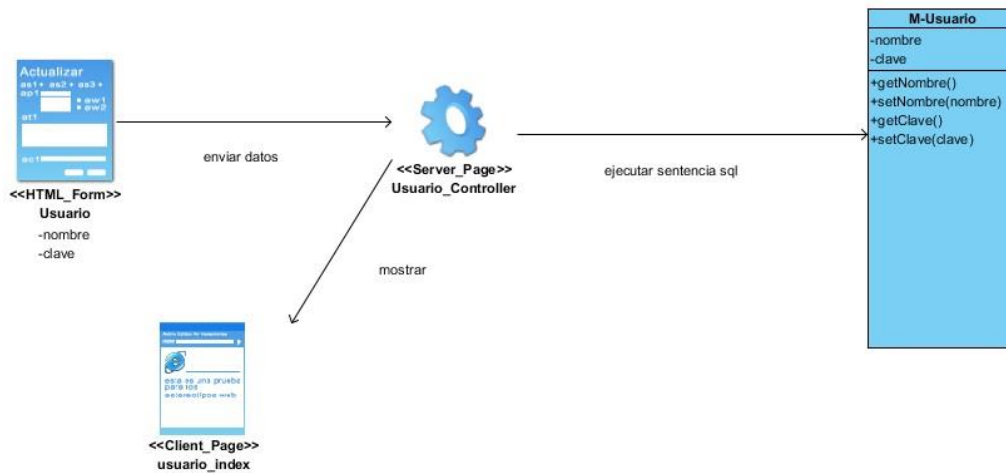


Figura 32 Diagrama de clases modificar usuario

Diagrama de clases modificar ventas

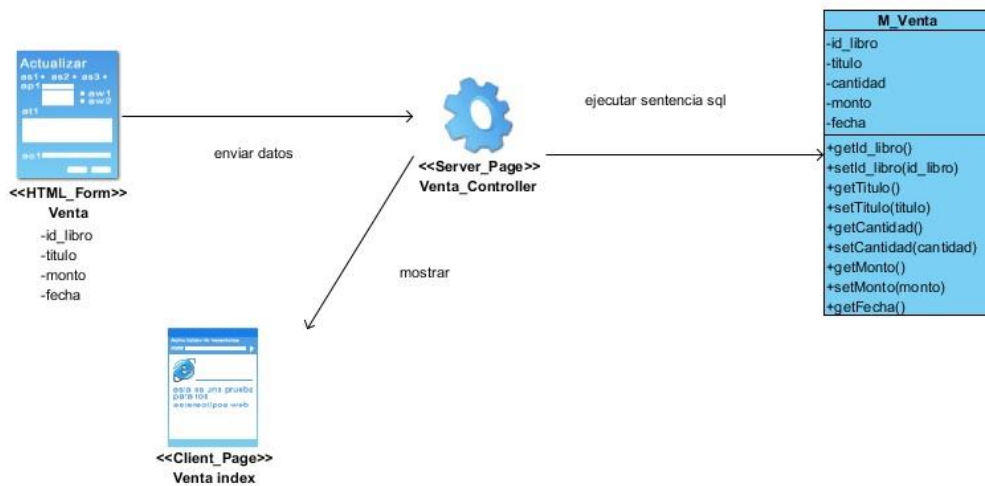


Figura 33 Diagrama de clases modificar venta

Diagrama de clases modificar rol

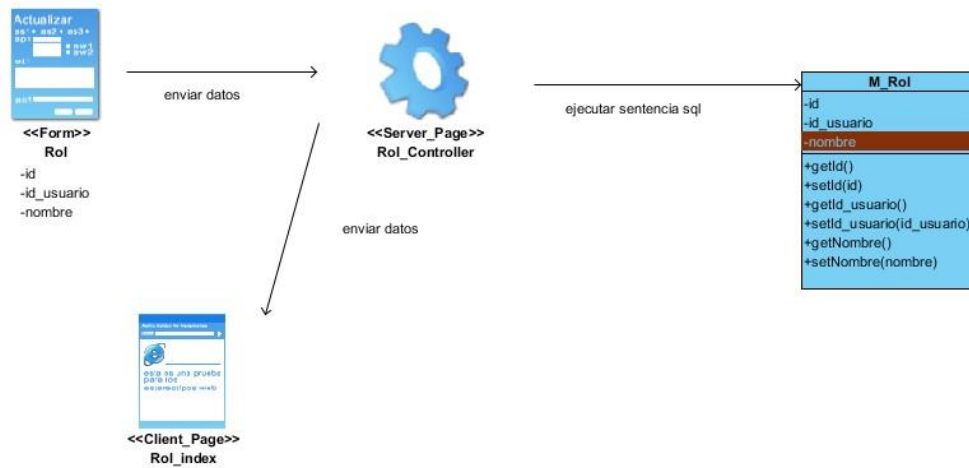


Figura 34 Diagrama de clases modificar rol

Preguntas de la entrevista:

- ¿Qué procesos se realizan en esta librería?
- ¿Cómo realizan estos procesos?
- ¿De qué forma se archiva el inventario?
- ¿Qué contiene una tarjeta de estiva?
- ¿Qué información maneja cada empleado?