

Universidad de las Ciencias Informáticas

Facultad 4



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Título: Módulo de reportes para Plataforma de recursos de la
Editorial Universitaria Félix Varela.

Autor(es):

Oscar Luis Ambruster Jiménez


Tutor(es):

Ing. Celia Indira Hidalgo Tagle

Ing. Yirka Céspedes Boch

Junio del 2017

"Año 59 de la Revolución"



*Las oportunidades grandes
nacen de haber sabido
aprovechar las pequeñas.*

Bill Gates

Declaración de autoría

Declaro ser único autor del presente trabajo y delego a la Universidad de las Ciencias Informáticas los derechos patrimoniales de este, con carácter exclusivo.

Para que así conste firmamos la presente a los ___ días del mes de ___ del año _____.

Firma del Autor

Oscar Luis Ambruster Jiménez

Firma del Tutor

Ing. Celia Indira Hidalgo Tagle

Firma del Tutor

Ing. Yirka Céspedes Boch

Dedicatoria

A mis padres por creer en mí, por su apoyo incondicional en todo momento, por sus consejos, por todo el amor que me han dado, por ser los mejores padres del mundo. Por las cosas que han enfrentado para que nunca me faltase nada, por la educación que me dieron. Son un ejemplo para mí, gracias por ser mis padres, gracias a ustedes soy hoy esta persona.

A mi hermana por ser la mejor hermana del mundo, por quererme mucho y apoyarme siempre.

A mis abuelos Ernesto y Emelina por todo el amor que siempre me han dado, por su apoyo y cuidado desde que era un bebé, gracias por ser los mejores abuelos del mundo.

A mi bisabuela por el amor inmenso que siempre me ha dado, por sus consejos y cuidados desde pequeño.

A mi bisabuelo que hoy quisiera estuviese aquí viéndome graduarme, por todo el amor que me dió, espero donde este se encuentre orgulloso de mí.

A mi abuelo José por todo el apoyo y cariño que siempre me dió a pesar que estaba lejos de mí.

A mi tío Miguel que esperaba estuviese presente el día de mi graduación celebrando conmigo, por su apoyo y amor siempre desde que era pequeño a pesar de la distancia. Por toda la ayuda y apoyo que me dió desde que comencé a estudiar en la UCI, donde quieras que estés este título es por ti.

A mi novia por el amor y apoyo que siempre me ha dado, te amo mucho.

Agradecimientos

A mis padres, mi hermana, mi bisabuela y mis abuelos por confiar en mí y apoyarme siempre.

A todos mis compañeros de aula, a los que se quedaron en el camino, en especial al Gato y los que hoy están graduándose conmigo por compartir esta aventura.

A mis amigos más cercanos en la uci Adrián, Pinar, Javier, El Flaco, Lennon, Carrión, Carlo, Yuniel, Jose, Norly, Manuel, Ernesto, Héctor por compartir momentos inolvidables.

A mis amigos de Villa Clara, Vladito y Daniel por mantener la comunicación y apoyarme siempre desde que comencé a estudiar en la UCI.

A mis entrenadores y compañeros del equipo de futsal de la UCI por compartir momentos maravillosos e inolvidables, por cumplir juntos ese sueño de ser campeones nacionales.

Resumen

Los sistemas de reportes tienen como objetivo principal mostrar una visión general de la situación de la empresa. Con los generadores de reportes se puede definir el diseño y la forma en que la información será visualizada de manera estructurada y/o resumida en un diseño atractivo y fácil de interpretar por los usuarios. La Plataforma de recursos de la Editorial Universitaria Félix Varela no cuenta con un módulo de reportes que permita conocer información sobre el acceso de los usuarios a los contenidos publicados en la misma. Como solución se propone desarrollar un módulo de reportes que brinde información al administrador y/o editor sobre las distintas actividades que ocurren en la plataforma. Para ello se decide integrar el Servidor Dinámico de Reportes (SDR) a la plataforma porque este brinda una capa de servicios que facilita su integración con cualquier otro sistema. Para el desarrollo de la presente investigación se emplean los métodos de investigación científica, entre los teóricos se utilizó histórico-lógico, analítico-sintético y modelación, y de los empíricos la entrevista. Como resultado se logró un módulo de reportes integrado a la Plataforma de recursos de la Editorial Universitaria Félix Varela, que brinda información sobre el acceso a los contenidos publicados. Se realizaron pruebas funcionales y de integración al software obteniendo resultados satisfactorios.

Palabras Claves: Reportes, Sistemas de reportes, Generadores de reportes

ÍNDICE

INTRODUCCIÓN	11
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	16
1.1 Introducción.....	16
1.2 Conceptos asociados al dominio del problema	16
1.3 Estudio de soluciones similares	18
1.4 Selección de la metodología	21
1.5 Herramientas y tecnologías	23
1.6 Conclusiones parciales	31
CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN	32
2.1 Introducción.....	32
2.2 Propuesta de solución	32
2.3 Modelo conceptual	34
2.4 Especificación de los requisitos del sistema	35
2.4.1 Requisitos Funcionales (RF):	35
2.4.2 Requisitos no funcionales (RNF):	37
2.5 Usuarios del sistema.....	37
2.6 Historias de usuario	38
2.7 Descripción de la arquitectura y los patrones de diseño del sistema	40
2.7.1 Arquitectura del sistema	40
2.7.2 Patrones de diseño	41
2.7.2.1 Patrones GRASP	41
2.7.2.2 Patrones GOF	43
2.8 Diagrama de clase del diseño (DCD).....	44
2.9 Diagrama de clases de secuencia del diseño (DCSD)	46
2.10 Diagrama de datos	46
2.11 Diagrama de despliegue	47
2.12 Conclusiones parciales.....	48
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN PROPUESTA	49
3.1 Introducción	49

3.2 Diagrama de componentes	49
3.3 Estrategia de pruebas	50
3.3.1 Método de Prueba	51
3.3.2 Resultados de las pruebas de caja negra	55
3.3.3 Resultados de las pruebas de integración	55
3.4 Conclusiones parciales	58
CONCLUSIONES	59
RECOMENDACIONES	60
REFERENCIAS BIBLIOGRÁFICAS	61
ANEXOS	65

ÍNDICE DE FIGURAS

FIGURA 1: DIAGRAMA CONCEPTUAL DEL NEGOCIO	34
FIGURA 2: FUNCIONAMIENTO DEL PATRÓN MVC	41
FIGURA 3: EJEMPLO DEL USO DEL PATRÓN CONTROLADOR.....	43
FIGURA 4: DCD-REPORTE DE LIBROS.....	45
FIGURA 5: DCD-EXPORTAR A FORMATO .PDF, .XLS Y .JPG LOS REPORTES	45
FIGURA 6: DCSD-REPORTE DE LIBROS.....	46
FIGURA 7: DCSD-EXPORTAR A FORMATO .PDF LOS REPORTES.....	46
FIGURA 8: DIAGRAMA DE DATOS	47
FIGURA 9: DIAGRAMA DE DESPLIEGUE	47
FIGURA 10: DIAGRAMA DE COMPONENTES DE LA PROPUESTA DE SOLUCIÓN.....	49
FIGURA 11: NIVELES DE PRUEBAS	50
FIGURA 12: RESUMEN DE LAS NO CONFORMIDADES DETECTADAS	55

ÍNDICE DE TABLAS

TABLA 1: FASES DE LA VARIACIÓN DE AUP PARA LA UCI.....	23
TABLA 2: DESCRIPCIÓN DE LOS REQUISITOS FUNCIONALES DEL SISTEMA.....	35
TABLA 3: DESCRIPCIÓN DE LOS USUARIOS DEL SISTEMA.....	38
TABLA 4: HU GENERAR REPORTE DE LIBROS	38
TABLA 5: HU EXPORTAR A FORMATO .PDF LOS REPORTEES	39
TABLA 6: DCP-GENERAR REPORTE DE LIBROS	52
TABLA 7: DCP-EXPORTAR A FORMATO .PDF, .XLS Y .JPG LOS REPORTEES.....	54
TABLA 8: CDP-GENERAR REPORTE DE LIBROS	56
TABLA 9: CDP-EXPORTAR A FORMATO .PDF, .XLS Y .JPG LOS REPORTEES.....	57

INTRODUCCIÓN

Conforme pasa el tiempo, la información que se maneja o se genera dentro de los procesos de una empresa u organización, se ha convertido en la base fundamental para la toma de decisiones de la misma. De igual forma es válido destacar que en ocasiones esta información no se presenta con la calidad requerida, a la vez que, debido al cúmulo de datos, resulta tedioso y difícil de entender por parte del usuario.

Sobre esta base y con el objetivo de facilitar el proceso de toma de decisiones se han ido creando mecanismos y formas de representar la información que facilite el entendimiento de la misma. Uno de estos mecanismos son los denominados reportes, cuyo objetivo fundamental es mostrar la información en un formato atractivo, organizado y fácil de entender por parte del usuario, valiéndose de gráficos, diagramas y tablas para la representación de los datos. A través de los reportes es posible comprobar si los objetivos trazados para un período determinado fueron cumplidos y en caso contrario determinar dónde radican los principales problemas.

La Editorial Universitaria Félix Varela (EUFV) es la encargada de la realización del proceso de edición de libros académicos que se corresponden con los programas de estudio de las carreras del Sistema de Educación Superior cubano. Entre los servicios brindados por la editorial se encuentra la publicación de libros de texto correspondientes a los planes de estudio de 54 carreras universitarias, vinculadas a las materias de Agronomía, Artes, Derecho, Filosofía, Historia, Geografía, Ingenierías, Educación, Psicología, Ciencias Exactas, de la Computación y la Información, entre otras. Además, gestiona la venta de libros y otros materiales poligráficos a los centros de educación superior cubanos, en ferias y eventos (1). Con el objetivo de mejorar sus servicios, la EUFV en conjunto con la Universidad de las Ciencias Informáticas (UCI) se propone desarrollar una plataforma para la gestión de libros y recursos utilizando tecnologías libres. Dicha plataforma se desarrolla en el Centro de Tecnologías para la Formación (Fortes) adscrito a la Facultad 4, que tiene como objetivo desarrollar productos y servicios para la implementación de soluciones de formación aplicando las Tecnologías de la Información y las Comunicaciones, a todo tipo de instituciones con diferentes modelos de formación y condiciones tecnológicas.

La plataforma permitirá la gestión y publicación de libros tanto de la misma editorial como de otras editoriales. Los libros se podrán clasificar en categorías y también podrán ser consultados por las instituciones, entiéndase por institución universidades o empresas, que contraten un número determinado de libros por un período específico. Tendrá espacios para que los profesores de las instituciones y los autores de los libros puedan subir recursos educativos a la plataforma. Los autores podrán dar

seguimiento a sus libros respondiendo las preguntas que realicen otros usuarios sobre los mismos. Una de las funcionalidades del proyecto consiste en permitir a los editores y administradores de la plataforma conocer el nivel de aceptación de un libro a través de un reporte de las consultas del mismo por parte de los usuarios. Se puede obtener la siguiente información de un libro determinado:

- Cantidad de visitas realizadas al libro.
- Cantidad de usuarios que tienen el libro como favorito.
- Evaluación general del libro.

Sin embargo, este reporte no permite a un editor o administrador conocer información general de los libros existentes en la plataforma o en una editorial específica, o sea, que para seleccionar el libro más visitado es obligatorio realizar un reporte por cada uno de los libros existentes en la plataforma y luego realizar un análisis de cada uno de estos. Otra de las deficiencias detectadas al reporte consiste en la ausencia de datos referentes al acceso de los usuarios a la plataforma, por ejemplo:

- La cantidad de usuarios que han accedido a la plataforma desde su publicación.
- Promedio de visitantes.
- Temas más solicitados por los usuarios.
- Los usuarios que más acceden a la plataforma.

A los elementos antes mencionados podemos agregar otros referentes a los contenidos que se publican en la plataforma, por ejemplo:

- Los libros con mayor y menor aceptación.
- Los recursos con mayor y menor aceptación.
- Los autores favoritos.
- Las editoriales más destacadas.

Al no contar con estos datos tanto el administrador de la plataforma como los editores no pueden realizar un análisis profundo y certero de las actividades que se llevan a cabo en cada una de sus áreas.

A partir de la problemática descrita surge como **problema de la investigación** la siguiente interrogante: ¿Cómo contribuir a la toma de decisiones sobre la Plataforma de recursos de la Editorial Universitaria Félix Varela?

Por lo que se define como **objeto de estudio**: Módulos de reportes en plataformas.

Para lo cual se plantea como **objetivo general** desarrollar un módulo de reportes para contribuir a la toma de decisiones de la Editorial Universitaria Félix Varela.

La investigación se enmarca en el reporte de la información generada de las distintas actividades que tienen lugar en la Plataforma de recursos de la Editorial Universitaria Félix Varela como **campo de acción**.

A partir del objetivo general definido se derivan las siguientes **preguntas científicas**:

- ¿Cuáles son los elementos teóricos y principales tendencias de los reportes y sistemas de reportes?
- ¿Cuáles son las herramientas y tecnologías existentes para el desarrollo de sistemas de reportes?
- ¿Cuáles son los requisitos con los que debe cumplir el módulo de reportes para la Plataforma de recursos de la Editorial Universitaria Félix Varela?
- ¿Cómo implementar el módulo de reportes para la Plataforma de recursos de la Editorial Universitaria Félix Varela?
- ¿Cómo realizar pruebas de software al módulo de reportes para la Plataforma de recursos de la Editorial Universitaria Félix Varela?

Para dar cumplimiento a las preguntas científicas, se definen las siguientes **tareas**:

- Análisis de los elementos teóricos y principales tendencias de los reportes y sistemas de reportes.
- Caracterización de la metodología, herramientas y tecnologías a utilizar en el desarrollo del módulo de reportes.
- Descripción de las características del sistema y levantamiento de los requisitos funcionales y no funcionales.
- Diseño de los prototipos de interfaz de usuario del sistema e implementación de las funcionalidades del sistema.
- Realización de las pruebas funcionales al módulo de reportes.

Métodos Teóricos

- **Histórico-Lógico:** se realizó un análisis sobre cómo ha evolucionado la utilización de los reportes hasta la actualidad y de igual forma las herramientas y tecnologías utilizadas en el desarrollo de módulos de reportes.

- **Analítico-Sintético:** se utilizó para el análisis de documentos, materiales y temas relacionados con el desarrollo de sistemas de reportes. Permitió la síntesis de conceptos, teorías y técnicas empleadas.
- **Modelación:** permitió reflejar la estructura, relaciones entre objetos, las actividades que intervienen en el proceso, relaciones internas y características de la solución a través de diagramas.

Métodos empíricos

- **Entrevista:** permitió entender cómo se realiza actualmente el control del acceso a los contenidos existentes en la Plataforma de recursos de la Editorial Universitaria Félix Varela. Además, se utilizó para identificar los requisitos funcionales y no funcionales que debe cumplir el sistema a desarrollar. Permitió conocer el funcionamiento de la herramienta seleccionada para la generación de los reportes. Ver anexo 1.

El documento consta de 3 capítulos que estructuran el informe de investigación, como se muestra a continuación:

Capítulo 1. Fundamentación teórica:

En este capítulo se precisan los elementos teóricos que conforman la base de la investigación y del desarrollo del tema propuesto. Se realiza un estudio sobre los reportes y los sistemas de reportes, así como la importancia de los mismos a nivel nacional e internacional. Se describen además las tecnologías actuales de desarrollo utilizadas para el análisis, diseño e implementación del sistema sobre las cuales se sustenta la propuesta.

Capítulo 2. Descripción de la propuesta de solución:

En este capítulo se describe el sistema que se propone, a partir de la metodología utilizada en la investigación. Se elaboran algunos de los siguientes artefactos propuestos por la metodología AUP en su variación para la UCI en las fases de Inicio y Ejecución: modelo del dominio, especificación de requisitos, definición y descripción de las historias de usuario, diseño de los prototipos de interfaz de usuario, descripción de la arquitectura y los patrones utilizados para el diseño del componente.

Capítulo 3. Implementación y pruebas de la solución propuesta:

En este capítulo se describe el flujo de implementación y las pruebas de la solución de esta investigación. Para ello se toma como punto de partida los conocimientos adquiridos en los capítulos anteriores. En esta parte de la investigación se construye el sistema teniendo en cuenta la arquitectura y los requisitos

definidos para el mismo. Se presenta además un resumen de las pruebas realizadas al software para garantizar su correcto funcionamiento.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se definen los conceptos fundamentales asociados al dominio del problema, para una mayor comprensión. Se realiza un estudio sobre los sistemas de reportes, así como la importancia de los mismos a nivel nacional e internacional. Se analizan soluciones existentes que pueden aportar elementos para la solución del trabajo y se describen las herramientas a utilizar en la implementación del módulo.

1.2 Conceptos asociados al dominio del problema

Reportes

Un reporte es una noticia o informe, que puede ser impreso, digital o audiovisual y pretende transmitir información de distintos tipos. Existen diversos tipos de reportes como los divulgativos, persuasivos, informativos y de otros tipos (2).

El reporte es un documento informativo que sirve para comunicar información que sea relevante, este puede ser visual o textual. Puede ser un material creado por una empresa, organización o un trabajo de clases que sirve para dar mayor información sobre un tema en específico. Entre la información que se puede mostrar en un reporte se encuentran: resultados básicos de una investigación, resultados de un proyecto técnico, la revisión del estado del arte de un tema en particular, la propuesta para una tesis, manuales, diagramas, notas de cursos o de seminario (3).

En el ámbito de la informática los reportes son informes que organizan y exhiben la información contenida en una base de datos. Su función es dar un formato determinado a los datos para mostrarlo a través de un diseño más atractivo y a su vez, fácil de interpretar por los usuarios (4).

Tipos de reportes

Los reportes pueden ser clasificados de distintas maneras, a continuación, se hace referencia a ejemplos de tipos de reporte:

- **Técnicos:** estos reportes son emitidos para entidades privadas o públicas a fin de responder a su pedido de investigación o estudio. Para la confección de estos informes se acude al rigor y se busca que sean accesibles a todos sus receptores, que no siempre tienen un manejo del lenguaje técnico de la entidad.
- **Científico:** estos son dirigidos a ciertos sectores, que sí manejan la jerga del tema a investigar. Es por esto que abundan los tecnicismos y la rigurosidad en el lenguaje y sus temáticas tienen que ver con algún área científica.

- **De divulgación:** los contenidos de estos informes son presentados para el público general. Es por esto que se caracterizan por tener un lenguaje para quienes poseen un nivel cultural medio.
- **Mixto:** estos informes son confeccionados para el público en general y, al mismo tiempo para una organización en particular, de allí su nombre. Al destinarse a un público tan amplio, el lenguaje no puede abundar en tecnicismos, sino ser más bien coloquial pero culto (5).

Formas de representar la información

Todos los días sobre todo al presentar un trabajo específico se utilizan diversas formas de mostrar la información, sin embargo, no siempre se está consiente de cuáles son o como se llaman. Entre las formas básicas de presentación de la información se encuentran (6):

Textual: es útil cuando se tiene poca información, muy utilizado en las revistas y periódicos. Consiste en introducir dentro del texto los datos que se consideran importantes o sobre los que se quiera llamar la atención del lector.

Semitabular: útil cuando se tienen pocos datos clasificados en forma simple. Se hace una explicación o se introduce una idea y cuando el lector está listo para comprenderla se le presentan los datos que la respaldan o fortalecen y se vuelve a reafirmar el concepto o idea comentando los datos.

Tabular: es la más utilizada por los investigadores para comunicar los datos o la información de los estudios realizados y lo hacen mediante el uso de cuadros. Existen dos tipos de cuadros:

1. Generales o de referencia: se utilizan para presentar resultados de censos, encuestas y otros con el mayor detalle. Abarcan toda la información disponible y se utilizan para análisis posteriores.
2. De resumen: son pequeños y hechos de forma que se resalten los datos con la mayor intensidad posible. Su propósito es facilitar el análisis de los datos.

Gráfica: los gráficos llaman la atención acerca de ciertas variaciones y similitudes en los datos numéricos y permiten apreciarlas mejor para obtener buenas conclusiones. Deben ser sencillos, cómodos de interpretar y adecuados al tipo de información disponible. Existen varios tipos de gráficos entre ellos se encuentran de barras, circular, lineal, bastones horizontales y verticales.

Luego del estudio de los tipos de reportes y formas de representar la información se decide utilizar los informes mixtos, seleccionando como formas para representar la información contenida en los mismos tabular y gráfica.

Sistemas de reportes

Los sistemas de reportes tienen como objetivo principal mostrar una visión general de la situación de la empresa. Consecuentemente, estos muestran la situación de las operaciones regulares de la empresa para que los directivos puedan controlar, organizar, planear y dirigir (7).

Los sistemas de reportes utilizan generadores de reportes, estos tienen en las bases de datos su principal fuente de alimentación, ya que a partir de la información almacenada en la misma se realizan consultas para obtener información en forma de reporte. Con estos generadores se puede definir el diseño y la forma en que la información será visualizada de manera estructurada y/o resumida en un diseño atractivo y fácil de interpretar por los usuarios.

1.3 Estudio de soluciones similares

Para el desarrollo y el buen funcionamiento del sistema se hace necesario un análisis de los sistemas de reportes que poseen características semejantes a las necesidades planteadas en el problema de la investigación y ver si es posible adaptarlas a los requerimientos del actual sistema en desarrollo. Hoy en día existen diversas soluciones tanto en Cuba como en el mundo, entre ellas se encuentran:

Janium

Es un sistema integrado de gestión muy completo e innovador. Es fácil de utilizar y de administrar. Permite automatizar en forma integrada la biblioteca tradicional, y la biblioteca digital. Puede administrar varias sedes en un mismo sistema 100% vía Web. Puede trabajar en diferentes idiomas, simultáneamente. Es ideal para instituciones que tienen varias ubicaciones distribuidas en diferentes lugares o aquellas instituciones que tienen usuarios que consultan documentos desde diferentes sitios a todas horas. Janium es un sistema diseñado completamente como una aplicación basada en WEB y Cloud Computing ¹(8).

Entre sus características principales se encuentran:

- Arquitectura de cómputo multi-capa, todo el sistema trabaja en ambiente web.
- Soporta los sistemas administradores de bases de datos relacionales más utilizados en el ambiente tecnológico como: PostgreSQL, MS SQL Server, y Oracle.
- Tanto el servidor como las estaciones de trabajo pueden funcionar en cualquier ambiente de cómputo: Windows, OS X, Linux, etcétera.

¹ Cloud Computing: (Computación de la nube) : es un término general para denominar cualquier cosa que tenga que ver con la provisión de servicios de hospedaje a través de Internet.

- Toda la información capturada es indizada² en línea en tiempo real por lo que no es necesario ejecutar ningún procedimiento posterior o tarea de mantenimiento en el sistema.
- Soporte de objetos digitales en diferentes formatos (JPG, JPG2, PDF, TXT, MPEG, WAV).
- Índices configurables y con excelentes tiempos de respuesta.
- Intercambio de información con XML, ISO 2709 y ASCII.
- El sistema está disponible en diferentes idiomas (8).

Koha

Software diseñado en Nueva Zelanda de código abierto desarrollado y actualizado por Liblime Koha. Está conformado por varios módulos que ayudan al responsable de la unidad de información en el control de la colección y en la interacción con sus usuarios. Su interfaz es de fácil uso, mediante su plataforma tipo Web permite el acceso tanto al área administrativa como a la parte pública (9).

Módulos de Koha:

Circulación: Su cómoda interfaz hace de este servicio una tarea ágil y eficaz. Permite el control de los préstamos, la identificación, ubicación y estado de cualquier publicación o usuario.

Catalogación: Un sistema de registro flexible hace de la catalogación de documentos un trabajo sencillo y rápido.

Autoridades: Posee control estricto ayudado por el sistema que permite monitorear cualquier detalle en la normalización de su información.

OPAC: Un sistema de búsqueda en plataforma Web que permite de manera dinámica recuperar la información, reservar libros y renovar préstamos.

D.S.I: Este módulo agiliza su servicio de alerta, para incrementar las consultas al sistema.

Informes: Cuenta con un conjunto completo de estadísticas y reportes personalizables (9).

Principales ventajas del software:

- Permite realizar un amplio número de reportes.
- Los resultados pueden ser exportados en archivo Excel (Office) o en Calc (OpenOffice.org).
- Se puede ingresar al sistema vía web o vía intranet.

² Indizada: Registrar ordenadamente datos e informaciones para elaborar su índice.

- Garantiza seguridad en la integridad de los registros.
- Permite el acceso concurrente e ilimitado de usuarios (9).

VERSAT Sarasola

Se distingue por ser el primer sistema de contabilidad cubano certificado, según las nuevas normativas establecidas por los Ministerios de Finanzas y Precios y de la Informática y las Comunicaciones, para este tipo de Software. Es un sistema económico integrado constituido por 10 módulos o subsistemas que incluyen configuración y seguridad, contabilidad general y de gastos, costos y procesos, finanzas y caja. Además, en el proceso intervienen activos fijos, planificación y presupuestos, control de inventarios, pago de salario (nómina), facturación y generador de reportes (10).

El generador de reportes, también llamado “Complementos del Versat”, es la herramienta que permite obtener información sin mayores complejidades de uso. Interactúa con Microsoft Excel, utilizando sus facilidades para el diseño de información. La concepción de estas funciones está ligada a los distintos subsistemas del Versat Sarasola.

Características del Versat Sarasola:

- Herramienta para la planificación económica, el control y el análisis de gestión.
- Diseñado para su empleo en cualquier tipo de entidad empresarial o presupuestada.
- Permite llevar el control y registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las entidades, así como exponer el estado financiero y toda la información económica y contable en este universo.
- Rapidez y fiabilidad, a partir de la configuración del proceso de contabilización de los documentos primarios y de las propias posibilidades de trabajo contenidas en cada subsistema (10).

Luego del estudio realizado a los sistemas homólogos Janium, Koha y Versat Sarasola se decide no hacer uso de estas soluciones debido a que no son compatibles con las tecnologías sobre las que está desarrollada la Plataforma de recursos de la Editorial Universitaria Félix Varela. Esta última utiliza el marco de trabajo Symphony y la arquitectura Xalix. Además, el sistema Janium cuenta con licencia privativa. Aunque no pueden ser utilizadas estas herramientas como solución al problema planteado, su estudio permitió detectar algunas características que pueden ser desarrolladas en la propuesta de solución, por ejemplo:

- La información que se muestre en los reportes esté disponible en diferentes idiomas.
- Los reportes se puedan exportar a diferentes formatos.

- Permita realizar un amplio número de reportes.
- Uso de generadores de reportes.

1.4 Selección de la metodología

Una metodología de desarrollo de software describe un entorno que es usado para organizar, planificar, y dirigir el proceso de desarrollo de un software. Existen varias metodologías de desarrollo de software, todas contienen algunas etapas básicas del ciclo de desarrollo de software como son la planificación, análisis, diseño, implementación y mantenimiento (11). Estas se dividen en dos grupos:

1. **Metodologías Tradicionales:** Rational Unified Process (RUP), Microsoft Solution Framework (MSF), entre otras. Cuentan con mayor énfasis en la planificación y el control del proyecto.
2. **Metodologías Ágiles:** Extreme Programming (XP), Agile Unified Process (AUP), entre otras. Estas con las características de fácil adaptabilidad a cambios en los requisitos por parte del cliente y hará la entrega del proyecto más satisfactoria tanto para el cliente como para el equipo de desarrollo (11).

Esta investigación tributa a un proyecto de desarrollo del centro Fortes. Para que la documentación asociada al desarrollo de la propuesta de solución sea compatible con la del proyecto, se utiliza, de acuerdo a las políticas de la Dirección de Calidad de la UCI, la metodología AUP-UCI como guía para el desarrollo. Esta metodología consiste en una versión de la metodología ágil AUP³, adaptada al proceso productivo de la Universidad de las Ciencias Informáticas.

1.4.1 Metodología de desarrollo de software AUP versión UCI

AUP-UCI surge a partir de la necesidad de unificar las buenas prácticas de las cerca de 20 metodologías que utilizaban los centros productivos para el desarrollo de software. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma y se unifican las restantes fases de la metodología de desarrollo de software AUP en una sola, nombrada Ejecución y agregándose también una nueva fase llamada Cierre (12). Para una mayor comprensión se muestra la siguiente tabla.

Fases AUP	Fases Variación AUP-UCI	Objetivos de las fases(Variación AUP-UCI)
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la

³ Por sus siglas en inglés AUP: *Agil Unified Process*.

		planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
Elaboración	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
Construcción		
Transición		
	Cierre	En esta fase se analizan tanto los resultados del proyecto como la ejecución y se realizan las actividades formales de cierre del proyecto.

Tabla 1. Fases de la variación de AUP para la UCI (12)

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

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

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

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

Escenario No 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos,

probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una historia de usuario (HU) no debe poseer demasiada información (12).

Para el desarrollo de la propuesta de solución se decidió optar por el escenario 4 porque es el adecuado para proyectos de corta duración. También se estará en constante interacción con el cliente para ultimar detalles de los requisitos y así poder implementar un software exitoso. Este coincide además con el escenario utilizado para el proceso de desarrollo de la Plataforma de recursos de la Editorial Universitaria Félix Varela por lo que los artefactos generados por la investigación podrán integrarse a la documentación del proyecto al que se adscribe.

1.5 Herramientas y tecnologías

Para el desarrollo de la propuesta de solución se adoptan las tecnologías utilizadas por el proyecto Plataforma de recursos para la Editorial Universitaria Félix Varela y se realiza un estudio sobre los generadores de reportes para seleccionar el que más se adapte al sistema para su posterior integración. A continuación, se presenta una breve descripción de las mismas.

Lenguaje programación del lado servidor:

PHP7

PHP es un lenguaje de programación de uso general de código del lado del servidor, originalmente diseñado para el desarrollo web de contenido dinámico (9). Esta versión incorpora nuevas mejoras a partir de las deficiencias de PHP5 y sus complementos (PHP5.6 la última) (13). Algunas de sus novedades son:

- **Un nuevo Zend Engine⁴:** Zend Engine 3.0 incorpora un nuevo motor, lo que mejora el rendimiento del lenguaje.
- **El doble de velocidad:** Gracias al nuevo motor PHPNG (PHP New Generation), la velocidad aumenta considerablemente, con la nueva característica que incorpora, la compilación Jit, que hace que el código se compile cuando se está ejecutando en lugar de antes de la ejecución.
- **Facilita el manejo de errores:** El manejar errores “Fatales” nunca ha sido una tarea fácil para los programadores en PHP. El nuevo motor de excepciones te permite reemplazar este tipo de errores con excepciones.
- **Incorporación de nuevos operadores:** un nuevo operador aterriza en PHP. Se trata del operador de comparación combinado “<=>”. Este devolverá 0 si los dos operandos son iguales, 1 si el de la izquierda es mayor, y -1 si el de la derecha es mayor. Además, se incluye el operador nulo coalescente “??” que se puede utilizar cuando quieres comprobar si existe algo y devolver un valor

⁴ Zend Engine: es el motor de scripting de código abierto que interpreta el lenguaje de programación PHP.

por defecto, en caso de que no sea cierto. El operador devuelve el resultado de su primer operando si existe y no es nulo, y el segundo operando en cualquier otro caso.

- **Soporte para sistemas Windows de 64 bits:** permite manejar archivos grandes o tener un soporte para grandes enteros (64 bits), así como otros tipos de datos nativos de 64 bits.
- **Las clases anónimas:** Una clase anónima es una clase sin nombre, definida en la misma línea de código donde se crea el objeto de la clase. PHP 7 te permite utilizar clases anónimas, una práctica bien establecida en otros lenguajes orientados a objetos como C++ y Java (14).

Tecnologías de lado del Cliente:

Html5

HTML v5.0 es un estándar y lenguaje de marcas de hipertexto que hace referencia al lenguaje de marcado, para la elaboración de páginas web en sus diferentes versiones. Se trata de un sistema para formatear el *layout* de las páginas, así como hacer algunos ajustes a su aspecto. Con HTML5, los navegadores como Firefox, Chrome, Explorer, Safari (15).

Este lenguaje proporciona para el sistema una mejor velocidad de navegación, debido a que las páginas creadas y maquetadas correctamente se arman más rápido en los exploradores. Además, proporciona un mejor entendimiento del código principalmente para los diseñadores web.

CSS3

Hoja de estilo en cascada o CSS⁵ es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML ⁶ y por extensión en XHTML. La versión CSS3 se divide en módulos, entre los más importantes se encuentran (16):

- Selectores.
- Modelo de caja.
- Fondos y fronteras.
- Valores de imagen y contenido reemplazado.
- Efectos de texto.
- Transformaciones 2D / 3D.

⁵ Por sus siglas en inglés CSS: *Cascading Style Sheets*.

⁶ Por sus siglas en inglés HTML o XML: *Extensible Markup Language*.

JavaScript

JavaScript, es un lenguaje de programación del tipo interpretado, completamente orientado a objetos, que permite la creación de pequeños programas de todo tipo y la programación con objetos complejos. Cuenta con una estructura simple y es considerado un lenguaje de alto nivel, ya que es muy simple de aprender, se adapta a la perfección a la sintaxis del lenguaje humano y está formado por pequeños textos o script. Entre las principales características con las que cuenta JavaScript, destaca su capacidad de integrarse a la perfección con el sistema operativo y con la mayoría de los navegadores web, ofreciendo una versatilidad absoluta que muy pocos lenguajes tienen y, al ser ejecutado del lado del cliente, acelera la carga del servidor creando aplicaciones y sitios web bastante ligeros (17).

JQuery

Es un marco de trabajo de JavaScript el cual cuenta con un conjunto de funciones implementadas para el desarrollo de aplicaciones web. Este nos permite agregar efectos y funcionalidades complejas como galerías de fotos dinámicas y elegantes, validación de formularios y calendarios. También nos brinda la posibilidad de utilizar la tecnología como JavaScript asíncrono y XML (AJAX por sus siglas en inglés Asynchronous JavaScript And XML) y Modelo de Objetos del Documento (DOM por sus siglas en inglés Document Object Model) sin tener en cuenta los detalles complejos de la programación (18).

Sistema de gestión de base de datos: PostgreSQL9.4

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. El conjunto de características maduras que tiene PostgreSQL no sólo rivaliza con sistemas de bases de datos propietarias, sino que los supera en características avanzadas, extensibilidad, seguridad y estabilidad (19).

Entre las principales novedades en PostgreSQL 9.4 se encuentran:

- Crear mecanismos para el seguimiento del progreso de la replicación, incluyendo métodos para identificar el origen de los cambios individuales durante la replicación lógica.
- Mejoras sustanciales de rendimiento para la clasificación.
- Mejoras sustanciales en el rendimiento de los equipos con varias CPU (20).

pgAdmin3

La herramienta pgAdmin3 es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en

cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres (21).

Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad (21).

Servidor web: Nginx

Nginx es un servidor web HTTP de código abierto que también incluye servicios de correo electrónico con acceso al Internet Message Protocol (IMAP) y al servidor Post Office Protocol (POP). Además, Nginx está listo para ser utilizado como un proxy inverso. Se utiliza para equilibrar la carga entre los servidores back-end, o para proporcionar almacenamiento en caché para un servidor back-end lento (22).

Tiene como principales características:

- Ser sumamente ligero.
- Permite servir aplicaciones web con una velocidad muy superior a la de sus competidores más directos como Apache y LiteSpeed.
- Es un servidor web/proxy multiplataforma, listo para trabajar en Linux, Unix, Windows, MAC OS, Solaris y BSD.
- Da una importancia relevante a la seguridad y cuenta con sistemas de protección y encriptado de datos que complementan todo el paquete de software (22).

Marco de trabajo: Symfony 2.7

Symfony es un framework PHP para la construcción de aplicaciones web robustas basado en el patrón Modelo Vista Controlador (MVC). Se encarga de separar la lógica del negocio, la lógica del servidor y la presentación de la aplicación web, así también nos provee de varias herramientas con el único objetivo de reducir el tiempo de desarrollo de una aplicación, así mismo automatiza las tareas más comunes permitiendo a los desarrolladores enfocarse por completo en requerimientos específicos de la aplicación (23).

Symfony 2.7 trae consigo algunas mejoras para ser más productivo, entre ellas se encuentran:

- Añadido el atajo *getParameter()* al controlador base para obtener el valor de cualquier parámetro del contenedor.
- Muestra información sobre el estado de tu versión Symfony.
- Añade el código de estado HTTP a los listados del *profiler* permitiendo localizar o descartar las peticiones erróneas.
- Reinicio automático del servidor web embebido.
- El comando *lint: twig* ahora soporta múltiples archivos y directorios (24).

ORM: Doctrine

Es un Object Relational Mapping (ORM) para PHP que proporciona persistencia transparente de objetos PHP. La principal tarea de los asignadores de objeto relacionales es la traducción transparente entre objetos PHP y las filas relacionales de la base de datos. Una de las características clave de Doctrine es la opción de escribir las consultas de base de datos en un dialecto Structured Query Language (SQL) propio, orientado a objetos llamados DQL, difiere ligeramente de SQL en que abstrae considerablemente Doctrine Query Language (DQL). Admite la asignación entre las filas de la base de datos y objetos, permitiendo a los desarrolladores escribir consultas de una manera sencilla y flexible (25).

Maquetador web: Bootstrap 3.3.7

Bootstrap es un framework de software libre, intuitivo y de gran alcance para el diseño de sitios y aplicaciones web más rápido y fácil. Con Bootstrap 2, se añaden funcionalidades adaptables a toda la estructura como una hoja de estilos opcional. Sobre esa base, con Bootstrap 3, se reescribe el framework una vez más para hacerlo adaptativo por defecto con un primer acercamiento móvil. Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tabletas y móviles a distintas escalas y resoluciones. Se integra perfectamente con las principales librerías JavaScript y JQuery (26).

NetBeans

Es un entorno de desarrollo integrado y una plataforma de desarrollo. Aunque inicialmente, NetBeans IDE sólo podía ser utilizado para desarrollar aplicaciones Java, a partir de la versión 6, NetBeans soporta varios lenguajes de programación mediante la instalación de complementos adicionales, algunos de estos lenguajes son Java, C, C ++, PHP, HTML, JavaScript, y Scala. NetBeans proporciona funcionalidades adicionales como la compilación en tiempo real, la comprobación de tipos, refactorización, navegadores de clase y soluciones rápidas para los errores en tiempo de compilación (27). Para el desarrollo de la

propuesta de solución se utilizó la versión 8.0 porque es un entorno de desarrollo gratuito y de código abierto. También cuenta con mejoras en PHP y en las referencias a métodos, y proporciona nuevas herramientas para HTML5.

Visual Paradigm for UML

Es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML (28). Se emplea esta herramienta de modelado en su versión 8.0 para el desarrollo de la propuesta de solución por ser una herramienta que soporta el ciclo de vida completo en el desarrollo de un software. Además, destacar que esta herramienta guarda todo el modelo en un solo fichero, así de simple, y basta con copiarse solo ese fichero se está seguro de que tiene todo el trabajo encapsulado en él. También agiliza la construcción de aplicaciones con calidad y permite generar informes en formato PDF/HTML y exportar los artefactos en .jpg posibilitando su documentación.

Generadores de reporte

Report-Writer

Herramienta desarrollada por ORACLE que permite a los usuarios obtener con facilidad información de cualquier base de datos. Además, proporciona a los usuarios el control sobre la apariencia y contenido de los reportes. Los resultados de las consultas se pueden presentar en un formato de reporte que se establece en forma automática por el software, o el usuario puede proporcionar las especificaciones que instruyan al sistema para preparar títulos específicos, descripciones de página y encabezados de columna (29).

Principales características:

- Rapidez en la elaboración de reportes sencillos y complejos.
- Filtra los datos para incluir sólo la información relevante en los reportes.
- Inserta elementos gráficos, tales como logotipos de compañías.

Reporteador EUREKA

Es una herramienta de software para producir reportes, obteniendo la información de una base de datos localizada en un servidor o bien de una PC⁷ local. El reporteador EUREKA produce reportes interactivos y distribuye la información a otros medios, dentro de los cuales se incluyen impresoras y faxes (29).

Características:

- Provee un ambiente sencillo para construir reportes empresariales.
- Los reportes pueden salvarse en diferentes formatos (HTML, XLS, CSD) y pueden ser publicados simultáneamente a través de diferentes medios de salida (incluyendo impresoras, faxes o medios electrónicos).
- Extrae datos de múltiples fuentes de datos, los combina y formatea en documentos fáciles de leer y de buena calidad.
- Reduce el tiempo dedicado para construir nuevos reportes.

Report Builder

Herramienta visual que permite seleccionar, analizar, resumir, colocar, formatear y presentar en una pantalla o como un reporte impreso datos de una base de datos del Sistema Manejador PROGRESS.

Características:

- Soporta tipos de letras, imágenes, vista preliminar de impresión y etiquetas de correos.
- Permite la manipulación de los datos para producir diferentes tipos de reportes.
- Permite darle diversas apariencias al reporte sin gran dificultad.
- Puede exportar la información en hojas de cálculo en ambiente de Windows y de Linux.
- No almacena cada reporte como un archivo individual, sino que crea cada reporte y lo almacena en una librería (cada una puede tener uno o más reportes individuales) para futuras sesiones (29).

Servidor Dinámico de Reportes V2.0

El Servidor Dinámico de Reportes (SDR) es un software desarrollado en el Centro de Tecnologías de Gestión de Datos de la Facultad 6 de la Universidad de Ciencias Informáticas, que permite de manera eficiente la gestión, compilación, publicación y exportación de reportes. El principio del SDR es brindar una capa de servicios que facilita su integración con cualquier otro sistema, web o de escritorio, con independencia de sus plataformas de desarrollo. A través de este los clientes pueden generar reportes en una gran cantidad de formatos sobre múltiples fuentes de datos. Estos reportes se mantienen

⁷ PC: computadora.

almacenados en el servidor, logrando con ello mejores tiempos de respuesta y acceso al historial de los reportes (30).

Principales características y funcionalidades:

- Permite la generación de reportes en múltiples formatos: PDF, XML, HTML, CSV, XLS, XLSX, RTF, JPG, ODT, PPTX, DOCX, ODS y XML4SWF, a partir de diferentes fuentes de datos.
- Permite la generación de subreportes.
- Visualiza las vistas previas de los reportes.
- Posibilita la selección de estilos de estructura para un grupo de reportes.
- Utiliza un gestor de base de datos, que puede ser PostgreSQL, SQLite, MySQL o SQL, para persistir la información del servidor y los datos de los reportes.
- Implementa un conjunto de políticas de seguridad a varios niveles que garantiza el acceso controlado a los recursos y servicios (30).

Se decide integrar el sistema SDR ya que es compatible con las tecnologías sobre la que está desarrollada la Plataforma de recursos de la Editorial Universitaria Félix Varela. Al ser un proyecto desarrollado en la universidad se puede contar con el apoyo del equipo de desarrollo si surge algún error tanto en el funcionamiento del sistema SDR como en la integración con la plataforma. Además, brinda una serie de características y funcionalidades que permitirán la elaboración, exportación y calidad de los reportes de una manera eficiente, por ejemplo:

- Posibilita la selección de diferentes gráficos que permitirán un mayor entendimiento y estructuración de los reportes.
- Permite la generación de reportes en formatos PDF, XML, HTML, CSV, XLS, XLSX, RTF, JPG, ODT, PPTX, DOCX, ODS y XML4SWF posibilitando documentar cualquier tipo de reporte seleccionado.
- Constituye un sistema multiplataforma.
- Implementa un conjunto de políticas de seguridad a varios niveles que garantiza el acceso controlado a los recursos y servicios.

iReport 5.3

JasperReports es una poderosa y flexible solución de código abierto para la generación y gestión de informes. iReport es un programa que ayuda a los usuarios y desarrolladores que usan la librería JasperReports para diseñar reportes visualmente. A través de una interfaz simple de usar, iReport provee funciones importantes para crear reportes amenos en poco tiempo. El iReport puede leer y modificar

ambos tipos de archivo, XML y jasper. A través de JasperReports, es capaz de compilar XML a archivos jasper y “ejecutar reportes”. Además, permite exportar el resultado a PDF, HTML, XLS, CSV (31). Esta herramienta es utilizada por el sistema SDR para generar y exportar los reportes a través de los ficheros .jrxml y .jasper.

1.6 Conclusiones parciales

1. El estudio de los conceptos básicos asociados al objeto de estudio posibilitó la adquisición de una mejor visión del problema a resolver.
2. El estudio de las soluciones existentes a nivel nacional e internacional arrojó como resultado que los sistemas estudiados no brindan una solución completa al problema a resolver, debido fundamentalmente a incompatibilidades con la Plataforma de recursos de la Editorial Universitaria Félix Varela.
3. La selección de las tecnologías, herramientas y metodología a utilizar permitió establecer el entorno de desarrollo en el que se implementará la solución a la problemática antes planteada.

CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

2.1 Introducción

En este capítulo se describe el sistema que se propone, a partir de la metodología utilizada en la investigación. Se elaboran algunos de los siguientes artefactos propuestos por la metodología AUP en su variación para la UCI en las fases de Inicio y Ejecución: modelo del dominio, especificación de requisitos, definición y descripción de las historias de usuario, diseño de los prototipos de interfaz de usuario y la descripción de la arquitectura y los patrones utilizados para el diseño del componente.

2.2 Propuesta de solución

La propuesta de solución consiste en la implementación de un módulo de reportes que brinde información al administrador de la plataforma o de una editorial sobre las actividades que ocurren en la misma. El módulo está destinado a los usuarios que tengan rol Administrador y/o Editor.

Usuario Administrador: Puede realizar todo tipo de reportes de cualquier editorial.

Usuario Editor: Puede realizar reportes de su editorial solamente.

La propuesta de solución permitirá realizar reportes acerca de los contenidos que se muestran en la plataforma (libros y recursos), así como información referente a las editoriales y al acceso de los usuarios a la plataforma. Estos datos permitirán al administrador comprobar si la plataforma ha tenido aceptación y en caso contrario tomar las decisiones y medidas necesarias para atraer la atención de los usuarios. También apoyarán al administrador a evaluar si los contenidos que se muestran en la misma son del agrado de los usuarios, además que proveen de información que puede ser utilizada para brindar nuevas posibilidades a las editoriales y autores destacados, así como, a exhortar a aquellos cuyos contenidos no han tenido gran aceptación. Si un administrador tuviera a su disposición uno o varios reportes donde se muestre información detallada sobre el acceso a la plataforma y/o contenidos de la misma, le sería mucho más fácil el proceso de toma de decisiones sobre aspectos tan importantes como, por ejemplo, la renovación de contratos con las editoriales entre otros.

En un marco más avanzado, a través de estos reportes el administrador pudiese hacer un estudio de las tendencias que tienen los usuarios de la plataforma, de esta forma sería mucho más sencillo seleccionar los contenidos a mostrar en la misma, puesto que se conocería de antemano si un contenido tendrá aceptación o no. Para ello se propone integrar el Servidor Dinámico de Reportes (SDR) a la Plataforma de recursos de la Editorial Universitaria Félix Varela ya que este brinda una capa de servicios que facilita su integración con cualquier otro sistema.

A continuación, se ofrece un resumen de la información contenida en los reportes que deben ser mostrados al usuario. La información que se muestra varía en dependencia del tipo de reporte realizado:

1. El reporte de libros permite determinar la cantidad de libros visitados en una editorial o en todas y en el rango de fecha especificada. Puede ser accedido por el administrador y los editores. Este contiene la siguiente información de los libros:

- Título.
- ISBN.
- Categoría.
- Año.
- Cantidad de visitas realizadas al libro.
- Cantidad de usuarios que han accedido al libro.
- Evaluación general del libro.
- Autor.

2. El reporte de recursos permite determinar la cantidad de recursos visitados en el rango de fecha especificada. Solo puede ser accedido por el administrador. Este contiene la siguiente información:

- Nombre.
- Carrera.
- Asignatura.
- Cantidad de visitas realizadas al recurso.
- Autor.
- Email del autor.

3. El reporte de editorial puede ser accedido por el administrador y los editores. Este contiene la siguiente información:

- Los 10 libros más solicitados.
- Los 5 usuarios más asiduos.
- Las 5 categorías favoritas.
- Los 5 autores favoritos.

4. El reporte de plataforma contiene información referente al acceso de los usuarios al sistema y a los contenidos. Puede ser accedido por el administrador y los editores, y contiene la siguiente información:

- Promedio de visitantes.
- Promedio de libros y recursos visitados.

- Usuarios más asiduos a la plataforma.
- Ranking de libros y/o recursos más solicitados.
- Editoriales destacadas.
- Autores destacados.

El sistema permitirá exportar a formato .pdf, .xls y .jpg los reportes antes mencionados.

2.3 Modelo conceptual

Un modelo conceptual o modelo de dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. No se trata de un conjunto de diagramas que describen clases de software, u objetos de software con responsabilidades, sino que modela clases conceptuales significativas en un determinado problema (32).

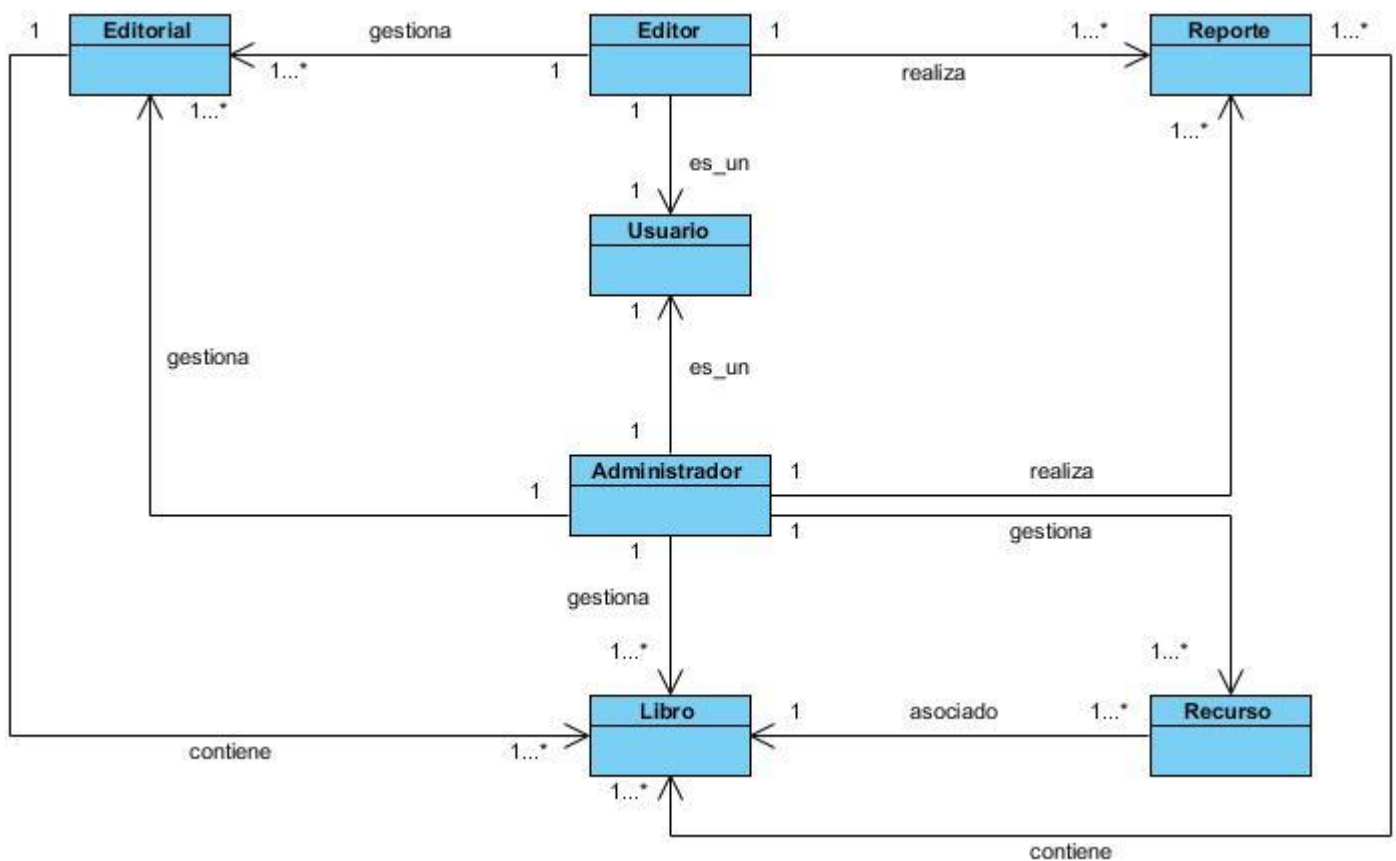


Figura 1: Diagrama conceptual del negocio.

Descripción de los elementos del modelo de dominio

Administrador: Es el máximo encargado de la gestión de la plataforma de recursos Félix Varela y de garantizar la calidad de las operaciones que se realizan en esta. Puede realizar reportes de libros y exportarlos a formato .pdf.

Editor: Encargado de la gestión de sus editoriales. Puede realizar reportes de libros y exportarlos a formato .pdf.

Usuario: persona que tiene acceso a navegar en la plataforma.

Editorial: conjunto de editoriales existentes en el sistema.

Libro: conjunto de libros existentes en el sistema.

Recurso: conjunto de recursos existentes en el sistema.

Reporte: funcionalidad que muestra un listado con información del acceso por parte de los usuarios a un libro determinado. Además, permite exportar a formato .pdf el reporte.

2.4 Especificación de los requisitos del sistema

La obtención y análisis de los requisitos constituye una de las etapas fundamentales en el proceso de ingeniería de requisitos. En esta actividad los ingenieros de software trabajan en conjunto con los clientes y usuarios finales, para determinar el dominio de la aplicación, los servicios que debe proporcionar, el rendimiento requerido para el sistema, así como las restricciones del hardware. (33).

Existen dos tipos de requisitos; funcionales y no funcionales. A continuación, se exponen cada uno de estos requisitos.

2.4.1 Requisitos Funcionales (RF):

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y como el sistema debe comportarse en distintas situaciones. En algunos casos estos requisitos también pueden declarar explícitamente lo que el sistema no debe hacer (33). En la siguiente tabla se muestran los requisitos funcionales identificados:

Nº	Requisito Funcional (RF)	Descripción	Complejidad	Prioridad para el cliente
1.	Generar reporte de libros.	El sistema debe permitir generar un reporte de uso de los libros, permitiendo seleccionar el rango de fecha del mismo, mostrando los siguientes datos: <ul style="list-style-type: none">• Título.	Alta	Media

		<ul style="list-style-type: none"> • ISBN. • Categoría. • Año. • Cantidad de visitas realizadas al libro. • Cantidad de usuarios que han accedido al libro. • Evaluación general del libro. • Autor. 		
2.	Generar reporte de recursos.	<p>El sistema debe permitir generar un reporte de uso de los recursos, permitiendo seleccionar el rango de fecha del mismo, mostrando los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre. • Carrera. • Asignatura. • Cantidad de visitas realizadas al recurso. • Autor. • Email del autor. 	Alta	Media
3.	Generar reporte de editorial.	<p>El sistema debe permitir generar un reporte de editorial, permitiendo seleccionar la editorial a la cual quiere realizarle el mismo, mostrando los siguientes datos:</p> <ul style="list-style-type: none"> • Libros más solicitados en la editorial. • Usuarios más asiduos a la editorial. • Categorías favoritas. • Autores favoritos. 	Alta	Media
4.	Generar reporte de plataforma.	<p>El sistema debe generar un reporte sobre las distintas actividades que se llevan a cabo en la plataforma, mostrando los siguientes datos:</p> <ul style="list-style-type: none"> • Promedio de visitantes. • Promedio de libros y recursos visitados. 	Alta	Media

		<ul style="list-style-type: none"> • Usuarios más asiduos a la plataforma. • Ranking de libros y/o recursos más solicitados. • Editoriales destacadas. • Autores destacados. 		
5.	Exportar a formato .pdf, .xls y .jpg los reporte.	El sistema debe permitir exportar a formato .pdf, .xls y .jpg los reportes.	Baja	Media

Tabla 2. Descripción de los requisitos funcionales del sistema.

2.4.2 Requisitos no funcionales (RNF):

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. En si se refiere a la fiabilidad, al tiempo de respuesta y la capacidad de almacenamiento que proporciona el sistema. A menudo son aplicados al sistema en su totalidad, aunque normalmente se aplican a características o servicios individuales del sistema (34). A continuación, se listan los requisitos no funcionales identificados:

Requisitos de software

RNF 1: Es necesario contar con el servidor de aplicaciones web Nginx instalado.

Requisitos de hardware

RFN 2: Requerir un servidor donde se instalará el sistema el cual deberá tener las siguientes prestaciones: al menos 2Gb de RAM, un microprocesador Dual-Core o superior.

Restricciones de diseño e implementación

RNF 3: Utilizar el lenguaje español e inglés para los mensajes y textos de la interfaz.

RFN 4: Para el desarrollo del sistema se deben utilizar los lenguajes JavaScript, CSS3 y PHP7.

RFN 5: Utilizar arquitectura Xalix.

RFN 6: El módulo debe visualizarse y ejecutarse correctamente en un navegador web moderno, como Firefox (a partir versión 42.0) o Google Chrome (a partir versión 56.0).

Requisitos de licencia

RNF 7: Utilizar la licencia de Visual Paradigm 8.

2.5 Usuarios del sistema

Los usuarios del sistema son aquellas personas que interactúan con la aplicación. La seguridad incluye un sistema de gestión por roles que son asignados a los usuarios que les permitirá acceder solo a las

funcionalidades correspondientes a su nivel de acceso. En la siguiente tabla se muestran los usuarios del sistema y sus responsabilidades.

Usuario	Responsabilidades
Administrador	1. Encargado de generar reportes de libros, recursos, editoriales y de plataforma. 2. Encargado de exportar a formato .pdf, .xls y .jpg los reportes creados.
Editor	1. Encargado de generar reportes de su editorial solamente. 2. Encargado de exportar a formato .pdf, .xls y .jpg los reportes de su editorial.

Tabla 3. Descripción de los usuarios del sistema.

2.6 Historias de usuario

La variación AUP-UCI utiliza en su escenario número 4 para describir los requisitos de software las Historias de Usuario (HU), estas son descripciones cortas de las necesidades del cliente para el software que está en desarrollo. A continuación, se muestra la HU: Generar reporte de libros y la HU: Exportar a formato .pdf, .xls y .jpg los reportes. Las restantes historias de usuario se muestran en Anexo #2.

Número: 1	Nombre del requisito: Generar reporte de libros.	
Programador: Oscar Luis Ambruster Jiménez	Iteración Asignada: 1	
Prioridad: Alta	Tiempo Estimado: 72 horas	
Riesgo en Desarrollo: Ausencia del desarrollador por enfermedad, problemas con la tecnología usada.	Tiempo Real: 72 horas	
Descripción: 1- Objetivo: Permitir generar el reporte del uso de varios libros en el sistema. 2- Acciones para lograr el objetivo (precondiciones y datos): - Para generar el reporte de varios libros deben haber sido creados previamente en el sistema. - El usuario debe tener rol Editor o Administrador. 3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos Editorial, Fecha de inicio y Fecha de fin son obligatorios. - (*) Editorial: Permite seleccionar la editorial a la cual se realiza el reporte. - (*) Fecha de inicio: Permite seleccionar la fecha a partir de la cual se realiza el reporte. - (*) Fecha de fin: Permite seleccionar la fecha hasta la cual se realiza el reporte.		

4- Flujo de la acción a realizar:

Para generar el reporte de varios libros el usuario debe seleccionar la opción **Reportes** del panel de Administración. Una vez que el usuario selecciona la editorial, el rango de fechas para el reporte y selecciona la opción **Generar reporte** se muestran los siguientes datos de los libros: título, isbn, año, categoría, cantidad de visitas realizadas al libro, cantidad de usuarios que tienen el libro como favorito, evaluación general del libro, autor. Si el usuario selecciona la opción Cancelar regresará a la página **Principal** del panel de Administración.

Prototipo de interfaz:

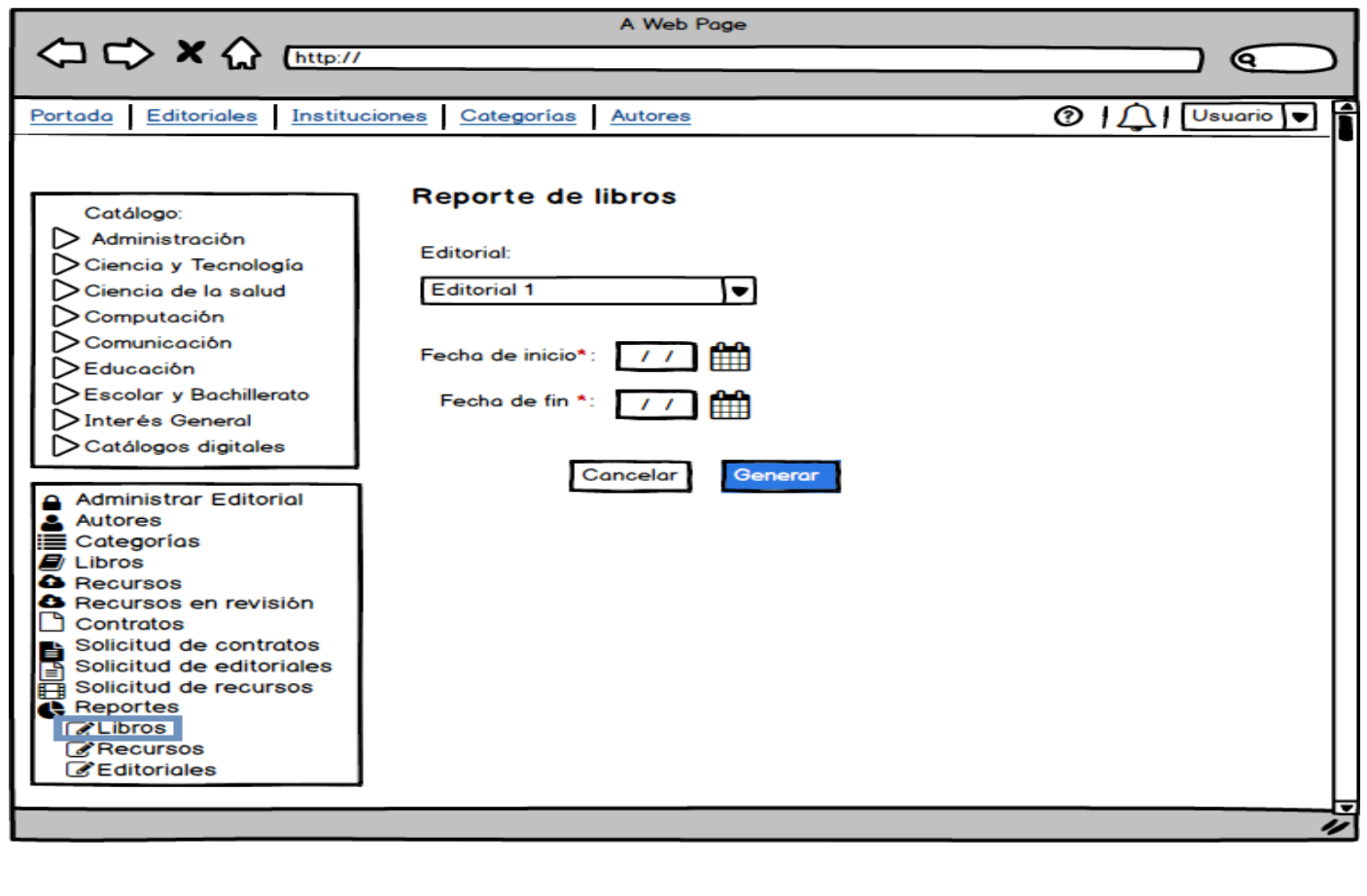


Tabla 4. HU Generar reporte de libros.

Número: 5		Nombre del requisito: Exportar a formato .pdf, .xls y .jpg los reportes.	
Programador: Oscar Luis Ambruster Jiménez		Iteración Asignada: 1	
Prioridad: Baja		Tiempo Estimado: 24 horas	
Riesgo en Desarrollo: Ausencia de miembros del equipo de trabajo y actividades extra productivas y sin carácter investigativo.		Tiempo Real: 24 horas	
Descripción:			

1- Objetivo:

Permitir exportar los reportes en formato .pdf, .xls y .jpg.

2- Acciones para lograr el objetivo (precondiciones y datos):

- El usuario debe tener rol Editor o Administrador.
- Debe haber sido creado un reporte previamente en el sistema.

3- Flujo de la acción a realizar:

Una vez que sea generado el reporte, el editor o administrador puede exportar el mismo a formato .pdf, .xls y .jpg seleccionando la opción Exportar. Una vez seleccionada la opción Exportar se muestran los tipos de formatos en que se puede exportar el reporte. Luego de seleccionar el formato se permite al editor o administrador descargar el reporte en la ubicación deseada.

Prototipo de interfaz:

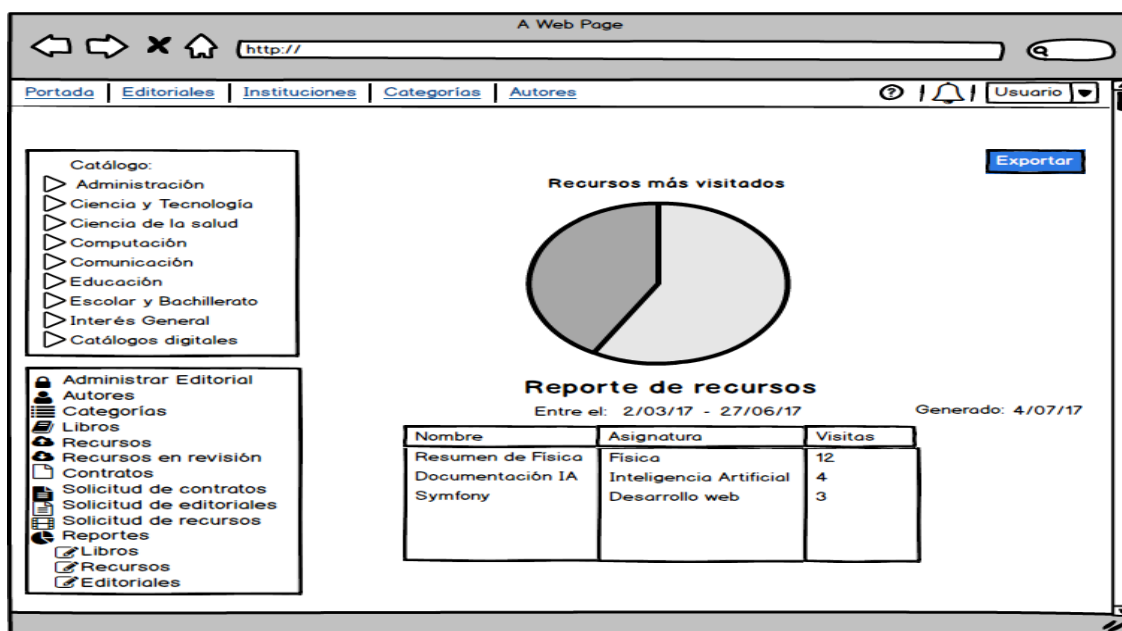


Tabla 5. HU Exportar a formato .pdf, .xls y .jpg los reportes.

2.7 Descripción de la arquitectura y los patrones de diseño del sistema

2.7.1 Arquitectura del sistema

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura Modelo Vista Controlador (MVC). Esta arquitectura separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si, por ejemplo, una misma aplicación debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo

utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación (35). A continuación, se muestra el funcionamiento del patrón MVC:

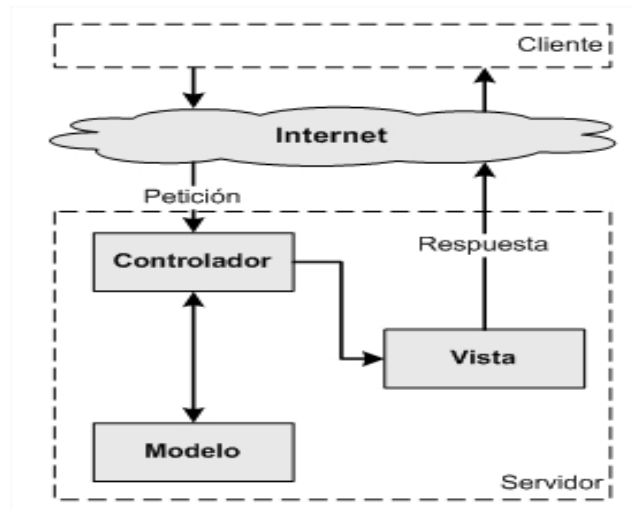


Figura 2. Funcionamiento del patrón MVC (35).

- **Modelo:** representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. Está representado por las clases Entity que posee Symfony.
- **Vista:** transforma el modelo en una página web que permite al usuario interactuar con ella. Está representado por las clases View que trae Symfony, las cuales poseen la extensión html.twig.
- **Controlador:** se encarga de procesar las interacciones del usuario procesando toda la información necesaria y realiza los cambios apropiados en el modelo o en la vista. Está representado por las clases Controller.

2.7.2 Patrones de diseño

Para el desarrollo de la propuesta de solución se utilizaron los Patrones Generales de Software para Asignar Responsabilidades (GRASP⁸) y los patrones GOF⁹ por ser considerados buenas prácticas en el proceso de desarrollo de un software.

2.7.2.1 Patrones GRASP

Los patrones GRASP describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. Constituyen un apoyo para la enseñanza que ayuda a

⁸ Por sus siglas en inglés GRASP: *General Responsibility Assignment Software Patterns*.

⁹ Por sus siglas en inglés GOF: *Gang of Four*.

entender el diseño de objetos, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable (36). A continuación, se brinda una descripción de los utilizados en el desarrollo de la propuesta de solución:

Experto

Se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Este patrón se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos (36).

El patrón experto puede verse en la implementación de los web scripts. El framework Web Script ofrece la posibilidad de crear una interfaz especializada para cada tipo de operación a través del protocolo http. Por lo que se adopta la siguiente convención y se crea cada interfaz experta según el método http que se utilice:

- **GET:** para obtener elementos sin necesidad de enviar un cuerpo de petición.
- **DELETE:** para eliminar recursos.
- **PUT:** para colocar elementos sin obtener mensajes de retroalimentación (donde no necesitas un cuerpo de respuesta).
- **POST:** cuando necesitas colocar elementos y necesitas un cuerpo tanto en la petición como en la respuesta.

En el sistema desarrollado se aprecia este patrón en las clases entidades, por ejemplo [book.php](#) es la entidad experta en conocer toda la información referente a los libros.

Controlador

Muchos de los frameworks que integra la editorial Félix Varela usan un patrón arquitectónico Modelo Vista Controlador (MVC), por lo que es común localizar objetos responsables de ejercer la función controladora. Ejemplo, cuando se necesita configurar una interfaz responsable de algún tipo de funcionalidad sobre un recurso, la responsabilidad de control se realiza de forma separada en un archivo *.php* usando el lenguaje PHP. En la investigación, el uso de este patrón puede evidenciarse en las clases controladoras ubicadas en [PRFV/src/PRFV/ReportBundle/Controller](#).



Figura 3. Ejemplo del uso del patrón controlador.

Bajo Acoplamiento

El patrón de Bajo Acoplamiento es un principio a tener en cuenta en todas las decisiones de diseño. Este patrón soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio. No se puede considerar de manera aislada a otros patrones como el Experto o el de Alta Cohesión, sino que necesita incluirse como uno de los diferentes principios de diseño que influyen en una elección al asignar una responsabilidad. En la solución se puede apreciar que las clases entidades son las más reutilizadas.

Alta Cohesión

En cuanto al diseño de objetos, la cohesión funcional es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento. Un elemento con responsabilidades altamente relacionadas, y que no hace una gran cantidad de trabajo, tiene alta cohesión (36). Entre los patrones Bajo Acoplamiento y Alta Cohesión existe una estrecha relación debido a que una mala cohesión causa un bajo acoplamiento y viceversa. En el caso del sistema en las clases controladoras se evidencia el uso de este patrón pues las mismas están formadas por varias funcionalidades que están estrechamente relacionadas, siendo estas las encargadas de definir las acciones y colaborar con otras para realizar tareas que las implican a todas.

2.7.2.2 Patrones GOF

Es un patrón de comportamiento a nivel de objeto, este permite disponer de varios procedimientos para realizar un requisito de software y elegir cuál es el que se va a aplicar en tiempo de ejecución. Con él se pueden definir un conjunto de algoritmos para un mismo método; pudiendo seleccionar a cada instante cómo debe comportarse la clase (37). A continuación, se describen los patrones GOF utilizados durante el desarrollo:

Abstract Factory¹⁰: es un patrón de tipo creación, su objetivo es facilitar una interfaz para crear "familias" de objetos relacionados o dependientes sin que sea necesario especificar su clase (38). Es utilizado cuando el marco de trabajo necesita crear un nuevo objeto para una petición, el cual busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea, como la definición por defecto de la factoría para las peticiones es "getRequest ()", Symfony2 crea un objeto de esta clase para tratar con las peticiones.

Decorator¹¹: es un patrón de tipo estructura, ya que permite determinar que clases y objetos serán utilizados para componer estructuras de mayor tamaño. Este patrón añade dinámicamente nuevas responsabilidades a un objeto (39). Cada una de las vistas generadas hereda su diseño de la plantilla "layout.html.twig" pues esta almacena el código HTML que es común a todas las páginas de la aplicación, siendo la plantilla contenedora de la estructura y el diseño básico de las vistas.

2.8 Diagrama de clase del diseño (DCD)

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones incluyendo herencia, agregación y asociación. Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño) (40). Se muestra a continuación los diagramas de clase del diseño correspondientes a HU: Generar reporte de libros y HU: Exportar a formato .pdf, .xls y .jpg los reportes. Los restantes diagramas de clase del diseño se muestran en Anexo #3.

¹⁰ **Abstract Factory**: Factoría Abstracta.

¹¹ **Decorator**: Decorador

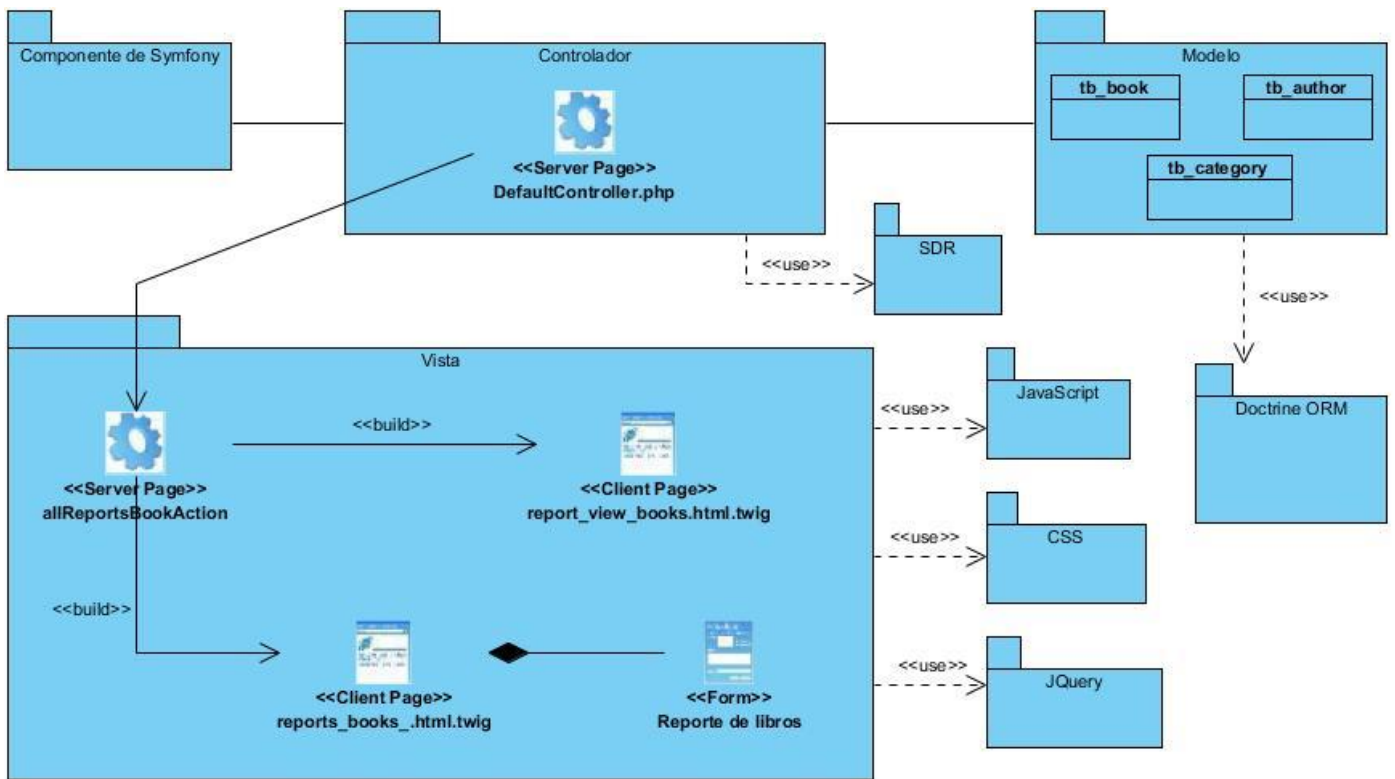


Figura 4. DCD-Reporte de libros.

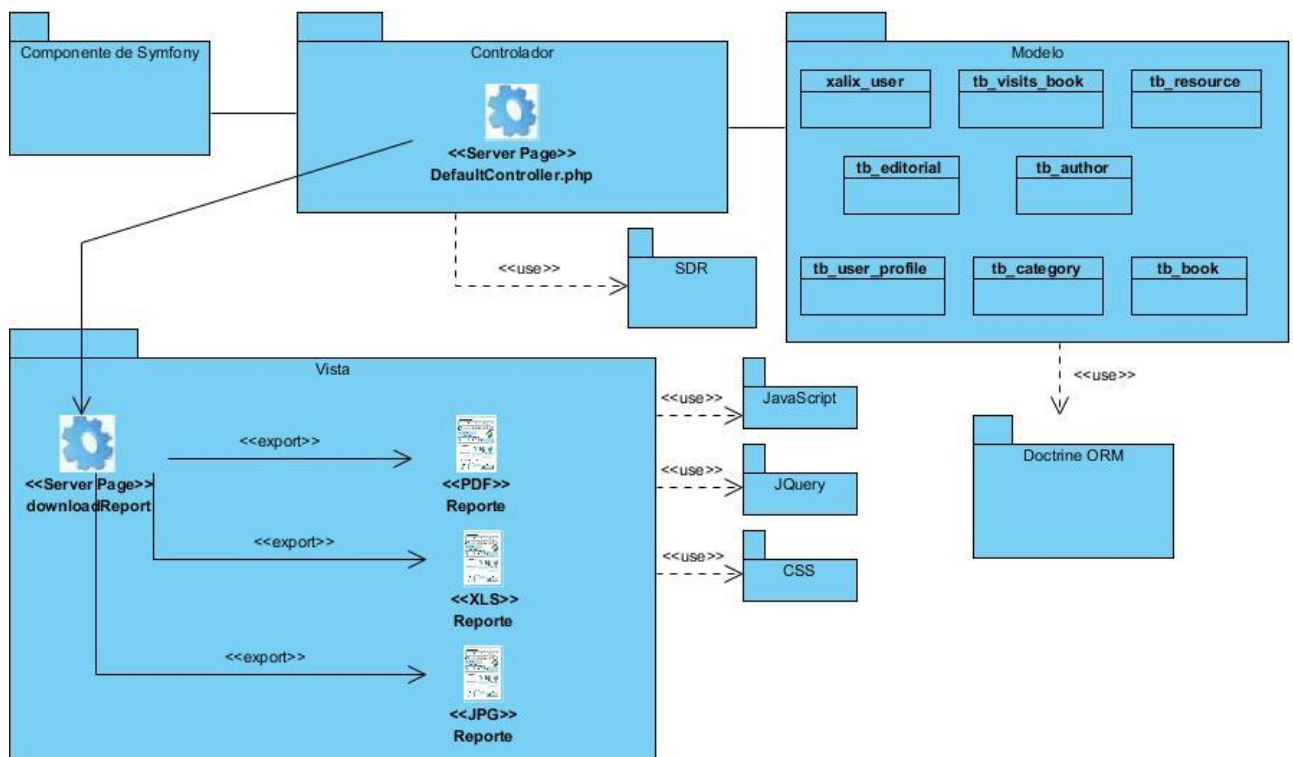


Figura 5. DCD-Exportar a formato .pdf, .xls y .jpg los reportes.

2.9 Diagrama de clases de secuencia del diseño (DCSD)

Los diagramas de secuencia muestran las interacciones entre objetos mediante transferencia de mensajes entre ellos (41). Se presenta a continuación los diagramas de secuencia del diseño correspondiente al DCD: Reporte de libros y DCD- Exportar a formato .pdf, .xls y .jpg los reportes. Los restantes diagramas de clase de secuencia del diseño se muestran en Anexo #4.

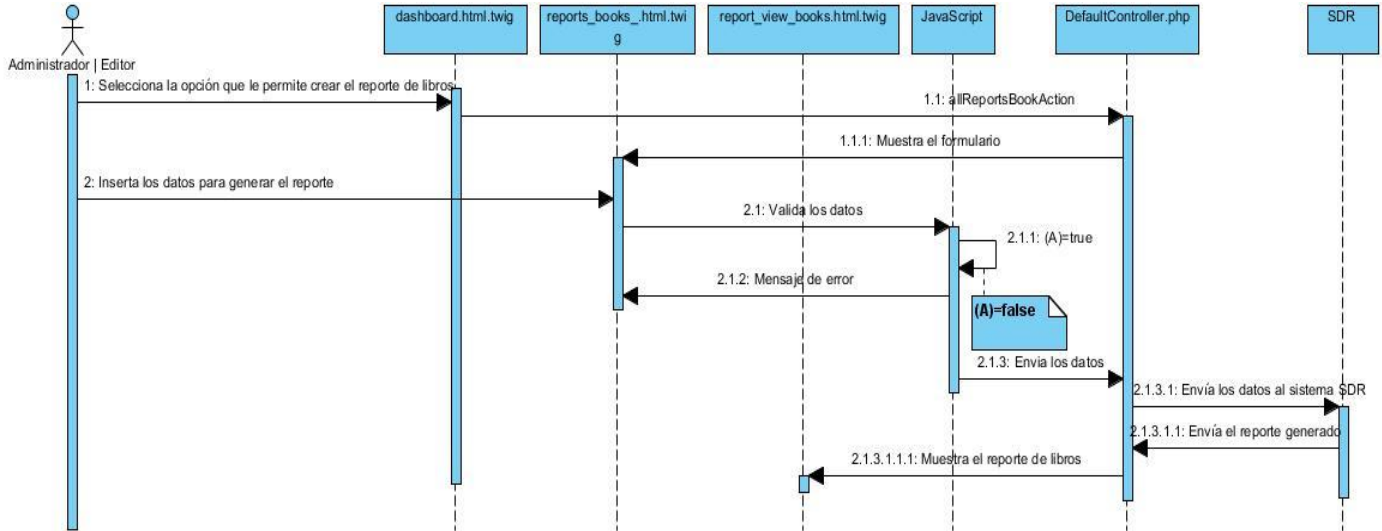


Figura 6. DCSD-Reporte de libros.

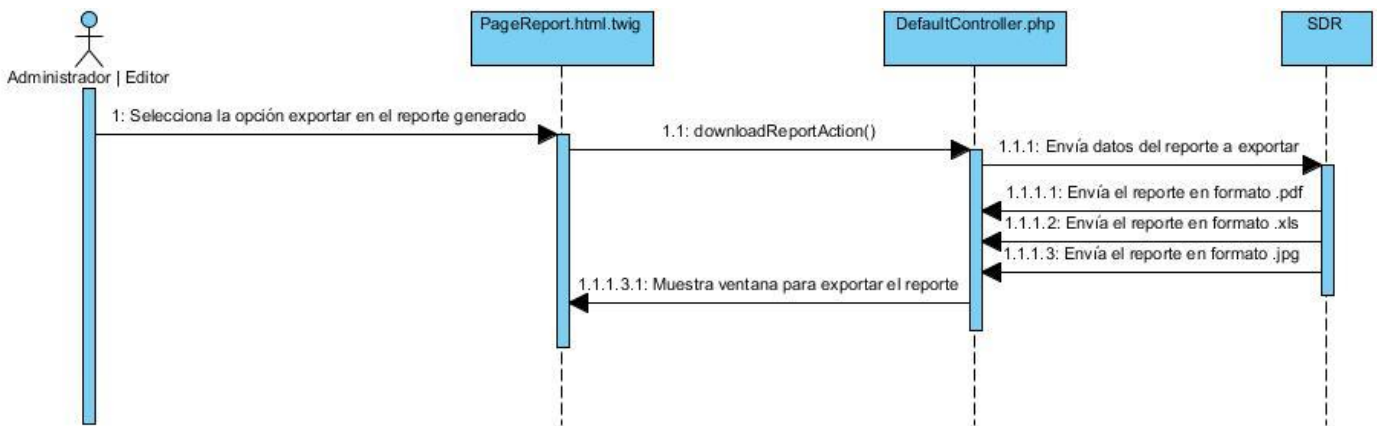


Figura 7. DCSD-Exportar a formato .pdf, .xls y .jpg los reportes.

2.10 Diagrama de datos

Un modelo de datos está orientado a representar los elementos que intervienen en la realidad o en un problema dado y la forma en que se relacionan dichos elementos entre sí (42). Se muestra a continuación la relación existente entre las entidades utilizadas en la propuesta de solución:

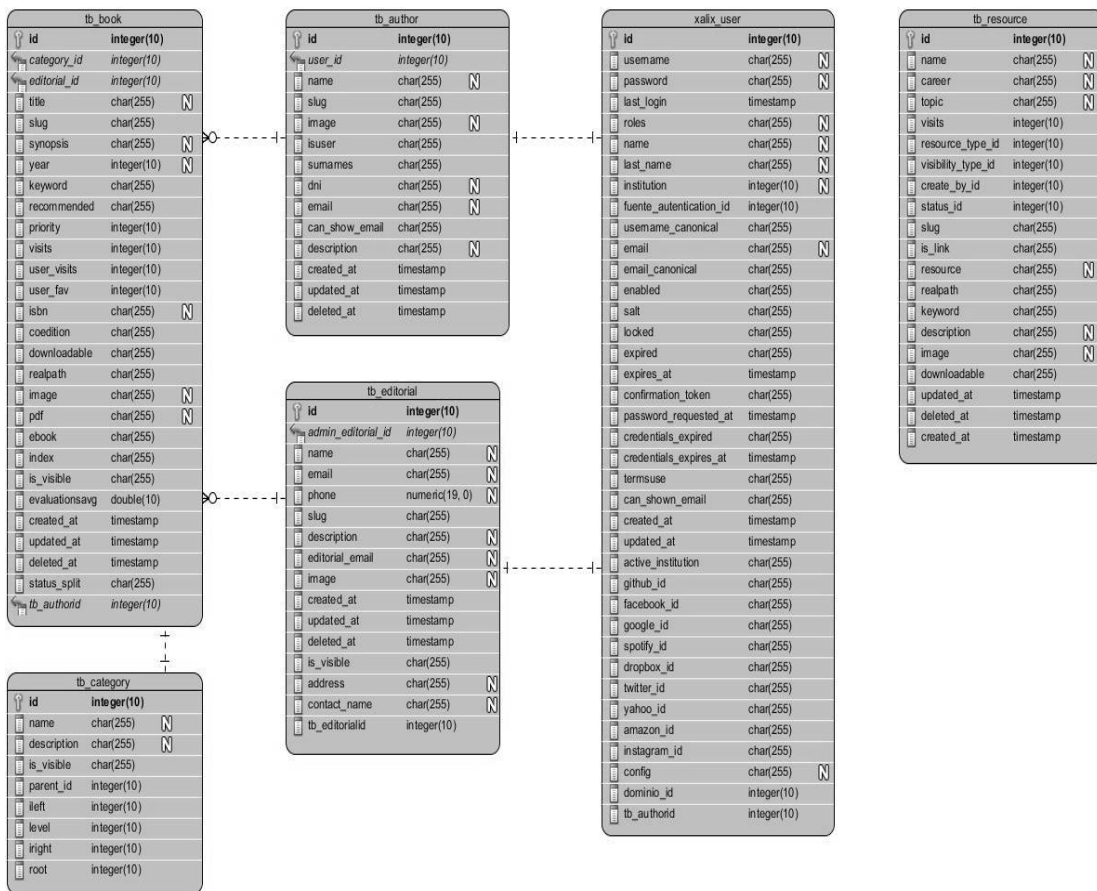


Figura 8. Diagrama de datos de la propuesta de solución.

2.11 Diagrama de despliegue

Representa de forma visual las relaciones físicas que existen entre los componentes de software y hardware en el sistema. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

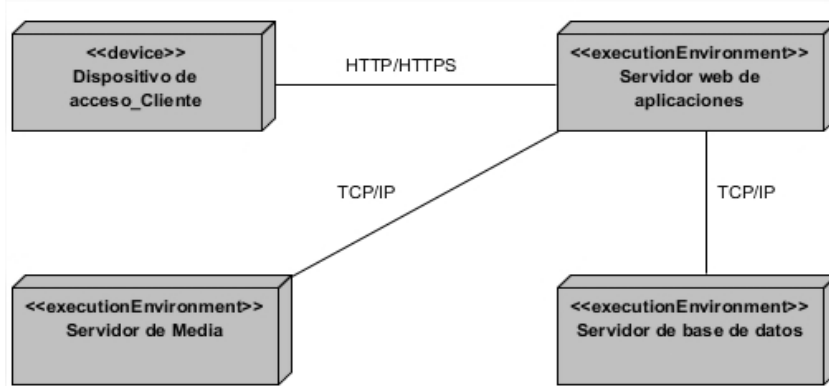


Figura 9. Diagrama de despliegue de la propuesta de solución.

Descripción de elementos:

Dispositivo de acceso_Cliente: La estación de trabajo necesita un navegador web para conectarse al sistema hospedado en el servidor de aplicaciones utilizando el protocolo de comunicación HTTP/HTTPS.

Servidor de aplicaciones: Es la estación de trabajo que hospeda el código fuente de la aplicación y que le brinda al usuario las interfaces para realizar los procesos del sistema. Esta estación se comunica con el servidor de base de datos y el servidor de media realizando la comunicación mediante el protocolo TCP/IP.

Servidor de base de datos: Este servidor es el encargado del almacenamiento de los datos del sistema. Se comunica con el servidor de aplicaciones del sistema, posibilitando el acceso mediante el usuario con privilegios para las operaciones determinadas a realizarse en el mismo.

Servidor de media: Este servidor es el encargado del almacenamiento de todos los recursos que se encuentran en el sistema. Se comunica con el servidor de aplicaciones del sistema, posibilitando el acceso mediante el usuario con privilegios para las operaciones determinadas a realizarse en el mismo.

2.12 Conclusiones parciales

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones parciales:

1. Los requisitos funcionales y no funcionales identificados a partir del proceso de determinación de requisitos permitieron desarrollar las distintas funcionalidades que debe cumplir el sistema para satisfacer las necesidades del cliente.
2. El modelo conceptual visualiza los conceptos fundamentales de la propuesta de solución en aras de un mejor entendimiento de la misma.
3. La descripción de las historias de usuario permitió identificar el comportamiento del sistema a partir de las acciones del usuario sobre el mismo.
4. La creación de los prototipos de interfaz de usuario sirvió de guía para el diseño de la propuesta de solución.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

En este capítulo se describe el flujo de implementación y las pruebas de la solución de esta investigación. Para ello se toma como punto de partida los conocimientos adquiridos en los capítulos anteriores. En esta parte de la investigación se construye el sistema teniendo en cuenta la arquitectura y los requisitos definidos para el mismo. Se presenta además un resumen de las pruebas realizadas al software para garantizar su correcto funcionamiento.

3.2 Diagrama de componentes

El diagrama de componentes muestra los componentes de un sistema de software conectados por las relaciones de dependencias lógicas entre cada uno de ellos. Provee una vista arquitectónica de alto nivel del sistema, ayudando a los desarrolladores a visualizar el camino de la implementación. Cada componente representa una unidad del código (fuente, binario o ejecutable), que permite mostrar las dependencias en tiempo de compilación y ejecución. La realización del diagrama posibilita tomar decisiones respecto a las tareas de implementación y los requisitos (43).

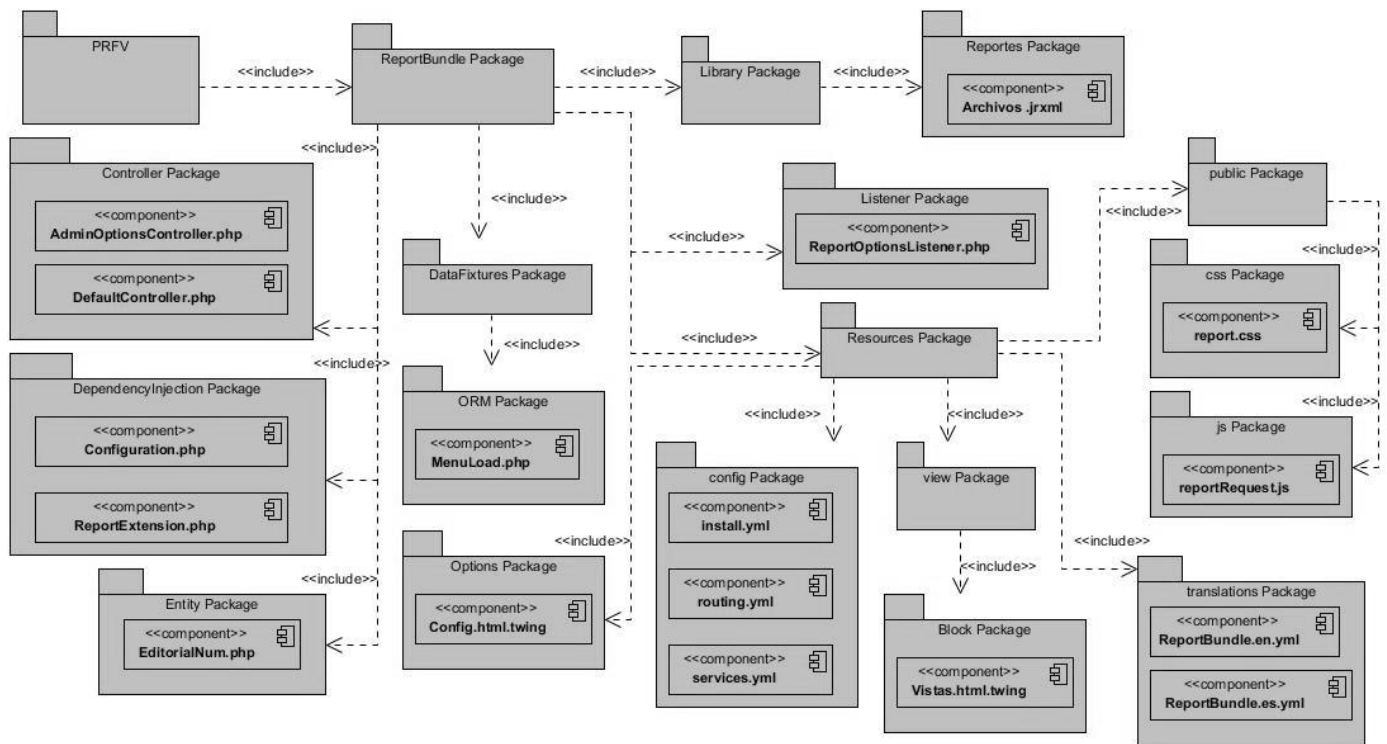


Figura 10. Diagrama de componentes de la propuesta de solución.

Descripción de los elementos principales:

Controller Package: Directorio que contiene las clases controladoras que posibilitan el funcionamiento del módulo.

Entity Package: Contiene las entidades creadas durante el desarrollo del módulo.

Reportes Package: Carpeta que contiene todos los reportes (archivos .jrxml) generados en el iReport.

config Package: Contiene los archivos de configuración del módulo para que este se integre correctamente a la plataforma Félix Varela.

view Package: Directorio donde se encuentran las vistas con las cual interactúa el usuario para generar los reportes.

3.3 Estrategia de pruebas

Una estrategia de prueba del software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. Y lo que es más importante, una estrategia de prueba del software proporciona un mapa a seguir para el responsable del desarrollo del software, a la organización de control de calidad y al cliente: un mapa que describe los pasos que hay que llevar a cabo como parte de la prueba, cuándo se deben planificar y realizar esos pasos, y cuánto esfuerzo, tiempo y recursos se van a requerir (44).

La estrategia que se ha de seguir a la hora de evaluar dinámicamente un sistema de software debe permitir comenzar por los componentes más simples e ir avanzando progresivamente hasta probar todo el software en su conjunto. Se muestran a continuación los niveles de pruebas de software:

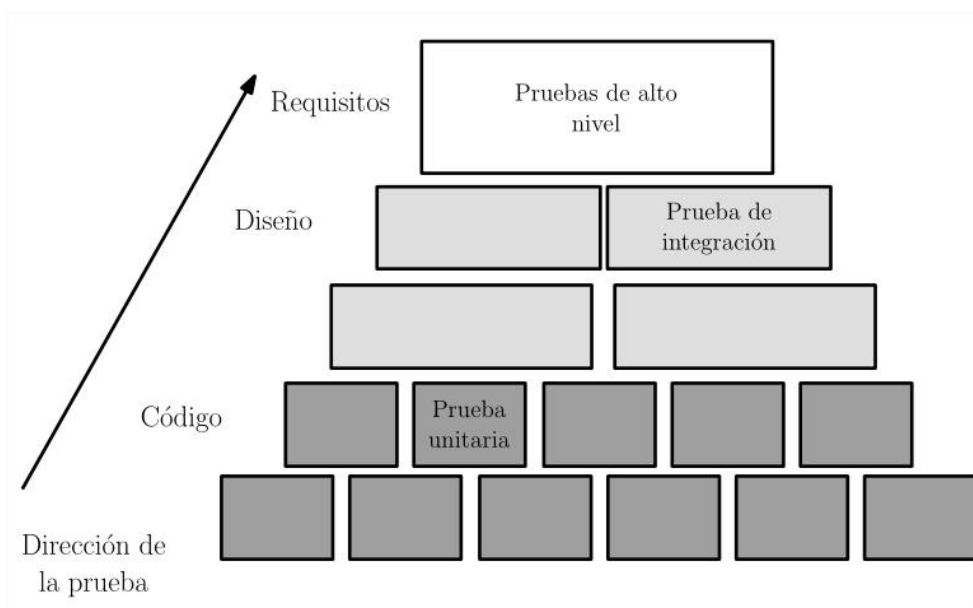


Figura 11. Niveles de pruebas (44).

Pruebas unitarias: se concentran en probar cada componente individualmente para asegurar que funcione de manera apropiada como unidad. Emplean técnicas de prueba que recorren caminos específicos en la estructura de control de los componentes.

Pruebas de integración: tienen dos objetivos principales, descubrir errores asociados con las interfaces de los módulos y ensamblar sistemáticamente los módulos individuales para formar subsistemas y al final un sistema completo.

Pruebas de validación: se enfocan en los requerimientos.

Pruebas del sistema: está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas (44).

Se decide aplicar en la propuesta de solución pruebas de validación mediante el método de caja negra o funcional utilizando la técnica partición de equivalencia y pruebas de Integración que permitieron ir comprobando la correcta integración del módulo con el sistema.

3.3.1 Método de Prueba

Las técnicas o métodos de prueba proporcionan distintos criterios para generar Casos de Pruebas que provoquen fallos en los programas. Se describe a continuación el método que se decide aplicar al sistema:

Pruebas de caja negra o funcional: también conocidas como pruebas de caja opaca, funcionales, de entrada/salida o inducidas por los datos. Este método es aplicado con el fin de verificar que las funciones son operativas a través de la interfaz del software. Valida que la entrada se acepta de forma adecuada y que se produce un resultado correcto, manteniendo la integridad de la información externa. Estas pruebas permiten encontrar: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a las bases de datos externas, errores de rendimiento, errores de inicialización y terminación (44).

Se describe a continuación en qué consiste la técnica de prueba seleccionada:

- **Partición de Equivalencia:** divide el dominio de entrada de un programa en un número finito de variables de equivalencia. Se definen dos tipos de variables de equivalencia, las válidas, que representan entradas válidas al programa, y las no válidas, que representan valores de entrada erróneos, aunque pueden existir valores no relevantes a los que no sea necesario proporcionar un valor real de dato (45).

Diseño de caso de prueba

El diseño de caso de prueba consiste en probar el sistema, se incluyen los datos de entrada y los resultados esperados. Éstos parten de las descripciones de requisitos por procesos, poseen la finalidad de encontrar la mayor cantidad de definiciones en las funcionalidades implementadas. Además de comprobar que el sistema satisface las necesidades del cliente. A continuación, se presentan los diseños de caso de prueba pertenecientes a HU: Generar reporte de libros y HU: Exportar a formato .pdf, .xls y .jpg los reportes utilizando la técnica partición en clases de equivalencia. Los restantes Diseño de Caso de Prueba (DCP) se muestran en Anexo #5.

DCP Generar reporte de libros

Descripción de las variables:

Los campos Fecha de inicio y Fecha de fin son obligatorios.

- Editorial: Permite seleccionar la editorial a la cual se realiza el reporte. Nunca el campo estará vacío pues se brinda una lista desplegable con el listado de editoriales.
- (*) Fecha de inicio: Permite seleccionar la fecha a partir de la cual se realiza el reporte.
- (*) Fecha de fin: Permite seleccionar la fecha hasta la cual se realiza el reporte.

Condiciones de ejecución

Para generar un reporte de libros hay que:

- Estar autenticado en el sistema con el rol administrador o editor.

Escenario	Descripción	Fecha de inicio	Fecha de fin	Respuesta del sistema	Flujo central
EC 1.1 Generar reporte de libros	Selecciona en el menú Reporte la opción Libro.	NA	NA	Se muestra una ventana donde el usuario debe seleccionar la editorial y el periodo en el que desea generar el reporte de libros. Los datos a seleccionar son los siguientes:	Inicio/Administración/Reporte/Libro/

				<ul style="list-style-type: none"> -Editorial - Fecha de inicio - Fecha de fin <p>Acciones:</p> <ul style="list-style-type: none"> - Cancelar - Generar 	
EC 1.2 Generar reporte de libros	Introduce el periodo en el que desea generar el reporte de libros.	V	V	<p>Se muestra una ventana con el reporte de los libros de la editorial seleccionada y en el periodo especificado por el usuario. Del mismo se muestran los siguientes datos:</p> <ul style="list-style-type: none"> -Título. - ISBN. - Categoría. - Año. - Cantidad de visitas realizadas al libro. - Cantidad de usuarios que han accedido al libro. - Evaluación general del libro. - Autor. <p>Y la acción:</p> <ul style="list-style-type: none"> -Exportar PDF -Exportar XLS -Exportar JPG 	Inicio/Administración/Reporte/Libro/Generar reporte

EC 1.3 Campos en blanco	El usuario deja campos obligatorios sin llenar.	I	NA	Se muestra un mensaje de información al usuario informándole del error ocurrido.	Inicio/Administración/Reporte/Libro/Generar reporte
		NA	I		
EC 1.4 Campos en incorrectos	El usuario introduce de forma incorrecta la fecha de inicio o fin en que se desea realizar el reporte de uso de un libro.	I	NA	Se muestra un mensaje de información al usuario informándole del error ocurrido.	Inicio/Administración/Reporte/Libro/ Generar reporte
		NA	I		
EC 1.5 Cancelar	El usuario selecciona la opción Cancelar	NA	NA	Regresa a la vista principal del panel de Administración.	Inicio/Administración/Reporte/Libro/ Generar reporte/Cancelar

Tabla 6. DCP-Generar reporte de libros.

DCP Exportar a formato .pdf los reportes			
Condiciones de ejecución			
Para exportar un reporte a formato .pdf, .xls y .jpg hay que:			
- Estar autenticado en el sistema con el rol administrador o editor.			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Exportar a formato .pdf, .xls y .jpg los reportes.	Selecciona la opción Exportar PDF o Exportar XLS o Exportar JPG.	Se muestra una ventana donde el usuario debe seleccionar la carpeta de destino del reporte de uso seleccionado en formato PDF o XLS o JPG. Las opciones varían en dependencia del navegador que utilice el usuario.	Inicio/Administración/Reporte/ Selecciona tipo de reporte/Introduce la fecha del reporte y el nombre de la editorial/Generar reporte/Exportar PDF, o Exportar XLS o Exportar JPG

Tabla 7. DCP-Exportar a formato .pdf, .xls y .jpg los reportes.

3.3.2 Resultados de las pruebas de caja negra

Los problemas detectados en el período de prueba se clasificaron en: No conformidades significativas (NCS) y en No conformidades no significativas (NCNS). Los aspectos que se tienen en cuenta en cada clasificación son:

NCS: las no conformidades en cuanto a las funcionalidades del componente: validaciones incorrectas o respuestas del sistema diferentes a lo descrito previamente en las historias de usuario.

NCNS: las no conformidades en cuanto al diseño de la propuesta de solución y errores ortográficos.

Para las pruebas funcionales fueron realizadas 3 iteraciones. En la primera iteración se detectaron 3 NCS y 8 NCNS asociadas a errores de interfaz y de funcionalidades incorrectas, las mismas fueron resueltas satisfactoriamente en la misma iteración. En la segunda iteración se detectaron 1 NCS y 3 NCNS asociadas a errores de interfaz y validaciones incorrectas siendo solucionadas de la misma forma que las anteriores. En la tercera iteración no se detectaron NCS ni NCNS por lo que se demostró que la aplicación cumplía con los requisitos funcionales establecidos y fue considerada concluida. A continuación, se muestra un gráfico con un resumen de los resultados obtenidos tras la realización de las pruebas funcionales:

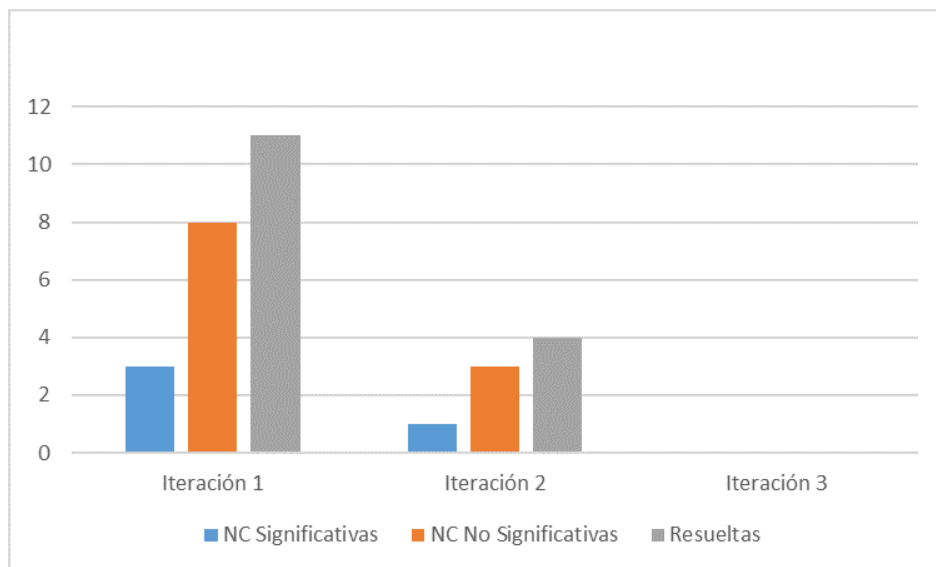


Figura 12. Resumen de las no conformidades detectadas.

3.3.3 Resultados de las pruebas de integración

Las pruebas de integración son definidas para verificar el correcto ensamblaje entre los distintos módulos que conforman un sistema informático (44). Para ello existen dos enfoques:

- **Big bang:** combinar todos los componentes y probar el sistema como un todo.
- **Integración incremental:** los componentes se integran y prueban poco a poco. Se encuentra la Integración descendente (componentes de funcionalidades) y la Integración ascendente (componentes de infraestructura) (44).

Se realizaron las pruebas en el sistema mediante el enfoque Integración incremental basado en la Integración descendente, donde a partir de 5 Casos de Pruebas (CDP) basados en las historias de usuarios se comprobó que el módulo de reportes se integra correctamente a la plataforma. Se muestran a continuación los CDP correspondiente a HU: Generar reporte de libros y HU: Exportar a formato .pdf, .xls y .jpg los reportes. Los restantes casos de pruebas se muestran en Anexo #6.

Caso de Prueba #1	
<p>Descripción de las variables:</p> <p>Los campos Fecha de inicio y Fecha de fin son obligatorios.</p> <ul style="list-style-type: none"> - (*) Fecha de inicio: Permite seleccionar la fecha a partir de la cual se realiza el reporte. - (*) Fecha de fin: Permite seleccionar la fecha hasta la cual se realiza el reporte. 	
<p>Condiciones de ejecución</p> <p>Para generar un reporte de libros hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol administrador o editor. 	
Historia de usuario:	Generar reporte de libros.
Variables de entrada:	Fecha de inicio, fecha de fin
Tablas implicadas:	tb_author, book_author, tb_book, tb_category
Consulta SQL realizada:	<pre>SELECT tb_book."title" AS tb_book_title, tb_book."year" AS tb_book_year, tb_book."visits" AS tb_book_visits, tb_book."user_fav" AS tb_book_user_fav, tb_book."evaluationsavg" AS tb_book_evaluationsavg, tb_book."updated_at" AS tb_book_updated_at, book_author."book_id" AS book_author_book_id, book_author."author_id" AS book_author_author_id, tb_author."name" AS tb_author_name, tb_book."isbn" AS tb_book_isbn, tb_category."name" AS tb_category_name</pre>

	<pre> FROM "pkt_book"."tb_author" tb_author INNER JOIN "public"."book_author" book_author ON tb_author."id" = book_author."author_id" INNER JOIN "pkt_book"."tb_book" tb_book ON book_author."book_id" = tb_book."id" INNER JOIN "pkt_editorial"."tb_category" tb_category ON tb_book."category_id" = tb_category."id" WHERE tb_book."updated_at" >= \${Fecha1} and tb_book."updated_at" <=\${Fecha2} OR tb_book."created_at" >= \${Fecha1} and tb_book."updated_at" <= \${Fecha2} ORDER BY tb_book."visits" DESC </pre>
Datos obtenidos:	Se obtuvieron 24 libros visitados entre las fechas entradas.
Resultado esperado:	La herramienta SDR crea el reporte y lo envía a la plataforma Félix Varela.
Resultado prueba:	Satisfactorio

Tabla 8. CDP-Generar reporte de libros.

Caso de Prueba #5	
<p>Condiciones de ejecución</p> <p>Para exportar un reporte a formato .pdf, .xls y .jpg hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol administrador o editor. 	
Historia de usuario:	Exportar a formato .pdf, .xls y .jpg los reportes
Variables de entrada:	fecha de inicio, fecha de fin
Tablas implicadas:	tb_author, book_author, tb_book, tb_category
Consulta SQL realizada:	<pre> SELECT tb_book."title" AS tb_book_title, tb_book."year" AS tb_book_year, tb_book."visits" AS tb_book_visits, tb_book."user_fav" AS tb_book_user_fav, tb_book."evaluationsavg" AS tb_book_evaluationsavg, tb_book."updated_at" AS tb_book_updated_at, book_author."book_id" AS book_author_book_id, book_author."author_id" AS book_author_author_id, tb_author."name" AS tb_author_name, tb_book."isbn" AS tb_book_isbn, </pre>

	<pre> tb_category."name" AS tb_category_name FROM "pkt_book"."tb_author" tb_author INNER JOIN "public"."book_author" book_author ON tb_author."id" = book_author."author_id" INNER JOIN "pkt_book"."tb_book" tb_book ON book_author."book_id" = tb_book."id" INNER JOIN "pkt_editorial"."tb_category" tb_category ON tb_book."category_id" = tb_category."id" WHERE tb_book."updated_at" >= \${Fecha1} and tb_book."updated_at" <=\${Fecha2} OR tb_book."created_at" >= \${Fecha1} and tb_book."updated_at" <= \${Fecha2} ORDER BY tb_book."visits" DESC </pre>
Datos obtenidos:	Se obtuvieron 24 libros visitados entre las fechas entradas.
Resultado esperado:	La herramienta SDR crea el reporte y lo exporta a formato .pdf, .xls y .jpg.
Resultado prueba:	Satisfactorio

Tabla 9. CDP-Exportar a formato .pdf, .xls y .jpg los reportes.

3.4 Conclusiones parciales

Como resultado del desarrollo de este capítulo se determinan las siguientes conclusiones parciales:

1. El diagrama de componentes permitió una vista arquitectónica de alto nivel del sistema, facilitando visualizar el camino de la implementación.
2. Se realizó la descripción de los casos de prueba con el objetivo que sean usados para verificar el correcto funcionamiento de la aplicación.
3. Se puede afirmar que una vez aplicadas las pruebas los resultados fueron satisfactorios ya que se identificaron a tiempo los problemas presentados por la aplicación y se les suministró solución de forma inmediata.

CONCLUSIONES

Con la realización de la presente investigación se brinda solución a los objetivos trazados obteniéndose como principales resultados:

1. El análisis de los elementos teóricos y principales tendencias de los reportes y sistemas de reportes permitió implementar una aplicación actualizada en tecnología, así como la selección adecuada de las herramientas para su desarrollo.
2. El análisis de la arquitectura y la aplicación de patrones de diseño permitieron comprender los aspectos relacionados con los requisitos del sistema, descomponer los trabajos de la implementación en partes creando así un punto de partida para la implementación.
3. Se integró el sistema SDR a la Plataforma de recursos de la Editorial Universitaria Félix Varela.
4. Mediante la puesta en práctica de las pruebas de caja negra y pruebas de integración, se pudo detectar y corregir los errores encontrados, logrando con esto verificar el perfecto funcionamiento de la solución propuesta.
5. Se obtuvo un módulo de reportes para la Plataforma de recursos de la Editorial Universitaria Félix Varela, que brinda apoyo a la toma de decisiones de la misma.

RECOMENDACIONES

Tomando como base la investigación realizada y el análisis de los resultados obtenidos se recomienda añadir nuevos reportes que contribuyan en el apoyo de toma de decisiones por parte de los administradores, por ejemplo:

- ✓ Promedio mensual de visitantes.
- ✓ Cantidad de consultas que recibe la plataforma por día, mes y año.
- ✓ Origen geográfico de la consulta, con un listado de las provincias que más acceden a la plataforma.
- ✓ Reportes de ventas.

REFERENCIAS BIBLIOGRÁFICAS

1. **Agencia del derecho de autor universitario.** Principal | Editorial Félix Varela. [En línea] 2013. [Citado el: 9 de Enero de 2017.] <http://www.epfv.com.cu/>.
2. **Fernández, Diana.** Reporte o Informe de investigación. [En línea] 2013. [Citado el: 16 Noviembre 2016] <https://prezi.com/e4ippy0basxk/reporte-o-informe-de-investigacion/>.
3. **¿Qué es reporte? Concepto y significado.** [En línea] 2007. [Citado el: 23 de Noviembre de 2016.] <https://diccionarioactual.com/reporte/>.
4. **Definición.** [En línea] 2007. [Citado el: 23 de Noviembre de 2016.] <http://definicion.de/reporte/>.
5. **Enciclopedia de Clasificaciones (2017).** "Tipos de informes". [En línea] 2017. [Citado el: 10 de Junio de 2017.] <http://www.tiposde.org/documentos/181-tipos-de-informes/>.
6. **León Solís, Prof.^a Miriam.** Curso-Taller de Redacción Artículos Científicos de SAN. Universidad de Costa Rica: 2005.
7. **Martínez, Prof. Edgar Martínez.** Sistemas de Reportes. - ITS EL GRULLO Inteligencia de Negocios. [En línea] 2009. [Citado el: 23 de Noviembre de 2016.] <https://sites.google.com/site/itsginteligenciadenegocios/home/1-2-componentes-de-la-inteligencia-de-negocios/1-2-4-sistemas-de-reportes>.
8. **Janium | Sistema de gestión de bibliotecas.** [En línea] 2014. [Citado el: 26 de Marzo de 2017.] <http://www.janium.com/janium/>.
9. **BITECA. KOHA para bibliotecas | Biteca Ltda.** [En línea] 2010. [Citado el: 26 de Marzo de 2017.] http://www.biteca.com/koha_bibliotecas.
10. **EI VERSAT-Sarasola: Sistema Cubano de Gestión Contable-Financiero.** [En línea] 2010. [Citado el: 20 de Noviembre de 2016.] http://www.betsime.disaic.cu/secciones/eco_enemar_07.htm.
11. **SAMPIERI, R, et al.** Metodología de la investigación. McGraw-Hill Education, 2014.
12. **Sánchez, Tamara Rodríguez.** *Metodología de desarrollo para la Actividad Productiva de la UCI.* La Habana, Cuba: UCI : s.n., 2014.

13. **Martínez, Prof. Miguel Vitala.** Artículos y tutoriales sobre php. [En línea] 2014. [Citado el: 15 de Diciembre de 2016.] <http://www.miguelvilata.com/blog/tag/php>.
14. **programcion.net.** [En línea] [Citado el: 17 de noviembre de 2016.] http://programacion.net/articulo/10_cosas_sobre_php_7_que_deberias_saber_1086.
15. **Qué es HTML5.** [En línea] 2013. [Citado el: 21 de Noviembre de 2016.] <https://hipertextual.com/archivo/2013/05/entendiendo-html5-guia-para-principiantes/>.
16. **THE WORLD´S LARGEST WEB DEVELOPER SITE.** CSS3 Introduction. [En línea] 2015. [Citado el: 24 de Noviembre de 2016.] http://www.w3schools.com/css/css3_intro.asp.
17. **Salgado, Victor Mijangos.** Para qué se utiliza JavaScript. [En línea] 2011. [Citado el: 15 de Diciembre de 2016.] <http://culturacion.com/para-que-se-utiliza-javascript/>.
18. **jQuery.** [En línea] 2015. [Citado el: 21 de Noviembre de 2016.] <http://jquery.com/>.
19. **Rafaelma.** Sobre PostgreSQL. [En línea] 2010. [Citado el: 15 de Diciembre de 2016.] http://www.postgresql.org.es/sobre_postgresql.
20. **PostgreSQL: Documentation: Release 9.5.** [En línea] 7 de Enero de 2016. [Citado el: 26 de Noviembre de 2016.] <https://www.postgresql.org/docs/9.6/static/release-9-5.html>.
21. **PgAdmin III - Guía Ubuntu.** [En línea] 2008. [Citado el: 22 de Noviembre de 2016.] http://www.guia-ubuntu.com/index.php/PgAdmin_III.
22. **¿Qué es nginx?.** [En línea] 2016. [Citado el: 27 de Noviembre de 2016.] <http://culturacion.com/que-es-nginx/>.
23. **Marín, Edson Mollericona.** Introducción e instalación del Framework Symfony 2. [En línea] 25 de Junio de 2015. [Citado el: 27 de Noviembre de 2016.] <http://www.edsonmm.com/introduccion-e-instalacion-del-framework-symfony-2/>.
24. **Nuevo en Symfony 2.7: Mejoras para ser más ProductivoPhpCuba | PhpCuba.** [En línea] 23 de Abril de 2015. [Citado el: 27 de Noviembre de 2016.] <https://php.uci.cu/nuevo-en-symfony-2-7-mejoras-para-ser-mas-productivo/>.

25. **Conceptos básicos (Symfony 1.2, guía definitiva).** [En línea] 2015. [Citado el: 21 de Noviembre de 2016.] http://librosweb.es/libro/symfony_1_2/capitulo_1/conceptos_basicos.html.
26. **About - Bootstrap.** [En línea] 2011. [Citado el: 24 de Noviembre de 2016.] <http://getbootstrap.com/about/#history>.
27. **Heffelfinger.** Java EE 7 Development with Netbeans 8. [En línea] 2015. [Citado el: 8 de Marzo de 2017.] https://books.google.com.cu/books?hl=es&lr=&id=VY92BgAAQBAJ&oi=fnd&pg=PP1&dq=Java+EE+7+Development+with+NetBeans+8+&ots=P4THdparyA&sig=EE_ZjEVFhxNIFqv7KbxRmMJiKzk&redir_esc=y#v=onepage&q=Java%20EE%207%20Development%20with%20NetBeans%208&f=false.
28. **Visual Paradigm.** [En línea] 2015. [Citado el: 8 de Marzo de 2017.] <http://www.visual-paradigm.com>.
29. **Silva Hernández, Lic. Iliana Amabely.** *Tesis de Maestría: Generador Automático de Reportes Dinámicos.* México, D.F. 2003.
30. **Cervantes, Ing. Beatriz Hernández.** *Manual de Usuarios SDRv2.0.* La Habana, Cuba: UCI : s.n., 2016.
31. **Contreras, Roger Armando.** *Manual de Ireport.* [En línea] 2012. [Citado el: 22 de Marzo de 2017.] <https://es.scribd.com/doc/37388195/Manual-de-Ireport>.
32. **Larman, Craig.** *Modelo del Dominio.* 2003.
33. **SOMMERVILLE, I.** *Ingeniería del software. Séptima edición.* S.I.: PEARSON EDUCACIÓN, S.A. Madrid, España: 2005. ISBN 84-7829-074-5.
34. **Departamento de ciencias de la computación-Universidad de Granada.** *Requeriment.* Granada, España : s.n., 2010.
35. **Francois, Potencier, Fabien y Zaninotto.** *El patrón MVC (Symfony 1.4).* [En línea] 2008. [Citado el: 13 de Marzo de 2017.] http://librosweb.es/libro/symfony_1_4/capitulo_2/el_patron_mvc.html.
36. **Larman, Craig.** *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado.* [En línea] 2003. [Citado el: 12 de Marzo de 2017.] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.

37. **Ignacio, Susana y Aedo.** Ingeniería de la web y patrones de diseño. [En línea] 2005. [Citado el: 13 de Marzo de 2017.] <https://www.casadellibro.com/libro-ingenieria-de-la-web-y-patrones-de-diseno/9788420546094/1046443>.
38. **Rodríguez, Erick.** *Patrones del "Gang of Four"*. Habana, Cuba : s.n., 2012.
39. **Sweat, Jason E.** *Guide to PHP Design Patterns*. 2005. ISBN 0-9735898-2-5.
40. **Linares, Manuel, otros.** Universidad Bicentenario de Aragua . [En línea] 2013. [Citado el: 26 de Marzo de 2017.] <http://es.slideshare.net/nedowwhaw/diagrama-de-clases-16208245>.
41. **García, Prof. Fabían.** Estudio de los diagramas de clases. Universidad de Chile: 2008.
42. **Jacobson.** *El proceso unificado de desarrollo de Software*. Madrid, España: s.n., 2000.
43. **Alva, Eduardo Rivera.** Arquitectura de software II. Diagramas de Componentes y Despliegue. [En línea] 2008. [Citado el: 7 de Marzo de 2017.] <http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.
44. **Rodríguez Tello, Dr. Eduardo A.** *Estrategias y técnicas de prueba del software*. Cinvestav, Tamaulipas: 2012.
45. **PRESSMAN, R.** Software Engineering A Practitioner's Approach Seventh Edition. [En línea] 2005. [Consultado el: 2 de abril 2017] <http://www.vumultan.com/Books/CS605-Software%20Engineering%20Practitioner%E2%80%99s%20Approach%20%20by%20Roger%20S.%20Pressman%20.pdf>

ANEXOS

Anexo #1: Entrevista a la jefa del proyecto Plataforma de recursos para la Editorial Universitaria Félix Varela.

Hablar del proyecto Plataforma de recursos para la Editorial Universitaria Félix Varela es hablar de responsabilidad, esfuerzo y sacrificio, es juntar un conjunto de especialistas con el objetivo de garantizar un software de calidad. Hablo con su jefa de proyecto Yenima Hernández Orozco.

¿Cómo se realiza actualmente el proceso de elaboración de reportes en la plataforma?

Actualmente los reportes son generados por el administrador de la plataforma y el editor de una editorial específica. Una vez desplegado el listado de libros se brinda la opción reporte en el menú derecho de cada libro. Al ejecutar esta acción se genera el reporte referente al libro seleccionado, brindando además poder exportarlo en formato PDF.

¿De qué forma se visualiza el reporte de libro?

El reporte se visualiza en forma de listado mostrando el título del libro, la portada, la cantidad de visitas, la cantidad de usuarios que lo tienen marcado como favorito, entre otros elementos del libro.

¿Qué tipos de reportes les gustaría añadir a la plataforma?

Reportes asociados al acceso de los usuarios al sistema, por ejemplo: ¿Cuáles son los usuarios con más visitas en el sistema? Otros reportes referentes al acceso a los contenidos publicados en la plataforma, por ejemplo: ¿Cuáles son los libros y recursos más visitados? y reportes de editorial que permitan por ejemplo conocer los autores favoritos, las categorías favoritas, los libros más visitados de la editorial.

¿Qué crees q se pueda hacer para mejorar el proceso?

Lo ideal sería la creación de un bundle enfocado solamente en la generación de reportes que nos permita la creación y gestión de estos.

Anexo #2: Historias de Usuario

Número: 2		Nombre del requisito: Generar reporte de recursos
Programador: Oscar Luis Ambruster Jiménez		Iteración Asignada: 1
Prioridad: Alta		Tiempo Estimado: 72 horas
Riesgo en Desarrollo: Ausencia del desarrollador por enfermedad y problemas con la tecnología usada.		Tiempo Real: 72 horas
Descripción:		
1- Objetivo:		

Permitir generar el reporte del uso de varios recursos en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

- Para generar el reporte de varios recursos deben haber sido creados previamente en el sistema.
- El usuario debe tener rol Editor o Administrador.

3- Comportamientos válidos y no válidos (flujo central y alternos):

Los campos Fecha de inicio y Fecha de fin son obligatorios.

- (*) Fecha de inicio: Permite seleccionar la fecha a partir de la cual se realiza el reporte.
- (*) Fecha de fin: Permite seleccionar la fecha hasta la cual se realiza el reporte.

4- Flujo de la acción a realizar:

Para generar el reporte de uso de varios recursos el usuario debe seleccionar la opción **Reportes** del panel de Administración. Una vez que el usuario selecciona el rango de fechas para el reporte y selecciona la opción **Generar reporte** se muestran los siguientes datos de los recursos: nombre, carrera, asignatura, cantidad de visitas realizadas al recurso, autor, email del autor. Si el usuario selecciona la opción Cancelar regresará a la página **Principal** del panel de Administración.

Observaciones:

Prototipo de interfaz:

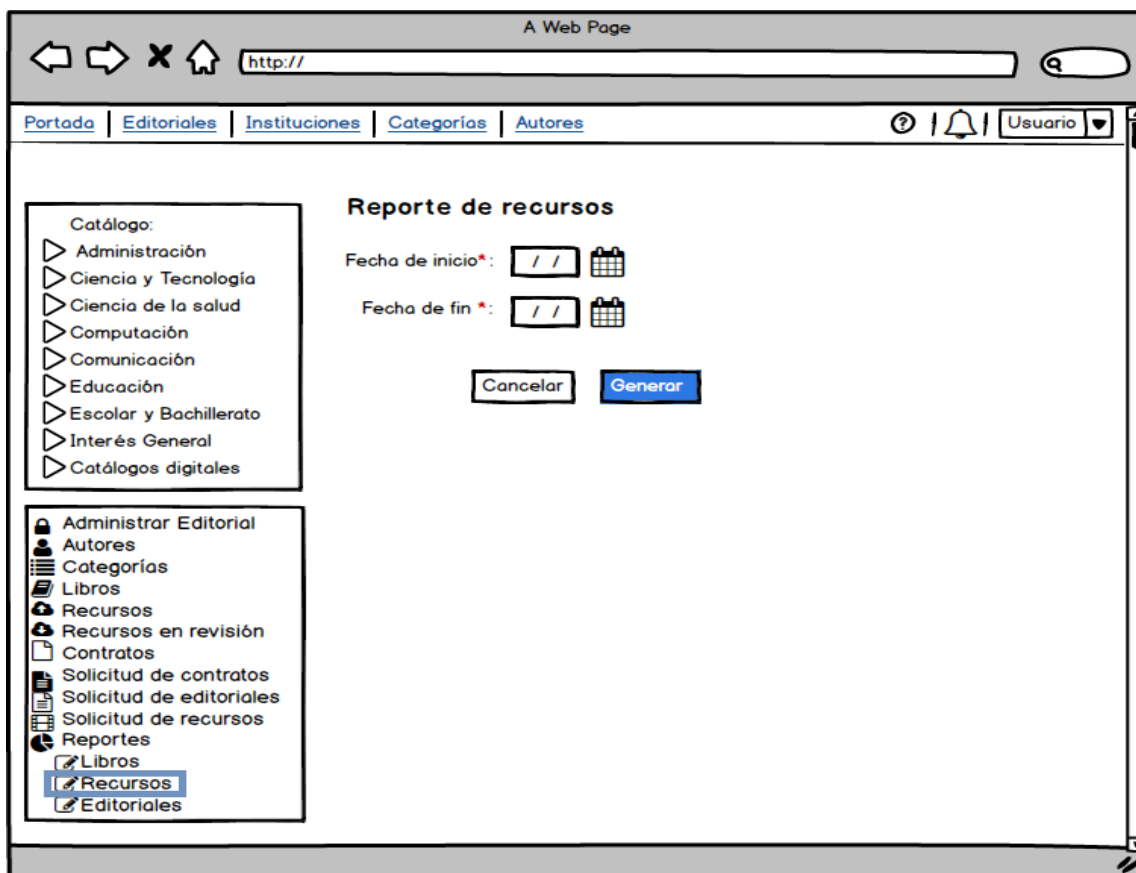


Tabla 10. HU Generar reporte de recursos.

Número: 3	Nombre del requisito: Generar reporte de editorial.	
Programador: Oscar Luis Ambruster Jiménez	Iteración Asignada: 1	
Prioridad: Alta	Tiempo Estimado: 72 horas	
Riesgo en Desarrollo: Ausencia del desarrollador por enfermedad y problemas con la tecnología usada.	Tiempo Real: 72 horas	
<p>Descripción:</p> <p>1- Objetivo: Permitir generar el reporte de una editorial.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <ul style="list-style-type: none"> - Para generar el reporte de una editorial debe haber sido creada previamente en el sistema. - El usuario debe tener rol Editor o Administrador. <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <p>Los campos Fecha de inicio y Fecha de fin son obligatorios.</p> <ul style="list-style-type: none"> - (*) Seleccionar editorial: Campo que permite seleccionar la editorial a la que se le quiere generar el reporte. <p>4- Flujo de la acción a realizar:</p> <p>Para generar el reporte de uso de una editorial el usuario debe seleccionar la opción Reportes del panel de Administración. Una vez que el usuario selecciona una editorial y selecciona la opción Generar reporte se muestran los siguientes datos de la editorial seleccionada: libros más solicitados en la editorial, mejor valorado; usuarios más asiduos, categorías favoritas, autores favoritos. Si el usuario selecciona la opción Cancelar regresará a la página Principal del panel de Administración.</p>		
Observaciones:		
Prototipo de interfaz:		

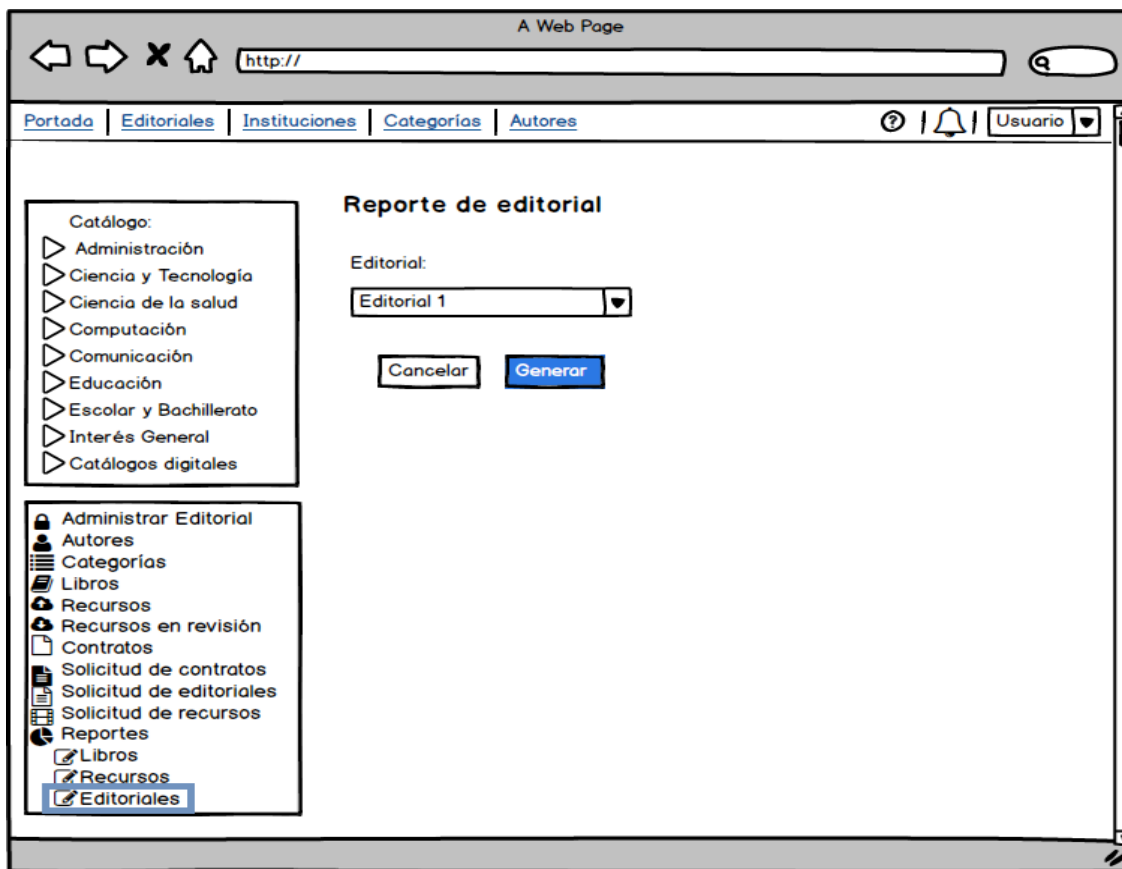


Tabla 11. HU Generar reporte de editorial.

Número: 4	Nombre del requisito: Generar reporte de plataforma.	
Programador: Oscar Luis Ambruster Jiménez	Iteración Asignada: 1	
Prioridad: Alta	Tiempo Estimado: 72 horas	
Riesgo en Desarrollo: Ausencia del desarrollador por enfermedad y problemas con la tecnología usada.	Tiempo Real: 72 horas	
Descripción:		
1- Objetivo:		
Permitir generar el reporte sobre las distintas actividades que se llevan a cabo en la plataforma.		
2- Acciones para lograr el objetivo (precondiciones y datos):		
- El usuario debe tener rol Administrador.		
3- Flujo de la acción a realizar:		
Para consultar el reporte de la plataforma el usuario debe entrar al panel de Administración. Una vez dentro se muestran los siguientes datos de la plataforma: clientes especiales, autores favoritos, ranking de libros y recursos más visitados, promedio de visitas a libros y recursos, editoriales destacadas, autores destacados.		

Observaciones:

Prototipo de interfaz:

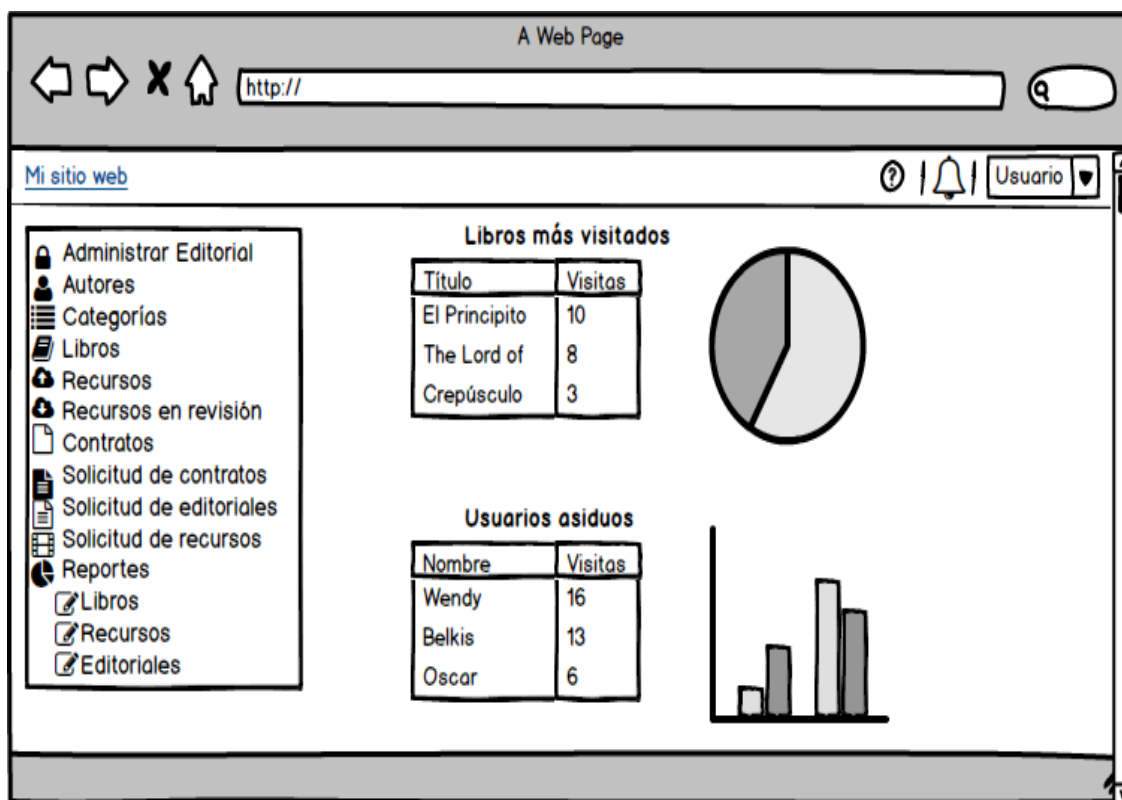


Tabla 12. HU Generar reporte de plataforma.

Anexo #3: Diagrama de clases del diseño

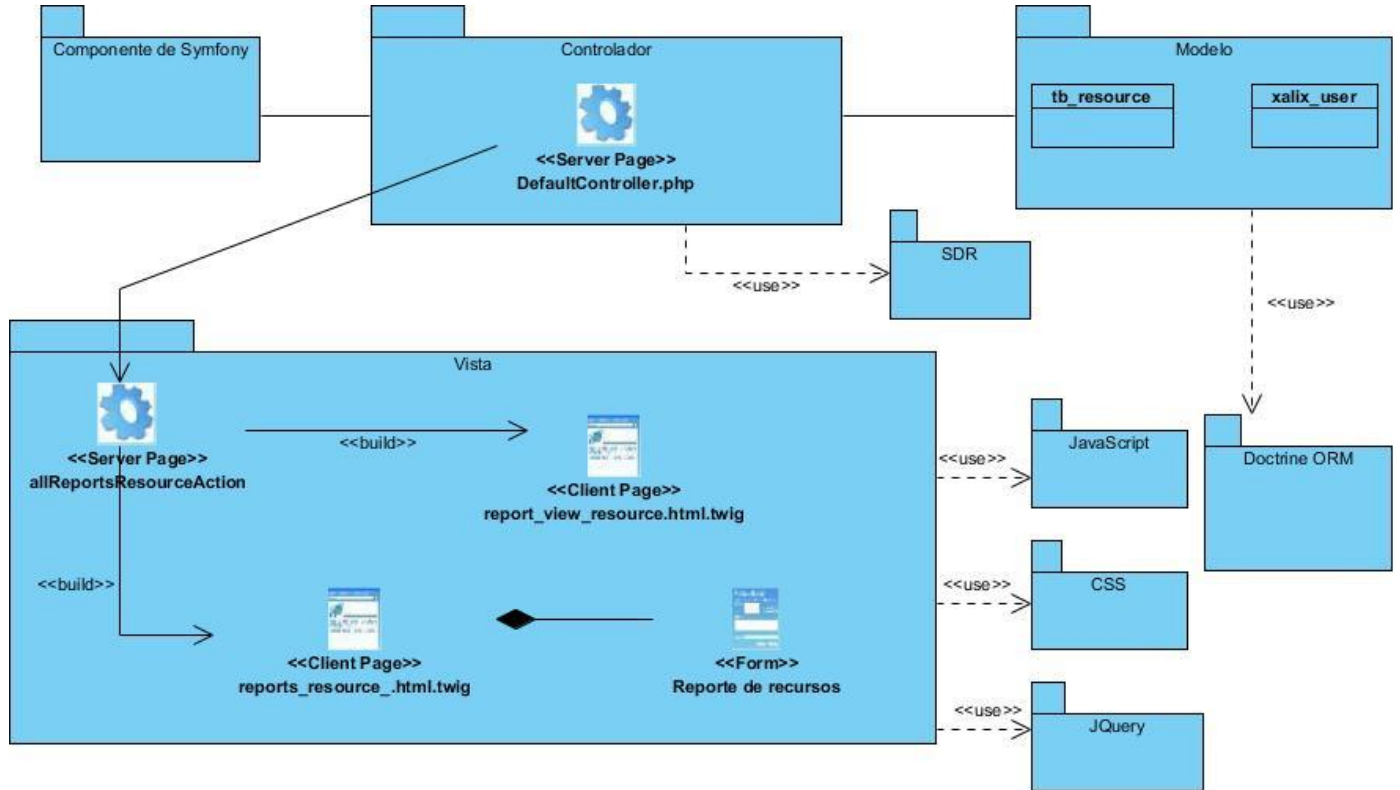


Figura 13. DCD Generar reporte de recursos.

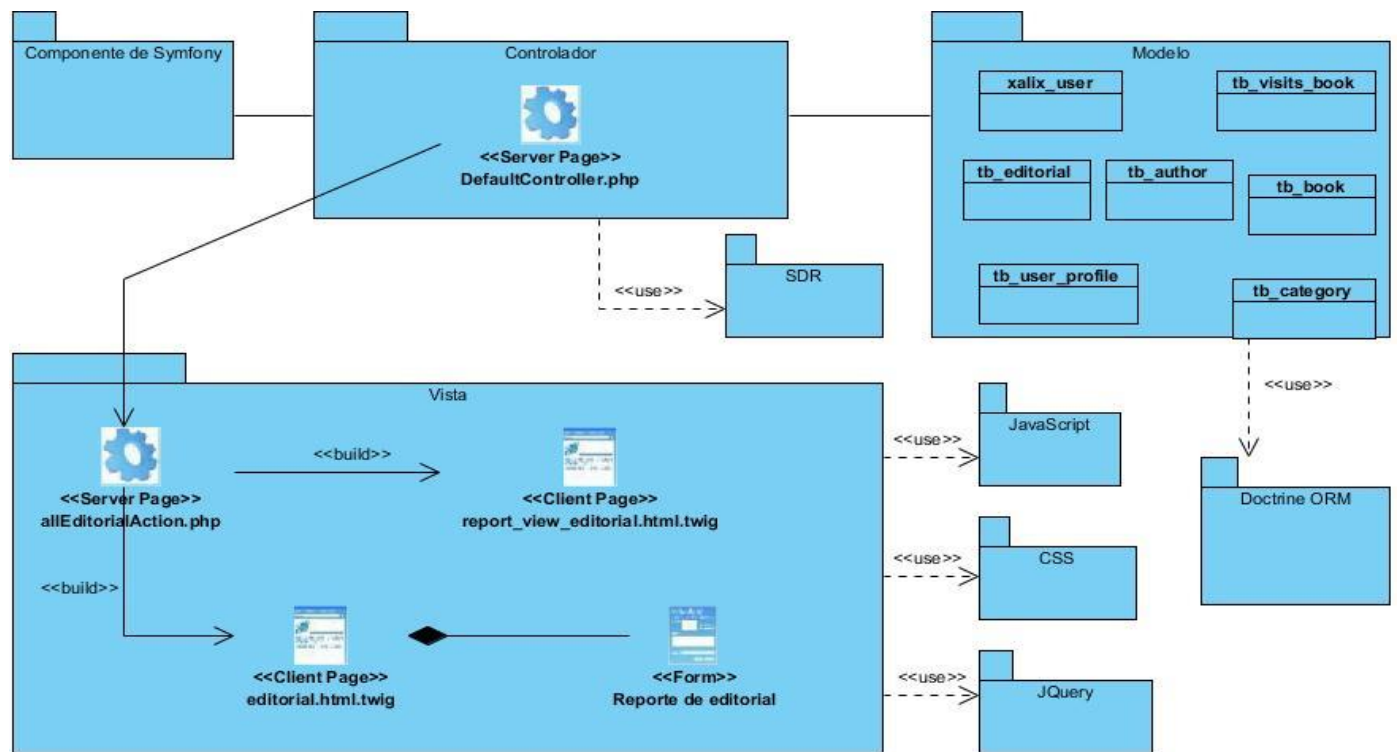


Figura 14. DCD Generar reporte de editorial.

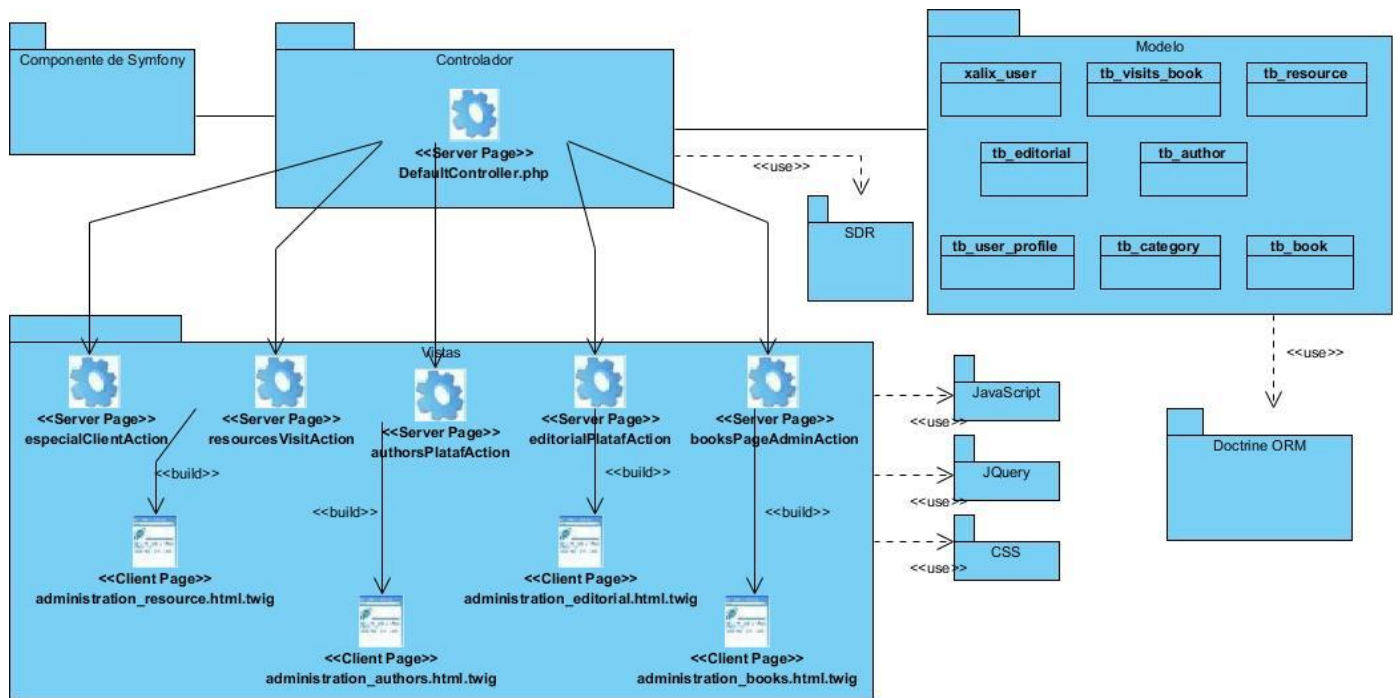


Figura 15. DCD Generar reporte de plataforma.

Anexo #4: Diagrama de clases de secuencia del diseño

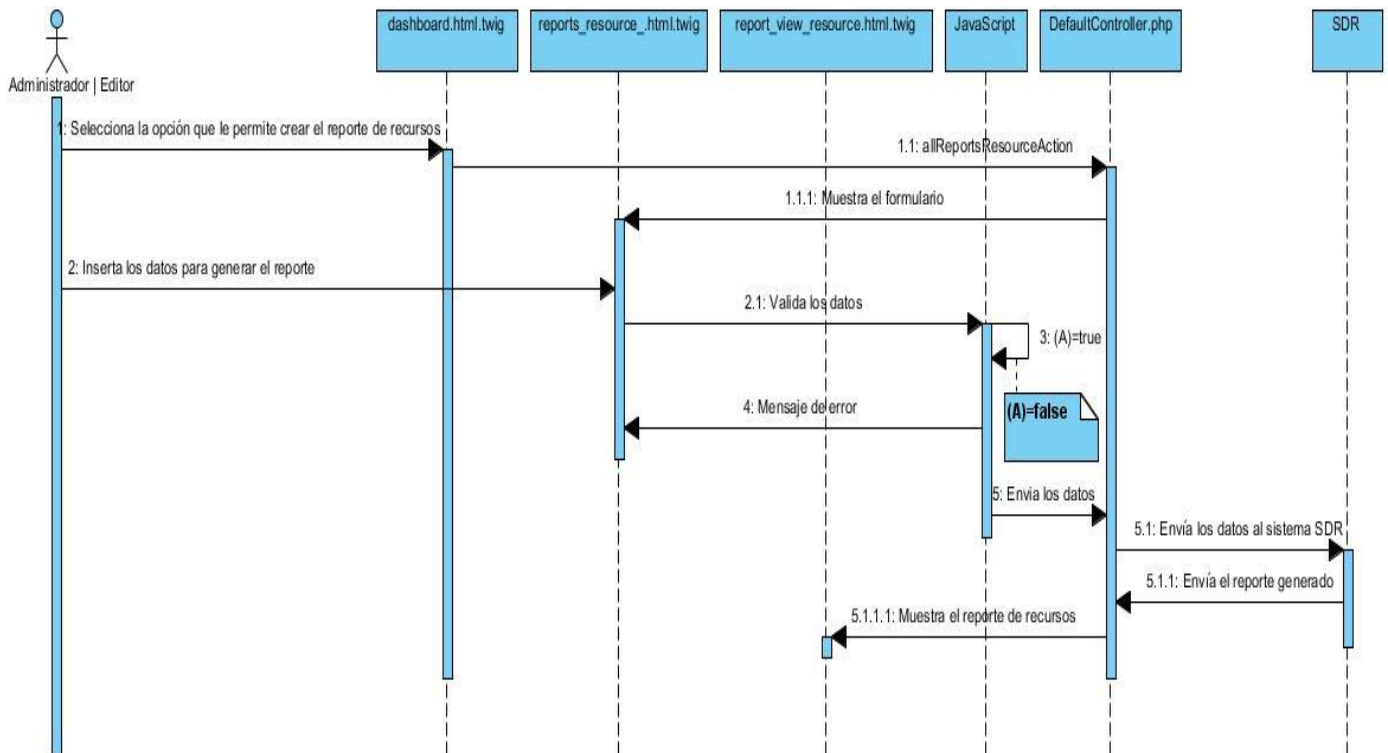


Figura 16. DCSD Generar reporte de recursos.

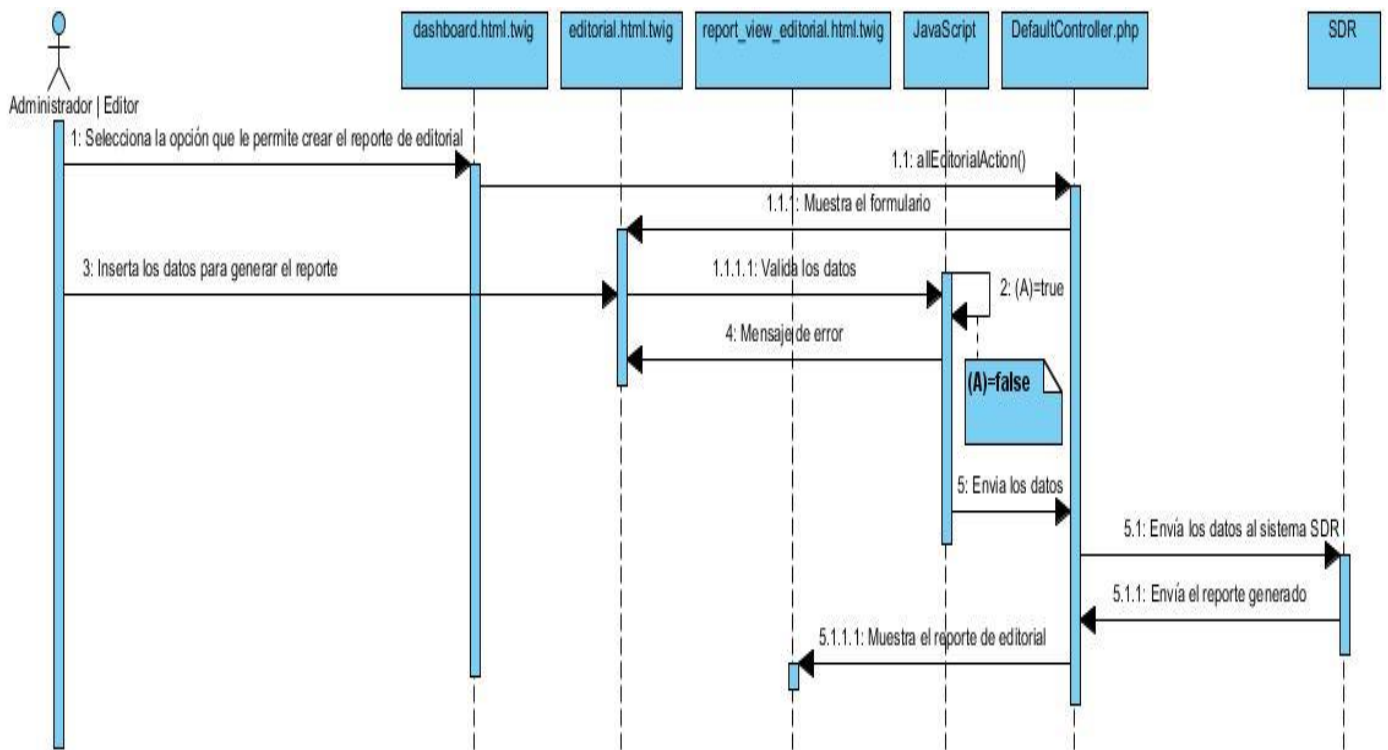


Figura 17. DCSD Generar reporte de editorial.

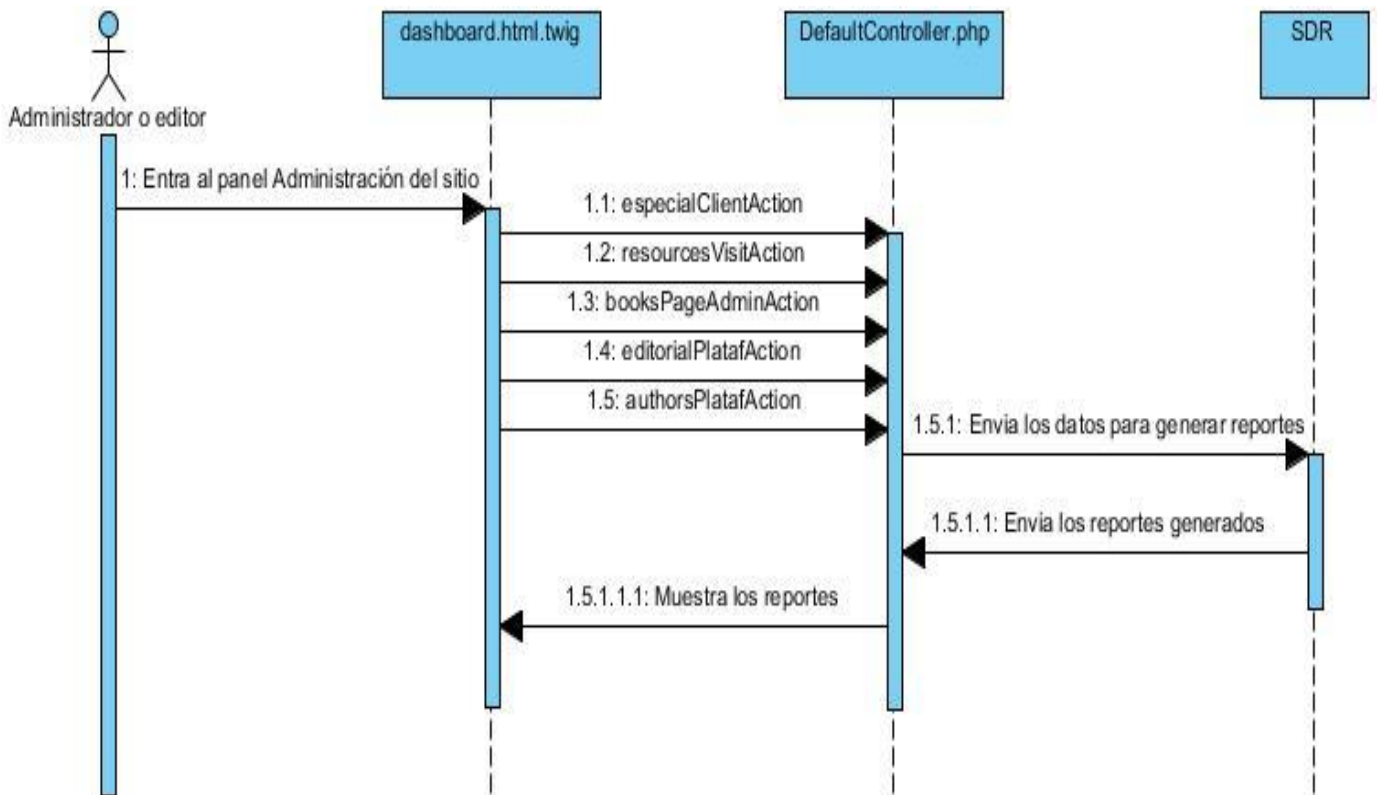


Figura 18. DCD Generar reporte de plataforma.

Anexo #5: Pruebas Funcionales

DCP Generar reporte de recursos

Descripción de las variables:

Los campos Fecha de inicio y Fecha de fin son obligatorios.

- (*) Fecha de inicio: Permite seleccionar la fecha a partir de la cual se realiza el reporte.
- (*) Fecha de fin: Permite seleccionar la fecha hasta la cual se realiza el reporte.

Condiciones de ejecución

Para generar un reporte de recursos hay que:

- Estar autenticado en el sistema con el rol administrador o editor.

Escenario	Descripción	Fecha de inicio	Fecha de fin	Respuesta del sistema	Flujo central
EC 1.1 Generar reporte de recursos	Selecciona en el menú Reporte la opción Recursos.	NA	NA	Se muestra una ventana donde el usuario debe ingresar el periodo en el que desea generar el reporte de recursos. Los datos a seleccionar son los siguientes: - Fecha de inicio - Fecha de fin Acciones: - Cancelar - Generar	Inicio/Administración/Reporte/Recursos/

EC 1.2 Generar reporte de recursos	Introduce el periodo en el que desea generar el reporte de recursos.	V	V	Se muestra una ventana con el reporte de los recursos en el período especificado por el usuario. Del mismo se muestran los siguientes datos: - Nombre. - Carrera. - Asignatura. - Cantidad de visitas realizadas al recurso. - Autor. - Email del autor. Y la acción: -Exportar PDF -Exportar XLS -Exportar JPG	Inicio/Administración/Reporte/Recursos/ Generar reporte
EC 1.3 Campos en blanco	El usuario deja campos obligatorios sin llenar.	I	NA	Se muestra un mensaje de información al usuario informándole del error ocurrido.	Inicio/Administración/Reporte/Recursos/Generar reporte
		NA	I		
EC 1.4 Campos en incorrectos	El usuario introduce de forma incorrecta la fecha de inicio o fin en que se desea realizar el reporte de uso de un libro.	I	NA	Se muestra un mensaje de información al usuario informándole del error ocurrido.	Inicio/Administración/Reporte/Libro/Generar reporte
		NA	I		
EC 1.5 Cancelar	El usuario selecciona la opción Cancelar	NA	NA	Regresa a la vista principal del panel de Administración.	Inicio/Administración/Reporte/Recursos/Generar reporte/Cancelar

Tabla 13. DCP Generar reporte de recursos.

Generar reporte de editorial

Descripción de las variables:

- Editorial: Permite seleccionar la editorial a la cual se realiza el reporte. Nunca el campo estará vacío pues se brinda una lista desplegable con el listado de editoriales.

Condiciones de ejecución

Para generar un reporte de editorial hay que:

- Estar autenticado en el sistema con el rol administrador o editor.

Escenario	Descripción	Número	Respuesta del sistema	Flujo central
EC 1.1 Generar reporte de editorial	Selecciona en el menú Reporte la opción Editorial.	NA	Se muestra una ventana donde el usuario debe seleccionar el nombre de la editorial a la que desea generar el reporte. Los datos a seleccionar son los siguientes: - Editorial Acciones: - Cancelar - Generar	Inicio/Administración/Reporte/Editorial/
EC 1.2 Generar reporte de libros	Introduce el número de la editorial a la que desea generar el reporte de libros.	V	Se muestra una ventana donde el usuario debe seleccionar el nombre de la editorial a la que desea generar el reporte. Del mismo se muestran los siguientes datos: - Libros más solicitados en la editorial, mejor valorado. - Usuarios más asiduos a la editorial. - Categorías favoritas. - Autores favoritos. Y la acción: -Exportar PDF -Exportar XLS -Exportar JPG	Inicio/Administración/Reporte/Editorial/Generar reporte

EC 1.3 Cancelar	El usuario selecciona la opción Cancelar	NA	Regresa a la vista principal del panel de Administración.	Inicio/Administración/Reporte/Editorial/Generar reporte/Cancelar
--------------------	--	----	---	--

Tabla 14. DCP Generar reporte de editorial.

Generar reporte de plataforma				
Condiciones de ejecución				
Para consultar un reporte de plataforma hay que:				
- Estar autenticado en el sistema con el rol administrador o editor.				
Escenario	Descripción	Número	Respuesta del sistema	Flujo central
EC 1.1 Generar reporte de plataforma	Selecciona el panel Administración.	NA	Se muestra una ventana con datos generales referentes a: -clientes especiales -autores favoritos -ranking de libros y recursos más visitados -promedio de visitas a libros y recursos. Y la acción: -Exportar PDF	Inicio/Administración/

Tabla 15. DCP Generar reporte de plataforma.

Anexo #6: Pruebas de Integración

Caso de Prueba #2	
Descripción de las variables:	
Los campos Fecha de inicio y Fecha de fin son obligatorios.	
- (*) Fecha de inicio: Permite seleccionar la fecha a partir de la cual se realiza el reporte.	
- (*) Fecha de fin: Permite seleccionar la fecha hasta la cual se realiza el reporte.	
Condiciones de ejecución	
Para consultar un reporte de recursos hay que:	
- Estar autenticado en el sistema con el rol administrador o editor.	
Historia de usuario:	Generar reporte de recursos.
Variables de entrada:	Fecha de inicio, fecha de fin
Tablas implicadas:	xalix_user, tb_resource

Consulta SQL realizada:	<pre> SELECT tb_resource."name" AS tb_resource_name, tb_resource."career" AS tb_resource_career, tb_resource."topic" AS tb_resource_topic, tb_resource."visits" AS tb_resource_visits, tb_resource."updated_at" AS tb_resource_updated_at, xalix_user."name" AS xalix_user_name, xalix_user."email" AS xalix_user_email FROM "public"."xalix_user" xalix_user INNER JOIN "pkt_educational_resource"."tb_resource" tb_resource ON xalix_user."id" = tb_resource."create_by_id" WHERE tb_resource."updated_at" >= \${Fecha1} and tb_resource."updated_at" <= \${Fecha2} ORDER BY tb_resource."visits" DESC </pre>
Datos obtenidos:	Se obtuvieron 4 recursos visitados entre las fechas entradas.
Resultado esperado:	La herramienta SDR crea el reporte y lo envía a la plataforma Félix Varela.
Resultado prueba:	Satisfactorio

Tabla 16. CDP Generar reporte de recursos.

Caso de Prueba #3	
Descripción de las variables:	
<ul style="list-style-type: none"> - Editorial: Permite seleccionar la editorial a la cual se realiza el reporte. Nunca el campo estará vacío pues se brinda una lista desplegable con el listado de editoriales. 	
Condiciones de ejecución	
Para consultar un reporte de editorial hay que: <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol administrador o editor. 	
Historia de usuario:	Generar reporte de editorial.
Variables de entrada:	Editorial

Tablas implicadas:	tb_category , tb_book , tb_editorial
Consulta SQL realizada:	<pre> SELECT tb_category."name" AS tb_category_name, sum(tb_book."visits" + tb_book."user_fav") AS mejores FROM "pkt_editorial"."tb_category" tb_category INNER JOIN "pkt_book"."tb_book" tb_book ON tb_category."id" = tb_book."category_id" INNER JOIN "pkt_editorial"."tb_editorial" tb_editorial ON tb_book."editorial_id" = tb_editorial."id" where tb_editorial."name"=\$P{E} Group by tb_category."name" order by mejores desc limit 5 </pre>
Datos obtenidos:	Se obtuvieron las 5 categorías más visitadas en la editorial.
Resultado esperado:	La herramienta SDR crea el reporte y lo envía a la plataforma Félix Varela.
Resultado prueba:	Satisfactorio

Tabla 17. CDP Generar reporte de editorial.

Caso de Prueba #4	
Condiciones de ejecución	
<p>Para consultar un reporte de plataforma hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol administrador o editor. 	
Historia de usuario:	Generar reporte de plataforma.
Tablas implicadas:	tb_book
Consulta SQL realizada:	<pre> SELECT tb_book."title" AS tb_book_title, tb_book."visits" AS tb_book_visits FROM </pre>

	<pre>"pkt_book"."tb_book" tb_book ORDER BY tb_book."visits" DESC limit 5</pre>
Datos obtenidos:	Se obtuvieron los 5 libros más visitados en la plataforma.
Resultado esperado:	La herramienta SDR crea el reporte y lo envía a la plataforma Félix Varela.
Resultado prueba:	Satisfactorio

Tabla 18. CDP Generar reporte de plataforma.