Facultad de Ciencias y Tecnologías Computacionales

Componente para la obtención de información estadística en forma de reportes en el Sistema de Gestión Académica XAUCE AKADEMOS para el Ministerio de Educación de la República de Cuba

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

Autor: Anás García Cardero
Tutores: Ing. Anaibis Alvarez Morales,
Ing. Raciel Perdomo Gómez
Consultante: Ing. Sandy Nuñez Padrón

La Habana, noviembre de 2022

"Año 64 de la Revolución"

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título ***Componente para la obtención de información estadística en forma de reportes en el Sistema de Gestión Académica XAUCE AKADEMOS para el Ministerio de Educación de la República de Cuba*** concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los 23 días del mes de noviembre del año 2022.

Anás García Cardero

_____
Firma del Autor

Raciel Perdomo Gómez

_____
Firma del Tutor

Anaibis Alvarez Morales

_____
Firma del Tutor

AGRADECIMIENTOS

A mis padres, por no dejar que me saliera del camino por difícil que fuera, a mis abuelos, por ser tantas veces escudo de regaños y fuente interminable de ternura, viejo sé que donde quiera que estés me miras orgulloso ahora, a mi hermana por darme ejemplo de estudio, aunque no lo haya tomado de la mejor manera, a mi esposa, que se ha convertido en el perfecto complemento, que tanto me ha ayudado a crecer a mi familia toda a la cual me debo, a mis amigos los cuales han sido los mejores psicólogos y a veces los peores ejemplos, a todas las casualidades que de una forma u otra fueron necesarias para que un día como hoy pudiera terminar esta investigación.

DEDICATORIA

A mi familia por no dejarme abandonar los estudios,

A mis amigos por incitarme a hacerlo,

A mi esposa por obligarme a ser mejor

A todos, porque que todos fueron necesarios.

RESUMEN

La presente investigación se centra en el desarrollo de un componente que sirva como apoyo al sistema de gestión académica XAUCE AKADEMOS para el Ministerio de Educación de la República de Cuba en lo referente a la obtención de información estadística en forma de reportes, lo cual brindará soporte a toma de decisiones de profesores y directivos de cada institución educativa que utilice el sistema. Previo a la implementación se realizó una búsqueda y análisis de los referentes teóricos asociados a la gestión de información, el análisis estadístico y los reportes en los sistemas de gestión, también se realizó un estudio de herramientas informáticas que generan reportes en la actualidad, propiciando estas infinidades de ventajas para el desarrollo del componente a implementar. Para guiar la investigación y el proceso de desarrollo se empleó la metodología de desarrollo AUP en su versión UCI. Para lograr una perfecta integración con el resto de componentes del sistema se decidió implementar el componente utilizando el framework Django 3.1 ya que es la tecnología utilizada para implementar el resto del back-end del sistema. Como resultado del trabajo se obtuvo un componente que se integra perfectamente con el resto del sistema y que permite generar reportes de manera rápida y sencilla, obteniéndose la información deseada por el usuario.

PALABRAS CLAVE: AKADEMOS, Django, AUP-UCI, Componente, Reportes, Información estadística.

ABSTRACT

This research focuses on the development of a component that serves as support for the academic

management system XAUCE AKADEMOS for the Ministry of Education of the Republic of Cuba in relation to obtaining statistical information in the form of reports, which will provide support to decision-making by teachers and directors of each educational institution that uses the system. Prior to the implementation, a search and analysis of the theoretical references associated with information management, statistical analysis and reports in management systems was carried out, a study of computer tools that currently generate reports was also carried out, promoting these infinities of advantages for the development of the component to be implemented. To guide the research and development process, the AUP development methodology was used in its UCI version. In order to achieve a perfect integration with the rest of the components of the system, it was decided to implement the component using the Django 3.1 framework, since it is the technology used to implement the rest of the back-end of the system. As a result of the work, a component was obtained that integrates perfectly with the rest of the system and that allows generating reports quickly and easily, obtaining the information desired by the user.

KEYWORDS: *AKADEMOS, Django, AUP-UCI, Component, Reports, Statistical information.*

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Con el desarrollo y evolución de la tecnología el hombre se ha visto en la necesidad de crear y aplicar masivamente las Tecnologías de la Información y las Comunicaciones (TIC), con el objetivo de mejorar sus condiciones de vida y de trabajo. Las empresas y organizaciones han visto mejorados los procesos que realizan con la aplicación de los llamados sistemas de gestión, herramientas que permite controlar los efectos económicos y no económicos de una empresa **[CITATION Fra05 \l 21514]**. Estos sistemas tienen como objetivo la optimización del proceso de gestión de información, desde su inicio, hasta su análisis y presentación. Lo cual permite mejorar los procesos de control, planificación y organización de las distintas instituciones. Sin organización y contexto los datos recopilados en los sistemas de gestión, no aportan valor alguno a las organizaciones, para procesar y agrupar estos datos, de forma tal que sean de utilidad, se hace uso de métodos de análisis estadístico.

La educación es uno de estos sectores que se han visto beneficiados con la aplicación actual de las TIC, permitiendo el desarrollo de softwares educativos orientados a facilitar y mejorar los procesos formativos que se imparten en los distintos centros. De igual manera se han creado varios sistemas de gestión con los cuales han sido visto mejorados los diferentes procesos educativos. Entre las actividades que se han visto optimizadas con estos sistemas están: la gestión del plan de estudio, la matrícula, el control de la asistencia y las evaluaciones de los estudiantes, ayudando también así a la toma de decisiones, y a una mejor gestión de la información al personal de cada institución educativa.

En la Universidad de las Ciencias Informáticas (UCI), en el Centro de Tecnologías para la Formación (FORTES) se desarrolla el Sistema de Gestión Académica XAUCE AKADEMOS para el Ministerio de Educación de la República de Cuba (MINED). Este software permite la gestión del proceso académico a partir de varios módulos, los cuales generan información acerca de los procesos de gestión de la información que se realizan en los centros educativos, las direcciones municipales y provinciales de educación y en el propio ministerio. Para obtener información estadística a diferentes niveles, ya sea de calificaciones de los estudiantes o asistencia a clases se deben realizar las siguientes acciones:

1. Acceder a cada una de las vistas que muestran listados relacionados con los datos que se necesiten analizar.
2. Filtrar los listados por varios criterios de búsqueda.
3. Cruzar los datos arrojados por cada una de las vistas consultadas.
4. Realizar operaciones matemáticas.
5. Representar la información generada en un reporte.

El procedimiento antes descrito se realiza de forma manual y resulta muy propenso a errores humanos, a causa de esto se crea también una resistencia al cambio de profesores y directivos. A raíz de los errores en los que se puede incurrir en el proceso, la información puede llegar a verse de manera incorrecta lo cual entorpece su seguimiento, la creación de nuevas estrategias y el análisis de la misma. Este proceso puede verse mejorado en gran medida con la implementación de un mecanismo más sencillo que permita obtener información estadística en forma de reportes en el sistema.

A raíz de la situación anteriormente expuesta se presenta el siguiente **problema de la investigación:** ¿Cómo obtener información estadística en forma de reportes para brindar soporte a la toma de decisiones en el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED?

Por lo que el **objeto de estudio** sería: Análisis estadístico en los sistemas de gestión. Como resultado el **campo de acción** es: Obtención de información estadística en el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED.

Para dar solución al problema existente se ha tomado como **Objetivo general:** Desarrollar un componente para la obtención de información estadística en forma de reportes que brinde soporte a la toma de decisiones a profesores y directivos en el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED.

Para dar cumplimiento al objetivo general definido anteriormente se trazan las siguientes **tareas de investigación**:

1. Determinación de los referentes teóricos al análisis de información estadística en un sistema de gestión académica.
2. Caracterización de los procesos necesarios para la gestión de reportes del sistema XAUCE AKADEMOS para el MINED.
3. Selección de la metodología de desarrollo de software, tecnologías, lenguajes de programación que son necesarios para implementar un componente para obtener información estadística en forma de reportes para sistema XAUCE AKADEMOS para el MINED, de forma tal que contribuya a mejorar la toma de decisiones al personal de las instituciones académicas del MINED.
4. Elaboración en correspondencia a la metodología de desarrollo de software escogida, los artefactos necesarios para el diseño, implementación y validación del componente implementado.
5. Implementación de un componente para obtener información estadística en forma de reportes para el sistema XAUCE AKADEMOS para el MINED de forma tal que contribuya a mejorar la toma de decisiones al personal de las instituciones académicas del MINED.
6. Aplicación de una estrategia de pruebas que permita valorar el nivel de calidad del componente desarrollado.

Posibles resultados:

Se espera obtener un componente para obtener información estadística en forma de reportes que mejore el proceso de gestión de información en el sistema XAUCE AKADEMOS y que brinde soporte a la toma de decisiones de profesores y directivos que utilicen el sistema.

Métodos Teóricos:

Analítico-Sintético: Este método permitió extraer conceptos esenciales relacionados a la obtención de información estadística en los sistemas de gestión, mediante consultas a diferentes fuentes bibliográficas.

Inductivo-Deductivo: A partir del estudio del proceso actual de generación de información estadística en el sistema XAUCE AKADEMOS para el MINED se definió la problemática y se determinaron los elementos particulares que dieron solución a la misma.

Histórico-Lógico: Se utilizó para analizar el surgimiento y la evolución de los sistemas de gestión y la generación de información estadística en ellos.

Métodos empíricos:

Entrevistas: Se realizaron entrevistas a personas dentro del centro FORTES con el fin de obtener conocimiento sobre la situación actual existente y establecer acuerdos.

Encuesta: Se efectuaron para evaluar periódicamente los resultados del sistema en ejecución. También se utilizaron para saber la opinión del personal acerca del componente a desarrollar.

La investigación está compuesta por **tres capítulos**:

- En el **Capítulo 1**, se expone un análisis crítico y valorativo sobre los elementos principales del objeto de estudio y el campo de acción, así como también la selección y justificación de las herramientas y metodología que se va a utilizar.
- En el **Capítulo 2**, se efectúa la descripción de la posible solución y a partir de ahí se realiza el levantamiento de requisitos funcionales y no funcionales, generándose de estos las historias de usuario pasando luego a especificar el diseño y la arquitectura de la solución a implementar.
- En el **Capítulo 3**, se presentan diagramas, y cuestiones referentes a seguridad, tratamiento de errores, excepciones, estrategias de codificación, integración y las pruebas de software que validen la propuesta de solución.

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL ANÁLISIS ESTADÍSTICO EN LOS SISTEMAS DE GESTIÓN

En este capítulo se abarcará todo lo relacionado con los referentes teórico-metodológicos sobre los sistemas de gestión y la generación de reportes en ellos. Se profundizará en la descripción de varios sistemas que gestionan reportes, mencionando algunas de sus principales características y funcionalidades. Enfatizando también algunas definiciones que se estiman convenientes para un mejor entendimiento de la posible propuesta de solución. Estará argumentado también en este capítulo la selección de la metodología, herramientas y tecnologías que se utilizarán en el proceso de desarrollo.

I.1 Conceptos asociados a la obtención de información estadística en los sistemas de gestión

Este epígrafe tiene como objetivo dejar claros conceptos fundamentales para un mejor entendimiento del dominio del problema.

La información

La información es el principal objeto de estudio cuando hablamos de ciencias informáticas. La Real Academia de la Lengua española (RAE) define informática de la siguiente manera “Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores” [CITATION RAE22 \l 21514]. La informática en la actualidad genera en el mundo entero un gran cúmulo de información, la cual puede ser información dudosa o información precisa. Muchas de estas informaciones viajan sobre una red privada o entre la red de redes integrándose unas con otras y formando un dominio que converge hacia nuevas vías de aprovechamiento.

Según Idalberto Chiavenato, información "es un conjunto de datos con un significado, o sea, que reduce la incertidumbre o que aumenta el conocimiento de algo. En verdad, la información es un mensaje con

significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones"[CITATION Chi06 \l 1033].

Según esta definición podemos entender la información como un conjunto de datos, que ordenados y en un contexto específico aumentan el conocimiento sobre algo. A partir de aquí se puede decir que los datos por si solos, sin contexto u organización no brindan ningún valor. Para obtener información utilizable se hace uso de la gestión de información.

La gestión de la información

El concepto de gestión de información, de acuerdo a Woodman, es considerado como el proceso para la obtención de la información adecuada, en la forma correcta, para la persona u organización indicada, al precio adecuado, en el tiempo oportuno y lugar apropiado, para tomar la decisión correcta[CITATION Woo09 \l 21514].

La gestión de la información abarca todos los conceptos genéricos de la gestión, incluyendo la planificación, organización, estructuración, procesamiento, control, evaluación y presentación de informes o reportes de actividades de información, todo lo cual es necesario para satisfacer las necesidades de aquellos que dependen de la información.

Con el avance de las tecnologías de la información y las comunicaciones han surgido los llamados sistemas de gestión con el objetivo de unificar todo el proceso de gestión de información.

Los sistemas de gestión

Un sistema de gestión es una herramienta que permite controlar los efectos económicos y no económicos de la actividad de la empresa. El control en este caso se define como aquella situación en que se dispone de conocimientos ciertos y reales de lo que está pasando en la empresa tanto internamente como en su entorno y permite planificar de cierta manera lo que ocurrirá en un futuro [CITATION Fra05 \l 21514].

Estos sistemas se utilizan para recopilar y manejar datos específicos de una manera organizada para generar a partir de estos, información útil para un negocio u organización. Entre tantos tipos de sistemas de gestión se encuentran los sistemas de gestión académica.

Los sistemas de gestión académica

La gestión académica en una institución educativa es un proceso mediante el cual se controla, organiza y dirigen todas las actividades propias del proceso docente; entre las que se encuentran la matrícula de los estudiantes, la gestión del plan de estudio y el control de la asistencia y evaluaciones de los estudiantes. En esta tarea intervienen directivos, personal de secretaría y profesores [CITATION Dal07 \l 21514].

La implementación de este tipo de sistemas en las instituciones académicas trae consigo una mejora sustancial al proceso de gestión académica en general. Mejorando así las condiciones de trabajo y el proceso de toma de decisiones en las instituciones académicas.

La estadística y el análisis estadístico

La RAE define la palabra estadística como: “Estudio de los datos cuantitativos de la población, de los recursos naturales e industriales, del tráfico o de cualquier otra manifestación de las sociedades humanas”[CITATION RAE22 \l 1033].

Algunos autores tienen definiciones de la estadística semejantes a la anterior, aunque otros difieren un poco. Para José Enrique Chacón, la estadística se define como —la ciencia que tiene por objeto el estudio cuantitativo de los colectivos [CITATION Muñ04 \l 1033].

Harald Cramér se limita a establecer que —el principal objeto de la teoría estadística consiste en la investigación de la posibilidad de obtener inferencias válidas a partir de los datos estadísticos, y en la construcción de métodos para realizar dichas inferencias [CITATION Crá51 \l 21514].

La más aceptada, sin embargo, es la de Óscar Vázquez Mínguez, que define la estadística como —La ciencia que tiene por objeto aplicar las leyes de la cantidad a los hechos sociales para medir su intensidad, deducir las leyes que los rigen y hacer su predicción próxima [CITATION Muñ04 \l 21514].

Una parte fundamental del trabajo estadístico práctico es el análisis de conjuntos de datos. El análisis de datos estadísticos es el proceso que nos permite interpretar los datos de los que disponemos. Partiendo de esta premisa se puede decir que el análisis de datos es una de las partes fundamentales de los sistemas de gestión de la información ya que, sin este, los datos generados en un sistema de gestión no serían de utilidad para ninguna organización o empresa. A partir del análisis de datos se pueden realizar predicciones o mejorar la toma de decisiones y así aumentar el rendimiento de la organización. En muchos sistemas de gestión, académicos o no, se opta por exportar esta información en forma de reporte o informe.

Los reportes

En la actualidad los reportes han tenido un creciente reconocimiento debido a su potencial para mejorar la toma de decisiones en las organizaciones, formando parte de las respuestas a las necesidades de éstas y de un mejoramiento en la transparencia dentro de los negocios u organizaciones.

En la bibliografía se definen el concepto de reportes de manera diferente al de informes, a diferencia de otros criterios que lo enfocan de igual manera.

- La real academia define el concepto de reporte como informe [CITATION RAE22 \l 21514].
- Briones define el concepto de reporte como informe claro y preciso, con la cantidad de detalles suficientes como para que cualquier persona que lo lea por primera vez pueda comprender a cabalidad aquello que se trata a través del proyecto y el estado de avance que este ha alcanzado [CITATION Bri10 \l 21514].
- El Lic. Pablo Legna de la Universidad de Abierta Interamericana en un estudio sobre informes de sostenibilidad, el cual sustenta que los reportes son herramientas de comunicación que transmiten resultados económicos, sociales y ambientales de la empresa a los distintos grupos de interés [CITATION Leg07 \l 21514].

A partir de estas definiciones, un reporte se puede definir como: información resumida y organizada, contenida en una base de datos, con un formato determinado, para mostrarlo a través de un diseño atractivo y fácil de interpretar por los usuarios. Los reportes permiten analizar la información de los proyectos tales como sus metas y propósitos en los negocios para los cuales hayan sido implementados o para los que se pretendan implantar, facilitando así la toma de decisiones en los proyectos dentro de las organizaciones.

La generación de reportes en los sistemas de gestión académica

Los generadores de reportes son herramientas complementarias de los sistemas de información. Por medio de estos se realizan consultas a la base de datos del sistema para obtener información en forma de reporte, además de realizar consultas a las bases de datos a través de un lenguaje transparente al usuario, permiten visualizar la información de manera estructurada y/o resumida en un diseño atractivo y fácil de interpretar, permiten además acoplarse a otras técnicas de análisis [CITATION Enn15 \l 21514].

En un sistema de gestión los reportes cumplen un rol muy importante, permiten a los usuarios del sistema tomar decisiones basadas en datos reales tomados directamente de la información que se tenga en el sistema, reduciendo así la probabilidad de ocurrencia de errores humanos a la hora de recopilar datos.

En el sector de la gestión académica los reportes son igual de valiosos que en cualquier otro tipo de sistema de gestión ya que facilitan la visualización de información estadística generada a partir de los datos recopilados en ellos.

I.2 Herramientas para la generación de reportes

Existen diferentes motores de generación de reportes, para seleccionar el que se utilizará en el presente trabajo se realizó un estudio sobre algunos de ellos. Entre los cuales se encuentran:

Jasper Reports

Jasper Reports es el motor de reportes más usado actualmente en el mundo del Código Abierto (*Open source*), puede ser embebido en todo tipo de aplicaciones, desde las que generan reportes a partir de una plantilla o modelo predeterminado hasta las que brindan más libertad al usuario para diseñar sus propios reportes y ejecutar otras operaciones complejas. Posee además una abundante variedad de formatos de salida e importación, así como una gran comunidad mundial que mantiene y desarrolla la biblioteca. Genera informes para formatos de impresión predeterminados existentes o para reportes continuos. Los reportes pueden ser exportados a formatos como: PDF, XML, HTML, CSV, XLS, RTF, TXT. A través de los sub-reportes es posible manipular y diseñar reportes con un di-

seño altamente complejo. Soporta a la misma vez varios orígenes de datos incluso siendo de tipos distintos. Es posible pasar parámetros desde una aplicación a Jasper Report, esto es muy simple de implementar y brinda una herramienta muy poderosa que permite clasificar, restringir o mejorar los datos que son enviados al usuario basándose en las condiciones de tiempo de ejecución. También pueden seguirse fácilmente los resultados obtenidos y la manera en que la actividad progresa en función de los objetivos fijados. Pueden descubrirse las tendencias y los factores claves que afectan a la actividad y a los resultados de la empresa [CITATION Jas22 \l 21514].

PhpReports

PhpReports es un motor de generación de reportes para PHP, es software libre y provee un alto nivel de productividad en este tipo de aplicaciones. Un reporte en PhpReports está dividido en capas que se contienen unas a otras. De esta forma un reporte sería un documento que contiene un encabezado, un pie y está formado por páginas, las que a su vez cuentan con un encabezado, un pie de página y puede contener grupos anidados en los que se cargarán los campos obtenidos de la consulta a la base de datos [CITATION Php22 \l 21514].

CrystalReports

CrystalReports es un producto de alta tecnología, además de ser una herramienta potente, es fácil de usar para el diseño y generación de reportes a partir de datos almacenados en una base de datos u otra fuente de información. CrystalReports es una solución de informes poderosa, dinámica y procesable que ayuda a diseñar, explorar, visualizar y entregar informes por medio de la web. Permite transformar rápidamente cualquier fuente de datos en contenido interactivo, integrar estrechamente capacidades de diseño, modificación y visualización en aplicaciones .NET, Java o COM, y además permite a los usuarios finales acceder e interactuar con los reportes a través de portales web, dispositivos móviles y documentos de Microsoft Office [CITATION SAP \l 21514].

Active Reports

Active Reports es un componente de informes .NET. El diseñador de informes de Active Reports se integra perfectamente en los entornos de desarrollo Visual Studio .NET. Los informes se crean dentro de Visual Studio .NET y se compilan directamente en el ejecutable. Incluye filtros para la exportación a formatos habituales como Adobe PDF, Microsoft Excel, RTF, HTML, texto y TIFF, tanto en aplicaciones Windows como en aplicaciones basadas en la web [CITATION Gra22 \l 21514].

Luego de estudiar las distintas soluciones para la generación de reportes se decidió utilizar Jasper Reports, ya que es de código abierto, multiplataforma, permite exportar en una mayor cantidad de for-

matos que los restantes motores de reportes, además cuenta con un mayor rango de fuentes de datos soportadas.

I.3 Soluciones homólogas

En este epígrafe se realizó un estudio de algunas soluciones similares desarrolladas a nivel internacional y se exponen las razones por las cuales no cumplen con todos los requerimientos necesarios para dar solución al problema de investigación.

Jasper-rails

Jasper-rails es un conector de Jasper Reports para el *framework* Ruby on Rails que permite una buena integración entre ambas tecnologías, fue desarrollado por el Grupo empresarial brasileño Fortes [CITATION For22 \l 21514] en 2011 [CITATION Gru \l 21514].

Vue-Jasper

Un componente basado en Vue.js, utilizado para consumir informes implementados en un servidor Jasper Reports. Al utilizar la API REST de Jasper, permite al usuario seleccionar informes disponibles en el servidor, seleccionar valores para los controles de entrada del informe, si los hay, y, por último, pero no menos importante, generar y descargar el informe en la máquina del usuario [CITATION Vue22 \l 21514].

Node-Jasper

Node-jasper [CITATION Agm22 \l 21514] es un componente desarrollado en JavaScript que permite integrar NodeJs con un servidor de Jasper Reports, facilitando la interoperabilidad de ambas tecnologías.

Jasper-reports-connector para WaveMaker

Jasper-reports-connector es una extensión de back-end basada en Java para aplicaciones WaveMaker. Los conectores se construyen como módulos de Java y exponen el SDK basado en Java para interactuar con la implementación del conector [CITATION Wav22 \l 21514].

Generador de reportes para la plataforma educativa XAUCE ZERA 2.0

Es un módulo que permite la integración de un motor para generación de reportes dentro la plataforma educativa XAUCE ZERA 2.0, que brinda una solución al proceso de organización y visualización de la información que se genera dentro de la aplicación a través de reportes.

De las soluciones anteriormente analizadas ninguna es compatible con las tecnologías utilizadas en el proyecto AKADEMOS, ya que están implementadas pensando en ciertos frameworks específicos, por lo que no se podrían integrar de manera óptima con el sistema, el cual está desarrollado en Python con el Framework Django. También se tuvo en cuenta que muchas de las soluciones son proyec-

tos o librerías desarrolladas para casos de uso específicos por lo que no cumplen con todos los requisitos necesarios para dar solución al problema planteado.

I.4 Herramientas y tecnologías

En este epígrafe se detallarán las tecnologías que se van a utilizar para el desarrollo de la solución.

Python v.3.8.1

Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo, ejemplos: Instagram, Netflix, Spotify, Panda 3D, entre otros. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma [CITATION Pyt22 \l 21514].

Django v.3.1

Django es un framework web Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático. Sigue la filosofía "Baterías incluidas" lo que significa que viene con muchas funcionalidades genéricas ya implementadas y fáciles de personalizar. Es muy confiable y seguro, permite crear software robusto y escalable en un tiempo relativamente corto. Es gratis y de código abierto [CITATION Dja22 \l 21514].

Django Rest Framework v.3.12

Django REST framework [CITATION Dja221 \l 21514] es un conjunto de herramientas potente y flexible para crear API web que engloba las siguientes características:

- Políticas de autenticación que incluyen paquetes para OAuth1a y OAuth2.
- Serialización que admite fuentes de datos ORM y no ORM.
- Amplia documentación y excelente apoyo de la comunidad.
- Utilizado por empresas reconocidas internacionalmente, incluidas Mozilla, Red Hat y Heroku.

JasperReports Server

TIBCO JasperReports® Server se basa en TIBCO JasperReports® Library como una familia integral de productos de Business Intelligence (BI), que proporciona sólidas capacidades de análisis de datos, servidor de informes y generación de informes estáticos e interactivos. Estas capacidades están disponibles como productos independientes o como parte de un paquete de BI integrado de extremo a extremo que utiliza metadatos comunes. El servidor expone interfaces públicas integrales que permiten una integración perfecta con otras aplicaciones y la capacidad de agregar fácilmente funciones personalizadas [CITATION Jas22 \l 21514].

Ireport

Ireport es el diseñador de reportes nativo de JasperServer, con licencia GPL (Licencia Pública

General) que se distribuye sin costo y por tiempo ilimitado. Permite a los usuarios editar visualmente cualquier tipo de informe complejo con gráficas, imágenes y sub reportes. Con esta herramienta visual de diseño es posible crear nuevos reportes fácilmente usando asistentes de configuración y plantillas, con la opción de poder tener una pre-visualización de la exportación de los reportes en PDF, HTML, XLS, CSV, TXT y XML. Es un diseñador visual de informes poderoso, intuitivo y fácil de usar. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes y sub reportes entre otros [CITATION Jas22 \l 21514].

PostgreSQL v12

PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. PostgreSQL se ha ganado una sólida reputación por su arquitectura comprobada, confiabilidad, integridad de datos, conjunto sólido de funciones, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer soluciones innovadoras y de alto rendimiento. PostgreSQL se ejecuta en todos los principales sistemas operativos y cumple con ACID desde 2001 [CITATION Pos22 \l 21514].

GIT v2.38.1

GIT es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes, con rapidez y eficiencia [CITATION GIT22 \l 21514]. GIT es un sistema de control de versiones distribuido, lo que significa que un clon local del proyecto es un repositorio de control de versiones completo. Estos repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma remota con facilidad. Los desarrolladores confirman su trabajo localmente y, a continuación, sincronizan su copia del repositorio con la copia en el servidor.

Ubuntu Server 18.04

Ubuntu Server es una versión del sistema operativo Ubuntu diseñada como una columna vertebral para Internet. Presenta gran estabilidad y cuenta con un amplio soporte, lo cual lo hace una gran opción como sistema operativo para servidores [CITATION Can22 \l 21514].

PyCharm v2022.1

Como entorno de desarrollo integrado (IDE), se decidió utilizar PyCharm Community en su versión 2022.1. PyCharm proporciona una finalización del código inteligente, inspecciones del código, indicación de errores sobre la marcha y arreglos rápidos, así como refactorización de código automática y completas funcionalidades de navegación [CITATION Jet22 \l 21514].

UML

UML es un lenguaje usado para especificar, visualizar, construir y documentar los componentes de un

sistema orientado a objetos. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, está apoyado en gran manera por el OMG (Object Management Group) [CITATION Lar03 \l 21514].

Además, ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables [CITATION Pon10 \l 21514].

Visual Paradigm Community Edition v17.0

Se utilizará el Visual Paradigm 17.0 como herramienta CASE (*Computer Aided Software Engineering*), esta es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. Ayuda a una construcción de aplicaciones de calidad más rápida y a menor costo. Permite construir todo tipo de diagramas de clases, generar código desde diagramas y generar documentación. Apoya los estándares más altos de las notaciones de Java y de UML. Soporta aplicaciones web, es fácil de instalar y actualizar. Está diseñado para distintos usuarios entre los que se incluyen ingenieros de software, analistas de sistemas, analistas de negocios, arquitectos y desarrolladores. Está orientado a la creación de diseños y se usa el paradigma de programación orientada a objetos [CITATION Vis22 \l 21514].

Visual Paradigm es una poderosa herramienta para visualizar y diseñar elementos de software, para ello utiliza UML (UML 2.1) y ofrece una gama de facilidades para el modelado de aplicaciones. Está orientada a la creación de diseños usando el paradigma de programación orientada a objetos. Provee soporte para la generación de código, tiene integración con diversos IDE como NetBeans (de SunMicrosystems), Developer (de Oracle), Eclipse (de IBM), JBuilder (de Borland), así como la posibilidad de realizarse la ingeniería inversa para aplicaciones realizadas en JAVA, .NET, XML e Hibernate [CITATION Vis22 \l 21514].

Postman

Se trata de una herramienta dirigida a desarrolladores web que permite realizar peticiones HTTP a cualquier API. Postman es muy útil a la hora de programar y hacer pruebas. Además, ofrece la posibilidad de comprobar el correcto funcionamiento de los desarrollos [CITATION Vic19 \l 21514].

Postman no es una herramienta de uso exclusivo para profesionales del entorno web. Es muy útil para todo aquel que interactúa con una API. Permite al desarrollador generar documentación acerca del proceso de consultas a la API para un mejor entendimiento a la hora de compartir el servicio con otros desarrolladores, mostrando información como los parámetros requeridos por las rutas de las consultas y los parámetros de respuesta.

Celery v4.6

Celery es una aplicación que nos permite crear tareas de trabajo asíncronas gestionadas por un gestor de colas que está basado en el envío de mensajes de manera distribuida. Se centra en tareas en tiempo real, pero también soporta calendarización de tareas, es decir, tareas que se ejecutan cada cierto tiempo. Las unidades de ejecución, llamadas tareas, se ejecutan de manera concurrente en uno o más nodos de trabajo. Estas tareas pueden ejecutarse tanto asíncrona como síncronamente, siendo el sistema en general altamente configurable [CITATION PyP22 \l 21514].

Todas las tecnologías antes mencionadas son las que se están utilizando actualmente para la implementación del sistema XAUCE AKADEMOS para el MINED por lo cual se decidió utilizarlas para mantener la coherencia e integración entre el componente y el sistema.

I.5 Metodología de desarrollo de software

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que cumple el objetivo por el cuál fue creado [CITATION Mai15 \l 21514].

Metodología de proceso unificado ágil (AUP)

El Proceso Unificado Ágil de Scott Ambler en inglés *Agile Unified Process* (AUP) es una versión simplificada del Proceso Unificado Racional (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (*test driven development* - TDD), Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad [CITATION Amb22 \l 21514].

Metodología de proceso unificado ágil versión UCI

Al no existir una metodología de software universal en los proyectos desarrollados en la UCI, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así́ que el proceso sea configurable, se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva. Con dicha adaptación se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3 las cuales se centran en el desarrollo de productos y servicios de calidad [CITATION Tam14 \l 21514].

Descripción de las Fases de AUP-UCI

Inicio: Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la 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 [CITATION Tam14 \l 21514].

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 [CITATION Tam14 \l 21514].

Cierre: En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto [CITATION Tam14 \l 21514].

Disciplinas de la Metodología Variación AUP-UCI

Modelado de negocio: Destinada a comprender los procesos de negocio de una organización. Se define como funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para modelar el negocio se proponen las siguientes variantes: Casos de Uso del Negocio (CUN), Descripción de Proceso de Negocio (DPN) y Modelo Conceptual (MC). A partir de las variantes anteriores se condicionan cuatro escenarios para modelar el sistema en la disciplina Requisitos [CITATION Tam14 \l 21514].

Requisitos: El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto. Existen tres formas de encapsular los requisitos: Casos de Uso del Sistema (CUS), historias de usuario (HU) y Descripción de requisitos por proceso (DRP), agrupados en cuatro escenarios condicionados por el Modelado de negocio [CITATION Tam14 \l 21514].

Análisis y diseño: Los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis [CITATION Tam14 \l 21514].

Implementación: En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema [CITATION Tam14 \l 21514].

Pruebas internas: Se verifica el resultado de implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberada. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posibles componentes de prueba ejecutables para automatizar las pruebas [CITATION Tam14 \l 21514].

Pruebas de liberación: Diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación [CITATION Tam14 \l 21514].

Pruebas de Aceptación: Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido [CITATION Tam14 \l 21514].

Descripción de los escenarios

A partir de que el Modelado de negocio propone tres variantes a utilizar en los proyectos (CUN, DPN o MC) y existen tres formas de encapsular los requisitos (CUS, HU, DRP), surgen cuatro escenarios para modelar el sistema en los proyectos, manteniendo en dos de ellos el MC, quedando de la siguiente forma [CITATION Tam14 \l 21514]:

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.

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.

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.

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.

El desarrollo del Componente de generación de reportes para el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED se va a regir por el **escenario 4** de la metodología AUP-UCI, ya que el negocio está muy bien descrito y definido y el cliente siempre va a estar junto al equipo de desarrollo para convenir los detalles de los requisitos a implementar.

Conclusiones del capítulo

En este capítulo se identifican conceptos asociados a los sistemas de gestión de la información y a la generación de reportes en ellos. Se estudiaron soluciones similares a nivel nacional e internacional lo cual permitió tener una mejor comprensión del problema y posibles métodos de solución al mismo. Se analizaron también distintos motores de generación de reportes y se seleccionó el más adecuado para dar solución al problema planteado. Además, se realizó una selección del stack de tecnologías que se utilizarán para implementar el componente y se seleccionó la metodología AUP-UCI para guiar el proceso de desarrollo de la solución a implementar.

CAPÍTULO II: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

En el actual capítulo se describe el procedimiento y los elementos que se tienen en cuenta en el desarrollo del componente para la generación de reportes estadísticos en el sistema XAUCE AKADEMOS para el MINED. Se presentan los conceptos asociados al modelo de dominio y la representación formal del mismo. Se identifican los requisitos no funcionales y funcionales, según los cuales se conforman las historias de usuario. También se realiza una propuesta de diseño mostrada en el diagrama de clases de diseño, incluyendo también la presentación del diagrama de componentes del sistema y el modelado de datos que se utilizará en la propuesta de solución.

II.1 Descripción de la propuesta de solución

Actualmente para obtener información estadística en el sistema XAUCE AKADEMOS se deben realizar las siguientes acciones:

1. Acceder a cada una de las vistas que muestran listados relacionados con los datos que se necesiten analizar.
2. Filtrar los listados por varios criterios de búsqueda.
3. Revisar y transcribir los datos que se desean analizar de forma manual.
4. Cruzar los datos arrojados por cada una de las vistas consultadas.
5. Realizar operaciones matemáticas.
6. Representar la información generada en un reporte

Este proceso resulta complejo, tardado y propenso a errores lo cual llevó al equipo de desarrollo a tomar la decisión de implementar un componente que permitiera obtener la información estadística en forma de reportes. Esto se logrará integrando una herramienta generadora de reportes, en este caso Jasper Reports, con el back-end del sistema. Exponiendo así las funcionalidades necesarias y optimizando en gran medida el proceso de obtención de información estadística.

Para darle solución a la problemática planteada se presenta el componente para la obtención de información estadística en forma de reportes en el sistema XAUCE AKADEMOS el cual mejorará el proceso de toma de decisiones en las instituciones educativas, simplificando el procedimiento hasta ahora utilizado para generar información en forma de reporte estadístico. Esto se logrará implementando un componente que permita una comunicación e integración de los servicios web ofrecidos por el servidor de JasperReports que sean necesitados por el sistema XAUCE AKADEMOS.

Figura 1: Propuesta de solución. Fuente: Elaboración Propia.

El software AKADEMOS está conformado por una aplicación Angular para el front-end y una API escrita en Django para el back-end se propone como solución implementar un componente de la API Django el cual permita integrar funcionalidades requeridas del servidor de JasperReports.

El proceso de gestión de reportes luego de implementar la solución comenzaría diseñando una plantilla de reporte en la herramienta Ireport. Después de creado y guardado el reporte en formato jrxml se procedería a incluir el reporte en el sistema, lo que se haría mediante la funcionalidad Incluir Reporte la cuál recibe como parámetro el nombre del reporte a incluir, el archivo jrxml y la descripción del mismo, mediante esta funcionalidad se añade el reporte a la base de datos de Django y a la base de datos de JasperReports Server. Teniendo el o los reportes ya incluidos en el sistema, cualquier usuario con los permisos pertinentes podrá ver los reportes que hay en el sistema, enviarle los parámetros que desea evaluar y así generar un reporte podrá también modificar el estado de uno o varios reportes a disponible o eliminado según necesite.

II.2 Análisis de Requisitos

El análisis de los requerimientos da como resultado la especificación de las características operativas del software, indica la interfaz de este y otros elementos del sistema, y establece las restricciones que limitan al software [CITATION Pre10 \l 21514].

II.2.1 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, la forma en que debe reaccionar ante ciertas entradas y como se debe comportar en situaciones particulares [CITATION Som07 \l 21514].

Se identificaron los siguientes requisitos funcionales:

- RF 1: Incluir Reporte.
- RF 2: Listar Reportes.
- RF 3: Modificar estado de reporte.
- RF 4: Filtrar reportes.
- RF 5: Buscar reporte.
- RF 6: Sincronizar Reportes.

II.2.2 Requisitos no funcionales

Los requisitos no funcionales constituyen restricciones de los servicios que brinda el sistema o el componente. No se refiere a lo que debe realizar el componente, sino a propiedades físicas. Incluyen aspectos del diseño, del proceso de desarrollo, del rendimiento de la solución informática, entre otras. Comprenden las características del componente que el usuario o cliente puede valorar como fiabilidad, seguridad, portabilidad, diseño de interfaces, capacidad de almacenamiento, entre otras [CITATION Som07 \l 21514].

Rendimiento:

- RNF 1: El sistema debe de tener un tiempo de respuesta a las peticiones menor a 4 segundos.

Compatibilidad:

- RNF 2: Para garantizar una perfecta compatibilidad del componente con el resto del sistema se va a implementar utilizando el framework Django y como gestor de base de datos PostgreSQL.

Seguridad:

- RNF 3: La información solo puede ser modificados por aquellos usuarios autorizados.

II.2.3 Historias de usuario

Una historia de usuario (HU) describe una funcionalidad que, por sí misma, aporta valor al usuario [CITATION Som07 \l 21514]. En el escenario 4 de la metodología de desarrollo de software AUP-UCI las HU constituyen el artefacto utilizado para describir las funcionalidades del sistema.

Las historias de usuario constituyen las bases para las pruebas funcionales ya que se utilizan para verificar si la propuesta desarrollada cumple con lo que especifica en ellas. A continuación, se representan las HU de cada requisito funcional del componente.

Tabla 1: HU del requisito Incluir Reporte.

HISTORIA DE USUARIO	
Número: HU-01	**Usuario:** Profesor o Directivo
Nombre de la historia: Incluir Reportes	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 3	**Iteración asignada:** 1
Programador responsable: Anás García Cardero	
Descripción: El software debe incluir nuevos reportes al sistema.	

Tabla 2: HU del requisito Listar Reportes

HISTORIA DE USUARIO	
Número: HU-02	**Usuario:** Profesor o Directivo
Nombre de la historia: Listar Reportes	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Bajo
Puntos estimados: 2	**Iteración asignada:** 1
Programador responsable: Anás García Cardero	
Descripción: El software debe permitir listar los reportes que contiene en ese momento la base de datos del sistema.	

Tabla 3: HU del requisito Modificar estado del reporte

HISTORIA DE USUARIO	
Número: HU-03	**Usuario:** Profesor o Directivo
Nombre de la historia: Modificar estado del Reporte	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 3	**Iteración asignada:** 2
Programador responsable: Anás García Cardero	
Descripción: El software debe permitir modificar el estado de los reportes que se encuentren en la base de datos del sistema.	

Tabla 4: HU del requisito Filtrar reportes

HISTORIA DE USUARIO	
Número: HU-05	**Usuario:** Profesor o Directivo
Nombre de la historia: Buscar Reporte	
Prioridad en negocio: Media	**Riesgo en desarrollo:** Medio
Puntos estimados: 1	**Iteración asignada:** 2
Programador responsable: Anás García Cardero	
Descripción: El software debe permitir buscar un reporte en la base de datos del sistema dado su nombre o parte de este.	

Tabla 5: HU del requisito Buscar reporte

HISTORIA DE USUARIO	
Número: HU-05	**Usuario:** Profesor o Directivo
Nombre de la historia: Filtrar Reporte	
Prioridad en negocio: Media	**Riesgo en desarrollo:** Medio
Puntos estimados: 1	**Iteración asignada:** 2
Programador responsable: Anás García Cardero	
Descripción: El software debe permitir Filtrar los reportes en la base de datos del sistema dado su nombre de clase(report_class_name) o parte de este.	

Tabla 6: HU del requisito Sincronizar reportes

HISTORIA DE USUARIO	
Número: HU-06	**Usuario:** Profesor o Directivo
Nombre de la historia: Sincronizar Reportes	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Medio
Puntos estimados: 4	**Iteración asignada:** 1
Programador responsable: Anás García Cardero	
Descripción: El software debe permitir sincronizar los reportes que estén en la base de datos de JasperReports Server con la base de datos del componente de Django	

II.3 Diseño de la solución

En esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis.

II.3.1 Diagrama de clases del diseño

Un Diagrama de Clases de Diseño (DCD) representa las especificaciones de las clases e interfaces software (por ejemplo, las interfaces de Java) [CITATION Som07 \l 21514].

En el siguiente DCD se muestra el funcionamiento del requisito Incluir Reporte. La *Client Page* es la parte del front-end de la aplicación que va a enviar una petición a la *Server page* con los datos y el fichero jrxml necesarios para incluir el nuevo reporte. La petición es recibida por la Server Page y esta busca que vista *(view)* es la encargada de gestionar esta petición, en este caso es add_report(), la vista consulta si el reporte ha sido creado en la base de datos del sistema, de no haber sido creada, crea un modelo Report con los datos recibidos en la base de datos del sistema y envía una petición al servidor de JasperReports para crear el reporte en su base de datos también.

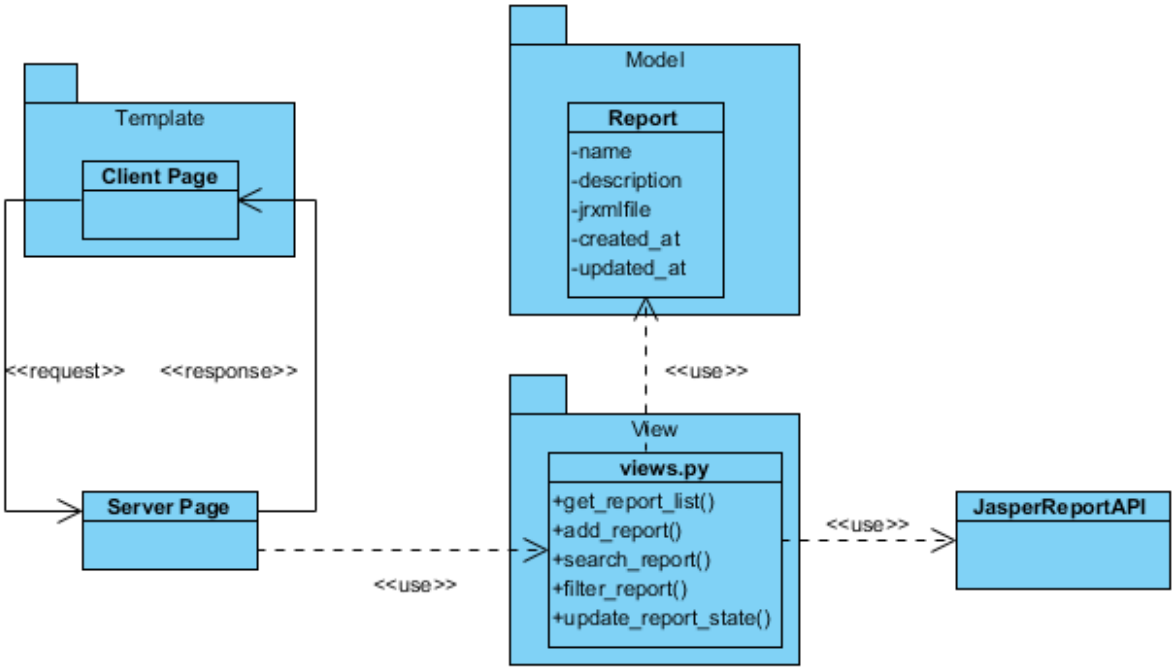

Figura 2: DCD de la HU-01 Incluir Reporte. Fuente: Elaboración propia.

En el caso del siguiente DCD se muestra el funcionamiento del requisito Listar Reportes el cual comienza con la petición recibida desde la Client Page que llega a la Server Page la cual decide cual será la vista (view) que la gestionará en este caso será get_report_list() esta vista comienza haciendo una sincronización con la base de datos de Jasper Reports en caso de que se haya añadido un nuevo reporte y no esté en la base de datos del sistema, esto se hace llamando a la tarea sync_reports(), esta es una tarea programada que se ejecutará automáticamente cada 1 hora o cuando se llame al método desde alguna vista como en este caso. Después de que se sincronicen los reportes de mane-

ra exitosa la vista realizará una consulta a la base de datos del sistema y devolverá una lista de reportes serializada en formato JSON[1].

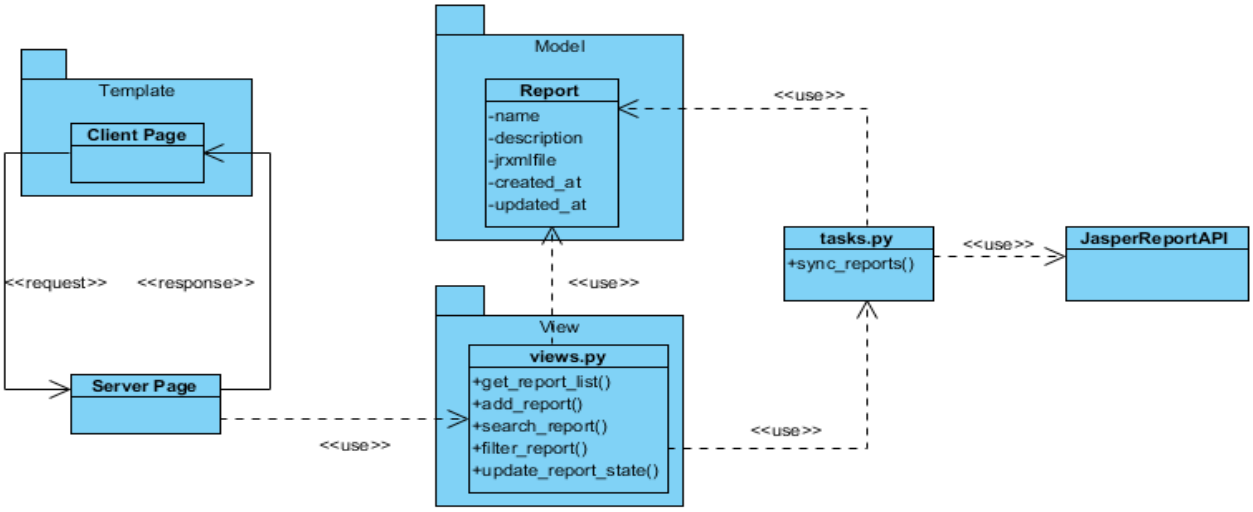

Figura 3: DCD de la HU-02 Listar Reportes. Fuente: Elaboración propia.

En el próximo DCD se presenta el funcionamiento del requisito Buscar Reporte, comienza, como los anteriores con la petición de la Client Page, la cual llega y es gestionada con la Server Page esta decide cual será la vista que gestionará la petición, en este caso será search_report(), esta sincroniza los reportes del servidor de JasperReports para luego comenzar a buscar reportes según el nombre o el fragmento de nombre que recibe de la Client Page para luego devolver una lista o un único resultado serializado en formato JSON.

1 JSON significa *Java Script Object Notation* JSON es un formato de texto para almacenar y transportar datos [CITATION W3S22 \l 21514].

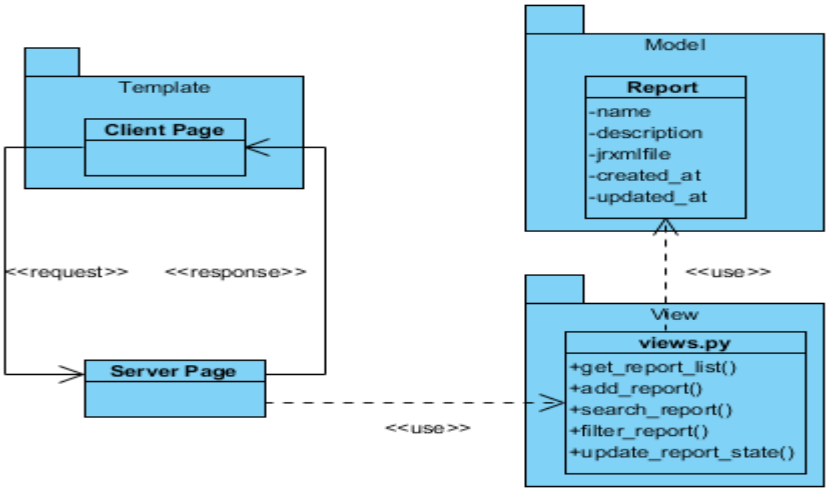

Figura 4: DCD de la HU-05 Buscar Reportes Fuente: Elaboración propia

II.3.2 Diagrama de componentes

El diagrama de componentes ayuda a modelar el aspecto físico de un sistema de software orientado a objetos. Ilustra las arquitecturas de los componentes de software y las dependencias entre ellos. Aquellos componentes de software que incluyen componentes en tiempo de ejecución, componentes ejecutables y los componentes de código fuente.

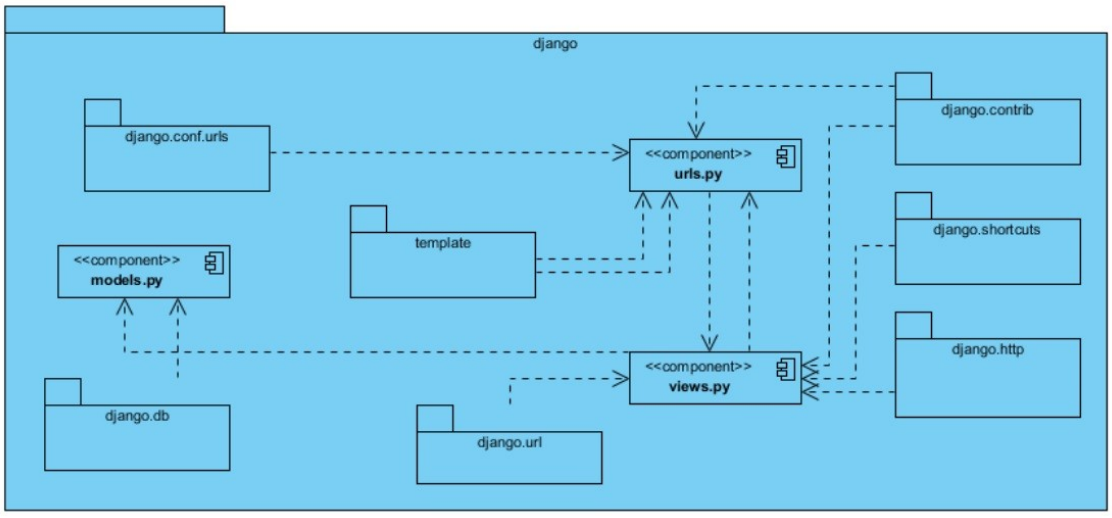

Figura 5: Diagrama de componentes. Fuente: Elaboración propia.

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

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

Django.shorcuts: Este paquete recopila funciones de ayuda y clases que "abarcan" múltiples niveles de MVT.

Django.http: Este paquete recopila una serie de parámetros http.

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

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

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

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

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

II.3.3 Modelado de datos

Otra de las etapas del diseño es la elaboración del modelo de datos, con el propósito de garantizar que los datos persistentes sean almacenados coherente y eficazmente y definir el comportamiento que debe ser implementado en la base de datos[CITATION Rum00 \l 21514].

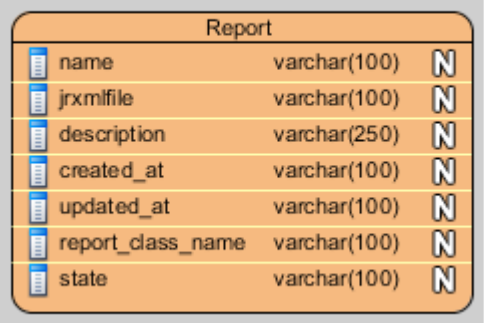

Figura 6: Modelo de datos. Fuente: Elaboración propia.

Tabla 7: Descripción de la entidad Report

TB_Report		
Descripción: en la siguiente tabla se agrupa la información correspondiente a los reportes existentes en la plataforma.		
Name	Varchar (100)	Nombre del reporte

Jrxmlfile	Varchar (100)	Nombre del archivo de reporte
Description	Varchar (250)	Descripción del reporte
Created_at	Varchar (100)	Fecha de creación
Updated_at	Varchar (100)	Fecha de modificación
Report_class_name	Varchar (100)	Nombre de la clase de reporte
State	Varchar (100)	Estado del reporte

II.3.4 Patrones

Un patrón es una colaboración parametrizada junto con las pautas sobre cuando utilizarlo. Un parámetro se puede sustituir por diversos valores, para producir distintas colaboraciones. Los parámetros señalan generalmente las ranuras para las clases. Cuando se instancia un patrón, sus parámetros están ligados a clases reales de un diagrama de clases o a los roles dentro de una colaboración más amplia [CITATION Coh04 \l 21514].

Patrón arquitectónico

Los patrones arquitectónicos son parte esencial en el desarrollo de software. Cada patrón describe un problema que ocurre una y otra vez en nuestro ambiente, y luego describe el núcleo de su solución de forma tal que es posible usarla un millón de veces sin elaborarla dos veces de la misma forma [CITATION Pre10 \l 21514]. Estos patrones definen la estructura general del software, indican las relaciones entre los subsistemas y los componentes del software y definen las reglas para especificar las relaciones entre los elementos (clases, paquetes, componentes, subsistemas) de la arquitectura.

Model-View-Template

Django se suele llamar un framework Modelo Vista Plantilla (MVT por sus siglas en inglés) debido a que está fuertemente influenciado por Modelo Vista Controlador (MVC) y es incluso posible argumentar que la terminología MVC es el único patrón de cambios en Django. Las tres capas básicas son el modelo, la vista, y la plantilla [CITATION Dja22 \l 21514].

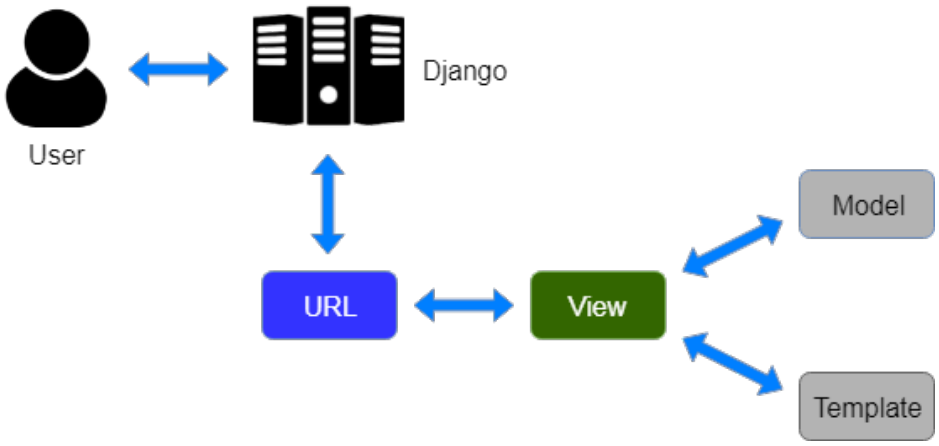

Figura 7: Diagrama de Model-View-Template. Fuente: Sitio web [CITATION Jav22 \l 21514].

A continuación, se explica el funcionamiento del MVT de Django que se ejemplifica en la imagen anterior.

1. El usuario envía una petición a través del navegador web.
2. El motor de Django relaciona la solicitud con una de las urls configuradas en urls.py.
3. La url llama a la vista que le corresponde.
4. La vista interactúa con el modelo para obtener datos.
5. La vista llama a la plantilla.
6. La plantilla se renderiza en respuesta a la solicitud del navegador.

El modelo: El modelo define los datos almacenados, se encuentra en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros y posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos [CITATION Dja22 \l 21514].

La vista: La vista se presenta en forma de funciones en Python, su propósito es determinar qué datos serán visualizados, entre otras cosas más. El ORM de Django permite escribir código Python en lugar de SQL para hacer las consultas que necesita la vista. También se encarga de tareas conocidas como el envío de correo electrónico, la autenticación con servicios externos y, la validación de datos a través de formularios [CITATION Dja22 \l 21514].

La plantilla: Es la encargada de recibir los datos de la vista y luego los organiza para la presentación al navegador web. La plantilla es básicamente una página HTML con algunas etiquetas extras propias de Django, en sí no solamente crea contenido en HTML también XML, CSS, JavaScript, CSV, etc. [CITATION Dja22 \l 21514].

Patrones de diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos [CITATION Cos16 \l 21514].

Estos patrones hacen más fácil la reutilización del código y los diseños. Facilitan también el aprendizaje al programador inexperto facilitando la decisión entre herencia y composición. Para el diseño del componente fueron analizados patrones de gran utilidad que proponen una forma de reutilizar la experiencia de los desarrolladores clasificando y describiendo formas de solucionar problemas frecuentes en la etapa del desarrollo.

Patrones GRASP

General Responsibility Asignment (GRASP por su acrónimo en inglés) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos. Favorecen la reutilización de código y ayudan a construir software basado en la reutilización [CITATION Jac00 \l 21514]. Los utilizados en la solución son los que a continuación se muestran.

Bajo Acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Las URLconfs de Django son un claro ejemplo de este principio en la práctica. En una aplicación de Django, la definición de la URL y la función de vista que se llamará están débilmente acopladas.

```
urlpatterns = [
    path('get_report_modal_data/<str:report_class_name>', get_report_modal_data),
    path('get_report/<str:report_class_name>', get_report),
    path('get_report_list', get_report_list),
    path('filter/<str:report_class_name>', filter_reports),
    path('search/<str:label>', search_reports),
    path('add', add_report),
    path('update', update_state),
]
```

Figura 8: Captura de las url del proyecto. Fuente: Elaboración propia.

Experto: Este patrón se basa en la asignación de responsabilidades, las cuales se asignan a las clases que posean la información necesaria para llevarlas a cabo. Es utilizado en la capa de abstracción del modelo de datos. Con el uso del *Object Relational Mapping* (ORM) de Django, se crean las clases que representan las entidades del modelo de datos. Asociado a cada una de estas clases, son generadas un conjunto de funcionalidades que las relacionan de forma directa con la entidad que repre-

sentan. Las clases contienen toda la información necesaria de la tabla que representan en la base de datos.

```
class Report(models.Model):
    label = models.CharField(max_length=100)
    report_class_name = models.CharField(max_length=100, null=True) #Para instanciar la clase
    state = models.CharField(blank=True, choices=StateType.choices, max_length=100)
    description = models.TextField()
    created_at = models.CharField(max_length=20, null=True, blank=True)
    updated_at = models.CharField(max_length=20, null=True, blank=True)
    jrxmlfile = models.CharField(max_length=100, null=True, blank=True)

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

Figura 9: Captura de la implementación parcial del modelo Report. Fuente: Elaboración propia.

Patrones GOF

Describen 23 patrones de diseño comúnmente utilizados y de gran aplicabilidad en problemas de diseño usando modelado UML. Se clasifican en tres categorías basadas en su propósito: creacionales, estructurales y de comportamiento [CITATION Lar03 \l 21514].

Decorador: Es un patrón estructural que extiende la funcionalidad de un objeto dinámicamente, de tal modo que es transparente a sus clientes, utilizando una instancia de una subclase de la clase original que delega las operaciones al objeto original. Provee una alternativa muy flexible para agregar funcionalidad a una clase [CITATION Lar03 \l 21514]. Este patrón Django lo incluye por defecto, ejemplo de ello lo constituye la función: *@login_required.*

```
@login_required
def get_report(request, report_class_name):
    report_params = json.loads(request.body)
    report_object = get_class_instance(report_class_name)
    return report_object.render_to_response(report_params)
```

Figura 10: Utilización del patrón decorador en la vista get_report(). Fuente: Elaboración propia.

Conclusiones del capítulo

En este capítulo se presentaron los artefactos generados en la fase de análisis y diseño en la metodología AUP versión UCI tales como el modelado de datos, las historias de usuario y el diagrama de clases de diseño. También se realizó el levantamiento de requisitos de software, se definió como patrón arquitectónico regente *Model-View-Template* y se explicó cómo se pusieron de manifiesto patrones GRASP y GOF en la implementación de la solución.

CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En este capítulo se definen los estándares de codificación utilizados para implementar la solución al problema de investigación. Se traza una estrategia de pruebas donde se validarán los resultados obtenidos con la implementación y se comprobarán y resolverán las no conformidades del cliente con respecto a la solución.

III.1 Implementación

En la etapa de implementación a partir de los resultados obtenidos en la fase de análisis y diseño se pasa a construir el sistema.

Estándar de codificación:

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Una faceta del desarrollo de software en la que todos los programadores influyen especialmente es en la técnica de codificación. “El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación” [CITATION Doc \l 21514]. Para el desarrollo del componente de generación de reportes se utilizó el estándar de codificación PEP 8 de Python. El mismo se puede consultar en el sitio web: Python Enhancement Proposals [CITATION PEP \l 21514].

```
def filter_reports(request, report_class_name):
    sync_reports()
    reports = JasperReportSync.objects.filter(
        report_class_name__startswith=report_class_name)
    data = serializers.serialize(
        'json', reports, fields=("label", "report_class_name"))
    return JsonResponse(json.loads(data), safe=False)


def search_reports(request, label):
    sync_reports()
    reports = JasperReportSync.objects.filter(label__startswith=label)
    data = serializers.serialize(
        'json', reports, fields=("label", "report_class_name"))
    return JsonResponse(json.loads(data), safe=False)
```

Figura 11: Fragmento de código con estándares de codificación aplicados. Fuente: Elaboración propia

En el fragmento de código anterior se pueden evidenciar algunas de las reglas definidas en el estándar PEP 8 como:

- Uso de 4 espacios de identación por línea de código.
- Separar las definiciones de funciones con dos líneas en blanco.
- No pasar de los 79 caracteres por línea de código.

III.2 Pruebas

La fase de pruebas tiene como objetivo verificar el sistema de software para comprobar si este cumple con sus requisitos. Dentro de esta fase se desarrollan varios tipos de prueba en función de los objetivos de las mismas. Algunos tipos son pruebas de caja blanca, caja negra, rendimiento, seguridad entre otras.

Estrategia de pruebas

Una estrategia de prueba de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados, además, debe ser suficientemente flexible para promover un uso personalizado de la prueba. Al mismo tiempo, debe ser suficientemente rígida para alentar la planificación razonable y el seguimiento de la gestión conforme avanza el proyecto [CITATION Pre10 \l 21514].

Pruebas nivel de Componente

Nivel de Componente: Es el primer nivel de pruebas, diseñadas e implementadas por el equipo de desarrollo, donde el programador es quien revisa su código fuente.

Método de prueba

Caja Blanca: Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que: garanticen que se ejercite por lo menos una vez, todos los caminos independientes de cada módulo; ejerciten todas las decisiones lógicas en sus vertientes verdaderas y/o falsas; ejecuten todos los bucles en sus límites y con sus límites operacionales, y que se ejerciten las estructuras internas de datos para asegurar su validez [CITATION Pre10 \l 21514]. Estas pruebas se realizan al código fuente para asegurar que la operación interna se ajuste a las especificaciones.

Técnica de prueba de caja blanca: Pruebas automatizadas: Pruebas unitarias, para comprobar el correcto funcionamiento de la implementación del componente desarrollado, se realizaron pruebas unitarias. Se utilizó la biblioteca de Python (*unit testing*) y la definición de las pruebas se realizaron dentro del archivo tests.py, creado por el marco de trabajo Django. Para ejecutar los test o iniciar el servidor de prueba se hace uso del comando Python manage.py test.

```
test_filter_report (jasper_connector.tests.JasperReportSyncTestCase) ... ok
test_get_report (jasper_connector.tests.JasperReportSyncTestCase) ... ok
test_get_report_list (jasper_connector.tests.JasperReportSyncTestCase) ... ok
test_get_report_modal_data (jasper_connector.tests.JasperReportSyncTestCase) ... ok
test_include_report (jasper_connector.tests.JasperReportSyncTestCase) ... ok
test_modify_report (jasper_connector.tests.JasperReportSyncTestCase) ... ok
test_search_report (jasper_connector.tests.JasperReportSyncTestCase) ... ok

----------------------------------------------------------------------
Ran 7 tests in 0.007s
```

Figura 12: Captura de la salida del comando manage.py test jasper_connector.tests. Fuente: Elaboración propia.

Como se puede apreciar en la captura las pruebas unitarias transcurrieron sin fallos, probando así el correcto funcionamiento del código implementado.

Prueba a nivel de Sistema

Prueba de Sistema: Son las pruebas que se hacen cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados.

Método de prueba:

Caja negra: Las pruebas de caja negra también conocidos como pruebas funcionales o pruebas de entrada y salida, son las que se ejecutan sobre la interfaz del software. Mediante el uso de estas se examinan todas las funcionalidades. Estas tienen poca relación con el comportamiento interno del software [CITATION Pre10 \l 21514]. Estas pruebas, permiten demostrar que las funciones del sistema, sean operativas; que la entrada se acepta de forma adecuada y que se produce una salida correcta.

Técnica de Prueba de caja negra:

Partición de equivalencia: La partición de equivalencia es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Un caso de prueba ideal descubre de primera mano una clase de errores que de otro modo podrían requerir la ejecución de muchos casos de prueba antes de observar el error general [CITATION Pre10 \l 21514].

El diseño de casos de prueba para la partición de equivalencia se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición booleana [CITATION Pre10 \l 21514].

Tabla 8: Variables empleadas en el caso de prueba basado en la HU Buscar reporte.

No.	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Nombre del reporte	Campo de texto	No	Se inserta una cadena de texto puede contener valores alfanuméricos no puede contener espacios.

Tabla 9: Diseño de caso de prueba correspondiente a la HU Buscar reporte

Escenario	Descripción	V1	Respuesta del sistema	Flujo Central
E1: Introducir el valor correctamente.	El usuario introduce el nombre del reporte que desea buscar de forma correcta.	Reporte1	El sistema devuelve el o los reportes cuyos nombres coinciden total o parcialmente con el criterio de búsqueda.	El usuario rellena el campo de búsqueda correctamente y luego da clic en el botón buscar, lo cual envía una petición al endpoint correspondiente en el componente de reportes.
E2: Introducir el valor incorrectamente (cadena de caracteres con espacio).	El usuario introduce el nombre del reporte que desea buscar de forma incorrecta.	Reporte 1	El sistema devuelve una respuesta de error para que la aplicación del cliente muestre el mensaje de error (El nombre del reporte a buscar no puede contener espacios).	El usuario rellena el campo de búsqueda incorrectamente y luego da clic en el botón buscar, lo cual envía una petición al endpoint correspondiente en el componente de reportes.
E3: Dejar el	El usuario hace	*null*	El sistema de-	El usuario no re-

campo vacío.	clic en el botón buscar con el campo vacío.		vuelve una respuesta de error para que la aplicación del cliente muestre el mensaje de error (El nombre no puede estar vacío).	llena el campo de búsqueda y luego da clic en el botón buscar.

Prueba de seguridad: La prueba de seguridad está enfocada al ámbito de la aplicación, asegurando que los usuarios solo accedan cuando estén registrados en el sistema.

Resultados:

Para la realización de las pruebas de seguridad al componente, primeramente, se identificaron los distintos tipos usuarios que están definidos en el sistema y sus respectivos niveles de acceso o privilegios definidos; luego se creó un usuario de cada tipo y se procedió con las pruebas.

Las primeras pruebas fueron intentos de violación de la seguridad del sistema, tratando de acceder a este con usuarios no existentes o utilizando usuarios reales con contraseñas erróneas.

Estas pruebas de acceso arrojaron resultados favorables al sistema, ya que no se logró acceder con usuarios no registrados previamente ni con falsas contraseñas, demostrando que el componente cuenta con un mecanismo de autenticación seguro, garantizando que solamente trabajen con el programa personas autorizadas previamente.

Se accede al sistema con un usuario específico y se intenta modificar información a la cual no debería de tener acceso ese determinado usuario, como son: Incluir reporte, Modificar estado de reporte.

Este mismo procedimiento se ejecutó con los tres tipos de usuarios con que cuenta la aplicación y los resultados de las pruebas fueron las siguientes. Los usuarios registrados en el sistema tienen acceso solamente a las funcionalidades predefinidas por los administradores del sistema, impidiendo en todo momento que se lleven a cabo funcionalidades por personas no autorizadas.

Rendimiento

Se aplicaron pruebas de carga y rendimiento con el software Apache JMeter se realizaron las pruebas con 100 usuarios concurrentes el sistema tuvo un tiempo de respuesta entre 600ms y 2s sin presentar caídas con un valor medio de 970ms y una desviación de 465ms.

Las prestaciones de hardware que posee la PC cliente y el servidor de la aplicación donde se realizaron las pruebas se muestran a continuación:

PC cliente:

- Ordenador Intel Core 2 Duo, con 2.0 GHz de velocidad de microprocesador.
- Memoria RAM de 4 GB.

PC servidor:

- Ordenador Intel Core i7, con 3.4 GHz de velocidad de microprocesador.
- Memoria RAM de 8 GB.

Muestra #	Tiempo de comie...	Nombre del hilo	Etiqueta	Tiempo de Muest...	Estado
1	07:04:53.149	Usuarios 1-1	getReportList	177	
2	07:04:53.249	Usuarios 1-2	getReportList	252	
3	07:04:53.349	Usuarios 1-3	getReportList	553	
4	07:04:53.449	Usuarios 1-4	getReportList	756	
5	07:04:53.550	Usuarios 1-5	getReportList	1163	
6	07:04:53.650	Usuarios 1-6	getReportList	1267	

Figura 13: Captura a Apache JMeter luego de correr las pruebas de rendimiento. Fuente: Elaboración propia.

No conformidades detectadas

El siguiente gráfico muestra las no conformidades detectadas durante todo el proceso de pruebas descrito anteriormente.

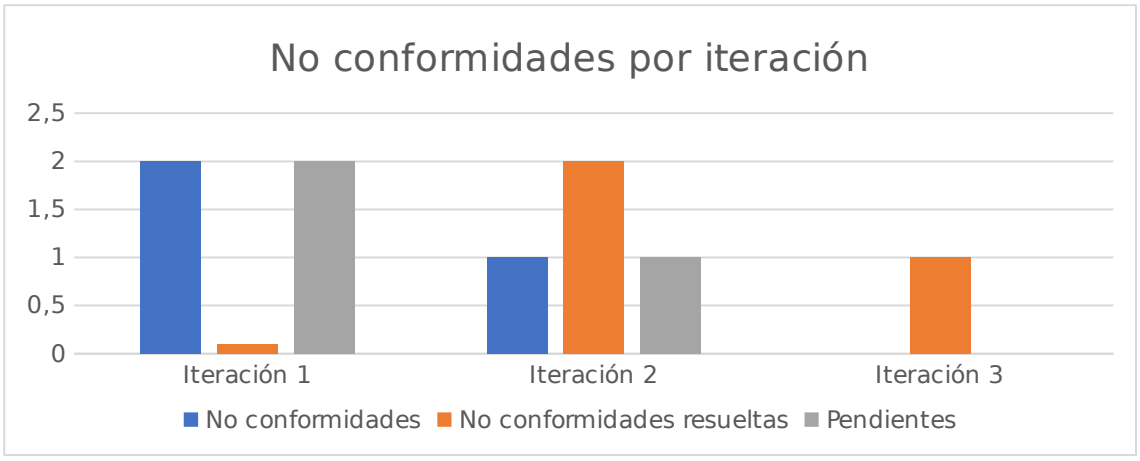

Figura 14: Gráfico de no conformidades. Fuente: Elaboración propia.

En el gráfico anterior se muestra la ejecución de 3 iteraciones de pruebas realizadas al sistema. Donde en cada una de estas se solucionan las no conformidades identificadas. Para así obtener un producto de calidad libre de errores y satisfacer las necesidades del cliente. En la primera iteración se

detectaron 2 no conformidades que fueron resueltas en la segunda iteración, en esta a su vez se detectó una no conformidad la cual fue resuelta en la tercera y última iteración.

Pruebas a nivel de aceptación

La prueba de aceptación de sistema por parte del personal de operaciones o de la administración del sistema se realiza, por lo general, en un entorno de producción (simulado) [CITATION IST18 \l 21514].

Las pruebas de aceptación se realizaron con el jefe y el analista principal del proyecto XAUCE AKADEMOS para el MINED, especialistas ambos del centro FORTES. Se probaron todas las funcionalidades del componente con la herramienta Postman, y se verificó que todas cumplían con los requisitos definidos. Se emitió un acta de aceptación oficial donde se deja constancia de que todos los requisitos del componente fueron implementados y probados correctamente y está 100% funcional.

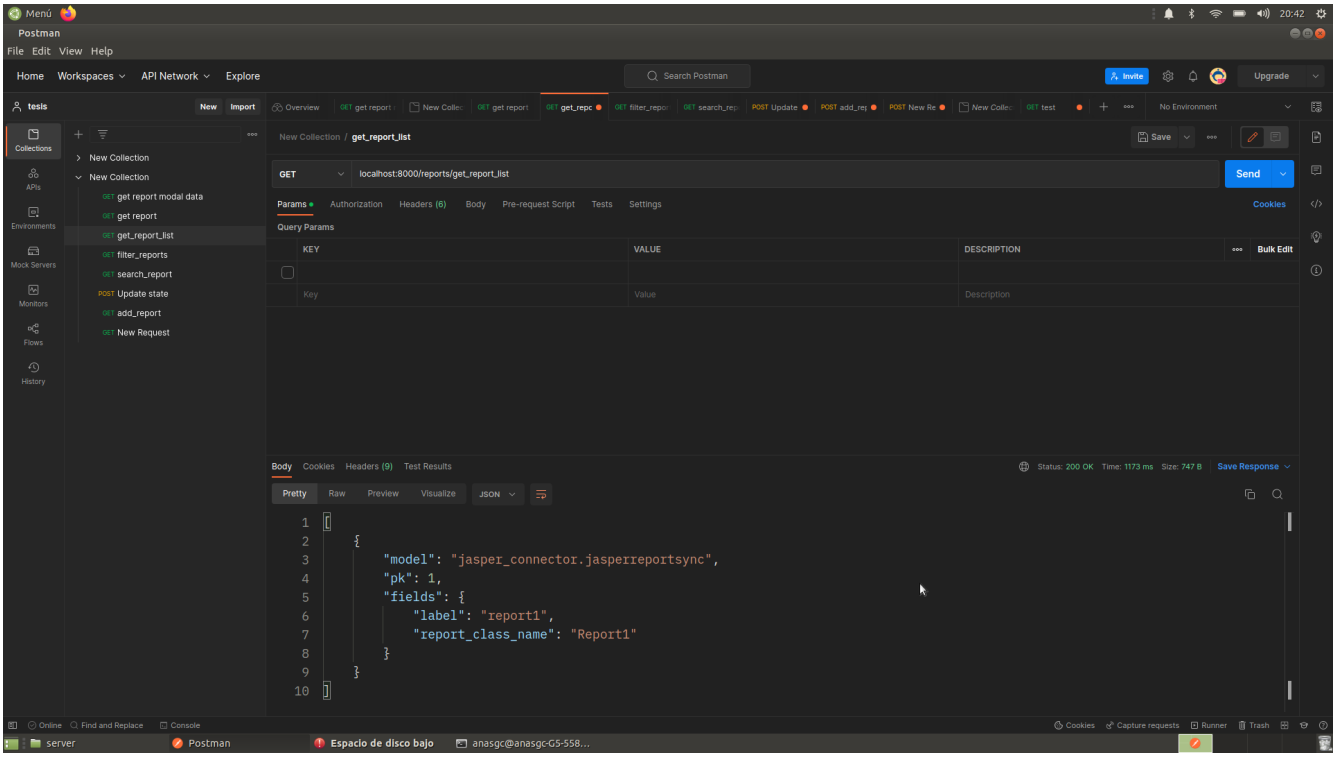

Figura 15: Captura de pantalla del Postman probando el requisito Listar reportes. Fuente: Elaboración propia.

En la imagen anterior se muestra una captura de pantalla del software Postman con la colección de peticiones creada para probar los requisitos funcionales del componte y la respuesta recibida luego de enviar la petición a la dirección del requisito Listar reportes.

Conclusiones del capítulo

En este capítulo se definieron los estándares de codificación que se usaron a la hora de programar la solución. Se trazó una estrategia de pruebas y se diseñaron distintos casos de prueba para poner en ejecución la misma, para de esa manera validar la correcta implementación de los requisitos y la conformidad del cliente.

CONCLUSIONES FINALES

Luego de la realización de esta investigación se concluye que:

- A partir de los objetivos específicos se logró implementar el componente para la obtención de información estadística en forma de reportes en el sistema XAUCE AKADEMOS MINED, para así ayudar a mejorar el proceso de toma de decisiones de profesores y directivos.
- La fundamentación de la metodología y herramientas empleadas para el desarrollo permitió identificar un ambiente de desarrollo de acuerdo a las necesidades del proyecto.
- El análisis y diseño dejaron como resultado los artefactos generados para la implementación.
- La implementación de las funcionalidades visibilizó el cumplimiento de los requisitos funcionales definidos para la solución.

RECOMENDACIONES

Los objetivos de esta investigación fueron logrados satisfactoriamente, sin embargo, se recomienda tener en cuenta la siguiente recomendación:

- Refactorizar el código para volver el componente reutilizable y así poderlo integrar en próximos proyectos que lo necesiten.

REFERENCIAS BIBLIOGRÁFICAS

Agmoyano. 2022. JasperReports from Node.js. [En línea] 2022. https://github.com/agmoyano/node-jasper.

Ambler, Scott. 2022. Ambysoft. [En línea] 2022. http://www.ambysoft.com/unifiedprocess/agile-UP.html.

Briones, Jose. 2010. Que es un reporte y como se hace. [En línea] 2010. http://pepishighs.blogspot.-com/2009/09/que-es-un-reporte-y-como-se-hace.html.

Cannonical. 2022. Ubuntu. [En línea] 2022. https://ubuntu.com/.

Chiavenato, Idalberto. 2006. *Introducción a la Teoría General de la Administración Séptima Edición.* s.l. : McGraw-Hill Interamericana, 2006.

Cohn, Mike. 2004. *User Stories Applied.* s.l. : Addison Wesley, 2004. 0-321-20568-5.

Community, Jaspersoft. 2022. JasperReports® Library | JaspersoftCommunity. [En línea] 2022. https://community.jaspersoft.com/project/jasperreports-library .

Crámer, Harald. 1951. *Mathematical Methods of Statistics.* 1951.

Cuervo, Victor. 2019. AiT. ArquitectoIt. [En línea] 2019. http://www.arquitectoit.com/postman/que-.

Dalmendray, Frank Benavides, Rojas, DarienCepero. 2007. *Estudio de Alternativas para la Migración del Sistema automatizado para la Gestión académica AKADEMOS a Software Libre, .* 2007.

Django Project. The web framework for perfectionists with deadlines. [En línea] https://www.django-project.com/ .

DjangoRest. 2022. Django Rest Framework. [En línea] 2022. https://www.django-rest-framework.org/.

Docs, G. *Recuperado el Estándar de Codificación.*

Ennis Bouly, Yanet Pérez Solá, Nara Lidia Abreu Medina, Aldis Joan. 2015. *Componentes para la Generación de Reportes Dinámicos en el GDR sobre Bases de Datos en Access y Oracle 11.* 2015.

Fortes. 2022. Grupo Fortes. [En línea] 2022. http://www.grupofortes.com.br/.

Fortes, Grupo. 2022. GitHub - fortesinformatica/jasper-rails: JasperReports on Rails. [En línea] 2022. https://github.com/fortesinformatica/jasper-rails.

GIT. 2022. Git --everything-is-local. [En línea] 2022. https://git-scm.com/.

GrapeCity. 2022. NET and JS Reporting Solutions | Design Custom Reports in Code | ActiveReports. [En línea] 2022. https://www.grapecity.com/activereports.

ISTQB. 2018. *Probador Certificado del ISTQB®.* 2018.

Jacobson, Ivar, Booch, Grady y James, Rumbaugh. 2000. *84-7829-036-2.* s.l. : Pearson Education, 2000. 84-7829-036-2..

JavaTPoint. 2022. JavaTPoint. [En línea] 2022. https://www.javatpoint.com/django-mvt.

JetBrains. 2022. JetBrains Herramientas para desarrolladores de software y equipos. [En línea] 2022. https://www.jetbrains.com/es-es/pycharm/features/.

Larman, Craig. 2003. *Modelo de dominio. UML y Patrones.* s.l. : Prentice Hall, 2003.

Legna, Pablo. 2007. *Reportes de sostenibilidad. Parte 1: concepto, beneficios y contenido.* 2007.

Maida, EG, Pacienzia. 2015. *J. Metodologías de desarrollo de software Tesis de Licenciatura en Sistemas y Computación. Facultad de Química e Ingeniería "Fray Rogelio Bacon".* s.l. : Universidad Católica Argentina, 2015.

Ministerio de Educación de la República de Cuba. 2022. Mined Cuba. [En línea] 2022. https://www.mined.gob.cu/derecho-a-la-educacion/.

Muñoz, David Ruiz. 2004. *Manual de Estadística.* 2004. 84-688-6153-7.

Patrones de Diseño. . **Cosmin, Puiu Ionut. 2016.** 2016, Vol. MoleQla: revista de Ciencias de la Universidad Pablo de Olavide.

PhpReports. 2022. PhpReports. [En línea] 2022. https://jdorn.github.io/php-reports/.

Pons, C, Roxana Giadini Gabriela PÉREZ. 2010. *Desarrollo de software dirigido por modelos. Conceptos teóricos y su aplicación práctica.* La Plata, Argentina : Universidad Nacional de la Plata, 2010. 978-950-34-0630-4..

PostgreSQL. 2022. PostgreSQL: The World's Most Advanced Open Source Relational Database. [En línea] 2022. https://www.postgresql.org/.

Pressman, Roger. 2010. *Ingeniería de Software, un enfoque práctico.* s.l. : McGraw-Hill, 2010. 978-607-15-0314-5.

Proposal, Python Enhancements. PEP 8 – Style Guide for Python Code. [En línea] https://peps.python.org/pep-0008/.

PyPi. 2022. PyPi. [En línea] 2022. https://pypi.org/project/celery/.

Python.org. 2022. Python. [En línea] 2022. https://www.python.org/.

RAE. 2022. Diccionario de la Lengua Española. [En línea] 2022. https://dle.rae.es/inform%C3%A1tico.

Rumbaugh J., Jacobson I., Booch G. 2000. *El lenguaje Unificado de Modelado. Manual de Referencia.* Madrid : Pearson Educación, 2000. 84-7829-037-0.

Sánchez, Tamara Rodríguez. 2014. *Metodología de desarrollo para la.* 2014.

Segura, Francisco Ogalla. 2005. *Sistemas de gestión una guía práctica.* España : Ediciones Díaz de Santos, 2005. 84-7978-695-7.

Solutions, SAP Crystal. 2022. SAP Crystal Solutions | Business Intelligence tool. [En línea] 2022. https://www.sap.com/products/technology-platform/crystal-bi.html.

Somerville, I. 2007. *Software Engineering.* s.l. : Pearson Education, 2007.

VisualParadigm. 2022. Visual Paradigm. [En línea] 2022. https://www.visual-paradigm.com/.

Vue-Jasper. 2022. A Vue.js componentto consume Jasper server reports». . [En línea] 2022. https://github.com/el95149/vue-jasper.

W3Schools. JSON Introduction. [En línea] [Citado el: noviembre 19, 2022.] https://www.w3schools.-com/js/js_json_intro.asp.

Wavemaker. 2022. GitHub - wavemaker/jasper-reports-connector. [En línea] 2022. https://github.com/wavemaker/jasper-reports-connector.

Woodman, Lynda. 2009. Information management in large organisations. [aut. libro] Suzanne, Matarazzo, James Connolly. *Knowledge and Special Libraries.* s.l. : Routledge, 2009.

ANEXOS

Anexo 1

Entrevista al líder del proyecto de XAUCE AKADEMOS en el centro FORTES:

Estimado especialista: Se necesita de su cooperación para la investigación de un trabajo de diploma. Por ello, sería de gran ayuda que respondiera lo siguiente:

1. ¿Existe en la actualidad algún mecanismo para generar reportes en el sistema XAUCE AKADEMOS para el MINED?

En caso positivo ¿De qué manera lo hace?

2. ¿Por qué usted considera necesario que se implemente un componente para obtener información estadística en forma de reportes en el sistema XAUCE AKADEMOS?

3. ¿Qué requisitos le gustaría a usted que brindara el software a desarrollar?

4. ¿Qué otras características consideran que deba presentar el sistema, en cuanto a la usabilidad, seguridad u otro aspecto que garantice su calidad?

Muchas gracias por su colaboración.

Anexo 2

Encuesta inicial de la investigación

Especialista, le invito a responder el siguiente cuestionario con el fin de conseguir su colaboración en la presente investigación.

Marque con una X en una sola opción. Por el tiempo brindado, muchas gracias.

1. ¿Considera importante la implementación de un componente que permita generar reportes en el sistema XAUCE AKADEMOS para el MINED?

Sí ___ No ___ No sé___

2. ¿Considera usted correcta la forma en que se generan reportes actualmente en el sistema XAUCE AKADEMOS?

Sí ___ No ___ No sé___

3. ¿Considera usted factible la implementación de un componente para generar reportes en el sistema XAUCE AKADEMOS?

Sí ___ No ___ No sé___