



FACULTAD DE CIENCIAS Y TECNOLOGÍAS COMPUTACIONALES

Título: Componente para generar Estadísticas del Personal en el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED.

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

Autor: Neyis Nieves Barrios Vega

Tutor (a): Msc. Yaneisis Pérez Heredia

La Habana, 21 de noviembre de 2022
Año 64 de la Revolución

No existe la suerte. Solo
hay preparación
adecuada o inadecuada
para hacer frente a una
estadística.

Robert A. Heinlein



DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título “**Componente para generar Estadísticas del Personal en el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED**”, 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 28 días del mes de noviembre del año 2022.

Neyis Nieves Barrios Vega

Firma del Autor

Msc. Yaneisis Pérez Heredia

Firma del Tutor

DATOS DE CONTACTO

Nombre y apellidos: Yaneisis Pérez Heredia

Títulos académicos: Máster.

Formación de postgrado recibida: Informática Aplicada.

Lugar de trabajo: UCI

Responsabilidades laborales asumidas: Profesor.

Experiencia profesional: 18 años.

Líneas de trabajo y/o investigación: Objetos de aprendizaje.

Correo electrónico: yheredia@uci.cu

DEDICATORIA

A mi madre, pues ella ha depositado toda su confianza en la concepción de la culminación de la carrera en ingeniería en ciencias informáticas, por estar siempre ahí cuando la necesité, apoyándome, por darme la vida, a ti dedico esta investigación.

El autor.

AGRADECIMIENTOS

Agradezco a todas las personas que de una forma u otra han hecho posible este sueño...

A mi madre por apoyarme incondicionalmente en todas las decisiones que he tomado en mi vida.

A mi tutora, por guiarme, por tener tanta paciencia, por toda la ayuda prestada y el tiempo dedicado.

A todos mis amigos, mencionarlos a todos sería no terminar, pues para cada uno de ellos existió una pregunta, una ayuda, una opinión, mil gracias por su tiempo y dedicación.

A cada uno de mis profesores durante toda mi carrera, cada uno diferente, pero todos especiales.

Muchas gracias a todos.

El autor.

RESUMEN

La Universidad de las Ciencias Informáticas (UCI), para gestionar los procesos de gestión académica, posee la aplicación automatizada Sistema de Gestión Académica Xauce Akademos, desarrollada en el Centro de Tecnologías para la Formación (FORTES). Este sistema está integrado por varios procesos, permitiendo la gestión del proceso académico a partir de varios módulos, dentro del cual se encuentra la gestión de trámites de personal. Actualmente sigue carente de funcionalidades en la parte estadística para visualizar la información captada del personal. El presente trabajo de diploma tiene como objetivo desarrollar un componente para generar información estadística de la gestión del personal en el Sistema de Gestión Académica XAUCE AKADEMOS. El amplio estudio realizado sobre aplicaciones que se asemejan al funcionamiento de la solución, definió la estructura y funcionalidades básicas del componente a realizar, seleccionando como metodología de software AUP-UCI y como arquitectura la orientada a servicio (SOA), así como las herramientas y tecnologías para el desarrollo de la solución. Se definió una estrategia de prueba plasmando los tipos de pruebas funcionales a partir de los niveles de prueba unidad y de integración, todo con el objetivo de encontrar y corregir posibles errores.

Palabras Claves: Gestión académica, Personal, Estadística.

ABSTRACT

The University of Informatics Sciences (UCI), to manage academic management processes, has the automated application Xauce Akademos Academic Management System, developed at the Center for Training Technologies (FORTES). This system is made up of several processes, allowing the management of the academic process from several modules, within which is the management of personnel procedures. Currently, it still lacks functionalities in the statistical part to view the information collected from the staff. This diploma work aims to develop a component to generate statistical information on personnel management in the XAUCE AKADEMOS Academic Management System. The extensive study carried out on applications that are similar to the operation of the solution, defined the structure and basic functionalities of the component to be carried out, selecting the AUP-UCI software methodology and service-oriented architecture (SOA), as well as the tools and technologies for the development of the solution. A test strategy was defined, capturing the types of functional tests from the unit and integration test levels, all with the aim of finding and correcting possible errors.

Keywords

Academic management, Personnel, Statistics.

TABLA DE CONTENIDOS

INTRODUCCIÓN 1

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICOS METODOLÓGICOS SOBRE LA GENERACIÓN ESTADÍSTICA DE PERSONAL EN EL SISTEMA XAUCE AKADEMOS PARA EL MINED. 6

1.1 Términos asociados a la investigación 6

1.1.1 Estadística:.....	6
1.1.2 Reportes:.....	7
1.1.3 Gestión Académica:.....	10
1.1.4 Sistema Gestión Académica:.....	11

1.2 Estado del Arte 12

1.3 Metodología a utilizar para la implementación de la solución 17

1.3.1 PROCESO UNIFICADO AGIL.....	17
1.3.2 Proceso Unificado Ágil versión UCI (AUP-UCI).....	18

1.4 Lenguaje de modelado 20

1.4.1 Herramienta de modelado.....	20
1.4.2 Sistema de gestión de bases de datos.....	21
1.4.3 Lenguaje de programación.....	21
1.4.4 Frameworks de desarrollo.....	22
1.4.5 Servidor web.....	24
1.4.7 Almacenamiento de datos en memoria.....	24

Conclusiones del capítulo 24

CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN PROPUESTA DE SOLUCIÓN AL COMPONENTE DE GESTIÓN ESTADÍSTICA DE PERSONAL PARA EL SISTEMA DE GESTIÓN ACADÉMICA XAUCE AKADEMOS PARA EL MINED. 25

2.1 Propuesta del Componente estadística de personal 26

2.1.1 Modelo conceptual.....	26
------------------------------	----

2.2 Requisitos del Componente de Gestión estadística de personal 27

2.2.1 Obtención de Técnicas de obtención de Requisitos del Componente de Gestión estadística de personal.....	27
2.2.1 Requisitos Funcionales.....	28
2.2.2 Especificación de requisitos funcionales.....	29
2.2.3 Requisitos no Funcionales.....	33
2.2.4 Descripción del requisito Generar Movimiento del personal.....	33
2.2.5 Descripción de los requisitos no funcionales.....	42
2.3 Descripción de la arquitectura y el diseño.....	43
2.3.1 Arquitectura Orientada a Servicios.....	43
2.3.2 Patrones de Diseño.....	44
Conclusiones del capítulo	49

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN 50

3.1 Estándares de codificación 50

3.2 Pruebas de software 50

3.2.1 Tipos de Pruebas 51

3.2.2 Métodos de Pruebas 55

Resultados obtenidos 56

Conclusiones del capítulo 58

CONCLUSIONES 59

RECOMENDACIONES 60

Referencias Bibliográficas 61

ANEXOS 66

ÍNDICE DE TABLAS

Tabla 1: Descripción de Requisitos Funcionales.....	32
Tabla 2: Descripción del Requisito Generar Movimiento del personal.....	42
Tabla 3: Descripción del Requisito no Funcional Usabilidad 1.1.....	42
Tabla 4: Estrategia de Pruebas de Software.....	51
Tabla 5: Caso de prueba unitaria de la funcionalidad Personal Docente.....	54
Tabla 6: Caso de Prueba con el Componente Estructura y Composición.....	55
Tabla 7: Prueba de caja negra.....	57
Tabla 8: Descripción del Requisito no Funcional Usabilidad 1.2.....	66
Tabla 9: Descripción del Requisito no Funcional Seguridad 1.2.....	67
Tabla 10: Descripción del Requisito Generar Reporte de datos sobre Personal Docente y no docente.	72

ÍNDICE DE FIGURAS

Fig. 1: Gráfico de Barra.....	9
Fig. 2: Gráfico de Pastel.....	9
Fig. 3: Gráfico de Línea.....	10
Fig. 4: Modelo Conceptual de la Gestión de Personal.....	27
Fig. 5: Patrón Controlador.....	46
Fig. 6: Patrón Alta cohesión.....	47
Fig. 7: Patrón Bajo acoplamiento.....	48
Fig. 8: Ejemplo de Caja Blanca.....	56
Fig. 9: Ejemplo de Caja Negra.....	56

INTRODUCCIÓN

En los últimos años los cambios que ha experimentado la sociedad moderna han sido enormes. La evolución de las Tecnologías de la Información y las Comunicaciones TIC han revolucionado no sólo las empresas sino también la vida cotidiana en general (Tecnología). Las Tecnologías de la Información y las Comunicaciones (TIC) son un conjunto de disciplinas científicas, tecnológicas, de ingeniería y de técnicas de gestión, utilizadas en el manejo y procesamiento de la información mediante el uso de distintos dispositivos electrónicos. Su uso en los diferentes niveles y sistemas educativos tienen un alto impacto significativo en el desarrollo del aprendizaje, mejorando así nuestras condiciones de vida y de trabajo.

El proceso de informatización de la gestión académica tiene como idea principal la gestión y visualización de la información de cada uno de los procesos informatizados, de manera que permita aportar reportes estadísticos en informes, de una forma rápida y precisa (Aguilar). Este proceso ha permitido en todos sus aspectos organizar, registrar, administrar y coordinar las diferentes actividades del proceso docente, entre las actividades existentes se destaca plan de estudio, la matrícula, el control de la asistencia y las evaluaciones de los estudiantes (Publicaciones, 2019).

La Universidad de las Ciencias Informáticas (UCI), para gestionar los procesos de gestión académica, posee la aplicación automatizada Sistema de gestión Académica Xauce Akademos para el MINED, desarrollado por un equipo de trabajo pertenecientes al Centro de Tecnologías para la Formación (FORTES) (Serie Científica de la Universidad de las Ciencias Informáticas, 2008). La misma contribuye al perfeccionamiento de los procesos académicos de una institución. Su uso permite el desarrollo coherente de una estrategia organizacional en los diferentes niveles de la estructura educacional con la que cuenta el Ministerio de Educación de la República de Cuba (Ramírez, 2021). Se le incorporaron varios procesos, permitiendo la gestión del proceso académico a partir de varios módulos, dentro del cual se encuentra la gestión de personal. Dicho módulo se encarga del control de los trámites de personal. Este módulo actúa de base para el apoyo en los procesos de análisis de información permitiendo, obtener información estadística a nivel de centro educacional, municipio y provincia facilitando el proceso de toma de decisiones a los directivos.

En este sistema, cada institución, a la hora de obtener información referente a los trabajadores, debe:

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.

De forma general, se ha llegado a la conclusión de que la versión de XAUCE AKADEMOS para el MINED¹, resulta un proceso complejo, engorroso, agotador y lento en el momento de realizar una operación sobre el estado de dicha información estadística para darle soporte a la toma de decisiones de directivos y profesores de los centros escolares. Esto trae como consecuencia las siguientes debilidades:

- Alto riesgo de errores humanos que pudieran provocar, dentro de los centros educacionales, la toma de decisiones basada en información incorrecta.
- Demora en la gestión y control del proceso de gestión académica a nivel de centro escolar.
- Inadecuada visualización de dichos indicadores a nivel de centro escolar, lo cual dificulta el seguimiento de los mismos, la creación de nuevas estrategias y el análisis de la información.
- Alto nivel de duplicidad de información provocando la pérdida de tiempo para el trabajador.
- Inadecuada manipulación de la información.

Luego de analizar la problemática anteriormente descrita se identifica como **problema a resolver**:

¿Cómo generar estadística asociada al personal de las instituciones educativas en el Sistema de Gestión Académica XAUCE AKADEMOS, de forma tal, que brinde soporte a la toma de decisiones?

La investigación en curso enmarca como **objeto de estudio**: Generación estadística asociada a recursos humanos en instituciones educativas y específicamente su **campo de acción**: la Generación estadística asociada a recursos humanos en el MINED.

Se define como **objetivo general**: Desarrollar un componente para generar estadísticas de personal en el Sistema de Gestión Académica XAUCE AKADEMOS para el MINED, de forma tal que brinde soporte a la toma de decisiones y mejore el control de la información en el proceso.

Se identificaron del objetivo general los siguientes **objetivos específicos** siguientes:

- Descripción del marco teórico conceptual de la generación estadística asociada a recursos humanos en instituciones educativas.
- Realizar el análisis y el diseño del componente generar estadística de personal.

¹ Ministerio de Educación de la República de Cuba

- Implementar las funcionalidades definidas en la generación estadística asociadas a recursos humanos con respecto a la gestión de personal.

-Validar el correcto funcionamiento de la solución desarrollada a partir del empleo de pruebas realizadas al componente.

La investigación se sustenta en la siguiente **idea a defender**: con el desarrollo del componente de generar estadística del personal en el Sistema de Gestión Académica XAUCE AKADEMOS, se optimiza el proceso de control de la información durante el proceso, contribuyendo a la toma de decisiones en el MINED, garantizando la obtención de reportes en gráficos estadísticos del personal existente.

Tareas de la Investigación:

- Elaboración del marco teórico de de la generación estadística asociada a recursos humanos en instituciones educativas.

- Análisis de las tecnologías a utilizar para el desarrollo de estadística asociadas a recursos humanos con respecto a la gestión de personal.

-Elección de los métodos, metodologías, tecnologías, herramientas y técnicas a utilizar para la implementación del componente generar estadística del personal dentro del sistema Akademos.

- Implementación de las funcionalidades definidas para la generación estadística asociada a la gestión de personal.

- Validación el componente, generar estadística del personal dentro del sistema Akademos para garantizar su funcionamiento adecuado.

Métodos Teóricos:

Analítico- Sintético: Este método permite extraer lo esencial de la bibliografía consultada, así como arribar a las conclusiones de la investigación. Además se utilizó con el objetivo de analizar los elementos bibliográficos y definiciones relacionadas con el tema de investigación.

Inductivo – deductivo: Este método es utilizado para el análisis de los sistemas existentes que gestionan información referente a la gestión estadística de personal internacional y nacional. Además permite llegar a conclusiones lógicas a partir de los conocimientos adquiridos en la investigación y así poder plantear una solución para la realización de la aplicación.

Histórico- Lógico: Permite determinar la trayectoria y las tendencias actuales de los estándares que gestionan la información académica.

Métodos empíricos:

Entrevista: Se realizaron entrevistas a personas dentro del centro con el fin de obtener conocimiento sobre la situación actual de la gestión del personal con el fin de obtener toda la información posible para un adecuado desarrollo del componente.

Instrumentos:

La **Guía de Entrevista** sirve como un contexto para el análisis de situación hipotética y actual de la gestión estadística de personal en el MINED. Ver Anexo 1.

El documento está estructurado por los siguientes capítulos:

Capítulo 1. Fundamentos teóricos: incluye el estudio del estado del arte del tema tratado, donde se exponen los principales conceptos empleados en el transcurso de la investigación. Se realiza la descripción y análisis de los sistemas informáticos de gestión estadística del personal existente y una caracterización de las herramientas, técnicas, lenguajes, metodología y el proceso de desarrollo a utilizar en la construcción de la propuesta de solución.

Capítulo 2. Concepción de la propuesta de solución: describe la propuesta de solución enmarcada en el proceso actual de gestión del personal en el sistema xauce Academos, permitiendo construir el modelo conceptual, la especificación de requisitos y definir mediante las técnicas de obtención de requisitos tanto funcionales como no funcionales. Además de describir la arquitectura y patrones de diseño, que se utilizan en la investigación.

Capítulo 3. Construcción y validación de la propuesta de solución: detalla la construcción de la solución propuesta. Se describen aspectos relacionados con la implementación, donde se refleja el empleo de diferentes tipos de pruebas, analizando los resultados de dichas pruebas como parte de la validación de la solución.

CAPÍTULO I: Fundamentos y referentes teóricos metodológicos sobre la Generación estadística de personal en el sistema Xauce Akademos para el MINED.

En el presente capítulo se definen los principales conceptos asociados a la gestión estadística de personal, así como el estudio a diversos sistemas informáticos referentes al tema, con el objetivo de estudiar sus funcionalidades y características dando lugar a una visión general del estado del arte. Para guiar la implementación de la propuesta de solución, se caracterizan las tecnologías y la metodología de desarrollo de software a utilizar para desarrollar el sistema.

1.1 Términos asociados a la investigación

Para el desarrollo del sistema es necesario el análisis de diferentes términos que ayuden a una mejor comprensión de la obtención de estadísticas del personal en el sistema de gestión académica Xauce Akademos. Estos conceptos se presentan a continuación:

1.1.1 Estadística:

La [estadística](#) es comúnmente considerada como una colección de hechos numéricos expresados en términos de una relación sumisa, y que han sido recopilados a partir de otros [datos](#) numéricos.

Según Kendall y Buckland definen la estadística como un [valor](#) resumido, calculado, como base en una [muestra](#) de observaciones que generalmente, aunque no por necesidad, se considera como una estimación de parámetro de determinada [población](#); es decir, una [función](#) de [valores](#) de muestra.

Murria R. Spiegel, (1991) dice: "La estadística estudia los [métodos](#) científicos para recoger, organizar, resumir y analizar datos, así como para sacar conclusiones válidas y tomar decisiones razonables basadas en tal [análisis](#)."

Para Chacón esta se define como "la [ciencia](#) que tiene por objeto el estudio cuantitativo de los colectivos"; otros la definen como "la expresión cuantitativa del [conocimiento](#) dispuesta en forma adecuada para el escrutinio y análisis". La más aceptada, sin embargo, es la de Minguez, 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" (Barrios, 2011).

La estadística se encarga de explicar además de mostrarle al operario o al administrador de recursos, si su producción es viable o no, dado a la utilización de probabilidades. Así por medio de técnicas de control de calidad y mejora de los procesos de producción se puede llevar a la funcionalidad perfecta. Además, se divide en dos grandes ramas, la inferencial y la descriptiva. Esta última consiste en la representación de datos en forma de tablas y gráficos, comprende cualquier actividad relacionada con estos y está diseñada para resumir o describir los mismos sin factores pertinentes adicionales.

La representación de datos estadísticos, constituye en sus diferentes modalidades, uno de los aspectos de más uso en la estadística, permitiendo visualizar la presentación de los mismos sobre el com-

portamiento de las diferentes variables. Una manera de representar estos datos es mediante reportes estadísticos.

1.1.2 Reportes:

Son informes que organizan y exhiben la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos, para mostrarlos por medio de un diseño atractivo y de fácil interpretación para los usuarios. Los reportes presentan diversos niveles de complejidad, desde una lista o enumeración, hasta gráficos mucho más desarrollados (Crisanto, 2017).

Existen diferentes representaciones de datos, la representación escrita, que es usada cuando la información o una serie de datos a representar contiene pocos valores, mediante esta forma, se resalta la importancia de las informaciones principales; la representación tabular, cuando los datos se presentan a través de un conjunto de filas y de columnas que responden a un ordenamiento lógico; y la presentación gráfica que proporciona mayor rapidez en la comprensión de los datos (Crisanto, 2017).

Un aspecto crucial cuando se decide representar los datos a través de gráficos, es la correcta elección del tipo a utilizar, debido a que no todos se ajustan a la naturaleza de lo que se desea representar. Los gráficos constituyen por sí mismos una poderosa herramienta para el análisis de los datos, siendo en ocasiones el medio más efectivo no sólo para describir y resumir la información, sino también para analizarla y llegar a un buen punto de partida para la toma de decisiones. Estos además tienen una función estética que atrae la atención del usuario y lo convence del mensaje que se quiere transmitir.

En comparación con otras formas de presentación de datos, los gráficos estadísticos permiten, de una mirada, comprender el comportamiento de cualquier tipo de variable independientemente del grado de dificultad que presente, por lo que es válido decir que ahorran un tiempo considerable al que analiza la información, ya que los mismos permiten usar las habilidades visuales para procesar información en un tiempo relativamente considerable.

La representación de los datos estadísticos mediante el uso de gráficos es de vital importancia para la comprensión e interpretación de la información, además es considerada una tarea importante en el proceso de comunicación. A continuación, se hace referencia a algunos tipos de gráficos existentes, teniendo en cuenta su uso para el desarrollo del componente para la gestión estadística de personal en el sistema de gestión académica Xauce Akademos.

Gráfico de barras: una representación gráfica de los resultados de un análisis estadístico. El gráfico consta de barras para cada dato representado. Las anchuras de estas barras son iguales, pero las longitudes varían según la importancia del valor. Estas barras se colocan generalmente en 2 ejes que pueden invertirse dependiendo de si se quiere hacer un gráfico de barras horizontal o vertical.

Ventajas

- Vista global de la frecuencia de una variable discreta.
- Es un tipo de gráfico muy fácil de crear y es muy útil para transferir información.

Desventajas

- En este tipo gráfico no se puede representar una variable continua.

Se utiliza una gráfica de barras en alguna de las siguientes situaciones:

- Para comparar dos o más valores de la misma categoría.
- Comprender cómo se relacionan múltiples conjuntos de datos similares entre sí (Tudashboard, 2022).

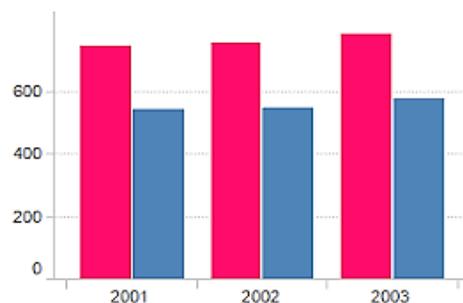


Fig. 1: Gráfico de Barra

Gráfico de pastel: es un tipo de representación para el análisis de datos estadísticos. Tiene la forma de un disco dividido en sectores, cuyas áreas son proporcionales a los porcentajes de los distintos componentes de la población estadística. Cada valor del carácter estudiado corresponde a un sector. Las medidas de los ángulos de los sectores son proporcionales a los números representados (o a las

frecuencias asociadas). La representación de números negativos es imposible con este tipo de diagrama y la frecuencia o el porcentaje se representa tanto visual como numéricamente, por lo que suele ser rápida de entender para los lectores.

Existen algunas limitaciones al utilizar una gráfica de pastel. Si hay demasiadas categorías, entonces habrá una multitud de pedazos de pastel. Algunos de ellos pueden ser probablemente muy delgados y pueden ser difíciles de comparar entre sí. Los gráficos circulares no deben utilizarse para evaluar los tamaños relativos de las categorías, comparar datos entre gráficas de pastel y visualizar porcentajes que no sumen el 100% (2022).

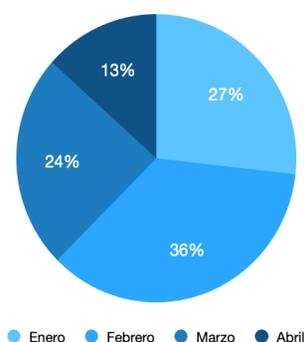


Fig. 2: Gráfico de Pastel

Gráficos de líneas: los gráficos de líneas ayudan a visualizar los datos en un formato compacto y preciso que facilita la exploración rápida de la información para comprender las tendencias. El uso adecuado del color en esta visualización es necesario porque las líneas de diferentes colores pueden facilitar aún más el análisis de la información por parte de los usuarios, además son ideales para mostrar tendencias y patrones de datos a lo largo del tiempo.

Un gráfico de líneas se utiliza para:

- Entender las tendencias, los patrones y las fluctuaciones de tus datos.
- Comparar conjuntos de datos diferentes pero relacionados con múltiples series.
- Hacer proyecciones más allá de tus datos (Tudashboard, 2022).

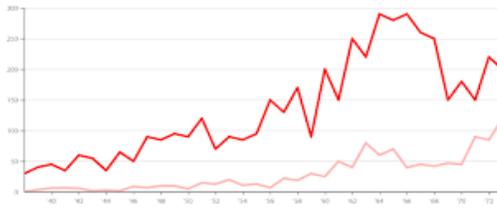


Fig. 3: Gráfico de Línea

Gráficos de Áreas: se usa para mostrar la tendencia de la información generalmente en un período de tiempo.

Gráficos Mixtos: se representan dos o más series de datos, cada una con un tipo diferente de gráfico. Son gráficos más vistosos y se usan para resaltar las diferencias entre las series.

Personal: Pertenciente o relativo a la persona. Conjunto de las personas que trabajan en un mismo organismo, dependencia, fábrica, taller, etc (Española).

1.1.3 Gestión Académica:

Es un ámbito de acción orientada a facilitar y mejorar los procesos formativos que imparten las instituciones de educación superior (acreditación). También conocido como un sistema abierto, flexible y dinámico que se compone de una serie de procesos que hacen posible el desarrollo de la actividad de un centro educacional. Es decir, se trata de prever, organizar, coordinar y hacer un seguimiento de los proyectos y acciones a través de los recursos humanos o los materiales.

Está constituida por cuatro áreas de gestión que son:

- Área de gestión directiva.
- Área de gestión pedagógica.
- Área de gestión de la comunidad.
- Área de gestión administrativa y financiera.

Para mejorar la gestión académica de una institución educacional son necesarios los parámetros que se definen a continuación:

- **Planificación:** Es un proceso muy importante en la gestión académica, y que conlleva sobre todo una previsión y fijación de objetivos para poder desarrollarlos con los recursos de los que

se dispone. Sin una planificación adecuada, los objetivos no se consiguen y sobre todo se gastarían demasiados recursos en tratar de hacerlo.

- **Organización:** Muy cerca de la planificación está la organización, que permite que exista un método concreto de aplicación de procedimientos, es decir, que muy pocas cosas se dejen al azar. De esta manera se consigue mejorar todas estas acciones y coordinar a todo el equipo para conseguir todas las metas.
- **Dirigir:** La dirección es una especie de ente mediador que hace posible tanto la planificación como la organización, programando, desarrollando y monitorizando los procesos. Una de las tareas más importantes que se encomiendan a la dirección es la de gestionar un equipo humano de calidad y eficiente, que permita el mayor rendimiento y elevar la calidad de la enseñanza (Educaweb, 2022).

1.1.4 Sistema Gestión Académica:

Herramienta de apoyo informático 100% web que a través de sus componentes atiende completamente el ciclo de vida del estudiante. Brinda mecanismos de control y gestión de la información en beneficio de los estudiantes, docentes y administrativos, logrando agilidad, oportunidad, seguridad y calidad en la información. En el marco de los planes de modernización y desarrollo institucional, el sistema de Gestión Académica se convierte en un medio eficaz para lograr los objetivos institucionales y de los programas, para facilitar la acreditación y en especial entregar productos y servicios de calidad a la comunidad.

Principales Características:

- Aporta medios de gestión y control de la información que se desprende de los procesos misionales y el Reglamento Institucional de la Institución.
- Facilita los procesos de Autoevaluación con miras a la Acreditación.
- Facilita herramientas informáticas que permiten llevar registro, control y seguimiento al ciclo de vida del estudiante cubriendo las principales etapas de los procesos académicos (CASEWER, 2018).

1.2 Estado del Arte

Normalmente, en un Sistema de Gestión Académica, los usuarios suelen ejecutar un gran número de acciones sobre los diferentes contenidos almacenados. Tener un control sobre dichas acciones, resulta sumamente importante debido a que permite tener constancia de la gestión de los contenidos. Por tal razón, diferentes Sistema de Gestión Académica han incorporado de una u otra manera, entre sus funcionalidades, la generación de datos estadísticos.

La estadística educativa, permite recolectar información para analizar y tomar decisiones en diferentes niveles. Se ocupa de la recolección, agrupación, representación, análisis e interpretación de datos. Es un método científico que pretende sacar conclusiones a partir de observaciones realizadas (Gonzalez, 2017).

El uso de la estadística aplicada a la gestión de los recursos humanos se constituye en una útil y poderosa herramienta. La administración de recursos humanos dentro de una empresa o institución, es una labor colmada de desafíos para los que se encargan de esta gestión. Todos estos desafíos son para que tanto la empresa como la institución sea eficiente, eficaz y para que la administración del personal sea uno de los eslabones más importantes dentro de estas áreas (Insignia, 2016).

El área de recursos humanos tiene como función gestionar la información del personal tal como: control de asistencia, vacaciones, permisos de salidas, comisiones, etc. También se encarga de la difusión de documentación institucional es decir políticas, normas, reglamentos, resoluciones, etc. Además, permite mantener los informes actualizados a modo de evaluar sus fortalezas y debilidades en cuanto a la manera que se están organizando con los empleados, es decir una herramienta que sirva de referencia en la toma de decisiones, es aquí donde la estadística se convierte en un aliado fundamental para el sector.

La estadística es mucho más que solo números y gráficos, es una forma de coleccionar información para obtener datos y luego analizar, resumir, organizar y llegar a las conclusiones necesarias en base a esos datos (Insignea, 2016).

A continuación, se realiza un estudio de algunos de estos sistemas, a fin de analizar la generación o no de datos estadísticos y de este modo identificar posibles funcionalidades a incorporar al Módulo estadístico para el Sistema de Gestión académica Xauce Akademos.

DocCF²: es una aplicación desarrollada para automatizar los procedimientos administrativos, académicos y comerciales de las instituciones educativas, así como facilitar la coordinación y comunicación entre padres, alumnos, docentes y cargos directivos para ofrecer información estadística sobre dichos procesos y facilitar la toma de decisiones en el proceso de gestión de la institución. Específicamente cuenta con módulos para la gestión de alumnos, registro de matrículas, gestión de docentes, evaluación de docencia, asignación y control de horarios, administración de la biblioteca, registro de calificaciones y generación de boletines escolares, generador de consultas, informes y estadísticas personalizadas (Guillot, 2018).

SIGA (Software Integrado de Gestión Académica Web): es un sistema modular para la administración académica y curricular, es una solución informática utilizada por la universidad de Córdoba para la automatización de la gestión académica. Esta aplicación, desarrollada por el servicio de informática de la universidad, se encuentra en continua evolución, ampliando nuevas funcionalidades y adaptándose a las nuevas tecnologías.

Las funcionalidades soportadas por SIGA son muy amplias, abarcan desde la gestión del acceso a la universidad hasta la tramitación de los títulos pasando por la matrícula, actas de examen, expedientes académicos, estadísticas, informes, gestión de becas y convalidaciones. Diseñado especialmente para instituciones de educación superior, funciona completamente en Internet, que integra tanto datos como procesos en una solución completa eliminando barreras de espacio y tiempo. SIGA cumple a cabalidad con las características necesarias que garantizan la calidad del mismo, ellas son: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; todo esto permite la integración con futuros desarrollos de manera económica y de alta calidad (Siga, 2017).

Se estructura en los siguientes módulos:

General: aparecen activados todos los módulos adquiridos por el cliente y en sombra los no adquiridos o desactivados por el supervisor. Instala automáticamente nuevos módulos, actualiza el sistema, la fecha y la hora, las estadísticas de utilización del programa y el *Copyright*³.

Definición de estudios: permite al usuario organizar sus estudios en varios planes, cada plan en varios cursos y cada curso con varias asignaturas. Los términos plan, curso, asignatura, grupo,

² Plataforma de Gestión Escolar

³ Derecho de Autor

turno, son personalizables por el usuario, es decir, puede llamarles, año, programa, módulo, provincia, área si lo considera, adecuándose la mayoría de los listados y menús a estos términos.

Opciones de alumnos: está dividido en cuatro sub módulos: Gestión de tutorías, Gestión de mensajerías, Gestión de asistencias y Gestión de calificaciones.

Gestión de Calificaciones: posibilita al usuario definir cuántos exámenes querrá hacer y de qué tipo y así hacer medias de exámenes con condiciones. Se generan listados para los expedientes académicos, calificaciones de un alumno/curso, observaciones, diplomas, entre otros. Profesores.

GENFOR (Generador de informes, estadísticas y gráficos).

Matrícula: permite la inscripción del estudiante en las asignaturas que desee y le estén permitidas, conforme a su expediente académico y a la normativa existente. Además, posibilita actualizar datos estadísticos, solicitar grupos de clase y renovar la tarjeta universitaria.

Auto-matrícula: permite que los alumnos se matriculen fácilmente vía Internet sin necesidad de realizar las gestiones antes pertinentes, utilizando para ello las aulas de informática de cada centro o incluso desde el ordenador de sus casas, por supuesto, los alumnos que lo deseen podrán acudir en cualquier momento a la secretaría de su centro para matricularse.

Cuba no se ha quedado atrás en el desarrollo de herramientas de gestión para el ámbito universitario, donde se puede mencionar:

El Sistema de Gestión de la Nueva Universidad (SIGENU) es un sistema que se desarrolla con el fin de ser una herramienta que permita la gestión de toda la información académica vinculada con la educación superior en Cuba. En correspondencia con su carácter nacional y la gran diversidad de sistemas de enseñanza superior con que cuenta la universidad cubana, este sistema se concibe de manera tal que sea capaz de brindar gran seguridad e integridad de la información, y a la vez, ser tan flexible que permita ser adaptado a todos los centros de educación superior del país con sus diversas particularidades y distintas maneras de realizar determinados procedimientos. La aplicación cliente cuenta con versiones de escritorio y para la Web.

Para el despliegue del sistema se utiliza como servidor web y servidor de aplicaciones *Tomcat*⁴, y el *jboss*:

Apache *Tomcat* es un contenedor de aplicaciones web de código abierto creado para ejecutar aplicaciones basadas en *Servlets* y páginas JSP. *Tomcat* es creado en Java por lo que es multiplataforma, puede funcionar en cualquier sistema operativo que cuente con la máquina virtual de Java. Incluye en su estructura un compilador *Jasper* que convierte las paginas JSP en *Servlets*⁵.

El *JBoss* es un servidor de aplicaciones de código abierto que implementa la especificación de la plataforma JEE en su totalidad. El mismo está basado en un diseño de micro-arquitectura dentro del cual los componentes pueden ser conectados en tiempo de ejecución para extender su comportamiento.

SIGENU en su nueva versión está compuesto por diferentes elementos dentro de los que se encuentran:

Cliente Secretaría (SIGENU-SEC): aplicación que constituye el elemento esencial que hace posible la inserción y actualización de toda la información que se registra en el sistema. Además, permite obtener un conjunto importante de reportes muy usados cotidianamente en el mundo de la educación superior.

Cliente Profesor (SIGENU-Profesor): aplicación que permite planificar a un profesor en una asignatura específica. Para realizar este proceso es necesario ingresar en la aplicación SIGENU_PROFESOR con un usuario con rol "Planificador".

Cliente Administrador (SIGENU-Admin): aplicación que permite la inserción y actualización de los usuarios y todas las funcionalidades que ejecutan los administradores para monitorear el correcto funcionamiento del sistema y su seguridad.

Cliente Archivo (SIGENU-Archivo): módulo que permite visualizar la información de todos los egresados por SIGENU de la Educación Superior en Cuba.

Cliente Servicios (SIGENU-servicios): es un sitio que ofrece un conjunto de servicios web que permitirán al desarrollador, utilizar la información que gestiona el SIGENU de forma segura (Sistema de gestión de la Nueva Universidad (SIGENUv4.0.0) como herramienta para la automatización de los procesos docentes y estadísticos de la Universidad de Granma, 2018).

⁴ Contenedor de *Servlets* que se puede usar para compilar y ejecutar aplicaciones web realizadas en Java.

⁵ Clases Java diseñadas para responder a solicitudes HTTP.

Valoración de los sistemas de gestión académica

Todos los sistemas internacionales estudiados se rigen por normas internacionales que garantizan la calidad y la eficiencia en su funcionamiento. Para lograr este aval internacional cuentan con toda su documentación en orden, lo que garantiza desde el buen funcionamiento y mantenimiento del sistema, hasta una correcta gestión del personal. Aunque los sistemas nacionales no están certificados por normas internacionales, si mantienen una correcta documentación que garantiza una buena gestión del personal y de la información que manejan.

Los sistemas estudiados poseen una amplia documentación de registro y control de toda la información y el personal que poseen, a pesar de ser sistemas eficientes ninguno de estos documentos se pudo utilizar para la elaboración de la propuesta de solución, debido a que dicha propuesta debe ser aplicable a la forma de desarrollo del componente estadístico del personal en el sistema Xauce Akademos.

1.3 Metodología a utilizar para la implementación de la solución

Resulta necesario establecer un enfoque sistemático y disciplinario para desarrollar un software además de seleccionar una metodología que permita el dominio del proceso descrito.

Una **metodología** es un enfoque, una manera de interpretar la realidad o la disciplina en cuestión. Se elabora a partir del marco definido por uno o varios ciclos de vida. Tienen como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo.

Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente, las cuales se clasifican en dos clases: las metodologías tradicionales o robustas y las ágiles o ligeras.

Metodologías tradicionales o robustas: impone una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Se centra especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada.

Metodologías ágiles o ligeras: permite adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

Ambas metodologías describen cómo se organiza un proyecto, establecen el orden en el que la mayoría de las actividades deben realizarse y los enlaces entre ellas. Indican cómo tienen que realizarse algunas tareas proporcionando las herramientas concretas e intelectuales.

1.3.1 PROCESO UNIFICADO AGIL

Proceso Unificado Ágil o *Agile Unified Process (AUP)*: es un marco de desarrollo software iterativo e incremental. A menudo es considerado como un proceso altamente ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto de software. 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.

Para el desarrollo de la solución se utilizará la metodología definida por el centro FORTES para el desarrollo de proyectos. La misma consiste en una versión desarrollada por la UCI basada en la metodología de desarrollo de software AUP.

1.3.2 Proceso Unificado Ágil versión UCI (AUP-UCI)

Metodología de Desarrollo AUP-UCI tiene como eje principal la aplicación de técnicas ágiles incluyendo:

- Desarrollo dirigido por pruebas,
- Modelado ágil,
- Gestión de cambios ágil,
- Refactorización de Base de datos para mejorar la productividad.

La metodología de software AUP-UCI a partir del modelado de negocio propone tres variantes a utilizar en los proyectos, como son: CUN (Casos de uso del negocio), DPN (Descripción de proceso de negocio) o MC (Modelo conceptual), a su vez existen tres formas de encapsular los requisitos los cuales son: CUS (Casos de uso del sistema), HU (Historias de usuario), DRP (Descripción de requisitos por proceso) y también surgen cuatro escenarios para modelar el sistema en los proyectos, los cuales son:

- Escenario No 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

- Escenario No 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- Escenario No 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.
- Escenario No 4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU (Sánchez, 2015).

AUP-UCI tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas.

Fases de la metodología 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.

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.

Cierre: En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Disciplinas de AUP-UCI:

Modelado de negocio: es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito.

Requisitos: el esfuerzo principal en la disciplina 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.

Análisis y diseño: en esta disciplina se modela el sistema y todos los requisitos, incluyendo los requisitos no funcionales.

Implementación: en esta disciplina a partir de los resultados del Análisis y Diseño se construye el sistema.

Pruebas internas: en esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas.

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 (Sanchez, 2015).

A partir del análisis e investigación realizada, se utiliza la metodología AUP-UCI en el escenario 3, debido a que es la apropiada para proyectos pequeños permitiendo disminuir las probabilidades de fracaso, por ser un producto de fácil uso y también por ser un proceso simplificado. Con la adaptación de AUP se propone para la gestión estadística de personal en la UCI estandarizar el proceso de desarrollo de software, lográndose hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos.

1.4 Lenguaje de modelado

Además de lo esencial que resulta una metodología de software como hilo conductor de todo el ciclo de vida del sistema, también es importante tener en cuenta que para desarrollar un proceso de desarrollo de software es necesario contar con algún elemento que describa el aspecto y la conducta del producto; estos elementos son llamados lenguaje de modelado.

Lenguaje Unificado de Modelado (UML)

Este lenguaje proporciona un vocabulario, reglas para permitir una comunicación y permite visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. A los creadores de sistema les permite generar diseños que capturen las ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas. Este lenguaje sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten (Enheror, 2019).

1.4.1 Herramienta de modelado

Visual Paradigm para UML⁶ V15.0

Es una herramienta profesional que soporta el ciclo completo de desarrollo del software: análisis y diseño orientados a objetos, construcción y despliegue a través de la representación de varios tipos de diagramas. Permite importar proyectos de otras herramientas de modelado como *Rational Rose*, *Erwin* y *Microsoft Visio*. Soporta la revisión ortográfica, brindando sugerencias para los idiomas: inglés, español, francés, alemán y portugués. Ayuda a los clientes a transformar con precisión sus requisitos del sistema en soluciones de software de alta calidad. Todos los productos de *Visual Paradigm* están diseñados y desarrollados para eliminar la complejidad, mejorar la productividad y comprimir el desarrollo de software según plazos emitidos por los clientes (Visual Paradigm, 2020).

1.4.2 [Sistema de gestión de bases de datos](#)

PostgreSQL

Es un sistema de bases de datos de código abierto, altamente estable, que proporciona soporte a diferentes funciones de SQL, como claves foráneas, subconsultas, disparadores y diferentes tipos y funciones definidas por el usuario. Además, aumenta el lenguaje SQL ofreciendo varias funciones que escalan y reservan meticulosamente las cargas de trabajo de datos. Se utiliza principalmente para almacenar datos para muchas aplicaciones móviles, web, geoespaciales y de análisis (Kinsta, 2022).

Posee numerosas ventajas entre ellas pueden mencionarse:

- Instalación ilimitada.
- Extensible pues el código fuente está disponible para todos sin costo.
- Multiplataforma.
- Diseñado para ambientes de alto volumen.

1.4.3 Lenguaje de programación

Python V3.8.6

⁶ Lenguaje para hacer modelos.

Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. A diferencia de otros lenguajes como Java o .NET⁷, se trata de un lenguaje interpretado, es decir, que no es necesario compilarlo para ejecutar las aplicaciones escritas en Python, sino que se ejecutan directamente por el ordenador utilizando un programa denominado interpretador, por lo que no es necesario “traducirlo” a lenguaje máquina (Python, 2021).

Ventajas de *Python*

Python tiene diversas aplicaciones en las empresas dedicadas al desarrollo de software, como *frameworks*, aplicaciones web, creación de prototipos, etc. Esto proporciona a *Python* una ventaja competitiva frente a otros lenguajes de programación utilizados en el área. Destacan las siguientes ventajas:

- **Lenguaje de alto nivel:** *Python* es un lenguaje de alto nivel, por lo que es más fácil de usar que los de bajo nivel, puesto que estos últimos no tienen mucha abstracción de lenguaje de máquina.
- **Bibliotecas y *frameworks*:** tiene una amplia colección de bibliotecas y *frameworks*. La biblioteca estándar de *Python* es muy extensa, puesto que contiene muchos módulos integrados.
- **Portabilidad:** *Python* es compatible con todos los sistemas operativos (*Linux* y *Windows*), y los programadores solo necesitan escribir código una vez y luego podrá ejecutarse en todas partes.
- **Gratis y de código abierto:** *Python* es un lenguaje de programación desarrollado bajo la licencia de código abierto aprobada por OSI, todos pueden usarlo y distribuirlo libremente.
- **Comunidad fuerte:** El hecho de que *Python* sea gratuito y de código abierto contribuye a crear una comunidad sólida. Los programadores de *Python* pueden descargar el código fuente, modificarlo y distribuirlo como deseen (KEEPCODING , 2021).

⁷ Plataforma de aplicaciones.

1.4.4 Frameworks⁸ de desarrollo

Para el desarrollo del sistema se utilizaron a gran escala de los frameworks y librerías: Django, *Django Rest Framework* y Angular, para desarrollar la parte administrativa, la API y la vista del sistema respectivamente.

Django V3.1.5

Django es un *framework* web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, *Django* se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago (22051).

Se trata del entorno de desarrollo de *Python* más popular, ya que permite un desarrollo ágil al encargarse de manejar los controladores.

Ventajas de *Django*

- Cuenta con un sistema de autenticación de usuarios.
- Manejo de versiones que permite una distribución simple de actualizaciones.
- Ofrece un gran rendimiento y flexibilidad, pudiendo escalar proyectos de forma sencilla.

Desventajas de *Django*

Entre los inconvenientes que se pueden encontrar al utilizar Django para desarrollar aplicaciones web con *Python* se destaca su extensa documentación, que puede llegar a ser confusa y dispersa. También, puede resultar difícil a la hora de realizar *API Rest* (estilo de arquitectura software para crear aplicaciones web respetando el protocolo HTTP) (2020).

Angular V8.2.7

Angular es una plataforma de desarrollo, construida sobre *TypeScript*. Es un *framework* basado en componentes para crear aplicaciones web escalables. Una colección de bibliotecas bien integradas que cubren una amplia variedad de características, que incluyen enrutamiento, administración de

⁸ Framework: Marco de trabajo. Es una estructura de software de componentes personalizables e intercambiables para el desarrollo de una aplicación.

formularios, comunicación cliente-servidor y más. Un conjunto de herramientas para desarrolladores que permiten desarrollar, compilar, probar y actualizar el código fuente de la aplicación (Hiberus, 2020).

Ventajas de utilizar Angular

- Arquitectura de Diseño mejorada.
- Directivas personalizadas.
- Servicios e Inyección de dependencia.

Desventajas

- Angular es detallado y completo.
- Opciones de SEO⁹ limitadas (Ichlese, 2022).

1.4.5 Servidor web

Un servidor web es un software que forma parte del servidor y tiene como misión principal devolver información (páginas) cuando recibe peticiones por parte de los usuarios, en otras palabras, es el software que permite que los usuarios que quieren ver una página web en su navegador puedan hacerlo (Webempresa, 2022).

El servidor HTTP **Apache** es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), *Microsoft Windows*, *Macintosh* y otras, que implementa el protocolo HTTP y la noción de sitio virtual. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido. Es modular, extensible y popular (fácil conseguir ayuda/soporte) (Ramírez, 2021).

1.4.6 Herramienta de monitoreo

Grafana es una herramienta de código abierto, específicamente con licencia Apache 2.0. Está escrita en Lenguaje Go (creado por Google) y *Node.js* LTS y con una fuerte Interfaz de Programación de Aplicaciones (API). Permite el análisis y visualización de datos de serie temporales y se utiliza

⁹ Optimización de motores de búsqueda.

frecuentemente para visualizar de una forma elegante series de datos en el análisis de infraestructuras y aplicaciones (Ramírez, 2021).

1.4.7 Almacenamiento de datos en memoria

Redis es un almacén de estructura de datos en memoria de código abierto (con licencia BSD), que se utiliza como base de datos, caché y agente de mensajes. El mismo proporciona estructuras de datos como cadenas, listas, conjuntos, conjuntos ordenados con consultas de rango, mapas de bits, índices geoespaciales y flujos. (Ramírez, 2021).

Conclusiones del capítulo

En este capítulo se realizó un estudio de los principales conceptos que guiaron la investigación fomentando las bases teóricas para comprender y analizar la gestión estadística de personal en un sistema de gestión académica, permitiendo caracterizar cada uno de los conceptos seleccionados, describiendo sus características, logrando identificar cada elemento principal dentro de cada descripción. Se analizaron los sistemas utilizados actualmente que permiten mejorar la gestión académica en centros educativos de Cuba y el mundo, adquiriendo un conjunto de conocimientos previos sobre las características fundamentales que tienen estos, además de tener conocimiento de sus ventajas y los inconvenientes por lo cual ninguno es escogido para ser solución a la problemática. Con el análisis del ambiente de desarrollo se permitió establecer la metodología, las herramientas, lenguajes y tecnologías a utilizar en el desarrollo del sistema de acuerdo a las características deseadas por el sistema Xauce Akademos.

CAPÍTULO 2: Diseño de la solución propuesta de solución al Componente de Gestión estadística de personal para el Sistema de gestión académica Xauce Akademos para el MINED.

En el presente capítulo se exponen las características de la propuesta de solución, iniciándose con el análisis del modelo de negocio para una mejor comprensión del entorno de la investigación. Se presenta una descripción detallada de los requisitos funcionales y no funcionales, así como las técnicas para su obtención. Además, se describe la arquitectura, el diseño del sistema y los patrones utilizados.

2.1 Propuesta del Componente estadística de personal

Se propone un componente de reportes estadísticos de personal para el sistema de gestión académica XAUCE AKADEMOS del MINED en el cual:

- Deben utilizarse herramientas libres capaces de permitir concluir la implementación de todas las funcionalidades requeridas.
- Contar con un entorno de fácil uso en cuanto a la manipulación de las principales acciones que intervienen en la misma.
- Poseer las funcionalidades que le permitan al usuario interactuar fácilmente con el reporte estadístico del personal en el menor tiempo posible.

El desarrollo de la propuesta de solución se contempla de forma general en estos pasos:

1. Consultar la base de datos del sistema de gestión de personal.
2. Extraer los datos solicitados para la generación de reportes estadísticos del personal.

3. Transformar los datos extraídos.
4. Generar los reportes estadísticos en representación ya sea escrita, tabular o en forma de gráfico.

2.1.1 Modelo conceptual

El modelo conceptual es una técnica de análisis de requisitos y de diseño de bases de datos que debe suministrar información al usuario acerca de qué hace el sistema y los mecanismos para llevarlo a cabo. Su importancia radica en que debe favorecer el aprendizaje del sistema y es una guía para predecir el comportamiento del mismo. El usuario utilizará este modelo para establecer estrategias encaminadas a resolver sus problemas (Invid, 2020).

La siguiente figura muestra el modelo conceptual de la problemática planteada.

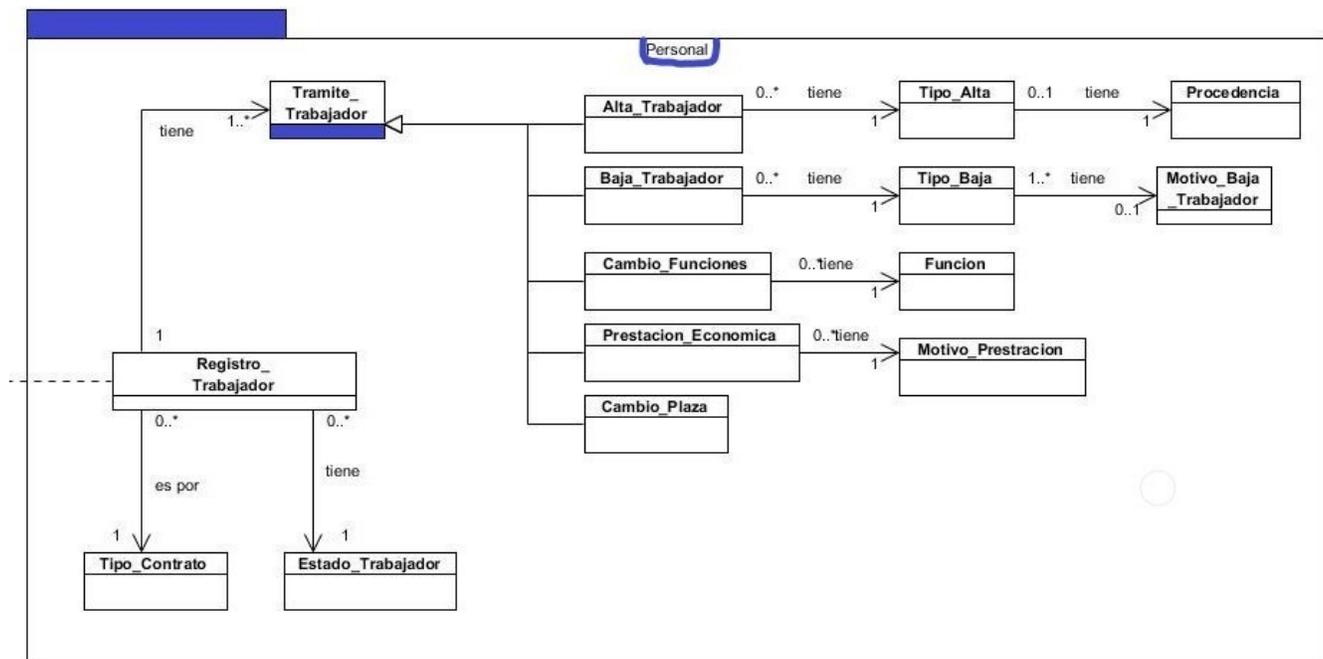


Fig. 4: Modelo Conceptual de la Gestión de Personal

2.2 Requisitos del Componente de Gestión estadística de personal

Requisitos del Sistema: descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar,

especificación u otro documento formalmente impuesto al inicio del proceso (La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software, 2017).

2.2.1 Obtención de Técnicas de obtención de Requisitos del Componente de Gestión estadística de personal

La obtención de requerimientos es una de las cuatro actividades que define la Ingeniería de Requisitos, para ayudar a reconocer la importancia que tiene una especificación y administración adecuada de los requerimientos de los usuarios. Es el proceso donde los interesados en un sistema de software descubren, revelan y entienden sus requerimientos, evidenciando la necesidad de trabajar con el cliente debido a que sus requisitos abarcan las tareas que se necesitan para llevar a cabo el desarrollo del sistema.

En la identificación de los requisitos de la propuesta de solución se utilizaron las siguientes técnicas:

Entrevistas

La entrevista es de gran utilidad para obtener información cualitativa como opiniones, o descripciones subjetivas de actividades. Es una técnica muy utilizada, y requiere una mayor preparación y experiencia por parte del analista. Debe quedar claro que no basta con hacer preguntas para obtener toda la información necesaria pues es muy importante la forma en que se plantea la conversación y la relación que se establece en la entrevista (Guerra, 2017).

Escenarios

Estos se utilizan para documentar el comportamiento del sistema cuando se le presentan eventos específicos. Cada evento de interacción distinto, o la selección de un servicio del sistema, se documentan como un escenario de eventos distinto. Los escenarios de eventos incluyen una descripción del flujo de datos y las acciones del sistema, y documenta las excepciones que puedan surgir.

Los Casos de Uso son una técnica que se basa en escenarios para la obtención de requerimientos. Actualmente se han convertido en una técnica fundamental que se utiliza para analizar y describir modelos de sistemas orientados a objetos. En su forma más simple, un caso de uso identifica a los actores involucrados en una interacción y nombra al tipo de ésta.

Desarrollo de Prototipos

Los prototipos suelen consistir en versiones reducidas, demos o conjuntos de pantallas (que no son totalmente operativos) de la aplicación pedida. Los prototipos de sistema permiten a los usuarios experimentar para ver cómo éste ayuda a su trabajo. Además de permitir a los usuarios mejorar las especificaciones de requerimientos (Guerra, 2017).

2.2.1 Requisitos Funcionales

Los requerimientos funcionales describen cualquier actividad que el sistema debe realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones. Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a ser desempeñados por el sistema y otros requerimientos de negocio, cumplimiento, seguridad u otra índole (PMO-Informatica, 2017). A continuación, se muestran los requisitos funcionales:

RF 1: Generar reportes de Composición del Personal.

RF 2: Exportar reportes de Composición del Personal.

RF 3: Filtrar información del reporte sobre Composición del Personal.

RF 4: Buscar información del reporte sobre Composición del Personal.

RF 5: Generar reportes de Datos sobre el Personal Docente y no docente.

RF 6: Exportar reportes de Datos sobre el Personal Docente y no docente.

RF 7: Filtrar información de Datos sobre el Personal Docente y no docente.

RF 8: Buscar información del reporte de Datos sobre el Personal Docente y no docente.

RF 9: Generar reportes de Movimiento de Personal.

RF 10: Exportar reportes de Movimiento de Personal.

RF 11: Filtrar información del reporte de Movimiento de Personal.

RF 12: Buscar información del reporte sobre el Movimiento de Personal.

2.2.2 Especificación de requisitos funcionales

N°	Nombre	Descripción	Prioridad	Complejidad	Referencias cruzadas
RF1.	Generar reportes de Composición del Personal.	El sistema debe generar reportes de la composición de personal seleccionado: -Provincia –Municipio –Estructura -Nivel -Cantidad de profesores por asignaturas -Cantidad de maestros y profesores - Cantidad de maestros y profesores por tipo de contrato - Cantidad de maestros y profesores por tipo de contrato Y las siguientes opciones: -Filtrar –Buscar -Exportar a PDF -Exportar a XLS –Cerrar	Alta	Mediana	FORTES_ Levantamiento_AKADEMOS_MINED_Descripcion_de_requisitos_por_proceso_Exportar_a_PDF_reporte_Total_de_trabajadores.odt
RF2.	Exportar reportes de Composición del Personal.	El sistema debe mostrar un listado de la composición de persona, mostrando los siguientes datos: -Provincia -Municipio -Cantidad de profesores -Cantidad de maestros -Cantidad de profesores por asignaturas Y las siguientes opciones: -Exportar a PDF -Exportar a XLS -Aceptar	Baja	Baja	FORTES_ Levantamiento_AKADEMOS_MINED_Descripcion_de_requisitos_por_proceso_Exportar_a_PDF_reporte_Total_de_trabajadores.odt
RF3.	Filtrar información del reporte sobre Composición del Personal.	El sistema debe permitir filtrar información del reporte sobre composición del personal seleccionado mostrando los siguientes datos: -Categoría -Estado -Cuadro -Cantidad de trabajadores por tipo de centro educacional - Cantidad de trabajadores por educación	Baja	Baja	FORTES_ Levantamiento_AKADEMOS_MINED_Descripcion_de_requisitos_por_proceso_Exportar_a_PDF_reporte_Total_de_trabajadores.odt

		- Cantidad de trabajadores por sexo Y las siguientes opciones: -Cerrar -Eliminar			
RF4.	Buscar información del reporte sobre Composición del Personal.	El sistema debe permitir la búsqueda de la información del reporte sobre Composición del personal solicitando los siguientes datos: -Cantidad de profesores por asignatura -Cantidad de maestros y profesores -Tipo de contrato Y las siguientes opciones: -Anterior –Siguiente –Aceptar -Cancelar	Baja	Baja	FORTES_ Levantamiento_AKADEMOS_MINED_Descripcion_de_requisitos_por_proceso_Exportar_a_PDF_reporte_Total_de_trabajadores.odt
RF5.	Generar reportes de Datos sobre el Personal Docente y no docente.	El sistema debe permitir generar los datos del personal docente y no docente mostrando los siguientes: -Fecha -Nivel -Estructura -Total de trabajadores -Datos del personal docente <ul style="list-style-type: none"> • Total trabajadores por categoría ocupacional • Total de trabajadores por categoría • Total de trabajadores por estado -Datos del personal no docente <ul style="list-style-type: none"> • Total de trabajadores por estado Y las siguientes opciones: -Aceptar –Filtrar –Buscar -Exportar a PDF -Exportar a XLS	Alta	Mediana	FORTES_Desarrollo_AKADEMOS_MINED_Descripcion_de_procesos_de_negocio_Organizacion_del_personal_docente_y_no_docente.odt
RF6.	Exportar reportes de Datos sobre el Personal Docente y no docente.	El sistema debe mostrar los datos del personal docente y no docente en formato PDF y XLS mostrando los siguientes: -Total de trabajadores -Fecha -Nivel –Estructura -Total trabajadores docentes por categoría ocupacional -Total de trabajadores docentes por categoría -Total de trabajadores docentes por estado -Total de trabajadores no docentes por estado Y las siguientes opciones:	Baja	Baja	FORTES_Desarrollo_AKADEMOS_MINED_Descripcion_de_procesos_de_negocio_Organizacion_del_personal_docente_y_no_docente.odt

		-Aceptar –Cerrar			
RF7.	Filtrar información de Datos sobre el Personal Docente y no docente.	El sistema debe mostrar la información a filtrar sobre el personal docente y no docente seleccionado mostrando los siguientes datos: -Estado –Cuadro –Miembro Consejo Dirección -Docente <ul style="list-style-type: none"> • Trabajadores por estado Y las siguientes opciones: -Cerrar	Baja	Baja	FORTES_Desarrollo_AKADEMOS_MINED_Descripcion_de_procesos_de_negocio_Organizacion_del_personal_docente_y_no_docente.odt
RF8.	Buscar información del reporte de Datos sobre el Personal Docente y no docente.	El sistema debe permitir buscar información del reporte de datos sobre el personal docente y no docente utilizando los siguientes datos: -Docente <ul style="list-style-type: none"> • Trabajadores docentes -No Docente <ul style="list-style-type: none"> • Estado Y las siguientes opciones: -Anterior –Siguiente –Aceptar -Cancelar	Baja	Baja	FORTES_Desarrollo_AKADEMOS_MINED_Descripcion_de_procesos_de_negocio_Organizacion_del_personal_docente_y_no_docente.odt
RF9.	Generar reportes de Movimiento de Personal.	El sistema debe permitir generar reportes de movimiento de personal utilizando los siguientes datos: -Fecha inicio -Fecha fin -Nivel -Estructura -Altas -Bajas Y las siguientes opciones: -Cancelar –Aceptar	Alta	Mediana	FORTES_Levantamiento_AKADEMOS_MINED_Descripcion_de_requisitos_por_proceso_Exportar_a_PDF_reporte_movimiento_del_personal.odt
RF10.	Exportar reportes de Movimiento de Personal.	El sistema debe permitir exportar reportes de personal utilizando los siguientes datos: -Resumen -Educación -Tipo de centro educacional -Periodo lectivo -Responsabilidad Y las siguientes opciones: -Cancelar –Aceptar	Baja	Baja	FORTES_Levantamiento_AKADEMOS_MINED_Descripcion_de_requisitos_por_proceso_Exportar_a_PDF_reporte_movimiento_del_personal
RF11.	Filtrar información del reporte de Movimiento de Personal.	El sistema debe permitir Filtrar información del reporte de Movimiento de Personal utilizando los siguientes datos: -Provincia -Municipio -Tipo de centro educacional Y las siguientes opciones: -Cancelar	Baja	Baja	FORTES_Levantamiento_AKADEMOS_MINED_Descripcion_de_requisitos_por_proceso_Exportar_a_PDF_reporte_movimiento_del_personal.odt

RF12.	Buscar información del reporte sobre el Movimiento de Personal.	El sistema debe permitir buscar información del reporte sobre el movimiento de personal utilizando los siguientes datos: -Alta <ul style="list-style-type: none"> • Tipo de Alta • Procedencia • Educación -Baja <ul style="list-style-type: none"> • Tipo de Baja • Procedencia • Educación Y las siguientes opciones: -Anterior –Siguiente –Cancelar -Aceptar	Baja	Baja	FORTES_Levantamiento_AKADMOS_MINED_Descripcion_de_requisitos_por_proceso_Exportar_a_PDF_reporte_movimiento_del_personal.odt
-------	---	---	------	------	---

Tabla 1: Descripción de Requisitos Funcionales

2.2.3 Requisitos no Funcionales

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc. A continuación, se muestran los requisitos no funcionales identificados para la propuesta de solución.

RNF 1- Usabilidad: el componente debe contar con una adecuada organización de las acciones permitiendo la fácil interacción con el usuario.

RNF 2- Confidencialidad: existencia de distintos roles que establezcan que la información solo sea vista por aquellos usuarios que posean los privilegios suficientes; restringir la ejecución de acciones a usuarios sin credenciales que intenten acceder a las mismas y la verificación de que el usuario esté autenticado antes de acceder a cualquier operación.

RNF 3- Interfaces externas: la interfaz debe ser simple, con una navegación por menú donde se pueda identificar de manera rápida las acciones a llevar. De este modo se logra una navegación efectiva en el módulo. Los mensajes deben ser claros, sin ambigüedades y con buena ortografía.

2.2.4 Descripción del requisito Generar Movimiento del personal

Con el objetivo de presentar los Requisitos por proceso de un modo más sólido y por ende más entendible, se realizó la descripción de cada Requisito por proceso. Se definieron secciones para descri-

bir detalladamente cada requisito y las acciones que se deben llevar a cabo por el usuario para cumplir con esa funcionalidad en el sistema.

A continuación, se muestra en la Tabla 2 una de las descripciones de requisitos diseñadas, para consultar otras descripciones remitirse al Anexo 2.

Precondiciones	Debe haber un usuario autenticado con permiso a la funcionalidad.
Flujo de eventos	
Flujo básico Generar reporte del movimiento de personal	
•	Se selecciona Estadísticas / AF Personal / Movimiento.
1.	Se introducen los datos solicitados.
2.	Se selecciona el Botón Generar.
3.	Se muestra un dashboard con los gráficos de la información general del reporte y los listados Alta y Baja.
4.	Se selecciona una de las siguientes opciones: <ul style="list-style-type: none"> • Ícono Exportar a PDF. Ver FORTES_Levantamiento_AKADEMOS_MINED_DRP_Exportar_PDF_reporte_movimiento_del_personal.odt. • Ícono Exportar a XLS. Ver FORTES_Levantamiento_AKADEMOS_MINED_DRP_Exportar_XLS_reporte_movimiento_del_personal.odt. • Botón Filtrar (General). Ver Flujo alternativo 5.a. • Botón Buscar (Altas). Ver Flujo alternativo 5.b. • Botón Filtrar (Altas). Ver Flujo alternativo 5.c. • Botón Buscar (Bajas). Ver Flujo alternativo 5.d. • Botón Filtrar (Bajas). Ver Flujo alternativo 5.e.
Post-condiciones	
1	Se generó el reporte movimiento del personal.
Flujo alternativo 5.a Filtrar.	
1	Se muestra el panel de filtros, con los filtros que se seleccionen y las opciones siguientes: Filtros: <ul style="list-style-type: none"> - Provincia - Municipio - Tipo de centro educacional Opciones: <ul style="list-style-type: none"> • Mostrar/Ocultar: Ver Flujo alternativo 2.f. • Icono Cerrar filtros (imagen de cruz): Ver Flujo alternativo 2.g. • Icono Cerrar filtro interno (imagen de signo menos): Ver Flujo alternativo 2.h.
2	Se seleccionan los valores en los filtros.
3	Se muestran los elementos del Movimiento de personal que coincidan con los valores se-

	leccionados en los filtros. - En caso de que no exista ninguna coincidencia en el Movimiento de personal con los filtros introducidos, se muestra el mensaje: "No hay datos".	
4	Se pasa al Paso 5 del Flujo básico.	
Flujo alternativo 5.b Buscar (Altas).		
1	Se introduce parámetro para la búsqueda (Tipo de alta, procedencia, educación)	
2	Se selecciona el botón Buscar .	
3	Se muestran los elementos del Movimiento de personal que coincidan con los parámetros introducidos. - En caso de que no exista ninguna coincidencia en el Movimiento de personal con los parámetros introducidos, se muestra el mensaje: " No hay datos".	
4	Se pasa al Paso 5 del Flujo básico.	
Flujo alternativo 5.c Filtrar (Altas).		
5	Se muestra el panel de filtros, con los filtros que se seleccionen y las opciones siguientes: Filtros: - Tipo de centro educacional Opciones: • Mostrar/Ocultar: Ver Flujo alternativo 2.i. • Icono Cerrar filtros (imagen de cruz): Ver Flujo alternativo 2.j. • Icono Cerrar filtro interno (imagen de signo menos): Ver Flujo alternativo 2.k.	
6	Se seleccionan los valores en los filtros.	
7	Se muestran los elementos del Movimiento de personal que coincidan con los valores seleccionados en los filtros. - En caso de que no exista ninguna coincidencia en el Movimiento de personal con los filtros introducidos, se muestra el mensaje: "No hay datos".	
8	Se pasa al Paso 5 del Flujo básico.	
Flujo alternativo 5.d Buscar (Bajas).		
5	Se introduce parámetro para la búsqueda (Tipo de alta, procedencia, educación)	
6	Se selecciona el botón Buscar .	
7	Se muestran los elementos del Movimiento de personal que coincidan con los parámetros introducidos. - En caso de que no exista ninguna coincidencia en el Movimiento de personal con los parámetros introducidos, se muestra el mensaje: " No hay datos".	
8	Se pasa al Paso 5 del Flujo básico.	
Flujo alternativo 5.e Filtrar (Bajas).		
9	Se muestra el panel de filtros, con los filtros que se seleccionen y las opciones siguientes: Filtros: - Tipo de centro educacional Opciones:	

	<ul style="list-style-type: none"> • Mostrar/Ocultar: Ver Flujo alternativo 2.i. • Icono Cerrar filtros (imagen de cruz): Ver Flujo alternativo 2.m. • Icono Cerrar filtro interno (imagen de signo menos): Ver Flujo alternativo 2.n. 	
10	Se seleccionan los valores en los filtros.	
11	Se muestran los elementos del Movimiento de personal que coincidan con los valores seleccionados en los filtros. <ul style="list-style-type: none"> - En caso de que no exista ninguna coincidencia en el Movimiento de personal con los filtros introducidos, se muestra el mensaje: "No hay datos". 	
12	Se pasa al Paso 5 del Flujo básico.	
Post-condiciones		
1	N/A	
Validaciones		
1	Campo de selección Fecha inicio	Muestra la estructura seleccionada en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_movimiento_del_personal.odt.
1	Campo de selección Fecha fin	Muestra la fecha inicio y la fecha fin seleccionadas en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_movimiento_del_personal.odt.
2	Campo de selección Nivel	Muestra el Nivel seleccionado en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_movimiento_del_personal.odt.
3	Campo de selección Estructura	Muestra el municipio de la estructura registrada en Estructura y seleccionada en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_movimiento_del_personal.odt.
4	Componente Filtrar	Muestra los filtros generales aplicados en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_movimiento_del_personal.odt.
5	Filtro-Provincia	Muestra la provincia de la estructura registrada en Estructura y seleccionada en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_movimiento_del_personal.odt.
6	Tabla filtros	Muestra los filtros generales aplicados en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_movimiento_del_personal.odt. Atributos que se muestran como columnas de la tabla: <ul style="list-style-type: none"> • Filtro y valor Se muestran organizados por el orden en que fueron aplicados.
7	Tabla Movimiento del personal por tipo de centro educacional	Atributos que se muestran como columnas de la tabla: <ul style="list-style-type: none"> • Concepto • Total • Tipo de centro educacional (una columna por cada tipo de centro educacional)

8	Tabla Movimiento del personal por educación	Atributos que se muestran como columnas de la tabla: <ul style="list-style-type: none"> • Concepto • Total • Educación (una columna por cada Educación)
9	Tabla Movimiento del personal por período lectivo	Atributos que se muestran como columnas de la tabla: <ul style="list-style-type: none"> • Concepto • Total • Período lectivo (una columna por cada Período lectivo)
10	Tabla Movimiento del personal por responsabilidad	Atributos que se muestran como columnas de la tabla: <ul style="list-style-type: none"> • Concepto • Total • Responsabilidad (una columna por cada Responsabilidad)
11	Fila Total de trabajadores al inicio del período	Muestra la cantidad de trabajadores de alta de la estructura seleccionada y sus subordinadas en la fecha_inicio del período seleccionado. Se calcula de la siguiente forma: $y = \sum_{n=1}^k f(a, b)_n$ $f(a, b) = \begin{cases} 1, & \sum_{n=1}^m a_n - \sum_{n=1}^p b_n \geq 1 \\ 0, & \sum_{n=1}^m a_n - \sum_{n=1}^p b_n = 0 \end{cases}$ <ul style="list-style-type: none"> • $y \in \mathbb{N}$, total de trabajadores al principio del período • $k \in \mathbb{N}$, cantidad de personas registradas en Persona que tienen al menos un trámite de alta registrado en Tramite_alta con <i>fecha_alta</i> ≤ <i>fecha de inicio del periodo seleccionado por el usuario</i> • $f(a, b) \in \{0 (baja); 1(alta)\}$, situación laboral de la persona • $m \in \mathbb{N}$, cantidad de trámites registrados en Alta_Trabajador de un trabajador registrado en Persona y con <i>fecha_alta</i> ≤ <i>fecha de inicio del periodo seleccionado por el usuario</i> • $a \in \mathbb{N}$, trámite de alta registrado en Alta_Trabajador de un trabajador registrado en Persona y con <i>fecha_alta</i> ≤ <i>fecha de inicio del periodo seleccionado por el usuario</i>

		<ul style="list-style-type: none"> • $p \in \mathbb{N}$, cantidad de trámites registrados en Baja_Trabajador de una persona registrada en Persona y con $fecha_baja \leq fecha\ de\ inicio\ del\ período\ seleccionado\ por\ el\ usuario$ • $b \in \mathbb{N}$, trámite de baja registrado en Baja_Trabajador de un trabajador registrado en Persona y con $fecha_baja \leq fecha\ de\ inicio\ del\ período\ seleccionado\ por\ el\ usuario$
12	Fila Total de altas	Muestra la cantidad de trámites de alta registrados en Alta_Trabajador con $fecha\ de\ inicio\ del\ período\ seleccionado \leq fecha_alta \leq fecha\ de\ fin\ del\ período\ seleccionado$
13	Filas tipo de alta	Muestra la cantidad de trámites de alta registrados en Alta_Trabajador de cada tipo_alta
14	Total de trabajadores al final del período	<p>Muestra la cantidad de trabajadores de alta de la estructura seleccionada y sus subordinadas en la fecha de fin del período seleccionado. Se calcula de la siguiente forma:</p> $z = y + \sum_{n=1}^q f(c, d)_n$ $f(c, d) = \begin{cases} 1, & \sum_{n=1}^h c_n - \sum_{n=1}^j d_n \geq 1 \\ 0, & \sum_{n=1}^h c_n - \sum_{n=1}^j d_n = 0 \end{cases}$ <p>Donde sea:</p> <ul style="list-style-type: none"> • z el total de trabajadores al final del período • y el total de trabajadores al inicio del período (ver validación Total de trabajadores al inicio del periodo) • $q \in \mathbb{N}$, cantidad de personas registradas en Persona que tienen al menos un trámite de alta registrado en Tramite_alta con $fecha\ de\ inicio\ del\ período\ seleccionado\ por\ usuario < fecha_alta \leq fecha\ de\ fin\ del\ período\ seleccionado\ por\ usuario$ • $f(c, d) \in \{0 (baja); 1 (alta)\}$, situación laboral de la persona • $h \in \mathbb{N}$, cantidad de trámites registrados en Alta_Trabajador de un trabajador registrado en Persona

		<p>y con <i>fecha de inicio del período seleccionado por usuario < fecha_alta ≤</i> <i>fecha de fin del período seleccionado por usuario</i></p> <ul style="list-style-type: none"> • $c \in \mathbb{N}$, trámite de alta registrado en Alta_Trabajador de un trabajador registrado en Persona y con <i>fecha de inicio del período seleccionado por usuario < fecha_alta ≤</i> <i>fecha de fin del período seleccionado por usuario</i>
15	Inactivos	<p>Muestra la cantidad de trabajadores de alta con estado_trabajador inactivo de la estructura seleccionada y sus subordinadas en la fecha de fin del período seleccionado. Se calcula de la siguiente forma:</p> $e = \sum g$ <p>Donde sea:</p> <ul style="list-style-type: none"> • \in , el total de trabajadores inactivos al final del período • \in persona registrada en Persona con al menos un trámite de prestación económica registrado en Prestacion_Economica con <i>ainicio ≤ fecha de fin del período seleccionado por usuario < fecha_fin</i>
16	Activos	<p>Muestra la cantidad de trabajadores de alta con estado_trabajador activo u otras funciones de la estructura seleccionada y sus subordinadas en la fecha de fin del período seleccionado. Se calcula de la siguiente forma:</p> $c = z - e$ <p>Donde sea:</p> <ul style="list-style-type: none"> • \in el total de trabajadores activos al final del período • \in el total de trabajadores inactivos al final del período (ver validación Inactivos) • \in el total de trabajadores al final del período (ver validación Total de trabajadores al final del período)
17	Columna Total	Muestra la suma de los valores numéricos de cada fila.
18	Gráfico de pastel Movimiento del personal	<p>Muestra la cantidad de movimientos del personal en el período seleccionado por el usuario.</p> <p>Atributos que se muestran en el gráfico:</p> <ul style="list-style-type: none"> • Altas (ver validación de Total de altas) • Bajas (ver validación de Total de bajas)
19	Gráfico de pastel Total de trabajadores	<p>Muestra la cantidad de trabajadores al final del período seleccionado por el usuario.</p> <p>Atributos que se muestran en el gráfico:</p> <ul style="list-style-type: none"> • Activos (ver validación de Activos)

		<ul style="list-style-type: none"> Inactivos (ver validación de Inactivos)
20	Gráfico de pastel Altas	<p>Muestra la cantidad de trámites de alta registrados en el período seleccionado por el usuario.</p> <p>Atributos que se muestran en el gráfico:</p> <ul style="list-style-type: none"> Tipo de alta (ver validación de Filas tipo de alta)
21	Gráfico de pastel de un tipo de alta con procedencias asociadas	<p>Muestra la cantidad de trámites de alta de un tipo de alta con procedencias asociadas y registrados en el período seleccionado por el usuario.</p> <p>Atributos que se muestran en el gráfico:</p> <ul style="list-style-type: none"> Procedencia (ver validación Filas procedencia)
22	Gráfico de pastel Total de trabajadores por educación	<p>Muestra por educación la cantidad de trabajadores al final del período seleccionado por el usuario (ver validación Tabla movimiento del personal por educación).</p> <p>Atributos que se muestran en el gráfico:</p> <ul style="list-style-type: none"> Educación
23	Gráfico de pastel Altas por educación	<p>Muestra por educación la cantidad de trámites de alta registrados en el período seleccionado por el usuario (ver validación Tabla movimiento del personal por educación).</p> <p>Atributos que se muestran en el gráfico:</p> <ul style="list-style-type: none"> Educación
24	Gráfico de pastel de un tipo de alta con procedencias asociadas por educación	<p>Muestra por educación la cantidad de trámites de alta de un tipo de alta con procedencias asociadas y registrados en el período seleccionado por el usuario (ver validación Tabla movimiento del personal por educación).</p> <p>Atributos que se muestran en el gráfico:</p> <ul style="list-style-type: none"> Educación
25	Gráfico de pastel Total de trabajadores por tipo de centro educacional	<p>Muestra por tipo de centro educacional la cantidad de trabajadores al final del período seleccionado por el usuario (ver validación Tabla movimiento del personal por tipo de centro educacional).</p> <p>Atributos que se muestran en el gráfico:</p> <ul style="list-style-type: none"> Tipo de centro educacional
26	Gráfico de pastel Altas por tipo de centro educacional	<p>Muestra por tipo de centro educacional la cantidad de trámites de alta registrados en el período seleccionado por el usuario (ver validación Tabla movimiento del personal por tipo de centro educacional).</p> <p>Atributos que se muestran en el gráfico:</p> <ul style="list-style-type: none"> Tipo de centro educacional
27	Gráfico de pastel de un tipo de alta con procedencias asociadas por tipo de centro educacional	<p>Muestra por tipo de centro educacional la cantidad de trámites de alta de un tipo de alta con procedencias asociadas y registrados en el período seleccionado por el usuario (ver validación Tabla movimiento del personal por tipo de centro educacional).</p> <p>Atributos que se muestran en el gráfico:</p> <ul style="list-style-type: none"> Tipo de centro educacional
28	Gráfico de pastel Bajas por tipo de centro educacional	<p>Muestra por tipo de centro educacional la cantidad de trámites de baja registrados en el período seleccionado por el usuario (ver validación Tabla movimiento del personal por tipo de centro</p>

		educacional). Atributos que se muestran en el gráfico: <ul style="list-style-type: none"> • Tipo de centro educacional
29	Gráfico de pastel de un tipo de baja con motivo de baja por tipo de centro educacional	Muestra por tipo de centro educacional la cantidad de trámites de baja de un tipo de baja con motivos de baja asociados y registrados en el período seleccionado por el usuario (ver validación Tabla movimiento del personal por tipo de centro educacional). Atributos que se muestran en el gráfico: <ul style="list-style-type: none"> • Tipo de centro educacional
30	Gráfico de pastel Total de trabajadores por período lectivo	Muestra por período lectivo la cantidad de trabajadores al final del período seleccionado por el usuario (ver validación Tabla movimiento del personal por período lectivo). Atributos que se muestran en el gráfico: <ul style="list-style-type: none"> • Período lectivo
31	Gráfico de pastel Altas por período lectivo	Muestra por período lectivo la cantidad de trámites de alta registrados en el período seleccionado por el usuario (ver validación Tabla movimiento del personal por período lectivo). Atributos que se muestran en el gráfico: <ul style="list-style-type: none"> • Período lectivo
32	Gráfico de pastel de un tipo de alta con procedencias asociadas por período lectivo	Muestra por período lectivo la cantidad de trámites de alta de un tipo de alta con procedencias asociadas y registrados en el período seleccionado por el usuario (ver validación Tabla movimiento del personal por período lectivo). Atributos que se muestran en el gráfico: <ul style="list-style-type: none"> • Período lectivo
33	Gráfico de pastel Bajas por período lectivo	Muestra por período lectivo la cantidad de trámites de baja registrados en el período seleccionado por el usuario (ver validación Tabla movimiento del personal por período lectivo). Atributos que se muestran en el gráfico: <ul style="list-style-type: none"> • Período lectivo
34	Gráfico de pastel de un tipo de baja con motivo de baja por período lectivo	Muestra por período lectivo la cantidad de trámites de baja de un tipo de baja con motivos de baja asociados y registrados en el período seleccionado por el usuario (ver validación Tabla movimiento del personal por período lectivo). Atributos que se muestran en el gráfico: <ul style="list-style-type: none"> • Período lectivo
35	Gráfico de pastel Total de trabajadores por responsabilidad	Muestra por responsabilidad la cantidad de trabajadores al final del período seleccionado por el usuario (ver validación Tabla movimiento del personal por responsabilidad). Atributos que se muestran en el gráfico: <ul style="list-style-type: none"> • Responsabilidad
36	Gráfico de pastel Altas por responsabilidad	Muestra por responsabilidad la cantidad de trámites de alta registrados en el período seleccionado por el usuario (ver validación Tabla movimiento del personal por responsabilidad). Atributos que se muestran en el gráfico: <ul style="list-style-type: none"> • Responsabilidad

Conceptos	Persona	Visibles en la interfaz: - N/A Utilizados internamente: - primer_nombre - segundo_nombre - primer_apellido - segundo_apellido
	Registró_Trabajador	Visibles en la interfaz: - N/A Utilizados internamente: - trabajador - estructura - responsabilidad - alta
	Estructura	Visibles en la interfaz: - N/A Utilizados internamente: - nombre
	Tipo_Estructura	Visibles en la interfaz: - N/A Utilizados internamente: - nombre - nivel
	Centro_Educacional	Visibles en la interfaz: - N/A Utilizados internamente: - nombre - tipo_centro_educacional - configuración_especialidad
	Configuracion_Especialidad	Visibles en la interfaz: - N/A Utilizados internamente: - educación
	Educación	Visibles en la interfaz: - N/A Utilizados internamente: - educación
	Tipo_Centro_Educacional	Visibles en la interfaz: - N/A Utilizados internamente: - nombre
	Responsabilidad	Visibles en la interfaz: - N/A Utilizados internamente: - nombre_responsabilidad
	Período_Lectivo	Visibles en la interfaz: - N/A Utilizados internamente: - período_lectivo
	Trámite_Trabajador	Visibles en la interfaz: - N/A Utilizados internamente: - trabajador - trámite - fecha_fin_trámite
	Alta_Trabajador	Visibles en la interfaz: - N/A Utilizados internamente: - trámite_trabajador - fecha_alta - tipo_alta - procedencia
	Tipo_Alta	Visibles en la interfaz: - N/A Utilizados internamente: - tipo_alta
	Procedencia	Visibles en la interfaz: - N/A Utilizados internamente: - procedencia
	Prestacion_Económica	Visibles en la interfaz: - N/A Utilizados internamente: - fecha_inicio - fecha_fin
	Estado_Trabajador	Visibles en la interfaz: - N/A Utilizados internamente: - estado_trabajador
Requisitos especiales	N/A	

Asuntos pendientes N/A

Prototipo elemental de interfaz gráfica de usuario

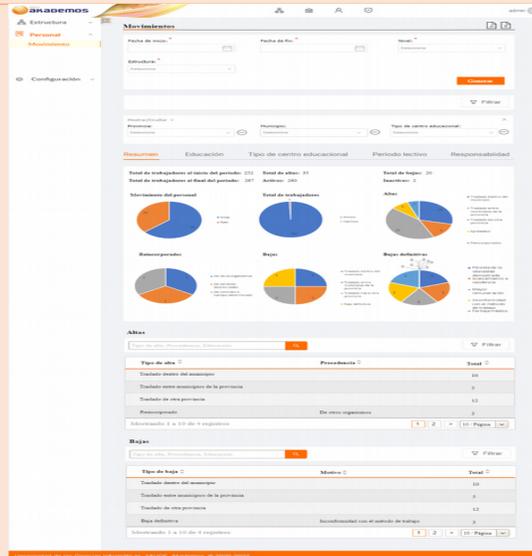


Tabla 2: Descripción del Requisito Generar Movimiento del personal

2.2.5 Descripción de los requisitos no funcionales

Con el objetivo de presentar los Requisitos no funcionales de forma más entendible. Se definieron secciones para describir detalladamente cada requisito no funcional que tienen que cumplir el sistema. A continuación, se muestra en la Tabla 3 una de las descripciones de requisitos diseñadas, para consultar otras descripciones remitirse al Anexo 2.

RNF 1. Requisito de Usabilidad 1.1. El componente debe contar con una adecuada organización de las acciones permitiendo la fácil interacción con el usuario.

Atributo de Calidad	Usabilidad
Sub-atributos/Sub-características	Comportamiento temporal
Objetivo	Proporcionar una adecuada organización de las acciones.
Origen	Humano
Artefacto	Sistema
Entorno	El sistema está funcionando correctamente

Tabla 3: Descripción del Requisito no Funcional Usabilidad 1.1

2.3 Descripción de la arquitectura y el diseño

La arquitectura de un software es la estructura u organización de un sistema que incluye los componentes del mismo, las propiedades visibles externas de esos componentes y las relaciones que existen entre ellos. Hace referencia a la estructura y la relación entre las diferentes partes de un software y sus propiedades visibles externas. Su principal objetivo radica en ofrecer cierta calidad al sistema de administración de datos, a partir de su desempeño, ahorro de tiempo, su disponibilidad y usabilidad, la capacidad de modificarse y adecuarse a las nuevas necesidades del sistema, entre otros atributos de calidad (Keepcoding, 2022).

2.3.1 Arquitectura Orientada a Servicios

La arquitectura orientada a servicios (SOA, por sus siglas en inglés) es un método de desarrollo de software que utiliza componentes de software llamados servicios para crear aplicaciones empresariales. Cada uno de estos servicios brinda una capacidad empresarial y, además, pueden comunicarse también con el resto de servicios mediante diferentes plataformas y lenguajes. Los desarrolladores usan SOA para reutilizar servicios en diferentes sistemas o combinar varios servicios independientes para realizar tareas complejas.

Existen diferentes componentes de la arquitectura orientada a servicios dentro de los que se encuentran:

Servicios: son los componentes básicos de la SOA. Pueden ser privados (disponibles únicamente para los usuarios internos de una organización) o públicos (accesibles para todos en Internet), donde cada servicio individual tiene tres características principales:

1. **Implementación de servicios:** es el código que crea la lógica para realizar la función de servicio específica, como la autenticación de usuarios o el cálculo de una factura.
2. **Contrato del servicio:** este define la naturaleza del servicio y sus condiciones y términos asociados, como los prerequisites para utilizar el servicio, su costo y la calidad del servicio proporcionado.
3. **Interfaz del servicio:** en SOA, otros servicios o sistemas se comunican con un servicio a través de su interfaz. Esta interfaz define la manera en que se puede invocar al servicio para llevar a cabo actividades o intercambiar datos.

Proveedor de servicios: crea, mantiene y proporciona uno o más servicios que otros pueden utilizar. Las organizaciones pueden crear sus propios servicios o adquirirlos de proveedores de servicios externos.

Consumidor de servicios: solicita al proveedor de estos poner en marcha un servicio específico. Puede ser un sistema completo, aplicación u otro servicio. El contrato de servicio especifica las reglas que el proveedor y el consumidor de servicios deben seguir al momento de interactuar entre sí.

Registro de servicios: es un directorio de servicios disponibles accesible a través de redes. Almacena documentos descriptivos sobre el servicio que pertenecen a los proveedores de servicios.

¿Cuáles son los beneficios de la arquitectura orientada a servicios?

La arquitectura orientada a servicios ofrece varios beneficios por encima de las arquitecturas monolíticas tradicionales, donde todos los procesos se ponen en marcha como una unidad única. Algunos de los principales beneficios que tiene son:

- **Reducción del plazo de comercialización:** Los desarrolladores pueden crear aplicaciones en menos tiempo con SOA en lugar de escribir código y llevar a cabo integraciones desde cero.
- **Mantenimiento eficiente:** Es más fácil crear, actualizar y corregir errores en servicios pequeños que en bloques grandes de código en aplicaciones monolíticas.
- **Excelente capacidad de adaptación:** La SOA se adapta de mejor manera a los avances tecnológicos. Puede modernizar sus aplicaciones de forma eficiente y rentable.

2.3.2 Patrones de Diseño

Patrón de diseño: es una solución estandarizada, probada y repetible a un problema común en el diseño de software. Es una descripción o plantilla sobre cómo resolver un problema que se puede utilizar en muchas situaciones diferentes.

El uso de patrones de diseño permite acelerar el proceso de desarrollo al proporcionar paradigmas probados y comprobados en el diseño de software. Estandarizar las soluciones a problemas comunes permite también mejorar la comunicación entre desarrolladores ya que se emplean nombres bien co-

nocidos y entendidos a la hora de describir un problema de diseño y como abordar su solución (Patrones de Diseño GOF , 2013)

Patrones GRASP: significan según sus siglas en inglés *General Responsibility Assignment Software Patterns*, es decir, patrones generales de asignación de responsabilidades. Se utilizan en la programación orientada a objetos, con lo que el punto de partida para aplicarlos siempre será usar POO¹⁰. Cómo definir estos objetos/clases será la aplicación de cada patrón (Patrones de Diseño GOF , 2013).

Experto: propone asignar una responsabilidad a la clase que cuenta con la información necesaria para cumplirla. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo. El uso del patrón se evidencia en la clase *Person*, que es la que cuenta con la información necesaria para cumplir las responsabilidades sobre los elementos de negocio referente a la parte del personal.

Creador: expresa la asignación a una clase la responsabilidad de crear una instancia de otra.

¹⁰ Programación Orientada a Objeto

```

26
27
28 class Person(BaseModel):
29     second_name = models.CharField(max_length=100, null=True, blank=True, verbose_name='second name', db_column='segundo_nombre')
30     first_last_name = models.CharField(max_length=100, verbose_name='first last name', db_column='primer_apellido')
31     second_last_name = models.CharField(max_length=100, verbose_name='second last name', db_column='segundo_apellido')
32     ci = models.CharField(max_length=11,
33                           validators=[RegexValidator(r"^\d{11}$", message="The CI must have 11 characters."),
34                                       RegexValidator(r"^[0-9]{2}((0[1-9]|11|12)|(1[0-9])|(2[0-9])|(3[0-1]))[0-9]{5}$",
35                                                     message="The CI is not valid.")], db_column='carne')
36     ci_serie = models.CharField(max_length=50, null=True, blank=True, db_column='no_serie_ci')
37     date_of_birth = models.DateField(null=True, blank=True, verbose_name='date of birth', db_column='fecha_nacimiento')
38     photo = models.ImageField(upload_to=person_photo_storage_path, default=None, null=True, blank=True,
39                               verbose_name='photo', db_column='foto')
40
41     # CONTACT DATA
42     address = models.TextField(default='', null=True, blank=True, verbose_name='address', db_column='direccion')
43     address_province = models.ForeignKey(Province, on_delete=models.CASCADE, null=True, blank=True,
44                                         verbose_name='address province', db_column='direccion_provincia')
45     address_municipality = models.ForeignKey(Municipality, on_delete=models.CASCADE, null=True, blank=True,
46                                             verbose_name='address municipality', db_column='direccion_municipio')
47     address_popular_council = models.ForeignKey(PopularCouncilType, on_delete=models.CASCADE, blank=True, null=True,
48                                               verbose_name='address popular council', db_column='direccion_consejo_popular')
49     address_location = models.ForeignKey(LocationType, on_delete=models.CASCADE, null=True, blank=True,
50                                       verbose_name='address location', db_column='direccion_localidad')
51     email = models.EmailField(default='', null=True, blank=True, verbose_name='email', db_column='correo_electronico')
52     mobile_phone = models.CharField(
53         max_length=16,
54         blank=True,
55         null=True,
56         validators=[
57             RegexValidator(
58                 regex=r"^[+]?1?d{8,15}$",
59                 message="Phone number must be entered in the format '+123456789'. Up to 15 digits allowed."
60             ),
61             verbose_name='mobile phone', db_column='telefono_movil'
62         ])
63     phone = models.CharField(max_length=16, blank=True, null=True,
64                             validators=[
65             RegexValidator(regex=r"^[+]?1?d{8,15}$", message="Phone number must be entered in the format '+123456789'")
66         ], verbose_name='phone', db_column='telefono')

```

Fig. 5: Patrón Controlador

Controlador: define que se le debe asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. El uso del patrón se evidencia en las clases controladoras que se encargan de obtener los datos y enviarlos a las librerías y las vistas.

Alta cohesión: la cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una clase con baja cohesión hace muchas tareas no afines o un trabajo excesivo. Esto se evidencia en la clase auxiliar *EducationalWorker*.

```

14
15
16 class EducationalWorker(BaseAkModel):
17     """ It relates all the data about a person who is an educational worker"""
18     name = models.CharField(max_length=100, null=True, blank=True, editable=False, verbose_name=_('name'))
19     structure = models.ForeignKey(Structure, on_delete=models.CASCADE, verbose_name=_('structure'), db_column='estructura')
20     person = models.ForeignKey(Person, on_delete=models.CASCADE, verbose_name=_('person'), db_column='persona')
21     # person_type = models.OneToOneField(PersonType, on_delete=models.CASCADE, related_name='educational_worker')
22     teaching_category = models.ForeignKey(TeachingCategoryType, on_delete=models.SET_NULL, null=True,
23     verbose_name=_('teaching category'), db_column='categoria_enseñanza')
24     specialty = models.ForeignKey(TeacherSpecialityType, on_delete=models.SET_NULL, null=True, blank=True,
25     verbose_name=_('specialty'), db_column='especialidad')
26     teaching_status = models.ForeignKey(TeacherStatusType, on_delete=models.SET_NULL, null=True, blank=True,
27     verbose_name=_('teaching status'), db_column='estado_enseñanza')
28
29     # these three fields are not in the document
30     scientific_grade = models.CharField(max_length=100, null=True, blank=True, verbose_name=_('scientific grade'),
31     db_column='grado_cientifico')
32     research_category = models.CharField(max_length=100, null=True, blank=True, verbose_name=_('research category'),
33     db_column='categoria_investigativa')
34     profession = models.CharField(max_length=100, null=True, blank=True, verbose_name=_('profession'), db_column='profesion')
35
36     groups = models.ManyToManyField(TeachingGroup, through='EducationalWorkerGroup', related_name='teachers',
37     through_fields=('teacher', 'group'), verbose_name=_('groups'), db_column='grupos')
38
39     def __str__(self):
40         return f'{self.person} -- Structure: {self.structure_id}'
41
42     class Meta:
43         verbose_name = _('Educational Worker')
44         verbose_name_plural = _('Educational Workers')
45         indexes = [
46             models.Index(fields=['structure']),
47         ]
48         constraints = [
49             UniqueConstraint(fields=['structure', 'person'], name='uniq_educ_worker')
50         ]
51         db_table = f'{APP_LABEL.lower()}_tbr_trabajador_docente'
52
53
54
55

```

Fig. 6: Patrón Alta cohesión

Bajo acoplamiento: el acoplamiento es una medida de la fuerza con que una clase está conectada con otras clases. Una clase con bajo acoplamiento no depende de muchas clases. El patrón propone que cada clase debe tener un bajo grado de dependencia con otras clases en la medida de lo posible. Un ejemplo es la clase *PersonStructure* en su implementación solo contiene sus atributos y el acceso a los datos de la clase *Person*, esta clase no tiene instancia de más ninguna clase que refleje dependencias.

```

1 from django.db.models.constraints import UniqueConstraint
2 from structure.models import *
3 from django.db import models, transaction, IntegrityError
4 from django.utils.translation import gettext as _
5 from core.generics.generic import BaseAkModel
6 from .mdl_person import Person
7 from core.models import PersonCategoryType
8 from person.utils import APP_LABEL
9
10 class PersonStructure(BaseAkModel):
11     """ It relates the person with an structure """
12     name = models.CharField(max_length=100, null=True, blank=True, editable=False, verbose_name=_('name'))
13     person = models.ForeignKey(Person, on_delete=models.CASCADE, verbose_name=_('person'), db_column='persona')
14     category = models.ForeignKey(PersonCategoryType, on_delete=models.CASCADE, default=None, verbose_name=_('category'), db_column='categoria')
15     structure = models.ForeignKey(Structure, on_delete=models.CASCADE, verbose_name=_('structure'), db_column='estructura')
16     category_order = models.IntegerField(default=0, verbose_name=_('category_order'), db_column='orden_categoria')
17
18     def __str__(self):
19         return f'{self.category} --{self.person} -- Structure: {self.structure_id}'
20
21     class Meta:
22         verbose_name = _('Person Structure')
23         verbose_name_plural = _('Person Structures')
24         indexes = [
25             models.Index(fields=['structure']),
26         ]
27         constraints = [
28             UniqueConstraint(fields=['structure', 'person', 'category'], name='uniq_pers_per_struct')
29         ]
30         db_table = f'{APP_LABEL.lower()}_tbd_persona_estructura'
31
32
33
34
35
36
37
38
39
40
41
42
43

```

Fig. 7: Patrón Bajo acoplamiento

Patrones GOF

Los patrones GOF describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos (Patrones de Diseño GOF , 2013), entre los patrones identificados se encuentran:

- **Patrón de comportamiento:** comprenden la asignación de responsabilidades entre objetos y algoritmos. Estos no solo conciernen a los objetos y las clases sino también la comunicación entre estas, además caracterizan flujos de control complejos que son difíciles de seguir en tiempo de ejecución.
- **Mediador:** define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto.

- **Patrones de creación:** se encargan de la creación de instancias de los objetos. Abstraen la forma en que se crean los objetos, permitiendo tratar las clases a crear de forma genérica, dejando para después la decisión de que clase crear o cómo crearla.
- **Instancia única (*Singleton*):** garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Restringe la instanciación de una clase o valor de un tipo a un solo objeto.

Conclusiones del capítulo

A través del presente capítulo se describieron las técnicas para la obtención de requisitos posibilitando comprender, identificar y describir la exigencia de 12 requisitos funcionales y 3 no funcionales que deberá cumplir la solución. Ofreciendo una visión general de los reportes estadísticos del sistema mediante los artefactos descritos. La utilización de patrones de diseño y arquitectura permitió diseñar una solución robusta, flexible y escalable. Se puede concluir que, a partir del análisis, el diseño realizado y de los artefactos generados quedan sentadas las bases para la implementación y validación de la solución propuesta.

CAPÍTULO 3: Implementación y validación de la solución

En el presente capítulo se lleva a cabo la etapa de pruebas para verificar la calidad de la propuesta de solución, mostrando a su vez los resultados de las mismas y las validaciones realizadas con el propósito de garantizar que el sistema cumple con los requisitos establecidos.

3.1 Estándares de codificación

La adopción de estándares de estilo y codificación es de vital importancia para asegurar la calidad del software y su uso tiene innumerables ventajas tales como:

- Asegurar la legibilidad del código entre distintos programadores para facilitar la depuración del programa.
- Provee una guía para el encargado de mantenimiento/actualización del sistema, con código claro y bien documentado facilitando la portabilidad entre plataformas y aplicaciones.

A continuación, se presentan algunos de los estándares que se utilizaron:

- Se deben usar siempre dos líneas en blanco entre las secciones de un fichero fuente o entre las definiciones de clases e interfaces.
- Se debe usar siempre una línea en blanco en las siguientes circunstancias:
 - a) Entre métodos.
 - b) Entre las variables locales de un método y su primera sentencia.
 - c) Antes de un comentario de bloque o de un comentario de una línea.
 - d) Entre las distintas secciones lógicas de un método para facilitar la lectura.
- Se debe usar un espacio en blanco entre una palabra clave del lenguaje y un paréntesis.
- Respecto a las normas de inicialización, declaración y colocación de variables, constantes, clases y métodos. Todas las instancias y variables de clases o métodos empezarán con minúscula. Las palabras internas que lo forman, si son compuestas, empiezan con su primera letra en mayúsculas.

3.2 Pruebas de software

Las pruebas son parte fundamental de cualquier proyecto, ya que ayudan a tener mejores resultados, ofrece una calidad mayor del producto y en consecuencia los clientes quedan satisfechos (Sánchez, 2019).

Las pruebas de software (*testing* en inglés) son los procesos que permiten verificar y revelar la calidad de un producto software antes de su puesta en marcha. Básicamente, es una fase en el desarrollo de software que consiste en probar las aplicaciones construidas. Estas se integran dentro

de las diferentes fases del ciclo de vida del software dentro de la Ingeniería de software. En este sentido, se ejecuta el aplicativo a probar y mediante técnicas experimentales se trata de descubrir qué errores tiene. Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema (Manrique, 2020).

Estrategia de Prueba de software

Para evaluar la calidad del módulo que se está desarrollando y verificar el cumplimiento de los objetivos trazados, se aplicaron un conjunto de pruebas definidas por *Pressman* en su libro de Ingeniería del software “Un enfoque práctico”, en su quinta edición (Pressman, 2015). En la siguiente tabla se muestra la estrategia de pruebas diseñadas para aplicar en la solución desarrollada:

Nivel	Tipo	Método	Técnica
Unitaria	Funcional	Caja Blanca	Camino Básico
Integración	Funcional	Caja Negra	Incremental

Tabla 4: Estrategia de Pruebas de Software

3.2.1 Tipos de Pruebas

La disciplina de pruebas es una de las más costosas del ciclo de vida software. En sentido estricto, deben realizarse las pruebas de todos los artefactos generados durante la construcción de un producto, lo que incluye especificaciones de requisitos, casos de uso, diagramas de diversos tipos y, por supuesto, el código fuente y el resto de productos que forman parte de la aplicación (por ejemplo, la base de datos), e infraestructura. Obviamente, se aplican diferentes técnicas de prueba a cada tipo de producto software (Silva, 2019).

Existen distintas técnicas de software que permiten verificar que el desarrollo cumple con los requerimientos explícitos definidos por el cliente. Es importante destacar que existen distintos tipos de pruebas que pueden ser aplicadas para detectar las desviaciones que afectan la calidad del entregable. Una de las más esenciales son las pruebas funcionales, que se recomienda aplicar en las distintas etapas del desarrollo de software, con el fin de garantizar de forma temprana que el sistema cumple con la calidad y funcionalidad esperada por el usuario (Verity, 2021).

¿Qué son de pruebas funcionales?

Las pruebas funcionales se definen teniendo como fuente los requisitos del sistema, estas pruebas validan y verifican que el producto cumple con lo especificado y hace lo que debe y cómo lo tiene que hacer dando también una idea del grado de calidad del software (Tester , 2018).

Existen diversos tipos de pruebas funcionales, sin embargo, las más críticas y que son imprescindibles realizar en el proceso de prueba son las siguientes:

Pruebas unitarias

Las pruebas unitarias de *software*, conocidas también como *unit testing* o test unitarios, pueden definirse como un mecanismo de comprobación del funcionamiento de las unidades de menor tamaño de un programa o aplicación en específico. Estas pruebas de *software* forman parte de la estrategia de metodología ágil del trabajo del desarrollo, donde se busca ofrecer piezas pequeñas de *software* en funcionamiento en un corto periodo de tiempo, con el objetivo de aumentar la satisfacción del cliente.

Es importante aclarar que este tipo de pruebas son de vital importancia para la detección de errores, ya que, sin este testeo, no podrían identificarse hasta fases más avanzadas del desarrollo, como, por ejemplo, la fase de integración (KeepCoding, 2022).

Las pruebas unitarias son realizadas por el equipo de desarrollo que consisten en comprobar el correcto funcionamiento de las unidades de código; en otras palabras, validan el funcionamiento de los elementos más pequeños del sistema (Verity, 2021).

Caso de Prueba unitaria en la funcionalidad Personal Docente
Probador: Neyis Nieves Barrios Vega
Código al que se le aplica: Clase personDocent

```

def set_order_person_category_structure(self, request, *args, **kwargs):
    datos=request.data
    srl = self.get_serializer(data=datos) } (1)
    if srl.is_valid(raise_exception=True): } (2)
        for item in datos['category_order']:
            tpla=PersonDocent.objects.filter(person=datos['person_id'],category=item['category_id']).first() } (3)
            tpla.category_order=item['order']
            tpla.save(update_fields=['category_order'])
    page = self.paginate_queryset(queryset) } (4)
    if page is not None:
        serializer = EducativeCenterSelectSerializer(page, many=True) } (5)
        return self.get_paginated_response(serializer.data)
    serializer = EducativeCenterSelectSerializer(queryset, many=True) } (6)
    return Response({'order update'}) } (7)

```

Complejidad ciclomática:

$$V(G) = (A - N) + 2$$

$$V(G) = (10 - 9) + 2 = 3$$

$$V(G) = P + 1$$

$$V(G) = 2 \text{ nodos predicados} + 1 = 3$$

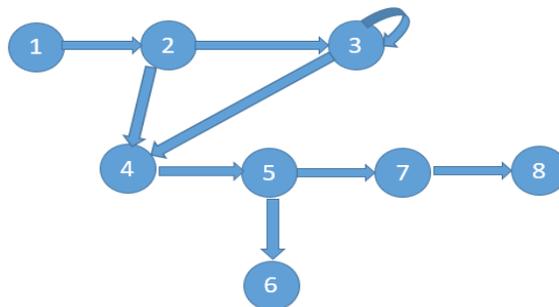
Caminos independientes:

1. 1- 2- 3- 4- 5- 6

2. 1- 2- 3- 4- 5- 7- 8

3. 1- 2- 4- 5- 6

Representación del grafo:



Caso de prueba para los caminos básicos

Tipo de dato esperado:

Persona

Función de evaluación:

```

class ModelsStr(BaseTest):
    """
    Test __str__ in models
    """
    fixtures = ['type.json', 'province.json', 'person.json', 'structure.json']

    def setUp(self):
        super().setUp()
        self.person = Person.objects.first()
        self.productive_activity = ProductiveActivityType.objects.create(
            name='actividad')

        self.educational_data = {
            'person_id': self.person.id,
            "teaching_category_id": TeachingCategoryType.objects.first().id,
            "specialty_id": TeacherSpecialityType.objects.first().id,
            "teaching_status_id": TeacherStatusType.objects.first().id,
            "structure_id": Structure.objects.get_school_queryset().first().id
        }
        self.educational_worker = EducationalWorker.objects.create(
            **self.educational_data)

        self.not_educational_data = {
            'person_id': self.person.id,
            "structure_id": Structure.objects.get_school_queryset().first().id
        }
        self.not_educational_worker = NotEducationalWorker.objects.create(
            **self.not_educational_data)

```

Resultado de prueba:

Nombre	Personal Docente
Tipo de datos	Persona
Tipo Esperado	Persona
Resultado	Pasado
Archivo	View_person_docent.py
Numero Linea	25

Evaluación del caso de prueba: Satisfactoria

Tabla 5: Caso de prueba unitaria de la funcionalidad Personal Docente.

Pruebas de integración

Las pruebas de integración se definen como un análisis de los procesos relacionados con el ensamblaje o unión de los componentes, sus comportamientos con múltiples partes del sistema (ya sea de archivos operativos) o de *hardware*, entre otras.

Las pruebas de integración están a cargo del examen de las interfaces entre los subsistemas o los grupos de componentes del programa o aplicación que se analiza, lo que contribuye a garantizar su funcionamiento correcto. Estas pruebas verifican que el sistema y sus componentes corran bien, cumplan con cada una de las labores asignadas y que se adapten a los requisitos establecidos. Permiten

el uso de sistemas reales, preparados y dedicados específicamente para este test, lo que hará más sencillo el proceso de aplicación de la prueba (KeepCoding, 2022).

Caso de prueba de integración con el componente Estructura y Composición	
Componente al que se le integra	Estructura y Composición.
Condiciones de la ejecución:	El componente de Estructura y Composición haya introducido los datos en la base de datos central y exista conexión con la misma.
Descripción de la prueba:	Comprobar que el SGA en el componente Personal es capaz de gestionar las estructuras y responsabilidades involucradas con las nuevas funcionalidades.
Entrada/Pasos de ejecución:	El componente Estructura y Composición introduce en la base de datos central los datos y el SGA en el componente Personal consulta estos y gestiona las estructuras y responsabilidades involucradas con las nuevas funcionalidades.
Resultado esperado:	Se gestionan las estructuras y responsabilidades involucradas con las nuevas funcionalidades
Evaluación:	Prueba satisfactoria

Tabla 6: Caso de Prueba con el Componente Estructura y Composición

3.2.2 Métodos de Pruebas

Prueba de Caja Blanca o estructural

Las pruebas de caja blanca pueden definirse como una técnica de monitorización o prueba de [software](#) en la se analiza el diseño, código y estructura interna, con el objetivo de mejorar propiedades como la seguridad y el uso eficiente del sistema. Estas pruebas se caracterizan principalmente porque son los propios sistemas y aplicaciones quienes exponen sus métricas para que el usuario pueda leerlas, analizarlas y tomar decisiones y acciones en función de la obtención de un resultado u otro (Keepcoding, 2022).

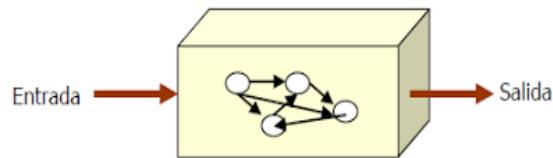


Fig. 8: Ejemplo de Caja Blanca

Para realizar la prueba de caja blanca se aplicó la técnica de camino básico que permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y utilizar esta medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

Prueba de Caja Negra o funcional

Las pruebas de caja negra verifican la funcionalidad de una aplicación ignorando la parte interna de la misma, es decir que se obvia la estructura del código, la arquitectura, los detalles relacionados con la implementación de los diferentes módulos, paquetes o rutas en la que se compone el código. Este tipo de prueba se basa por completo en los requisitos de las aplicaciones y en sus especificaciones técnicas.

En las pruebas de caja negra el *Tester* solo se centra en las entradas y salidas de la aplicación, sin preocuparse por el contenido interno. Lo que pase por dentro es indiferente, solo importa que, si se realiza cierta acción, la salida sea la indicada según los requerimientos (Testing, 2020).



Fig. 9: Ejemplo de Caja Negra

Resultados obtenidos

Resultados de las pruebas unitarias

Durante el desarrollo del componente de integración se llevaron a cabo pruebas unitarias para verificar que el código implementado de una funcionalidad específica cumple con el resultado esperado.

Resultados de las pruebas de integración

La necesidad de realizar las pruebas de integración viene dada por el hecho de que los módulos que forman un programa suelen fallar cuando trabajan de forma conjunta, aunque previamente se haya demostrado que funcionan correctamente de manera individual. Con el uso de estas pruebas se consigue ir formando el programa global a medida que se comprueba como los distintos componentes interactúan y se comunican libres de errores.

Durante la implementación de las funcionalidades se realizaron las pruebas de integración y las realizaciones de las pruebas unitarias permitieron afirmar que las funcionalidades establecidas a integrar y demás elementos del sistema operaron satisfactoriamente, ajustándose a los requisitos especificados.

Resultado de la Prueba de caja Negra realizada

Descripción	Variable 1	Respuesta del sistema	Flujo central
Permite generar reporte de Movimiento de personal	V (Introducir valores en los campos)		1-Introducir los valores en los campos. 2-Se selecciona la opción "Generar". 3-El Sistema genera el reporte.
	I(Introducir valores numéricos)	El Sistema muestra "Existen parámetros con errores"	
	I(Introducir campos vacíos)	El Sistema muestra el mensaje "Existen campos Vacíos"	

Tabla 7: Prueba de caja negra **Conclusiones del capítulo**

La definición de los estándares de codificación y de las técnicas de programación empleadas para la implementación permitió describir el proceso de creación del código de la solución. Las distintas pruebas realizadas demostraron que el módulo desarrollado a partir de las pruebas internas provee un rendimiento satisfactorio para los caminos estudiados por los diferentes métodos de caja blanca y caja negra, cumpliendo con los requisitos funcionales identificados.

Conclusiones

- Se desarrolló un módulo de reportes estadísticos de personal para el sistema de gestión académica Xauce Akademos para el MINED que es capaz de generar reportes.
- La elaboración del marco teórico de la investigación permitió obtener conceptos fundamentales que permitieron un mayor entendimiento para dar solución al problema planteado.

- El estudio teórico de las herramientas para los reportes estadísticos a nivel internacional permitió diseñar un módulo sobre aplicaciones libres que visualice la información referente al proceso de personal.
- Los resultados del módulo de reportes estadísticos para el sistema de gestión académica Xauce Akademos para el MINED, permitió constatar mediante las pruebas que el componente está listo para ser desplegado.

Recomendaciones

Tras haber finalizado la investigación y desarrollado la propuesta de solución surgen ideas que podrían ser implementadas en futuras versiones del módulo. Se recomienda:

- Ampliar las funcionalidades módulo estadístico de personal, de forma que se puedan enviar por correo los recursos a fin de que los usuarios puedan obtener y enviar la información sin

necesidad de abrir el sistema, para una obtención de reportes más precisos al momento de tomar decisiones y dar conclusiones referentes a algún tema.

- Diversificar las formas de graficar los resultados con el objetivo de facilitar al administrador una mayor comprensión de los mismos.

Referencias Bibliográficas

[En línea] [Citado el: 19 de 05 de 2022.] <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>.

[En línea]

[En línea] [Citado el: 09 de 06 de 2022.] https://www.unece.org/fileadmin/DAM/stats/documents/writing/MDM_Part2_Spanish.pdf.

[En línea] [Citado el: 20 de 06 de 2022.] <https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/>.

[En línea] <http://estadisticacrisanto.blogspot.com/2013/10/representacion-tabular-y-grafica-de.html>.

[En línea] [Citado el: 20 de 10 de 2022.] <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/pruebas/acceptacion/>.

Academia. [En línea] [Citado el: 26 de 10 de 2022.] https://www.academia.edu/8813345/Pruebas_de_Software.

acreditacion, Comision Nacional de. Comision Nacional de acreditacion CNA-Chile. [En línea] [Citado el: 02 de 06 de 2022.] <https://www.cnachile.cl/noticias/Paginas/GESTI%C3%93N-ACAD%C3%89MICA.aspx>.

Aguilar, Lic. Eddy Ismael Tito. Pedagogiacuba. [En línea] [Citado el: 02 de 06 de 2022.] <https://trabajos.pedagogiacuba.com/trabajos/54Eddy%20Ismael%20Tito%20Aguilar%20C.%20Zapata.pdf>.

AWS. 2021. AWS. [En línea] 2021. [Citado el: 20 de 9 de 2022.] <https://aws.amazon.com/es/what-is/service-oriented-architecture/>.

2022. AyerViernes. [En línea] 2022. [Citado el: 20 de 06 de 2022.] <https://www.ayerviermes.com/blog/herramientas-ux-mapa-de-navegacion>.

Barrios, Erika Josefina Sulbaran Silva y Trino. 2011. Monografias. [Online] Ministerio Popular para la Educacion Superior , Enero 2011. [Cited: 06 02, 2022.] <https://www.monografias.com/trabajos84/definicion-estadistica/definicion-estadistica>.

Brainlabs. [En línea] [Citado el: 03 de 06 de 2022.] <https://www.brainlabs.com.ar/novedad/staruml-una-herramienta-para-modelado/>.

CASEWER. 2018. CASEWER Tecnologia. [En línea] 2018. [Citado el: 02 de 06 de 2022.] <http://www.casewaresa.com.co/sistema-gestion-academica/>.

Clibre. 2021. Clibre. [En línea] 2021. [Citado el: 20 de 06 de 2022.] <https://clibre.io/blog/por-secciones/codigo/item/433-patrones-de-diseno-de-software>.

Crisanto. 2017. [En línea] 2017. [Citado el: 09 de 06 de 2022.] <https://definicion.de/reporte/>.

Crisanto, Abraham Jimenez. 2017. Estadistica. [En línea] 2017. <http://estadisticacrisanto.blogspot.com/2013/10/representacion-tabular-y-grafica-de.html>.

2022. Digite. [En línea] 2022. [Citado el: 20 de 10 de 2022.] <https://www.digite.com/es/agile/pruebas-de-acceptacion/>.

2022. Ebooksonline. [En línea] 2022. [Citado el: 26 de 10 de 2022.] <https://ebooksonline.es/que-es-una-prueba-de-regresion-definicion-casos-de-prueba-ejemplo/>.

Educaweb. 2022. educaweb. [En línea] 2022. [Citado el: 09 de 06 de 2022.] <https://gestioneducativa.educaweb.com/gestion-academica-escuela-formas-mejorar/>.

Enheror. 2019. [En línea] 2019. <http://www.disca.upv.es/Enheror/Pdf/Actauml.Pdf>.

Española, Real Academia. Real Academia Española. [En línea] Asociación de Academias de las Lenguas Españolas. [Citado el: 02 de 06 de 2022.] <https://dle.rae.es/personal>.

Gonzalez, Diego. 2017. El Oriente. [En línea] abril de 2017. [Citado el: 09 de 06 de 2022.] <http://www.eloriente.net/home/2017/04/17/educacion-uso-estadisticas-aula-diego-gonzalez-algara/>.

Guerra, César Arturo. 2017. Software Guru. [En línea] 2017. [Citado el: 20 de 06 de 2022.] <https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>.

—. 2017. Software Suru (SG). [En línea] 2017. [Citado el: 20 de 06 de 2022.] <https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>.

Guillot, Javier. 2018. Researchgate. [En línea] 2018. [Citado el: 08 de 06 de 2022.] https://www.researchgate.net/publication/264037218_SIGUA_Sistema_Informatico_de_Gestion_Universitaria?enrichId=rgreq-00ab0a6273bf777dd8c0e80868571b5f-XXX&enrich-Source=Y292ZXJQYWdIOzI2NDZAzNzIxODtBUzoxMjAyNjA4NjI2ODEwODIAMTQwNTY4Mzgz-MjgxMA%3D%3D&el=1_x_2&_e.

Hiberus. 2020. [En línea] 2020. [Citado el: 19 de 05 de 2022.] <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/>.

Ichlese. 2022. Ichlese. [En línea] 2022. [Citado el: 02 de 06 de 2022.] <https://es.ichlese.at/what-are-advantages>.

2020. Ilimit . [En línea] 30 de 07 de 2020. [Citado el: 02 de 06 de 2022.] <https://www.iliimit.com/blog/flask-vs-django/>.

Insignea. 2016. El Insignea. [En línea] 2016. [Citado el: 03 de 06 de 2022.] <https://elinsignea.com/2016/11/28/la-estadistica-utilizada-area-recursos-humanos/>.

Insignia. 2016. El insignia. [En línea] 2016. [Citado el: 03 de 06 de 2022.] <https://elinsignia.com/2016/11/28/la-estadistica-utilizada-area-recursos-humanos/>.

Invid. 2020. [En línea] 2020. [Citado el: 23 de 06 de 2022.] <https://mpiua.invid.udl.cat/fases-mpiua/diseno/modelo-mental-y-modelo-conceptual/>.

KEEPCODING . 2021. KEEPCODING Tech School. [En línea] 2021. [Citado el: 02 de 06 de 2022.] <https://keepcoding.io/blog/ventajas-y-desventajas-de-python/>.

Keepcoding. 2022. Keepcoding. [En línea] 2022. [Citado el: 20 de 06 de 2022.] <https://keepcoding.io/blog/que-es-arquitectura-software/>.

—. 2022. Keepcoding. [En línea] 03 de 06 de 2022. [Citado el: 26 de 10 de 2022.] <https://keepcoding.io/blog/que-son-pruebas-de-caja-blanca/>.

KeepCoding. 2022. Keepcoding. [En línea] 05 de 08 de 2022. [Citado el: 26 de 10 de 2022.] <https://keepcoding.io/blog/que-son-las-pruebas-de-integracion/>.

KeepCoding, Redacción. 2022. Keepcoding. [En línea] 01 de 08 de 2022. [Citado el: 26 de 10 de 2022.] <https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/>.

Kinsta. 2022. Kinsta. [En línea] 25 de Abril de 2022. [Citado el: 02 de 06 de 2022.] <https://kinsta.com/es/base-de-conocimiento/que-es-postgresql/>.

La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. **Arias, Chaves Michael. 2017.** 1-13, s.l. : InterSedes: Revista de las Sedes Regionales, 2017, Vol. 10. ISSN: 2215-2458.

Manrique, Silva Julio Cesar. 2020. Academia. [En línea] 2020. [Citado el: 26 de 10 de 2022.] https://www.academia.edu/8813345/Pruebas_de_Software.

MENZINSKY, Alexander, et al. Historias de usuario. Ingeniería de requisitos ágil, 2018. [En línea]

MENZINSKY, Alexander, et al. Historias de usuario. Ingeniería de requisitos ágil, 2018. [En línea]

—. [En línea]

Monografías. [En línea] <https://www.monografias.com/docs/Representacion-Oral-Y-Escrita-F3DV2STPJ8U2Y>.

Patrones de Diseño GOF. **Guerrero, Carlos. 2013.** Colombia : Ciudadela Real de Minas, 2013. 0718-0764.

PMO-Informatica. 2017. PMOinformatica.com La oficina de proyectos de informática. [En línea] 2017. [Citado el: 20 de 06 de 2022.] <http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>.

Pressman, Roger. *Ingeniera de software, Un enfoque practico.* s.l. : 5ta.

—. *Ingeniera del Software: Un enfoque práctico.* s.l. : 5ta.

—. **2015.** *Ingeniería de Software, Un enfoque práctico.* s.l. : 5ta, 2015.

Publicaciones. 2019. [En línea] 2019. <https://publicaciones.uci.cu/index.php/serie/article/download/253/149/>.

Python. 2021. [En línea] 2021. [Citado el: 19 de 05 de 2022.] <https://www.becas-santander.com/es/blog/python-que-es.html>.

Ramírez, Ing. Miguel Medina. 2021. *Proyecto Desarrollo del Sistema.* La habana : s.n., 2021.

—. **2021.** *Proyecto Desarrollo del Sistema.* 2021.

—. **2021.** *Proyecto Desarrollo del Sistema.* 2021.

Ramírez, Miguel Medina. 2021. *Proyecto Desarrollo del Sistema.* La Habana, Cuba : s.n., 2021.

Redes de computadores. [En línea] [Citado el: 20 de 06 de 2022.] <http://profesores.elo.u-tfsm.cl/~agv/elo322/1s16/projects/reports/Proyecto%20Cliente%20-%20Servidor.pdf>.

Redhat. [En línea] [Citado el: 20 de 9 de 2022.] <https://www.redhat.com/es/topics/cloud-native-apps/what-is-service-oriented-architecture>.

Sánchez, José. 2019. [En línea] 2019. [Citado el: 26 de 10 de 2022.] https://oa.upm.es/40012/1/PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf.

Sánchez, Tamara Rodríguez. 2015. *Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana, Cuba : s.n., 2015. ISSN.

Sanchez, Tamara Rodríguez. 2015. Metodología de desarrollo para la Actividad productiva en la UCI. [En línea] 2015. [Citado el: 18 de 05 de 2022.]

Serie Científica de la Universidad de las Ciencias Informáticas. **Costa, Grisel Infante. 2008.** 1, La habana : Universidad de las Ciencias Informáticas, 2008, Vol. 1, p. 1 y 2.

2022. Servicio de Informática. [En línea] Universidad de Alicante, 2022. [Citado el: 20 de 06 de 2022.] <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>.

Siga. 2017. SIGA. [En línea] 2017. [Citado el: 09 de 06 de 2022.] <http://www.siga.com.-co/sitio/>.

Silva, Julio Cesar Manrique. 2019. [En línea] 2019. [Citado el: 26 de 10 de 2022.] https://www.academia.edu/8813345/Pruebas_de_Software.

Sistema de gestión de la Nueva Universidad (SIGENUv4.0.0) como herramienta para la automatización de los procesos docentes y estadísticos de la Universidad de Granma. **Lic. Rubisnel Almeida Saavedra, Gisber Miguel Góngora Mora. 2018.** 4, s.l. : REDEL. Revista Granmense de Desarrollo Local, 2018, Vol. 2. RNPS: 2448..

Tecnología, Equipo de Expertos en Ciencia y. La evolución de las TICs. [En línea] <https://www.universidadviu.com/int/actualidad/nuestros-expertos/la-evolucion-de-las-tics-desde-el-abaco-al-smartphone>.

Tester . 2018. Tester house. [En línea] 2018. [Citado el: 26 de 10 de 2022.] <https://testerhouse.com/teoria-testing/pruebas-funcionales/>.

2019. Tester house. [En línea] 06 de 03 de 2019. [Citado el: 26 de 10 de 2022.] <https://testerhouse.com/teoria-testing/pruebas-funcionales/>.

Testing. 2020. How to testing. [En línea] 2020. [Citado el: 26 de 10 de 2022.] <https://howtotesting.com/testing-funcional/pruebas-de-caja-negra/>.

2022. Tudashboard . *Inteligencia de Negocios*. [En línea] 2022. [Citado el: 09 de 06 de 2022.] <https://tudashboard.com/grafica-de-pastel/>.

Tudashboard. 2022. Tudashboard. [En línea] 2022. [Citado el: 09 de 06 de 2022.] <https://tudashboard.com/grafica-de-barras/>.

—. **2022.** Tudashboard. [En línea] 2022. [Citado el: 09 de 06 de 2022.] <https://tudashboard.com/grafica-de-lineas/>.

2021. Verity. [En línea] 09 de 04 de 2021. [Citado el: 26 de 10 de 2022.] <https://www.verity.cl/blog/5-pruebas-funcionales-que-todo-software-necesita>.

Verity. [En línea] [Citado el: 26 de 10 de 2022.] <https://www.verity.cl/blog/5-pruebas-funcionales-que-todo-software-necesita>.

Verity. 2021. Verity. [En línea] 09 de 04 de 2021. [Citado el: 26 de 10 de 2022.] <https://www.verity.cl/blog/5-pruebas-funcionales-que-todo-software-necesita>.

—. **2021.** Verity. [En línea] 09 de 04 de 2021. [Citado el: 26 de 10 de 2022.] <https://www.verity.cl/blog/5-pruebas-funcionales-que-todo-software-necesita>.

Visual Paradigm. 2020. Visual Paradigm. [En línea] 2020. [Citado el: 02 de 06 de 2022.] <https://www.visual-paradigm.com/aboutus/newsreleases/vp122.jsp>.

Webempresa. 2022. Webempresa. [En línea] 2022. <https://www.webempresa.com/hosting/que-es-servidor-web.html>.

ANEXOS

Anexo 1

Guía de Entrevista

1. ¿Qué objetivo tiene el módulo?
2. ¿Qué personas proveerán la información, cuál es su grado de autoridad y que disponibilidad de tiempo tienen?
3. ¿Qué metodología usa el Sistema Xauce Akademos? ¿En qué escenario?
4. ¿Con qué tecnología se trabaja en el sistema Akademos? ¿En qué versiones?
5. ¿Cómo se va a hacer la representación de la información?
6. ¿Cuáles son los tipos de centros educacionales?
7. ¿Cuáles son los tipos de educación?

Anexo 2

RnF 1. Requisito de Usabilidad 1.2. El sistema debe poseer un diseño “Responsable”.

Atributo de Calidad	Usabilidad
Sub-atributos/Sub-características	Estética de la interfaz de usuario
Objetivo	Garantizar la adecuada visualización y organización de las acciones en múltiples dispositivos de acceso.
Origen	Sistema
Artefacto	Sistema
Entorno	El sistema está funcionando correctamente
Estímulo	Respuesta: Flujo de eventos (Escenarios)
1.a Realizar una acción en el sistema	
El usuario accede a una vista del sistema.	Se muestra la interfaz correspondiente.
Medida de respuesta	
Se muestra la interfaz correspondiente adaptada al tamaño de la pantalla del dispositivo.	

Tabla 8: Descripción del Requisito no Funcional Usabilidad 1.2

RnF 2. Requisito de Seguridad 1.2. El Sistema debe restringir la ejecución de acciones a usuarios sin credenciales que intenten acceder a las mismas.

Atributo de Calidad	Seguridad
Sub-atributos/Sub-características	Confidencialidad
Objetivo	Proteger los datos y la información contra

	accesos sin credenciales
Origen	Interno / Externo / No autorizado
Artefacto	Sistema
Entorno	El sistema está funcionando correctamente
Estímulo	Respuesta: Flujo de eventos (Escenarios)
1.a Acceso no autorizado	
Intentos de acceso a una vista sin credenciales.	<ol style="list-style-type: none"> 1. Se muestra el mensaje de información. 2. Se deniega el acceso a la acción.
Medida de respuesta	
No se muestra la vista	

Tabla 9: Descripción del Requisito no Funcional Seguridad 1.2

Precondiciones	Debe haber un usuario autenticado con permiso a la funcionalidad.
Flujo de eventos	
Flujo básico	Generar reporte de datos sobre el personal docente y no docente.
	Se selecciona Estadísticas / AF Personal / Docente y no Docente.
5.	Se introducen los datos solicitados.
6.	Se selecciona el Botón Generar.
7.	Se muestra un dashboard con los gráficos de la información general del reporte y los listados del personal docente y no docente.
8.	<p>Se selecciona una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Ícono Exportar a PDF. Ver FORTES_Levantamiento_AKADEMOS_MINED_DRP_Exportar_reporte_sobre_el_personal_docente_y_no_docente.odt. • Ícono Exportar a XLS. Ver FORTES_Levantamiento_AKADEMOS_MINED_DRP_xportar_reporte_sobre_el_personal_docente_y_no_docente.odt. • Botón Filtrar (General). Ver Flujo alternativo 7.a. • Botón Buscar (Docentes). Ver Flujo alternativo 7.b. • Botón Filtrar (Docentes). Ver Flujo alternativo 7.c. • Botón Buscar (no Docentes). Ver Flujo alternativo 7.d. • Botón Filtrar (no Docentes). Ver Flujo alternativo 7.d.
Pos-condiciones	
	Se generó el reporte sobre el personal docente y no docente.
Flujos alternativos	
Pos-condiciones	
	Se generó el reporte de datos sobre el personal docente y no docente.
Flujo alternativo 7.a Filtrar.	
1	<p>Se muestra el panel de filtros, con los filtros que se seleccionen y las opciones siguientes:</p> <p>Filtros: - Estado - Cuadro - Miembro del Consejo de Dirección</p>

	Opciones:	
	<ul style="list-style-type: none"> • Mostrar/Ocultar: Ver Flujo alternativo 2.f. • Icono Cerrar filtros (imagen de cruz): Ver Flujo alternativo 2.g. • Icono Cerrar filtro interno (imagen de signo menos): Ver Flujo alternativo 2.h. 	
2	Se seleccionan los valores en los filtros.	
3	Se muestran los elementos del personal docente y no docente que coincidan con los valores seleccionados en los filtros. <ul style="list-style-type: none"> - En caso de que no exista ninguna coincidencia en el personal docente y no docente con los filtros introducidos, se muestra el mensaje: "No hay datos". 	
4	Se pasa al Paso 5 del Flujo básico.	
Flujo alternativo 5.b Buscar (Docentes).		
9	Se introduce parámetro para la búsqueda (Tipo de alta, procedencia, educación)	
10	Se selecciona el botón Buscar .	
11	Se muestran los elementos del personal docente y no docente que coincidan con los parámetros introducidos. <ul style="list-style-type: none"> - En caso de que no exista ninguna coincidencia en el personal docente y no docente con los parámetros introducidos, se muestra el mensaje: " No hay datos". 	
12	Se pasa al Paso 5 del Flujo básico.	
Flujo alternativo 5.c Filtrar (Docentes).		
5	Se muestra el panel de filtros, con los filtros que se seleccionen y las opciones siguientes: Filtros: - Categoría -Estado -Cuadro -Educación -Tipo de centro educacional Opciones: <ul style="list-style-type: none"> • Mostrar/Ocultar: Ver Flujo alternativo 2.i. • Icono Cerrar filtros (imagen de cruz): Ver Flujo alternativo 2.j. • Icono Cerrar filtro interno (imagen de signo menos): Ver Flujo alternativo 2.k. 	
6	Se seleccionan los valores en los filtros.	
7	Se muestran los elementos del personal docente y no docente que coincidan con los valores seleccionados en los filtros. <ul style="list-style-type: none"> - En caso de que no exista ninguna coincidencia en el personal docente y no docente con los filtros introducidos, se muestra el mensaje: "No hay datos". 	
8	Se pasa al Paso 5 del Flujo básico.	
Flujo alternativo 5.d Buscar (no Docentes).		
13	Se introduce parámetro para la búsqueda (Tipo de alta, procedencia, educación)	
14	Se selecciona el botón Buscar .	
15	Se muestran los elementos del personal docente y no docente que coincidan con los parámetros introducidos. <ul style="list-style-type: none"> - En caso de que no exista ninguna coincidencia en el personal docente y no docente con los parámetros introducidos, se muestra el mensaje: " No hay datos". 	
16	Se pasa al Paso 5 del Flujo básico.	
Validaciones		
1	Campo de selección Fecha	Muestra la fecha_inicio y la fecha_fin seleccionadas en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_personal_docente_y_no_docente.odt.

37	Campo de selección Nivel	Muestra los filtros seleccionados en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_personal_docente_y_no_docente.odt
38	Campo de selección Estructura	Muestra la estructura seleccionada en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_personal_docente_y_no_docente.odt.
39	Componente Filtrar	Muestra los filtros seleccionados en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_personal_docente_y_no_docente.odt.
40	Filtro-Provincia	Muestra la provincia de la estructura registrada en Estructura y seleccionada en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_personal_docente_y_no_docente.odt.
41	Check Nacional	Se muestra marcado si la estructura seleccionada en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_personal_docente_y_no_docente.odt tiene nivel Nacional.
42	Tabla filtros	Muestra los filtros generales aplicados en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_personal_docente_y_no_docente.odt. Atributos que se muestran como columnas de la tabla: - Filtro y valor Se muestran organizados por el orden en que fueron aplicados.
43	Período	Muestra la fecha_inicio y la fecha_fin seleccionadas en FORTES_Levantamiento_AKADEMOS_MINED_DRP_Generar_reporte_personal_docente_y_no_docente.odt.
44	Tabla Personal Docente	Atributos que se muestran como columnas de la tabla: - Concepto -Total
45	Tabla Personal no Docente	Atributos que se muestran como columnas de la tabla: - Concepto -Total
46	Tabla Personal Docente por tipo de centro educacional	Atributos que se muestran como columnas de la tabla: - Concepto -Total - Tipo de centro educacional (una columna por cada tipo de centro educacional)
47	Tabla Personal no Docente por tipo de centro educacional	Atributos que se muestran como columnas de la tabla: - Concepto -Total Tipo de centro educacional (una columna por cada tipo de centro educacional)
48	Tabla Personal Docente por educación	Atributos que se muestran como columnas de la tabla: - Concepto -Total - Educación (una columna por cada Educación)
49	Tabla Personal no Docente por educación	Atributos que se muestran como columnas de la tabla: - Concepto - Total - Educación (una columna por cada Educación)
50	Fila Total de trabajadores al inicio del	Muestra la cantidad de trabajadores de alta de la estructura seleccionada y sus subordinadas en la fecha_inicio del período seleccionado. Se calcula de la siguiente forma:

período	$y = \sum_{n=1}^k f(a, b)_n$ $f(a, b) = \begin{cases} 1, & \sum_{n=1}^m a_n - \sum_{n=1}^p b_n \geq 1 \\ 0, & \sum_{n=1}^m a_n - \sum_{n=1}^p b_n = 0 \end{cases}$ <ul style="list-style-type: none"> • $y \in \mathbb{N}$, total de trabajadores al principio del período • $k \in \mathbb{N}$, cantidad de personas registradas en Persona que tienen al menos un trámite de alta registrado en Tramite_alta con <i>fecha_alta</i> ≤ fecha de inicio del período seleccionado por el usuario • $f(a, b) \in \{0 \text{ (baja)}; 1 \text{ (alta)}\}$, situación laboral de la persona • $m \in \mathbb{N}$, cantidad de trámites registrados en Alta_Trabajador de un trabajador registrado en Persona y con <i>fecha_alta</i> ≤ fecha de inicio del período seleccionado por el usuario
51 Inactivos	<p>Muestra la cantidad de trabajadores de alta con estado_trabajador inactivo de la estructura seleccionada y sus subordinadas en la fecha de fin del período seleccionado. Se calcula de la siguiente forma:</p> $e = \sum g$ <p>Donde sea:</p> <ul style="list-style-type: none"> • \in, el total de trabajadores inactivos al final del período • \in persona registrada en Persona con al menos un trámite de prestación económica registrado en Prestacion_Economica con <i>fecha_inicio</i> ≤ fecha de fin del período seleccionado por usuario < <i>fecha_fin</i>
52 Activos	<p>Muestra la cantidad de trabajadores de alta con estado_trabajador activo u otras funciones de la estructura seleccionada y sus subordinadas en la fecha de fin del período seleccionado. Se calcula de la siguiente forma:</p> $c = z - e$ <p>Donde sea:</p> <ul style="list-style-type: none"> • \in el total de trabajadores activos al final del período • \in el total de trabajadores inactivos al final del período (ver validación Inactivos) • \in el total de trabajadores al final del período (ver validación

		Total de trabajadores al final del periodo)
53	Columna Total	Muestra la suma de los valores numéricos de cada fila.
54	Gráfico de pastel Total Personal Docente y no Docente	Muestra la cantidad de movimientos del personal en el período seleccionado por el usuario. Atributos que se muestran en el gráfico: - Docentes (ver validación de Total de docentes) - No Docentes (ver validación de Total de no docentes)
55	Gráfico de pastel Total no Docente	Muestra la cantidad de trabajadores al final del período seleccionado por el usuario. Atributos que se muestran en el gráfico: - Activos (ver validación de Activos) - Inactivos (ver validación de Inactivos)
Conceptos	Persona	Visibles en la interfaz: - N/A Utilizados internamente: - primer_nombre - segundo_nombre - primer_apellido - segundo_apellido
	Registrador	Visibles en la interfaz: - N/A Utilizados internamente: - trabajador - estructura - responsabilidad - alta
	Estructura	Visibles en la interfaz: - N/A Utilizados internamente: - nombre
	Tipo_Estructura	Visibles en la interfaz: - N/A Utilizados internamente: - nombre - nivel
	Centro_Educacional	Visibles en la interfaz: - N/A Utilizados internamente: - nombre - tipo_centro_educacional -configuración_especialidad
	Configuracion_Especialidad	Visibles en la interfaz: - N/A Utilizados internamente: - educación
	Educación	Visibles en la interfaz: - N/A Utilizados internamente: - educación
	Tipo_Centro_Educacional	Visibles en la interfaz: - N/A Utilizados internamente: - nombre
	Responsabilidad	Visibles en la interfaz: - N/A Utilizados internamente: - nombre_responsabilidad

	Tramite _Trabajador	Visibles en la interfaz: - N/A Utilizados internamente: - trabajador - tramite - fecha_fin_tramite
	Alta_Trabajador	Visibles en la interfaz: - N/A Utilizados internamente: - tramite_trabajador - fecha_alta - tipo_alta - procedencia
	Tipo_Alta	Visibles en la interfaz: - N/A Utilizados internamente: - tipo_alta
	Procedencia	Visibles en la interfaz: - N/A Utilizados internamente: - procedencia
	Baja_Trabajador	Visibles en la interfaz: - N/A Utilizados internamente: - tramite_trabajador - fecha_baja - tipo_baja -motivo_baja_trabajador
	Tipo_Baja	Visibles en la interfaz: - N/A Utilizados internamente: - tipo_baja
	Estado_Trabajador	Visibles en la interfaz: - N/A Utilizados internamente: - estado_trabajador
Requisitos especiales	N/A	
Asuntos pendientes	N/A	

Tabla 10: Descripción del Requisito Generar Reporte de datos sobre Personal Docente y no docente.