



Facultad de Ciencias y Tecnologías Computacionales

Componente cuadro de mando para XAUCE AKADEMOS MINED

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

Autor:Wendy Rosell Darias

Tutores:Maily Andrea Santos Sánchez
Sandy Nuñez Padrón

Co-tutor:Yamilka Gómez León

La Habana, noviembre de 2022

Año 64 de la Revolución



Considero más valiente al que conquista sus deseos, que al que conquista a sus enemigos, ya que la victoria más dura es la victoria sobre uno mismo.

Aristóteles

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título “**Componente cuadro de mando para XAUCE AKADEMOS MINED**”, concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como únicos autores de su contenido. Para que así conste firman la presente a los ~~< día >~~ 1 días del mes de ~~< mes >~~ diciembre del año ~~< año >~~ 2022.

~~< nombre del autor >~~ Wendy Rosell Darias Maily Andrea Santos Sánchez ~~< nombre del autor >~~

Firma del Autor

Firma del ~~Autor~~ Tutor

Sandy Nuñez Padrón ~~< nombre del tutor >~~ Yamilka Gómez León ~~< nombre del tutor >~~

Firma del Tutor

Firma del Co-t Tutor

AGRADECIMIENTOS

A mi mayor tesoro en todo el mundo, mi familia. Gracias por estar siempre ahí y darme el mayor apoyo cuando me derrumbaba.

Dedicatoria

DEDICATORIA

A ti, mi mayor impulso, donde quiera que estés espero que sigas estando orgulloso de mi.

RESUMEN

En la inteligencia de negocio, utilizar indicadores claves de desempeño en un cuadro de mando para monitorizar todos los parámetros de la empresa, tiene el fin de disponer de una imagen real de lo que ocurre dentro y fuera de los centros u organizaciones. La investigación realizada posee como objetivo desarrollar un componente de cuadro de mando para XAUCE AKADEMOS para el Ministerio de Educación de la República de Cuba (MINED), que permita a los usuarios/clientes capacitados realizar el monitoreo de los principales indicadores claves de desempeño de una manera más rápida y sencilla. Se realizó un estudio ~~a-de~~ un conjunto de sistemas homólogos que permitió determinar varios aspectos importantes a tener en cuenta en el desarrollo del componente. Además, para regir el proceso de desarrollo se utilizó la metodología AUP-UCI, adaptada a los procesos productivos de la ~~universidad~~ Universidad de Ciencias Informáticas (UCI). La solución desarrollada es capaz de mostrar la información deseada proporcionando un entorno de trabajo más cómodo y fácil para los usuarios; y así facilitar una representación mucho más exacta. Antes de desplegar el producto se sometió a diferentes pruebas de acuerdo a la metodología de desarrollo de software realizando varias iteraciones hasta no encontrar no conformidades, demostrando que el módulo está apto para ser utilizado.

Palabras Claves: inteligencia de negocio, indicadores claves de desempeño, cuadro de mando.

ABSTRACT

In business intelligence, using key performance indicators in a dashboard (circuit board) to monitor all the parameters of the enterprise (company) intends to make use of a real image of what occurs inside and outside centers and organizations. The conducted research aims to develop a component of dashboard for XAUCE AKADEMOS for the Cuban Ministry of Education that allows qualified users and clients to monitor the key performance indicators quicker and easier. A study of a set of homologue systems was made, and it allowed to determine some important aspects that should be taken into account in the development of the component. Furthermore, to rule the process of development, the methodology AUP-UCI, adapted to the productive process of the university, was used. The developed solution is able to show the wanted information, providing users with an easier and more comfortable work environment, and in that way, easing a much more exact representation. Before displaying the product, it was subjected to different tests according to the methodology of software development, making some interactions until no inconvenience is found, proving that the module is suitable to be used.

Key words: *business intelligence, key performance indicators, dashboard*

Con formato: Centrado, No dividir palabras, Punto de tabulación: No en 1,62 cm + 3,23 cm + 4,85 cm + 6,46 cm + 8,08 cm + 9,69 cm + 11,31 cm + 12,92 cm + 14,54 cm + 16,16 cm + 17,77 cm + 19,39 cm + 21 cm + 22,62 cm + 24,23 cm + 25,85 cm

Contenido

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICOS.....	6
I.1 Conceptos asociados	6
I.2 Estudio de soluciones similares.....	12
I.3 Metodología, lenguajes y tecnologías	16
Conclusiones del capítulo.....	22
CAPÍTULO II: ANÁLISIS Y DISEÑO DEL COMPONENTE CUADRO DE MANDO PARA	
XAUCE AKADEMOS.	24
II.1 Disciplina de modelado del negocio.....	24
II.2 Disciplina de requisitos.....	25
II.3 Disciplina de análisis y diseño	33
Conclusiones del capítulo.....	41
CAPÍTULO III: VERIFICACIÓN DE LA CALIDAD DEL COMPONENTE DE CUADRO DE	
MANDO XAUCE AKDEMOS PARA EL MINED.	42
III.1 Estrategia de pruebas.	42
III.2 Resultado de las pruebas.	44
Conclusiones del capítulo.....	52
CONCLUSIONES FINALES	53
BIBLIOGRAFÍA	54
ANEXOS.....	56

ÍNDICE DE TABLAS

Tabla 1 AUP- UCI.....	17
Tabla 2 Escenarios para la disciplina de requisitos	18
Tabla 3 Requisitos funcionales del componente	26
Tabla 4 Descripción del RF5. Mostrar cantidad de estudiantes.....	30
Tabla 5 Descripción del requisito RnF 9. El sistema debe funcionar de forma exitosa en un entorno de hardware y software.....	33
Tabla 6 Heurísticas de usabilidad	46
Tabla 7 Clasificación de las NC.....	51
Tabla 8 Descripción de variables del Caso de Prueba correspondiente al requisito Mostrar la cantidad de estudiantes.....	63
Tabla 9 DCP Mostrar la cantidad de estudiantes.....	64

ÍNDICE DE FIGURAS

Ilustración 1 Implicaciones de BI (Tikait, 2022)	7
Ilustración 2 Diferencia entre KPI y Métricas	10
Ilustración 3 Ejemplo creado con Data Studio por Cyberclick.	12
Ilustración 4 Soluciones que ofrece Classlife	13
Ilustración 5 Ejemplo del uso de Cronacademy	14
Ilustración 6 Clickedu Analytics.....	15
Ilustración 7 Arquitectura tradicional - Arquitectura Microservicio	35
Ilustración 8 Patrón arquitectónico N- capas de Grafana	37
Ilustración 9 Diagrama de clase del diseño	38
Ilustración 10 Diagrama de despliegue.....	41
Ilustración 11 Integración del cuadro de mando a XAUCE AKADEMOS para el MINED	45
Ilustración 12 Integración del cuadro de mando a XAUCE AKADEMOS para el MINED.	45
Ilustración 13 Cumplimiento de los indicadores de las categorías de usabilidad	49
Ilustración 14 Gráfico de no conformidades detectadas al realizar las pruebas.	50
Ilustración 15 NC nivel de aceptación	51
Ilustración 16 Diagrama de clases del dominio Control Docente.....	58
Ilustración 17 Diagrama de clases del dominio Plan de estudio	61
Ilustración 18 Diagrama de clases del dominio Titulación	61
Ilustración 19 Diagrama de clases del dominio Trámites docentes	62

INTRODUCCIÓN

Las Tecnologías de la Información y la Comunicación (TIC) han facilitado la interconexión entre las personas, así como entre instituciones a nivel mundial, eliminando barreras espaciales y temporales. Además, Estas ocupan un lugar primordial en el desarrollo social y económico de todo país debido a los múltiples beneficios que brindan y a la elevada cantidad de datos que son capaces de generar. (Few, 2006)

Las empresas a nivel mundial ~~se han visto tenido en~~ la necesidad de buscar en el almacenamiento de datos soluciones que ayuden a la toma de decisiones dentro del negocio, debido al alto nivel competitivo ~~con el que se cuenta en esta~~ existente en cada área. No obstante, las empresas no pueden darse el lujo de solamente operar y ser eficientes, deben ir un paso adelante de la competencia visualizando el futuro, con la ventaja competitiva que supone contar con información oportuna, certera y en tiempo real, para el análisis de datos y brindar respuestas inmediatas a las demandas de los clientes. (Few, 2006)

Las TIC tienen un papel muy relevante al permitir la recolección, almacenamiento y procesamiento de datos generados por la operación de las empresas. Así se Con ellas se facilita el acceso a la información y se reduce el margen de error ~~que pudiera puede existir al realizar una misma captura en varias ocasiones~~ (Few, 2006). Tan importante como Por lo que es vital ser cautelosos con la información que se genera ya que es igual de importante contar con la información, como es saber manejarla e interpretarla, por ello desde de aquí que en la década de los 90 surgieron las herramientas de análisis y reportes.

Las TIC ofrecen un potencial enorme de transformación de los sistemas educativos y de difusión de sus aplicaciones productivas; que lo cual se logra mediante una actividad sistemática de usos de las capacidades de la mente, ~~(conocida como conocimiento e investigación).~~ Lo anterior lo que hace crucial que su implementación y desarrollo académico demande una gestión administrativa con modelos organizativos, que impliquen calidad para que sus procesos y recursos interactúen con el uso intensivo de la tecnología. (Altamirano, 2017)

La Gestión Académica es aquel ámbito de acción está orientada a facilitar y mejorar los procesos formativos que imparten se desarrollan en las instituciones educacionales. Esto incluye la definición, implementación y seguimiento de procesos de mejoramiento continuo que les permitan responder y anteponerse a las necesidades formativas de sus estudiantes, a las exigencias del entorno, del mercado laboral y la política pública. (Chile, 2018)

Comentario [Y1]: Captura de qué???

Los [softwares](#) de gestión académica brindan mecanismos de control y gestión de la información en beneficio de los estudiantes, egresados, docentes y administrativos; logrando agilidad, oportunidad, seguridad y calidad en la información. Se caracterizan por ser un medio eficaz para lograr los objetivos institucionales y de los programas, facilitando así la acreditación y en especial entregar productos y servicios de calidad a la comunidad.

Estos aportan medios de gestión y control de la información que se desprende de los procesos institucionales y el [Reglamento-reglamento](#) de la [Institucióninstitución](#). Facilitan los procesos de autoevaluación, igualmente permiten llevar registro, control y seguimiento al ciclo de vida del estudiante cubriendo las principales etapas de los procesos académicos.

Con el fin de aprovechar los beneficios de la utilización de las nuevas tecnologías, surgen instituciones como la Universidad de las Ciencias Informáticas (UCI). Esta está estructurada en varios centros de desarrollo, entre los cuales se encuentra el Centro de Tecnologías para la Formación (FORTES) perteneciente a la Facultad de Tecnologías Educativas (FTE). En este centro se desarrolla el Sistema de Gestión Académica XAUCE AKADEMOS para el Ministerio de Educación de la República de Cuba(MINED).

Este sistema es una herramienta multiplataforma que 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 MINED. A pesar de las ventajas que supone su uso, actualmente este sistema no cuenta con un espacio donde se puedan visualizar los indicadores claves que intervienen en la consecución de los objetivos del proceso de gestión académica de los centros escolares.

Los indicadores claves constituyen métricas que ayudan a analizar el rendimiento de una determinada acción o estrategia. Para obtener información referente a uno de estos indicadores en este sistema, cada centro escolar debe:

- Acceder a cada una de las vistas que muestran listados relacionados con los datos que se necesiten analizar.
- Filtrar los listados por varios criterios de búsqueda.
- Cruzar los datos arrojados por cada una de las vistas consultadas.
- Realizar operaciones matemáticas.
- Representar la información.

En ese sentido, se ha corroborado que en la versión de XAUCE AKADEMOS para el MINED, resulta un proceso complejo y engorroso realizar una evaluación sobre el estado de dichos indicadores para darle soporte a la toma de decisiones de directivos y profesores de los centros escolares. Esto trae como consecuencia las siguientes debilidades a nivel de centro escolar:

- Alto riesgo de errores humanos que pudieran provocar la toma de decisiones basada en información incorrecta.
- Demora en la gestión y control del proceso de gestión académica.
- Inadecuada visualización de los indicadores, lo cual entorpece el seguimiento de los mismos, la creación de nuevas estrategias y el análisis de la información.
- Alta curva de aprendizaje que contribuye a la resistencia al cambio de directivos y profesores.

A partir del análisis de la situación anteriormente descrita se identificó el siguiente **problema científico**: ¿Cómo brindar soporte a la toma de decisiones basada en indicadores claves a nivel de centro escolar en XAUCE AKADEMOS?

Para brindar una solución acorde con la problemática planteada se centra el estudio de la investigación en el **siguiente objeto de estudio**: la inteligencia de negocios en los sistemas de gestión académica. Acotando. Dentro de este objeto se acota el **campo de acción** a la gestión de indicadores claves en los centros escolares del MINED.

En la presente investigación se define como **objetivo general**, ; desarrollar un componente de cuadro de mando basado en indicadores claves a nivel de centro escolar en XAUCE AKADEMOS para el MINED. Con su cumplimiento se pretende, a fin de obtener como **posibles resultados** un componente de cuadro de mando que permita brindar soporte a la toma de decisiones basado en indicadores claves a nivel de centro escolar en XAUCE AKADEMOS para el MINED.

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación para obtener los fundamentos y referentes teóricos sobre los cuadros de mando, así como de las tecnologías empleadas.

Con formato: Fuente: Negrita

2. Seleccionar las tecnologías a utilizar para el desarrollo del componente de cuadro de mando para brindar soporte a la toma de decisiones basado en indicadores claves a nivel de centro escolar en XAUCE AKADEMOS para el MINED.
3. Aplicar las fases, disciplinas y/o etapas ~~etcétera~~, según corresponda; de la metodología de desarrollo para ~~desarrollar la elaboración de~~ elaboración del componente.
4. Verificar la calidad del componente de cuadro de mando obtenido a partir del diseño y aplicación de pruebas de calidad de software ~~al componente de cuadro de mando~~.

Los **métodos científicos de investigación** que se utilizan con el fin de dar cumplimiento a los objetivos del presente trabajo son: (Hernández León, y otros, Noviembre 2002)

Métodos Teóricos

Analítico - Sintético: Permite la descomposición de un todo complejo en sus partes y cualidades. La síntesis, por su parte, establece la unión entre las partes, previamente analizadas y posibilita descubrir relaciones y características generales entre los elementos de la realidad.

Este método se utiliza en la investigación para la evaluación de soluciones que respondan al problema y permite realizar una valoración crítica y detallada de cada una de ellas. También para seleccionar las herramientas y tecnologías a utilizar durante el desarrollo de la herramienta.

Histórico - Lógico: El método histórico estudia la trayectoria real de los fenómenos y acontecimientos en el transcurso de su historia. El método lógico investiga las leyes generales del funcionamiento y desarrollo de los fenómenos. Se emplea para analizar la evolución de los conceptos asociados la inteligencia de negocios en los sistemas de gestión académica, permitiendo formar una definición propia. Se utiliza para realizar un análisis del estado del arte relacionado con la inteligencia de negocio, los cuadros de mando y los indicadores claves de desempeño, así como sus principales conceptos y elementos que los componen.

Métodos Empíricos

Observación: Se emplea para realizar valoraciones y obtener información a partir de lo observado. Se manifiesta principalmente en ~~la realización de observaciones~~ observación sobre

del funcionamiento de sistemas similares al desarrollar, dando una visión de cómo tiene que ser el sistema a realizar en su forma externa.

Técnica de recogida de información.

Entrevista: Se utiliza con el objetivo de entrevistar a algunos profesionales del proyecto para un mayor dominio del tema a investigar y para obtener algunas características y funcionalidades a tener en cuenta para el desarrollo del componente de control de mando para XAUCE AKADEMOS.

El presente trabajo de diploma está estructurado per-en tres capítulos:

Capítulo 1: “Fundamentos y referentes teóricos.”

En este capítulo se define el marco teórico de la investigación reflejado a través de los principales conceptos asociados a la temática definida. Se describen las principales herramientas homólogas más importantes que aportan conocimientos e información necesarios para el desarrollo del módulo. Además, se describen las tecnologías, herramientas y la metodología de desarrollo que se utilizan en la elaboración del componente para XAUCE AKADEMOS.

Capítulo 2: “Análisis y diseño del componente cuadro de mando para XAUCE AKADEMOS MINED.”

Se realiza la presentación de la solución del componente ~~propuesta del componente~~, así como los requisitos tanto funcionales como no funcionales que debe cumplir el mismo y los artefactos generados por la metodología de desarrollo de software seleccionada.

Capítulo 3: “Implementación y verificación de la calidad del componente cuadro de mando para XAUCE AKADEMOS MINIED.”

Describe las funcionalidades del componente de acuerdo a su desarrollo. Propone las pruebas de software a realizar y muestra los resultados obtenidos, con el fin de verificar la calidad de la solución propuesta.

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICOS

En el presente capítulo se hará referencia a una serie de elementos con el objetivo de conformar el marco teórico conceptual asociado al objeto de la investigación. Se analizarán soluciones similares con el fin de buscar aspectos que sirvan de base para el desarrollo del componente y que respondan a las necesidades del problema. Se describirán la metodología de desarrollo y las herramientas y tecnologías que serán empleadas en la propuesta de solución.

I.1 Conceptos asociados

Inteligencia de Negocios (Business Intelligence)

En 1865, Richard Millar Devens presentó la frase “*Business Intelligence*” (BI) en la *Cyclopædia of Commercial and Business Anecdotes*. Lo estaba usando para describir cómo Sir Henry Furnese, un banquero, se benefició de la información al recopilarla y actuar ante su competencia. (Foote, 2017)

Más recientemente, en 1958, un científico informático de IBM llamado Hans Peter Luhn escribió un artículo que describía el potencial de recopilar *Business Intelligence* mediante el uso de la tecnología. La inteligencia empresarial, tal como se entiende hoy, utiliza la tecnología para recopilar y analizar datos, traducirlos en información útil y actuar sobre ellos “antes que la competencia”. (Foote, 2017)

Esencialmente, la versión moderna de BI se enfoca en la tecnología como una forma de tomar decisiones de manera rápida y eficiente, con base en la información correcta en el momento correcto. (Foote, 2017) La inteligencia de negocios en resumen no es más que estrategias, acciones y herramientas enfocadas en administrar y crear ~~de~~ conocimiento mediante el análisis de datos de una organización o empresa. La BI analiza datos y brinda información procesable que ayuda a los gerentes y ejecutivos a tomar decisiones informadas. (Tikait, 2022)

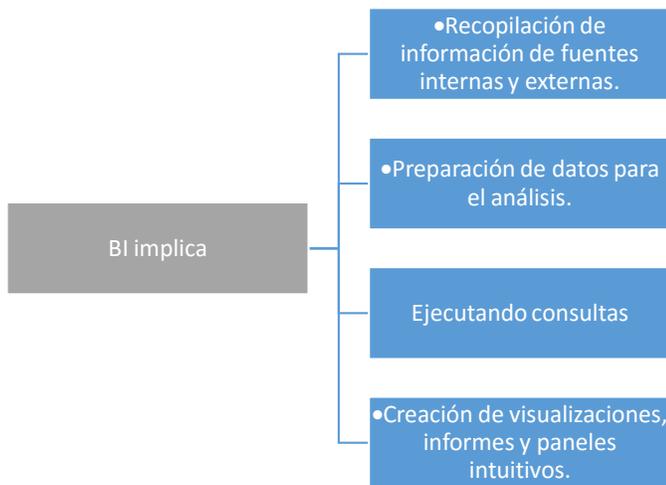


Ilustración 1 Implicaciones de BI (Tikait, 2022)

Procesos de inteligencia empresarial(Tikait, 2022)

- Los *Data warehouses* almacenan los datos de toda una organización, mientras que los *data marts* contienen subconjuntos de información para departamentos específicos.
- Los datos de BI incluyen información histórica y en tiempo real de diferentes fuentes, incluidos archivos de registro, texto y datos de sensores consolidados en almacenes. Puede analizar esta información para descubrir tendencias y patrones mientras realiza pronósticos precisos.
- Este proceso filtra información irrelevante, sopesa los riesgos actuales y futuros, realiza análisis de costo-beneficio y descubre acciones plausibles.
- Representar los resultados del análisis como visualizaciones sólidas mientras teje historias de datos convincentes.

Indicadores claves de desempeño (key performance indicator)

La definición de un indicador clave de rendimiento (KPI), es "un valor medible que se utiliza para evaluar qué tan exitosa es una persona u organización para alcanzar un objetivo". (Klipfolio, 2019) En las empresas, los KPI pueden ser de alto nivel o profundizar en un departamento o individuo específico. Los de alto nivel generalmente analizan el rendimiento

de su negocio como un todo. Cuando profundiza en procesos que son específicos de departamentos, equipos o individuos, esos son KPI de bajo nivel. ~~(Klipfolio, 2019)~~ ~~(Klipfolio, 2019)~~

Antecedentes y evolución

En el siglo III los emperadores de la dinastía Wei China (221-265AD) comenzaron a calificar el desempeño de los miembros oficiales de la familia. Con el transcurso de los siglos siguieron midiendo el desempeño con métodos bastantes rudimentarios y no es hasta el siglo XX, que el rendimiento de la gestión comenzó a medirse como un propio conjunto de procesos. ~~D~~ desarrollándose los conceptos de gestión científica (Frederick Taylor) y de estudios de tiempo y movimiento (Frank y Lillian Gilbreth). No fue hasta principios de la década de 1990, que el rendimiento se supervisó individualmente y se utilizó el primer cuadro de mando equilibrado verdadero (BSC) creado por el Dr. Robert Kaplan y el Dr. David Norton para monitorear el rendimiento. (Román, 2019)

En el año 1992 (febrero) Kaplan y Norton introdujeron el *Balanced Score Card* (BSC), presentando el concepto como una herramienta de medición del rendimiento, utilizado por las organizaciones para capturar además de las medidas financieras, las actividades de creación de valor de los activos intangibles de una organización. Los KPI y BSC fueron evolucionando hasta llegar a ser utilizados extensamente en todas las empresas y han avanzado mucho más allá de las mediciones simples que solían ser. Se cuenta con aplicaciones y tecnología para facilitar la toma de decisiones, representando una evolución constante. (Román, 2019). (Ver Anexo1 Infografía de la Evolución de los KPI)

Ventajas de los KPI(TWIN, 2022):

- *Informar a la gerencia sobre problemas específicos:* su enfoque basado en datos proporciona información cuantificable útil en la planificación estratégica y garantiza la excelencia operativa.
- *Ayudan a responsabilizar a los empleados:* en lugar de depender de sentimientos o emociones, los KPI están respaldados estadísticamente y no pueden discriminar entre empleados. Cuando se usan adecuadamente, los KPI pueden ayudar a alentar a los empleados, ya que los vendedores pueden darse cuenta de que sus números están siendo monitoreados de cerca.

- *Son el puente que conecta las operaciones comerciales y los objetivos reales.* Una empresa puede establecer objetivos, pero sin la capacidad de realizar un seguimiento del progreso hacia esos objetivos, hay poco o ningún propósito en esos planes. En cambio, los KPI permiten a las empresas establecer objetivos y luego monitorear el progreso hacia esos objetivos.

Limitaciones de los KPI(TWIN, 2022):

- *Es posible que se requiera un marco de tiempo prolongado para que proporcionen datos significativos.* Por ejemplo, una empresa puede necesitar recopilar datos anuales de los empleados durante años para comprender mejor las tendencias en las tasas de satisfacción durante largos períodos de tiempo.
- *Requieren un monitoreo constante y un seguimiento cercano para que sean útiles.* Un informe de KPI que se prepara, pero nunca se analiza no sirve para nada. Además, los KPI que no se monitorean continuamente para verificar su precisión y razonabilidad no fomentan la toma de decisiones beneficiosas.
- *Abren la posibilidad de que los gerentes "jueguen" con los KPI.* En lugar de centrarse en mejorar realmente los procesos o los resultados, los gerentes pueden sentirse incentivados a centrarse solo en mejorar los KPI vinculados a las bonificaciones por rendimiento. Además, la calidad puede disminuir si los gerentes están muy enfocados en los KPI de productividad, y los empleados pueden sentirse demasiado presionados para cumplir con las medidas de KPI específicas que simplemente no son razonables.

KPIs	Metrics
<ul style="list-style-type: none"> All KPIs are Metrics 	<ul style="list-style-type: none"> All Metrics are not KPIs
<ul style="list-style-type: none"> KPIs give a holistic view of the performance of different functions in your organization 	<ul style="list-style-type: none"> Metrics give you a picture of how different individual activities rolled out within the functions are progressing
<ul style="list-style-type: none"> KPIs tell you where exactly your teams stand with respect to the overall business goals 	<ul style="list-style-type: none"> Individual Metrics do not give any insights on their own
<ul style="list-style-type: none"> Examples: Pre-sales KPIs, Email Marketing KPIs, Customer Success KPIs 	<ul style="list-style-type: none"> Examples: Open Rate, Conversations in the last 2 weeks, Deals lost last quarter

Ilustración 2 Diferencia entre KPI y Métricas

Cuadro de mando

Un tablero o cuadro de mando es una pantalla visual de todos sus datos. Si bien se puede usar de muchas maneras diferentes, su intención principal es proporcionar información de un vistazo, como los KPI. (Adjust, 2022)

Un tablero generalmente se encuentra en su propia página y recibe información de una base de datos vinculada. En muchos casos, es configurable, lo que le permite elegir qué datos desea ver y si desea incluir tablas o gráficos para visualizar los números. (Adjust, 2022)

Antecedentes del cuadro de mando integral

En 1992, David Norton y Robert Kaplan presentaron por primera vez la metodología Cuadro de Mando Integral, en un artículo para el *Harvard Business Review*. El artículo original se basó en un proyecto de investigación multiempresa para estudiar la medición del desempeño en organizaciones cuyos activos intangibles eran un factor esencial en la ecuación de creación de valor. Aunque han pasado 17 años desde entonces, los principios presentados en la metodología del Cuadro de Mando Integral o *Balanced Scorecard* siguen teniendo gran relevancia. (Roncancio, 2020)

A finales de los 80 e inicio de los 90, se preparó el escenario, para pensar en un marco general mediante el cual, tanto los equipos ejecutivos de alto nivel, como los trabajadores de producción de primera línea, obtendrían la información financiera y no financiera necesarias para ejecutar sus labores. Kaplan planteo la evolución de la idea inicial de Cuadro de Mando Integral la cual se ha nutrido y transformado a medida que más organizaciones de diferentes naturalezas la adoptan. (Roncancio, 2020)

El *dashboard* o cuadro de mando es como los indicadores de la cabina de un avión. Se trata de una herramienta para obtener información de los datos y centralizar los KPI que se necesitan para saber lo que está pasando realmente con tu negocio. (Ortiz, 2022)

Un *dashboard* es una herramienta de gestión de la información que monitoriza, analiza y muestra de manera visual los ~~indicadores clave de desempeño (KPI)~~, métricas y datos fundamentales para hacer un seguimiento del estado de una empresa, un departamento, una campaña o un proceso específico. (Ortiz, 2022)

Estas son algunas de las características que debe tener este centro de control (Ortiz, 2022):

- *Personalizado*. Un dashboard debe contener únicamente los KPI que sean relevantes para el departamento, campaña o proceso que nos ocupa. Para orientarlo, podemos pensar en las preguntas principales a las que queremos responder. Por ejemplo, cuáles son las principales fuentes de tráfico a nuestra web, cómo está funcionando nuestro embudo de ventas o cuáles son los 5 productos que nos generan más ingresos.
- *Visual*. La idea de un dashboard es que podamos obtener la información que buscamos a golpe de vista. Por ello, los datos se presentan en forma de gráficos y debemos contar con indicadores rápidos a través de claves de color, flechas hacia arriba o abajo o cifras destacadas, por ejemplo.
- *Práctico*. La función principal de un dashboard siempre debe ser orientar las acciones de nuestro equipo. Por tanto, debe facilitarnos la información necesaria para que podamos saber cuáles son los siguientes pasos a seguir para mejorar los resultados.
- *En tiempo real*. A día de hoy, las acciones de marketing digital evolucionan con gran rapidez y aprovechar el momento clave es esencial. Por eso, la información debería estar actualizada al momento en todas las fuentes y mostrarse en el dashboard en tiempo real.

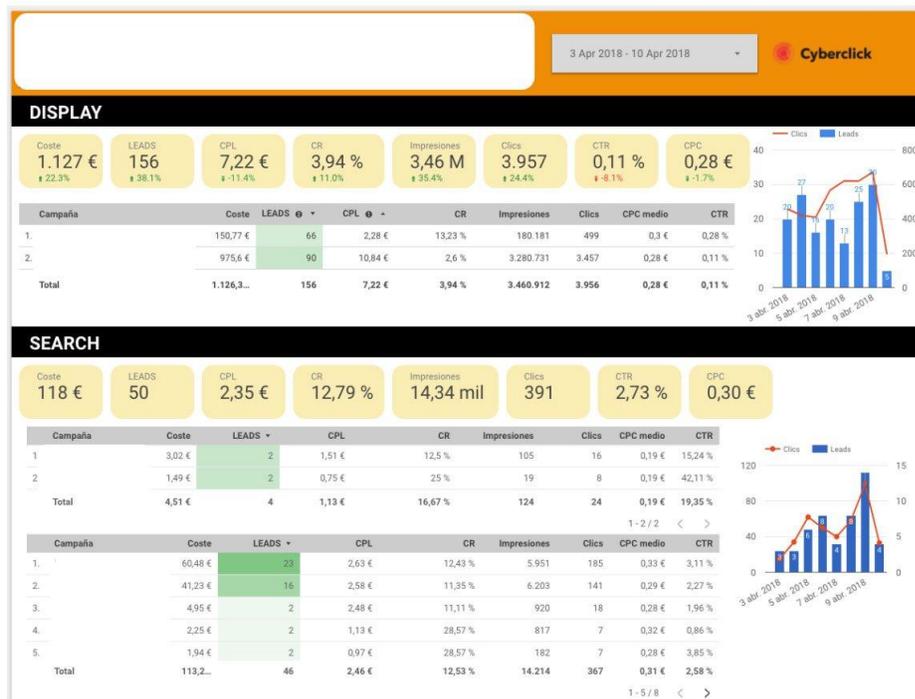


Ilustración 3 Ejemplo creado con Data Studio por Cyberclick.

I.2 Estudio de soluciones similares

Como parte del estudio ~~realizada~~ realizado se analizaron un conjunto de sistemas similares con el objetivo de identificar componentes que se ajusten a la solución que persigue XAUCE AKADEMOS para el desarrollo de su cuadro de mando.

Classlife

Esta plataforma SaaS (Software como servicio) facilita la administración de los centros educativos: desarrollo de los cursos, expedientes académicos, control de contactos y matrículas, gestión de cobros y pagos, asignación de tareas, calendarios y todo el resumen global del centro con reportes, gráficas y estadísticas. Desde su Campus Virtual, es posible conectar con todos los recursos formativos y de comunicación para ofrecer una mejor experiencia de aprendizaje y fomentar la participación de profesores y alumnos. Cuenta con

varias soluciones adaptadas a diferentes tipos de instituciones: universidades, institutos de educación superior, escuelas de negocios, institutos, centros de formación ocupacional, academias. (CLASSLIFE EDUCATION, 2022)

Se adapta a la realidad y a las necesidades de todo tipo de centros para ayudarlos a optimizar su gestión. *Classlife* integra dentro de una misma plataforma diferentes herramientas para que puedas gestionar el día a día de tu centro educativo sin duplicar esfuerzos y optimizando al máximo todos los recursos, dando cobertura a todas las áreas de gestión administrativa del centro. Entre sus principales herramientas se encuentran: (CLASSLIFE EDUCATION, 2022)

- *Listas Inteligentes*: Crea listas inteligentes en base a filtros detallados que muestra información actualizada.
- *Plantillas Dinámicas*: Sistema de plantillas con etiquetas dinámicas que permite personalizar cada documento generado en la plataforma.
- *Progresión en el Aula*: Visualiza el porcentaje de entregas realizadas en cada aula y progresión de los alumnos.
- *Control de Entregas*: Accede a informes detallados sobre las entregas programadas en cada aula.

¿Qué te ofrecemos?

Una plataforma educativa integral con múltiples soluciones.

Plataforma integral Conecta a todas las personas y procesos en un único espacio.	Gestión multi-sede Control centralizado de las diferentes sedes de trabajo del centro.	Cuadros de mando Seguimiento detallado de todo lo que pasa en tu centro en tiempo real.
Tu red social privada Toda la comunicación de tu centro en un espacio privado y seguro.	Documentación e Informes Genera todos los documentos, listados e informes para una gestión óptima.	Entorno multi-idioma Facilita la accesibilidad y comunicación con tus usuarios.
Módulos personalizados Entendemos que cada centro es único, puedes solicitar desarrollos a medida.	Integración con terceros Conecta con tus herramientas favoritas a través de nuestro Marketplace API.	Movilidad Accede desde cualquier lugar y en tiempo real a través de la APP Móvil.

Ilustración 4 Soluciones que ofrece Classlife

Cronacademy

Permite la digitalización de academias y centros de formación: desde la gestión económica, ventas e inscripciones online, hasta el servicio integrado de videoconferencias para ofrecer cursos online o mixtos online-presencial. Vinculada a la nube, se puede acceder a ella desde cualquier dispositivo y a cualquier hora. (Cronacademy, 2022)

Su principal objetivo es ayudar a las academias a maximizar las inscripciones de alumnos y todos sus servicios: por ejemplo, la gestión de recibos está vinculada a la venta online y a las inscripciones o matrículas. Por otro lado, cuenta con una emisión de certificados y diplomas a través de un código QR de verificación de forma automática. (Cronacademy, 2022)

Cronacademy permite gestionar tu centro con toda la información en una sola pantalla. Información al instante del rendimiento de tus cursos, la evolución de las inscripciones o la facturación. Herramientas inteligentes que te permiten tomar decisiones con datos reales y precisos. (Cronacademy, 2022)

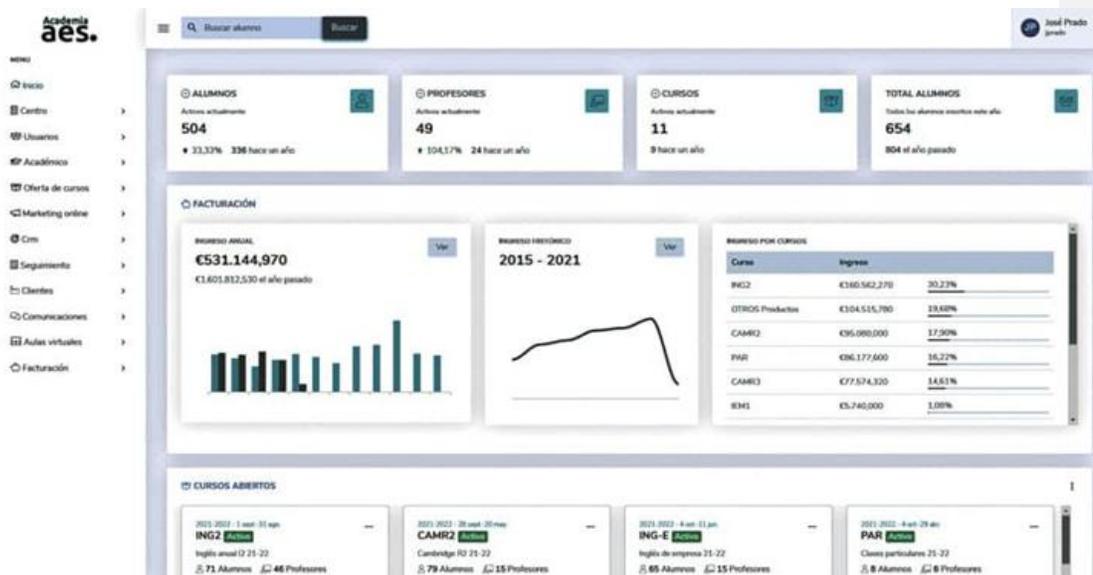


Ilustración 5 Ejemplo del uso de Cronacademy

Clickedu

Plataforma escolar en la nube que cuenta con una variada gama de funciones: gestión tutorial, académica, administrativa, económica y del profesorado; el entorno virtual de aprendizaje con conexión a contenidos propios o libros digitales; la comunicación con las familias; garantizar la calidad del centro; administrar las actividades extraescolares o el comedor; disponer de una tienda online donde las familias pueden comprar desde libros de texto hasta lotes de material escolar, reservar salidas del curso desde cualquier dispositivo con conexión a Internet. (EDUCACIÓN 3.0, 2022)

Con *Clickedu Analytics*, el nuevo módulo de la plataforma escolar en la nube, se pueden analizar distintos datos registrados en el centro educativo relacionados con el ámbito académico, financiero y de recursos humanos. Permite asistir la toma de decisiones a partir del análisis de los datos registrados de los diversos centros educativos que conforman un grupo y trabajan en Red. Con el desarrollo de esta nueva tecnología de Business Intelligence, el equipo directivo de una institución puede consultar en tiempo real los datos de uso de la plataforma para desarrollar y matizar los planes estratégicos. (EDUCACIÓN 3.0, 2022)



Ilustración 6 Clickedu Analytics

A partir del estudio de las soluciones existentes ~~abordadas~~ abordado anteriormente, se concluye que las mismas no posibilitan una solución factible al problema de la investigación, debido a que no se adaptan completamente a la plataforma. Al no representar una solución de código abierto presentan imposibilidad de copias, modificación y redistribución de la misma. Además, de forma general, las soluciones analizadas están desarrolladas para iniciar una plataforma de gestión académica desde cero, por lo que no servirían a las necesidades del componente, además de que para su uso es necesario pagar.

Por lo antes mencionado se sustenta la necesidad del desarrollo del componente de cuadro de mando. ~~Sin embargo, es importante destacar que las soluciones analizadas aportaron funcionalidades a tener en cuenta en el desarrollo de la propuesta de solución como la visualización y análisis de información en tiempo real y contar con interfaces intuitivas.~~

I.3 Metodología, lenguajes y tecnologías

Existe una amplia variedad de metodologías, lenguajes y herramientas que permiten la creación y el desarrollo de aplicaciones web. La selección de las que se van a utilizar para la solución que se propone está sustentada por la pertenencia del producto XAUCE AKADEMOS para el MINED al centro FORTES, por lo que existen políticas establecidas y definidas por la arquitectura de este producto. No obstante, se exponen un conjunto de características que ejemplifican por qué fueron escogidas para el desarrollo de XAUCE AKADEMOS.

Metodología de desarrollo a utilizar para el desarrollo del componente

Una metodología de software es un marco de trabajo que lo integran varios procedimientos agrupados con el objetivo de organizar, controlar y planear el proceso de desarrollo de un sistema. La misma está compuesta por técnicas, herramientas y artefactos que crean toda la base documental necesaria en el desarrollo de un software (Pressman, 2010).

Existen metodologías ágiles y robustas. Las ágiles son convenientes para guiar proyectos de escaso volumen que demanden una rápida implementación. Por otro lado, las robustas pueden ser empleadas para guiar el proceso de desarrollo de proyectos grandes o pequeños (Ramírez Martín, y otros, 2009).

La metodología empleada para el desarrollo de la presente investigación se basó en el método de desarrollo ágil: Proceso Unificado Ágil (AUP). ~~Esta es una versión simplificada de RUP~~

~~(Proceso Unificado de Software), que describe la forma de desarrollar aplicaciones de software de negocio, usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP.~~

En la UCI, se decide hacer una variación de esta metodología, de forma tal que se adapte al ciclo de vida definido para la actividad productiva en la universidad, logrando estandarizar el proceso de desarrollo de software. De las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición), se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola titulada Ejecución y se agrega la fase de Cierre (Rodríguez Sánchez, 2015).

A continuación, se muestra la ~~comparación entre de AUP y la variante AUP-UCI con relación a sus las fases y los objetivos específicos de las mismas para esta última (Ver Tabla 1)-.~~

Tabla 1 ~~Comparación entre AUP y AUP- UCI~~

Fases AUP-UCI	Objetivos de las fases (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. Se realizan estimaciones de tiempo, esfuerzo y costo para decidir si se ejecuta o no el proyecto.
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. Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del software.
Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Tomado de (Rodríguez Sánchez, 2015)

Con formato: Espacio Antes: 0 pto, Después: 0 pto

Con formato: Normal, Justificado, Interlineado: 1,5 líneas, No conservar con el siguiente, No ajustar espacio entre texto latino y asiático, Punto de tabulación: 0 cm, Izquierda

Tabla con formato

Escenarios para la disciplina Requisitos

El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto. Existen tres formas de encapsular los requisitos: [Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de Requisitos por Proceso (DRP)]; agrupados en cuatro escenarios condicionados por el Modelado de Negocio (Rodríguez Sánchez, 2015)

A continuación, se presentan los escenarios para la disciplina Requisitos de acuerdo a la variante AUP-UCI (Ver Tabla 2).

Tabla 2 Escenarios para la disciplina de requisitos

Escenarios	Características
Escenario No 1	Proyectos que modelen el negocio con Casos de Uso del Negocio solo pueden modelar el sistema con Casos de Uso del Sistema.
Escenario No 2	Proyectos que modelen el negocio con Modelo Conceptual solo pueden modelar el sistema con Casos de Uso del Sistema.
Escenario No 3	Proyectos que modelen el negocio con Diagrama de Procesos del Negocio solo pueden modelar el sistema con Descripción de Requisitos por Proceso.
Escenario No 4	Proyectos que no modelen negocio solo pueden modelar el sistema con Historias de Usuario.

Tomado de (Rodríguez Sánchez, 2015)

Dentro de la metodología se seleccionó el escenario número tres, ya que está definido en el proyecto al que pertenece la propuesta de solución. ~~Además de que e~~Este escenario se aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. ~~Se debe tener presente que e~~Este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados.

Herramientas y tecnologías a utilizar para el desarrollo del componente.

- Lenguaje de modelado UML 2.1

El Lenguaje Unificado de Modelado (UML) es un lenguaje estándar de modelado introducido en la informática en el año 1997 que sirve para escribir los planos del software. El UML sirve para especificar, visualizar, construir y documentar los artefactos de los sistemas software, así como para el modelado del negocio y otros sistemas no software (Object Management Group, 2001).

El UML puede usarse para modelar desde sistemas de información hasta aplicaciones distribuidas basadas en Web. Debido a su estandarización y su definición, aunque no sea un lenguaje de programación puede ser conectado de manera directa a lenguajes de programación como Java, C++ o Visual Basic. Mediante esta ingeniería directa se puede obtener el código fuente partiendo de los modelos, además es posible reconstruir un modelo en UML partiendo de la implementación ~~(, lo que se conoce como la~~ ingeniería inversa).

Este lenguaje tiene la capacidad de modelar actividades de planificación de proyectos y de sus versiones, expresar requisitos y las pruebas sobre el sistema, representar todos sus detalles, así como la propia arquitectura. Mediante estas capacidades se obtiene una documentación que es válida durante todo el ciclo de vida de un proyecto (Eriksson, 2000).

UML ofrece una serie de ventajas según (Fuentes Hernández, y otros, 2013), las cuales se enuncian a continuación:

- ✓ Es estándar, facilita la comunicación.
- ✓ Es independiente del proceso de desarrollo.
- ✓ Cubre las diferentes vistas de la arquitectura de un sistema mientras se evoluciona a través del ciclo de vida del desarrollo de software.
- ✓ Proporciona vistas de software (estáticas, dinámicas, etc.).
- ✓ Es un lenguaje de propósito general para el modelado orientado a objetos.

Para realizar los modelos del sistema propuesto se hará uso del Lenguaje Unificado de Modelado en su versión 2.1.

- *Herramienta CASE: Visual Paradigm for UML 15.1*

Las herramientas CASE (*ComputerAided/Assisted Software/System Engineering*) o (Ingeniería de Software Asistida por Computadora) son un conjunto de herramientas que brindan soporte a un enfoque de ingeniería en el desarrollo de software en alguna o en todas las fases de este proceso (Kendall, 2005).

Visual Paradigm es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (Rosales Morales, y otros, 2013).

Dentro de sus principales características se encuentran: posibilita el uso de varios idiomas, soporta aplicaciones Web, es un producto de calidad, es fácil de instalar, actualizar y tener gran compatibilidad. Es una herramienta profesional multiplataforma, que proporciona abundantes tutoriales de UML, así como demostraciones interactivas y proyectos de este lenguaje de modelado. Posee como peculiaridad sobre el resto de las demás herramientas que cuenta con una potente funcionalidad para la creación de interfaces de usuarios de las aplicaciones (Visual Paradigm International, 2010).

- *PostgreSQL 14*

Según (Moreno, 2012) un Sistema Gestor de Base de Datos (SGBD) ~~no es más que~~ una colección de datos interrelacionados y un conjunto de programas para acceder a los mismos. Permite el almacenamiento, manipulación y consulta de datos pertenecientes a una base de datos organizada en uno o varios ficheros. En el modelo más extendido (bases de datos relacionales) la base de datos consiste, de cara al usuario, en un conjunto de tablas entre las que se establecen relaciones.

Durante el transcurso de la investigación, se emplea PostgreSQL 12, el cual es un ~~sistema de gestión de bases de datos~~ SGBD objeto-relacional. Se encuentra distribuido bajo licencia BSD (*Berkeley Software Distribution: Licencia de Software Libre*) y tiene su código fuente disponible libremente. Es uno de los ~~sistemas de gestión de bases de datos~~ SGBD de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Esto ofrece como ventaja que un

fallo en uno de los procesos no afecte al resto, garantizando así que el sistema continúe funcionando (PostgreSQL, 2018).

Es multiplataforma, soporta múltiples transacciones e integridad de datos. Presenta una estabilidad alta, gran seguridad de los datos y facilidad de administración e implementación de estándares (Zea Ordóñez, y otros, 2017).

- *PGAdmin IV*

PGAdmin IV es la más popular y completa plataforma de administración y desarrollo de código abierto para PostgreSQL, ~~la base de datos~~ [el SGBD](#) de código abierto más avanzada del mundo. La aplicación puede utilizarse en Linux, Solaris, Mac OSX y Windows para administrar PostgreSQL 7.3 y superiores y funciona en cualquier plataforma.

Está diseñado para responder a las necesidades de todos los usuarios, desde escribir simples consultas SQL hasta crear bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita su administración. La aplicación también incluye un resaltado de sintaxis SQL y un editor de código del lado del servidor.(POSTGRESQL, 2012)

La conexión con el servidor se puede hacer a través de TCP / IP o Unix Domain Sockets (en plataformas *Unix), y puede utilizar encriptado SSL para la seguridad. No se requieren controladores adicionales para comunicarse con el servidor de base de datos. Desarrollado por una comunidad de expertos de PostgreSQL en todo el mundo, está disponible en más de una docena de idiomas y publicado bajo la licencia PostgreSQL (POSTGRESQL, 2012).

- *Grafana 9.2.2*

Grafana permite consultar, visualizar, alertar y comprender sus métricas sin importar dónde estén almacenadas. Crea, explora y comparte tableros y fomenta una cultura basada en datos. Torkel Ödegaard CGO, Co-founder de Grafana Labs expreso *“El objetivo al crear Grafana era hacer que las cosas que yo encontraba difíciles y otras personas encontraban difíciles, fueran más fáciles y accesibles. De esa forma, más personas podrían comenzar a instrumentar sus aplicaciones y crear tableros por sí mismos.”*.(Labs, 2022)

Grafana está escrita en Lenguaje Go (creado por Google) y Node.js LTS y con una fuerte Interfaz de Programación de Aplicaciones (API). Es una aplicación que ha venido escalando posiciones, con una comunidad entusiasta de más de 600 colaboradores bien integrados

(son 7 desarrolladores líderes -Torkel a la cabeza- y 5 a tiempo parcial para poder coordinar tal grupo de personas). Su código fuente está publicado en GitHub. (Olano, 2022)

Se destacan las siguientes características de Grafana (Olano, 2022):

- Conecta con todas las fuentes de datos posibles, comúnmente conocidas como bases de datos como Graphite, Prometheus, Influx DB, Elasticsearch, MySQL, PostgreSQL, y otros.
- Solución de código abierto que permite escribir complementos desde cero para su integración con varias fuentes de datos diferentes.
- Ayuda a estudiar, analizar y monitorear datos durante un período de tiempo, técnicamente llamado análisis de series de tiempo.
- Rastrea el comportamiento del usuario y de la aplicación; la frecuencia de errores que aparecen en producción o en un entorno previo a producción; el tipo de errores que aparecen y los escenarios contextuales al proporcionar datos relativos.
- Puede ser implementado localmente por organizaciones que no desean que sus datos se transmitan a un proveedor de nube por razones de seguridad y otras. Por otro lado, tener un modelo híbrido entre nube privada y pública.
- existen paneles, donde contienen una gama de opciones de visualización, como mapas geográficos, mapas de calor, histogramas, toda la variedad de cuadros y gráficos que una empresa normalmente requiere para estudiar datos.
- Un tablero contiene varios paneles individuales diferentes en la cuadrícula. Cada panel tiene diferentes funcionalidades.
- Provee una plataforma de complementos que facilita a todos los usuarios la creación de complementos de alta calidad. Diseñada para todos, desde usuarios avanzados que han estado usando Grafana durante años hasta personas que recién están comenzando con Grafana.

Conclusiones del capítulo

El estudio de las principales definiciones relacionadas con los cuadros de mando permitió sentar las bases para el desarrollo de la solución y la necesidad de la realización del presente trabajo. La búsqueda y el análisis de sistemas que utilizan los cuadros de mando evidenciaron que el uso de los mismos resuelve determinados problemas, pero se comprobó

que los sistemas homólogos existentes no constituyen un recurso para resolver la problemática de la investigación. Esto determina la necesidad de desarrollar un componente de control de mando para AKADEMOS MINED.

La selección de la metodología AUP variación UCI y el uso del lenguaje de modelado UML con el apoyo de la herramienta CASE Visual Paradigm para el desarrollo del módulo permitió la creación del entorno de trabajo para el desarrollo de la solución. Además, la utilización de la multiplataforma Grafana permitirá crear un cuadro de mando que ayudará a visualizar los indicadores claves de la plataforma acorde a las exigencias del sistema XUACE AKADEMOS para el MINED.

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL COMPONENTE CUADRO DE MANDO PARA XAUCE AKADEMOS.

En el presente capítulo se describe la solución propuesta mediante la identificación de los requisitos funcionales y no funcionales, la selección de la arquitectura adecuada y la representación de los artefactos generados en la etapa correspondiente al modelado del sistema. De estos últimos se presentan los respectivos modelos de diseño y de datos.

II.1 Disciplina de modelado del negocio

El modelado del negocio esta orientado a entender los procesos de negocio de una organización determinada. Su objetivo es comprender el funcionamiento del negocio que se va a informatizar para tener la seguridad de que el software desarrollado va a cumplir con su propósito. (Pressman, 2010)

Para la presente investigación se plantea el modelo conceptual como artefacto para describir el negocio y este a su vez consta de un diagrama de clases del dominio, ya que se desea modelar exclusivamente los conceptos fundamentales, que tiene como objetivo primario la presentación de la información de XAUCE AKEDEMOS.

Modelo Conceptual

Un modelo conceptual es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. El mismo se representa con un conjunto de diagramas de clases en los que no se define ninguna operación. En el modelo se pueden mostrar objetos del dominio o clases conceptuales, asociaciones entre las clases conceptuales y atributos de las clases conceptuales. Para su confección no son adecuados artefactos del software, como una ventana o una base de datos, a menos que el dominio que se esté modelando sea de conceptos de software, como un modelo de interfaces de usuario gráficas (LARMAN, 2003).

Modelo del Dominio

Según Sommerville (2016) un modelo de dominio es una definición de abstracciones de dominio, como políticas, procedimientos, objetos, relaciones y eventos que sirve como base de conocimiento sobre algún área problemática. De igual forma define un dominio como un problema específico o área de negocios donde se usan sistemas de software.

Se hizo un estudio de los ~~diagramas de diseño~~ modelos conceptuales realizados por los especialistas de XAUCE AKADEMOS para el MINED con el fin de obtener conocimientos sobre el funcionamiento de las demás partes de la plataforma en especial la de los componentes asociados a los ~~indicadores claves de desempeño~~ KPI utilizados en el cuadro de mando para lograr un mejor entendimiento de los mismos. En el Anexo 3 se pueden ver los Diagramas de ~~Clases del Diseño~~ Dominio correspondientes a los modelos conceptuales de Trámites docentes, Plan de estudio, Titulación y Control Docente.

II.2 Disciplina de requisitos

Los requisitos son la base de todos los proyectos que involucren el desarrollo de software. Al acordar los requisitos para el sistema, se proporcionan las bases para la planificación del desarrollo de un proyecto de software. La ingeniería de requisitos sirve para determinar las necesidades que un cliente tiene en la implementación o desarrollo de software. Estas necesidades se obtienen mediante el proceso de recopilación, análisis y verificación de la especificación de los requisitos del software de una manera completa y correcta (García Velázquez, 2016).

Según Pressman (2010) los requisitos de software constituyen las necesidades de los clientes, las funcionalidades y las restricciones que debe cumplir el sistema. Los mismos se clasifican en funcionales y no funcionales.

Técnica de captura de requisitos.

La captura de requisitos es la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. Para que este proceso sea eficiente y de manera precisa se han desarrollado técnicas, una técnica es una operación especial para recolectar, procesar o analizar los datos que se necesitan; un ejemplo de ello es la entrevista (González, et al., 2003).

La entrevista resulta una técnica muy reconocida dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Le permite al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural (ver Anexo 2) (Escalona, et al., 2002).

Especificación de requisitos

La especificación de requisitos consiste en describir de forma detallada los requisitos de software para poder implementar la solución propuesta. Según (Young, 2004) un requisito es un atributo necesario en un sistema, una afirmación que identifica a una capacidad, característica o cualidad de un sistema en orden de que este tenga valor y utilidad para un cliente o usuario. El propio autor clasifica los requisitos de software en funcionales y no funcionales.

Requisitos funcionales.

Los requisitos funcionales son los que definen las funciones que el sistema será capaz de realizar, ya que describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el qué y no el cómo se deben hacer esas transformaciones. Estos requisitos, al mismo tiempo que avanza el proyecto de software, se convierten en los algoritmos, la lógica y gran parte del código del sistema. (García Velázquez, 2016)

Definen una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requisitos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone que un sistema debe cumplir. Especifican lo que debe hacer o los servicios que debe proporcionar el sistema. Deben describir también cómo responderá el sistema ante estas distintas entradas, y su comportamiento frente a situaciones particulares. (Pressman, 2010).

Los requisitos funcionales identificados para el desarrollo del componente de control de mando para AKADEMOS MINED son los siguientes (Ver Tabla 3):

Tabla 3 Requisitos funcionales del componente

No	Requisitos	Descripción	Prioridad	Complejidad
RF1.	Representar cantidad de especialidades modalidad.	El sistema debe permitir mostrar la cantidad de especialidades existentes.	Baja	Baja
RF2.	Representar cantidad de especialidades	El sistema debe permitir mostrar la cantidad de especialidades	Baja	Baja

	modalidad por educación.	por modalidad.		
RF3.R	Representar cantidad de planes de estudio.	<i>El sistema debe permitir mostrar la cantidad de planes de estudio por modalidad de estudio.</i>	Media	Baja
RF4.R	Representar la cantidad de estudiantes.	<i>El sistema debe permitir mostrar la cantidad de estudiantes en el sistema hasta el momento. Debe permitir la opción de mostrar la cantidad de estudiantes por provincia, municipio, especialidad-modalidad, estado docente, sexo, raza, modalidad de estudio y tipo de curso.</i>	Alta	Alta
RF5.R	Representar la cantidad de estudiantes en proceso de traslado.	<i>El sistema debe permitir mostrar la cantidad de estudiantes en proceso de traslado.</i>	Alta	Baja
RF6.R	Representar la cantidad de estudiantes en proceso de cambio de educación.	<i>El sistema debe permitir visualizar la cantidad de docentes en proceso de cambio educacional.</i>	Media	Baja
RF7.R	Representar la cantidad de estudiantes pendientes de ubicación.	<i>El sistema debe permitir mostrar la cantidad de estudiantes en trámite de ubicación.</i>	Media	Media
RF8.R	Representar la cantidad de estudiantes registrados.	<i>El sistema debe permitir mostrar la cantidad de estudiantes registrados en el sistema.</i>	Media	Baja
RF9.R	Representar la canti-	<i>El sistema debe permitir mostrar</i>	Media	Baja

	dad de estudiantes de baja	<i>la cantidad de estudiantes de baja en el sistema.</i>		
RF10.F	Representar la cantidad de estudiantes graduados.	<i>El sistema debe permitir mostrar la cantidad de estudiantes graduados por curso escolar.</i>	Media	Media
RF11.F	Representar la el estado de los trámites docentes.	<i>El sistema debe permitir mostrar la cantidad de trámites de acuerdo a su estado.</i>	Media	Baja
RF12.F	Representar la cantidad de estudiantes por modalidad-especialidad.	<i>El sistema debe permitir mostrar la cantidad de estudiantes por modalidad- especialidad.</i>	Baja	Baja
RF13.F	Representar la cantidad de tramites por motivo.	<i>El sistema debe permitir mostrar la cantidad de tramites por motivos.</i>	Media	Baja
RF14.F	Representar la cantidad de estudiantes aprobados y suspensos.	<i>El sistema debe permitir mostrar la cantidad de estudiantes aprobados y suspensos.</i>	Alta	Alta
RF15.F	Representar el estado de los trámites docentes.	<i>El sistema debe permitir mostrar el estado de los trámites docentes.</i>	Media	Media
RF16.F	Representar cantidad de grupos de certificación.	<i>El sistema debe permitir mostrar la cantidad de grupos de certificación.</i>	Media	Media

Requisitos no Funcionales

Los requisitos no funcionales son restricciones en los servicios o funciones ofrecido por el sistema los cuales incluyen restricciones de tiempo, restricciones en el proceso de desarrollo y restricciones impuestas por los estándares. Estos a menudo se aplican al sistema como un todo en lugar de características o servicios individuales del sistema. Suelen ser más críticos

que los requisitos funcionales ya que los usuarios de un sistema generalmente pueden encontrar formas de evitar una función del sistema que realmente no satisface sus necesidades. A pesar de ello, no cumplir con un requisito no funcional puede significar que todo el sistema sea inutilizable. (Sommerville, 2016)

Para el desarrollo del cuadro de mando fueron definidos los requisitos no funcionales que se muestran a continuación:

- *Requisito de Funcionalidad*

RnF 1. El Sistema debe cumplir con el marco legal vigente relacionado con los procesos que gestiona.

- *Requisito de Eficiencia de desempeño*

RnF 2. El sistema debe responder las peticiones del usuario en un tiempo inferior o igual a 3 segundos.

- *Requisito de Usabilidad*

RnF 3. El sistema debe permitir que se accedan a las funcionalidades con 3 clics como máximo.

RnF 4. Las vistas del sistema deben ser homogéneas en cuanto a la distribución, colores y diseño de gráficos.

RnF 5. El sistema debe proveer retroalimentación o respuesta a cada una de las acciones realizadas por el usuario.

RnF 6. El sistema debe mostrar mensajes de error indicando la causa del mismo.

- *Requisito de Seguridad*

RnF 7. El Sistema debe garantizar que los usuarios accedan solo a aquellas acciones que pueden realizar según los roles y permisos que posean.

- *Requisito de Mantenibilidad*

RnF 8 Se deben analizar todos los cambios a realizar y su impacto

- *Requisito de Portabilidad*

RnF 9. El sistema debe funcionar de forma exitosa en un entorno de hardware y software.

Descripción de los requisitos funcionales

Los requisitos funcionales son la entrada esencial para realizar el análisis, diseño, implementación y pruebas del sistema. Es por ello que realizar la descripción de los mismos,

es de gran importancia. A continuación, se presenta la descripción del requisito funcional RF5 a partir del modelo propuesto por la Descripción de Requisitos por Proceso (Ver Tabla 4). El resto de las descripciones de los requisitos funcionales del módulo se encuentra en el artefacto “AKADEMOS_Componente_de_Control_de_Mando_de_Requisitos_de_Software”, el cual se encuentra adjunto a la presente investigación.

Tabla 4 Descripción del RF5. Mostrar cantidad de estudiantes.

Precondiciones	-El usuario debe poseer los permisos para acceder al tablero.
Flujo de eventos	
Flujo básico Mostrar la cantidad de estudiantes.	
1	Se visualiza el total de estudiantes registrados en la base de datos.
2	Se muestra un cajón de información con el total de estudiantes. Y permite, además, realizar las siguientes opciones de mostrar la cantidad de estudiantes por: <ul style="list-style-type: none"> a) Provincia b) Municipio c) Especialidad-modalidad d) Estado docente e) Sexo f) Raza g) Modalidad de estudio h) Tipo de curso
Postcondiciones	
1	Se muestra satisfactoriamente la cantidad de estudiantes.
Flujos alternativos	
Cantidad de estudiantes por provincia.	
1	El usuario selecciona la opción de filtrar provincia.
2	Se escoge la provincia de la cual se quiere conocer la información.
3	Se muestra dicha información en el cajón de información.
4	Concluye así el requisito.

Postcondiciones	
1	Se mostró satisfactoriamente dicha información.
Flujos alternativos	
Cantidad de estudiantes por municipio.	
1	El usuario selecciona la opción de filtrar por municipio.
2	Se escoge el municipio del cual se quiere conocer la información.
3	Se muestra dicha información en el cajón de información.
4	Concluye así el requisito.
Postcondiciones	
1	Se mostró satisfactoriamente dicha información.
Flujos alternativos	
Cantidad de estudiantes por especialidad - modalidad.	
1	El usuario selecciona la opción de filtrar por especialidad - modalidad.
2	Se escoge la especialidad – modalidad de la cual se quiere conocer la información.
3	Se muestra dicha información en el cajón de información.
4	Concluye así el requisito.
Postcondiciones	
1	Se mostró satisfactoriamente dicha información.
Flujos alternativos	
Cantidad de estudiantes por estado docente.	
1	El usuario selecciona la opción de filtrar por estado docente.
2	Se escoge el estado docente del cual se quiere conocer la información.
3	Se muestra dicha información en el cajón de información.
4	Concluye así el requisito.
Postcondiciones	
1	Se mostró satisfactoriamente dicha información.
Flujos alternativos	
Cantidad de estudiantes por sexo.	
1	El usuario selecciona la opción de filtrar por sexo.

2	Se escoge el sexo del cual se quiere conocer la información.
3	Se muestra dicha información en el cajón de información.
4	Concluye así el requisito.
Postcondiciones	
1	Se mostró satisfactoriamente dicha información.
Flujos alternativos	
Cantidad de estudiantes por raza.	
1	El usuario selecciona la opción de filtrar por raza.
2	Se escoge la raza de la cual se quiere conocer la información.
3	Se muestra dicha información en el cajón de información.
4	Concluye así el requisito.
Postcondiciones	
1	Se mostró satisfactoriamente dicha información.
Flujos alternativos	
Cantidad de estudiantes por modalidad de estudio.	
1	El usuario selecciona la opción de filtrar por modalidad de estudio.
2	Se escoge la modalidad de estudio de la cual se quiere conocer la información.
3	Se muestra dicha información en el cajón de información.
4	Concluye así el requisito.
Postcondiciones	
1	Se mostró satisfactoriamente dicha información.
Flujos alternativos	
Cantidad de estudiantes por tipo de curso.	
1	El usuario selecciona la opción de filtrar por tipo de curso.
2	Se escoge el tipo de curso del cual se quiere conocer la información.
3	Se muestra dicha información en el cajón de información.
4	Concluye así el requisito.
Postcondiciones	
1	Se mostró satisfactoriamente dicha información.

Requisitos especiales	
Asuntos pendientes	NA

Tabla 5 Descripción del requisito RnF 9. El sistema debe funcionar de forma exitosa en un entorno de hardware y software.

Atributo de Calidad	Portabilidad
Sub-atributos/Sub-características	Adaptabilidad
Objetivo	Utilizar el sistema de forma exitosa
Origen	Humano
Artefacto	Sistema
Entorno	<ul style="list-style-type: none"> • Sistema operativo: Ubuntu Server 20.04 x64 • Docker 19.03 • NFS • HDD: 100 GB • RAM: 16 GB • Microprocesador: 8 núcleos
Estímulo	Respuesta: Flujo de eventos (Escenarios)
1.a Acceder al sistema	
Se accede al sistema o a una funcionalidad del mismo.	Se muestra la interfaz correspondiente.
Medida de respuesta	
No se interrumpe el funcionamiento del sistema	

II.3Disciplina de análisis y diseño

El modelado del sistema permitirá de alguna manera visualizar el software que se va a construir. Es un medio para tratar con la complejidad propia de un proceso de construcción de software con un mayor nivel de abstracción. [\(Pressman, 2010\)](#)

Con formato: Fuente: (Predeterminado) Arial, Sin Cursiva, No revisar la ortografía ni la gramática

Diseños de Casos de Prueba

Un Diseño de Casos de Prueba (DCP) es, en ingeniería del software, un conjunto de condiciones o variables bajo las cuales un analista determinará si una aplicación o una característica de éstos son parcial o completamente satisfactoria. (Pressman, 2010). El DCP consiste en probar el sistema, se incluyen los datos de entrada y los resultados esperados. Estos parten de las descripciones de requisitos por procesos, poseen la finalidad de encontrar la mayor cantidad de definiciones en las funcionalidades implementadas. Además de comprobar que el sistema satisface las necesidades del cliente. (Sommerville, 2016)

En el Anexo 3 se presenta el DCP perteneciente a la DRP- Mostrar la cantidad de estudiantes en el sistema.

Modelo de diseño

El modelo de diseño se encarga de refinar la arquitectura definida en el modelo de análisis adaptándola al ambiente de implementación. Se requiere un modelo de diseño, ya que el modelo de análisis no es suficientemente formal para alcanzar el código fuente. Se obtendrá una abstracción del código fuente, es decir, una descripción de cómo debe estructurarse, administrarse y escribirse el código fuente (González, 2008).

Patrón arquitectónico: Arquitectura de Software y Patrón Arquitectónico

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos. Además, se refiere a las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos. (Arias, y otros, 2016)

Al diseñar una arquitectura de software se debe crear y representar componentes que interactúen entre ellos y tengan asignadas tareas específicas, además de organizarlos de forma tal que se logren los requisitos establecidos. Se puede iniciar con patrones de soluciones ya probados y utilizar modelos que han funcionado. Estas soluciones probadas se conocen como patrones arquitectónicos, que van de lo general a lo particular. (Valdés Fernández, et al., 2017). Para el desarrollo del componente se decide hacer uso de la arquitectura de software Microservicios y del patrón arquitectónico N capas.

Con formato: Fuente:
(Predeterminado) Arial, 12 pto

Con formato: Título 4, Espacio
Después: 0 pto

Arquitectura de software Microservicios

La arquitectura de microservicios (*Micro Services Architecture*) es un método de desarrollo de software que consiste en construir una aplicación como un conjunto de pequeños servicios, con operaciones bien definidas e independientes entre sí. Cada microservicio ejecuta su propio proceso y se encarga de implementar una funcionalidad completa del negocio. Puede estar programado en distintos lenguajes y usar diferentes tecnologías de almacenamiento de datos. A la hora de hacer un despliegue, cada servicio se hace de forma independiente. (Hiberus, 2021)

Las principales diferencias entre una estructura tradicional y una de microservicios, es que la tradicional se realiza de forma monolítica, es decir, todas las partes de la aplicación que podrían implementarse se encontraban en esa única aplicación de forma integral. Al contrario, una arquitectura microservicios descompone una aplicación en sus funciones principales. Cada función se denomina microservicio y se puede diseñar e implementar de forma independiente. Esto permite que funcionen de forma aislada a los demás servicios de la plataforma. (Hiberus, 2021)

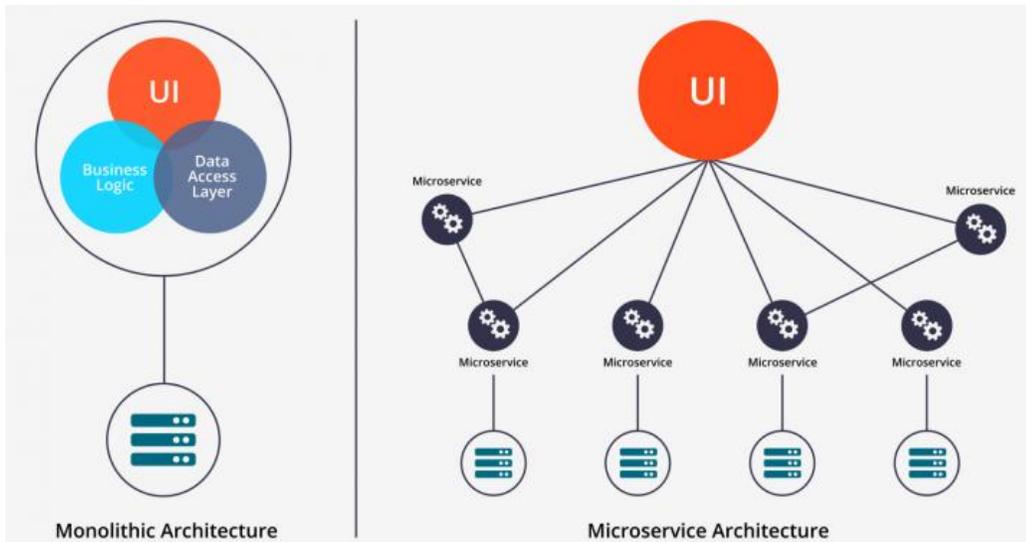


Ilustración 7 Arquitectura tradicional - Arquitectura Microservicio

Patrón arquitectónico N capas

Este tiene las ventajas que a continuación se describen según (Lizardo, 2010):

- Se puede entender una capa como un todo, sin considerar las otras.
- Las capas se pueden sustituir con implementaciones alternativas de los mismos servicios básicos
- Se minimizan dependencias entre capas.
- Las capas posibilitan la estandarización de servicios
- Luego de tener una capa construida, puede ser utilizada por muchos servicios de mayor nivel.

Las capas son una forma de separar responsabilidades y administrar dependencias. Cada capa tiene una responsabilidad específica. Una capa superior puede utilizar los servicios de una capa inferior, pero no al revés. Los niveles están físicamente separados y se ejecutan en máquinas diferentes. Un nivel puede llamar a otro nivel directamente o usar mensajería asincrónica (cola de mensajes). Aunque cada capa se puede hospedar en su propio nivel, esto no es necesario. Se pueden hospedar varias capas en el mismo nivel. La separación física de los niveles mejora la escalabilidad y la resistencia, pero también agrega latencia a la comunicación de red adicional. (Microsoft, 2022)

Una aplicación tradicional de tres niveles tiene un nivel de presentación, un nivel intermedio y una capa de datos. El nivel intermedio es opcional. Las aplicaciones más complejas pueden tener más de tres niveles. (Microsoft, 2022)

Grafana hace uso del patrón arquitectónico N capas haciendo uso de las capas que se muestran a continuación:

Con formato: Normal, Justificado

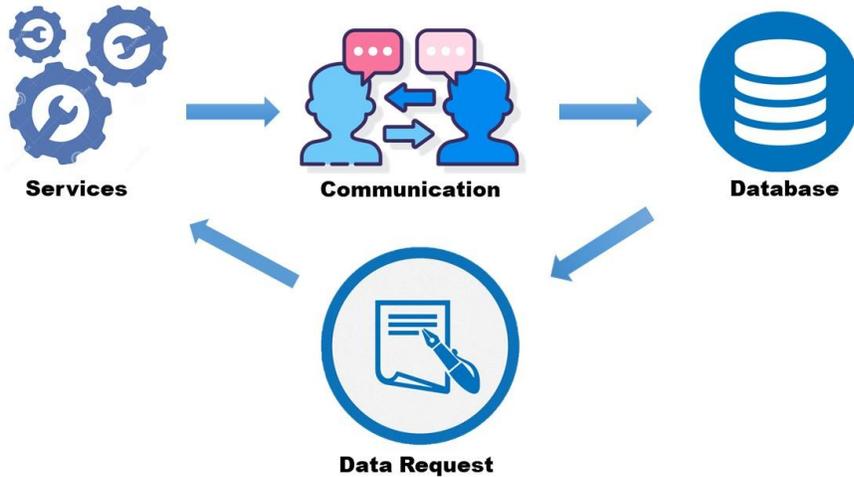


Ilustración 8 Patrón arquitectónico N- capas de Grafana

Capas que componen el patrón arquitectónico N-capas en Grafana:

- **Servicios:** Un servicio de encapsula y expone la lógica de la aplicación al resto de la aplicación, a través de un conjunto de operaciones relacionadas.
- **Comunicación:** Usa un bus para pasar mensajes entre diferentes partes de la aplicación. Toda la comunicación a través del bus ocurre sincrónicamente. Grafana estructura argumentos para servicios utilizando una separación de comando/consulta donde los comandos son instrucciones para una mutación y las consultas recuperan registros de un servicio.
- **Base de datos:** Usa una base de datos para conservar la configuración entre reinicios. De hecho, si no especifica uno, Grafana crea un archivo de base de datos SQLite3 en su disco local. Grafana admite las siguientes bases de datos: MySQL, PostgreSQL, SQLite3.
- **Solicitudes de datos:** maneja todas las solicitudes salientes de Grafana.

Diagrama de clases del diseño

A diferencia del modelo de dominio, un diagrama de clases de diseño muestra definiciones de entidades software más que conceptos del mundo real. Muestra la especificación para las

clases software de una aplicación, este diagrama incluye las asociaciones, atributos, métodos y dependencias de las clases, así como la navegabilidad (UML, 2016).

Entre los diagramas de UML los diagramas de clases del diseño ocupa un lugar destacado ya que son un tipo de diagrama estático que describen y estructuran gráficamente los datos asociados con el área de aplicación y el software (Arango Isaza, et al., 2016). Para el desarrollo del módulo se utilizan los diagramas de clases del diseño, ya que los mismos brindan a los desarrolladores una aproximación de lo que se desea implementar.

A continuación, se expone la representación del Diagrama de Clases del Diseño (Ver Figura 9). Esta representación contribuye a una correcta implementación del producto.

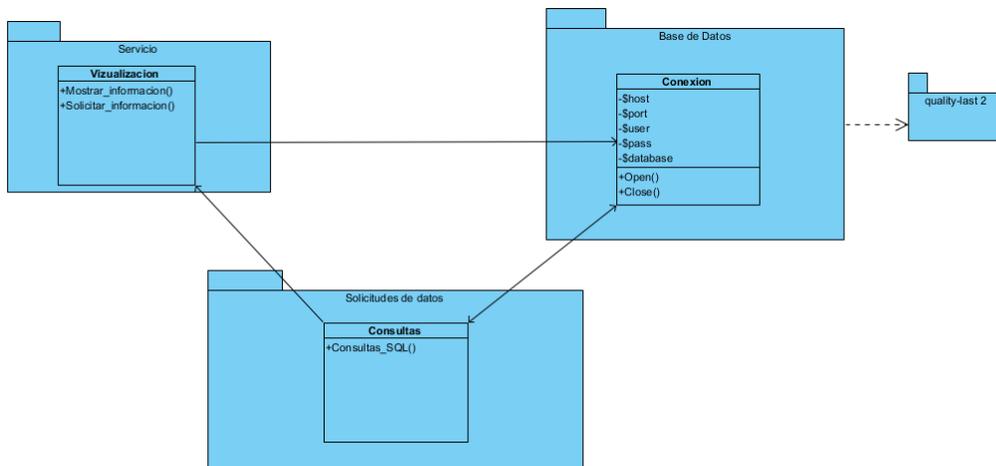


Ilustración 9 Diagrama de clase del diseño

Descripción de las clases

La clase visualización en la capa servicio se encarga de solicitar y mostrar la información en el tablero además de establecer la conexión con la base de datos a utilizar.

La clase conexión en la capa de datos se encarga de que Grafana establezca correctamente la conexión con el servidor de base de datos y la conservación de la configuración en cada reinicio.

La clase información obtiene información de la base de datos y recibe las solicitudes de la capa servicio.

La clase consultas es la encargada de almacenar las consultas SQL que se realizan a la base de datos y de enviar dicha información a través de los bus de comunicación a la capa servicio para que el usuario visualice la información deseada.

Patrones de diseño de software empleados en el desarrollo del módulo.

Los patrones de diseño son una descripción de problemas que ocurren en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces, así como la base para la búsqueda de soluciones a estos problemas, de tal modo que se pueda aplicar esta solución un millón de veces, sin repetir lo mismo dos veces. (González, 2016)

Patrones para la asignación de responsabilidades (GRASP).

Los patrones GRASP son una colección de principios de diseño orientados a objetos que guían la asignación de responsabilidades sobre los objetos y un intento de documentar lo que los diseñadores expertos probablemente conozcan intuitivamente. (Trellini, 2015). Para el desarrollo del componente se hizo uso de los patrones siguientes:

Controlador: Es el encargado de asignar a clases específicas la responsabilidad de controlar el flujo de eventos de la aplicación.

La utilización del patrón se evidencia en la clase Visualización, la cual controla el flujo de servicios.

Experto: Es el encargado de asignar responsabilidades a las clases que tienen la información

necesaria para cumplir dicha responsabilidad. De este modo se obtendrá un diseño manteniendo un bajo acoplamiento, incrementar la claridad, el encapsulamiento y la reusabilidad.

Este patrón se aplica en la clase Conexión, pues esta cuenta con la información suficiente para llevar a cabo una responsabilidad, que en este caso es establecer las conexiones.

Creador: Es el responsable de crear una nueva instancia de una clase con la información necesaria para realizar la creación del objeto. De este modo se ayuda al mantener un bajo acoplamiento, incrementar la claridad, el encapsulamiento y la reusabilidad (Gil, 2015).

El mismo se ve aplicado en la clase Consulta, la cual utiliza la información de la base de datos a través de la clase Información para crear objetos de ella.

Bajo Acoplamiento: Es el encargado de lograr una menor dependencia y bajo impacto de cambio entre las clases. De tal forma que, en caso de producirse una modificación en alguna

de ellas, no incremente el acoplamiento a un nivel que produzca resultados negativos típicos del alto acoplamiento entre las clases (Trellini, 2015) .

La utilización de dicho patrón se ve empleado en todo el diseño del componente ya que cada clase solo se relaciona con otras que necesite, de esta forma si se realiza algún cambio en algunas de ellas se afecte lo menos posible las otras clases.

Alta Cohesión: Es el encargado de que la información que contiene una clase debe de ser coherente, altamente relacionada con dicha clase y que la misma no realice una excesiva cantidad de trabajo. De este modo se garantiza una mejor claridad y facilidad de comprensión del diseño, se simplifica el mantenimiento y la introducción de mejoras, se logra una menor dependencia y se especifican los propósitos de cada objeto en el sistema (Sabará, 2015).

La aplicación de dicho patrón se ve reflejado en todo el diseño del componente ya que cada clase contiene la información y los métodos correspondientes a ellas y no se sobrecargan con información de otras clases.

Diagrama de despliegue.

El diagrama de despliegue es un tipo de diagrama del UML que muestra la arquitectura física de un sistema informático, este permite representar a los equipos y a los dispositivos y también mostrar sus interconexiones y el software que se encontrará en cada máquina. Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. (Delgado, 2015). En la siguiente figura se muestra el diagrama de despliegue del módulo desarrollado.

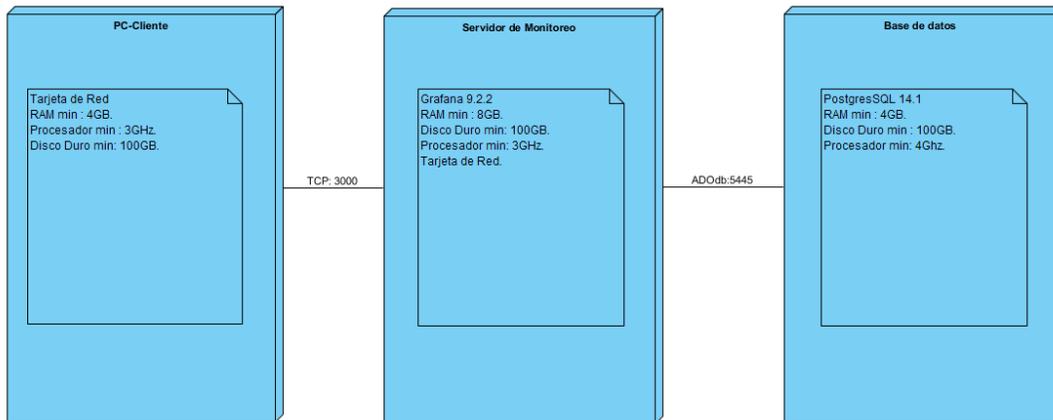


Ilustración 10 Diagrama de despliegue

Conclusiones del capítulo

La identificación de los requisitos del módulo, su modelado a partir de la descripción de requisitos y la descripción de casos de prueba propició un mejor entendimiento de la solución propuesta al equipo de desarrollo en cuanto a las funcionalidades y características que debe poseer. Los diagramas de clases del diseño sirvieron de guía para la implementación del módulo brindando una representación de la estructura que se desea implementar. La selección del patrón arquitectónico N-capas [contribuyó](#) al diseño de la solución, evidenciando la utilización de buenas prácticas durante el desarrollo de la solución propuesta.

CAPÍTULO III: VERIFICACIÓN DE LA CALIDAD DEL COMPONENTE DE CUADRO DE MANDO XAUCE AKDEMOS PARA EL MINED.

En el presente capítulo se define y realiza el proceso de aplicación de las pruebas para comprobar el correcto funcionamiento del componente con el objetivo de detectar y enmendar la mayor cantidad de errores posibles en la implementación de los requisitos funcionales del módulo.

III.1 Estrategia de pruebas.

Una estrategia de pruebas de software debe establecer pruebas de bajo nivel necesarias para verificar que un pequeño segmento de código fuente se haya implementado correctamente; así como pruebas de alto nivel que validan las principales funciones del sistema contra requerimientos del cliente. Estas integran los niveles, tipos, métodos y técnicas de diseño de casos de prueba en una serie de pasos bien planificados que llevan a la construcción correcta del software. Constituyen una parte fundamental del proceso de validación y verificación del software.

Comentario [SNP2]: Referenciar.

La verificación es una actividad que permite asegurar que las distintas partes del software cumplen con la función para la cual fueron diseñadas. En este sentido la verificación se encarga de revisar el funcionamiento de los módulos del software, mientras que la validación se encarga de comprobar que los módulos verificados cumplen con los requisitos que el cliente ha expresado. (Pressman, et al., 2015)

Niveles de pruebas.

Para la realización de las pruebas ~~los autores~~ definieron los siguientes niveles de pruebas que a continuación se describen según:

- Pruebas de Integración: Verifican la correcta interacción entre dos o más componentes que se desarrollan aisladamente en un sistema para comprobar que funcionan correctamente juntos, es decir, se prueba la interacción entre las distintas partes del software. (Báez, 2016)
- Pruebas de Aceptación: se efectúan cuando el cliente prueba de manera formal un sistema para decidir si debe aceptarse del proveedor del sistema, o si se requiere más desarrollo. (Sommerville, 2016)

Tipos de pruebas.

- Pruebas de Funcionalidad: Las pruebas funcionales son pruebas diseñadas tomando como referencia las especificaciones funcionales de un componente o sistema (lo que se va a testear, el software o una parte de él). Se realizan para comprobar si el software cumple las funciones esperadas y se enfocan sobre un conjunto de pruebas que intentan descubrir errores en el mismo. (Valdés Fernández, et al., 2017) .
- Pruebas no Funcionales: Para la realización de las pruebas no funcionales ~~al componente módulo de edición~~ los desarrolladores optaron por aplicar la siguiente prueba que a continuación se describe:
 - Pruebas de Usabilidad: se basan principalmente en determinar cuán bien el usuario podrá usar y entender la aplicación además de identificar las áreas de diseño que hacen al sistema de difícil uso para el usuario. (Maceo, 2015)

Con formato: Fuente: 12 pto

Métodos de pruebas.

- Método de Caja Negra: El método de caja negra pone a prueba la funcionalidad y el rendimiento del sistema para el cumplimiento de los requisitos funcionales documentados en la especificación del producto. Implica adicionalmente diseñar casos de prueba en los que el programa no actúe como está esperado que lo haga si se desea utilizar este método para encontrar todos los errores en el programa, el factor determinante es que las entradas sean aceptadas de forma adecuada y que se produzca una salida correcta; así como que la integridad de la información externa se mantenga. (López, 2015). Para la realización de este método se utiliza la técnica de Partición de Equivalencia. ~~Para la realización de este componente se optó por la utilización de la técnica de Partición de Equivalencia.~~
- Método de Usuarios Expertos: Los usuarios expertos contribuyen a las pruebas de usabilidad detectando errores del sistema, basando sus opiniones en su propia experiencia. (Ruis Eguino, et al., 2013). Para la realización de este método se optó por la utilización de la técnica de Evaluaciones Heurísticas.

Con formato: Fuente: (Predeterminado) Arial, 12 pto, Color de fuente: Énfasis 1

Con formato: Título 4 Car, Fuente: (Predeterminado) Arial, 11 pto, Sin Cursiva, Color de fuente: Automático, Español (España, internacional)

Código de campo cambiado

Con formato: Izquierda, Espacio Después: 0 pto, Agregar espacio entre párrafos del mismo estilo, Sin viñetas ni numeración

Técnicas de pruebas.

- *Partición de Equivalencia:* Esta técnica consiste en clasificar las entradas de datos del sistema en grupos que presentan un comportamiento similar. Las particiones también pueden definirse en función de las salidas de datos, valores internos, valores relacionados antes o después de ciertos eventos, y también para los valores que reciben las interfaces. Además, se definen pruebas para cubrir todos o parte de las particiones de datos válidos y datos inválidos. Es aplicable a entradas de datos realizadas por personas o vía interfaces con otros sistemas. (Terrera, 2017)
- *Evaluaciones heurísticas:* Una evaluación heurística es un método de inspección de usabilidad de software que ayuda a identificar problemas en la interfaz de usuario. Supone que los evaluadores examinan y juzgan su conformidad con los principios reconocidos los cuales son llamados también heurísticas de usabilidad. Estas generalmente se llevan a cabo por un pequeño grupo (de uno a tres) de probadores, los cuales de forma independiente examinan una interfaz de usuario y juzgan el cumplimiento de un conjunto de principios de usabilidad. (Prieto, et al., 2013)

III.2 Resultado de las pruebas.

La realización de **pruebas de integración**, se ejecutaron con la finalidad de validar la compatibilidad y el funcionamiento de las interfaces que comunican las diferentes partes que componen la solución informática. Para su realización se probó la unificación del componente con la plataforma XAUCE AKADEMOS para el MINED, empleando el método de integración ascendente, ejecutándose correctamente todas las funcionalidades del mismo como son el caso de la correcta integración de Grafana con la base de datos de la plataforma, así como la comprobación de que se permita acceder correctamente a los usuarios autorizados y no solo al administrador del sistema. El desarrollo de las pruebas garantizó que el componente presentaba un correcto funcionamiento una vez integrado al sistema.

Con formato: Abstract Text, Ninguno,
Espacio Antes: 0 pto, Después: 0 pto,
No conservar con el siguiente, No
conservar líneas juntas



Ilustración 11 Integración del cuadro de mando a XAUCE AKADEMOS para el MINED.



Ilustración 12 Integración del cuadro de mando a XAUCE AKADEMOS para el MINED.

~~Las pruebas funcionales al componente verificado que las funcionalidades implementadas se corresponden de acuerdo a los requisitos definidos. Para su realización se probó el componente, ejecutándose correctamente todas las funcionalidades del mismo; además, se mostraron satisfactoriamente la información deseada.~~

Con el fin de comprobar la usabilidad del sistema, se empleó las pruebas con usuarios expertos basada en la técnica evaluaciones heurísticas. Fueron seleccionados 3 especialistas. Estos especialistas interactuaron con el sistema y ejecutaron las funcionalidades para lo cual

Comentario [SNP3]: ¿NC por iteración? ¿Cantidad de NC? ¿clasificación de las NC detectadas?

fue diseñado, con el fin de evaluar la interfaz a través de las heurísticas de diseño descritas por Jakob Nielsen, en su libro Usability Engineering (Rojas Rodríguez, et al., 2017). El equipo de desarrollo fue tomando nota de los errores que tenía el sistema en función de sus funcionalidades. Estas heurísticas estuvieron definidas por 10 criterios dados por dicho autor, las cuales se muestran a continuación:

Tabla 6 Heurísticas de usabilidad

ID	Principio	Definición
1	Visibilidad del sistema	El sistema debe mantener siempre informado al usuario sobre lo que está ocurriendo, a través de retroalimentación dentro de un tiempo razonable.
2	Coincidencia entre el sistema y el mundo real	El sistema debería "hablar" el idioma del usuario, con palabras, frases y conceptos familiares, más que términos orientados al sistema. Seguir convenciones, de modo que la información parezca lógica y natural.
3	Control y libertad del usuario	Los usuarios eligen a veces funciones del sistema por error, y necesitarán una salida de emergencia, con opciones de deshacer y rehacer.
4	Consistencia y estándares	Los usuarios no deberían preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Se deben seguir las convenciones de la plataforma.
5	Prevención de errores	Mucho más adecuado que mostrar mensajes de error entendibles, es un diseño cuidadoso que evite la ocurrencia de problemas. Se deben eliminar estas situaciones presentando una opción de confirmación a los usuarios antes de que realicen la acción.
6	Minimizar la carga de memoria	Reconocimientos, más que recordatorios. Minimizar la carga de memoria de los usuarios mediante el uso de objetivos, acciones y opciones visibles. El usuario no debería recordar información de una parte del sistema a otra.

7	Flexibilidad y eficacia del uso	Aceleradores que pasan desapercibidos para los usuarios novatos, pero que deben agilizar la interacción con el sistema a los usuarios expertos. Debe facilitar a los usuarios la ejecución de acciones frecuentes.
8	Diseño estético y minimalista	El sistema no debe mostrar información que sea poco relevante o que raramente sea utilizada por el usuario.
9	Ayuda al usuario para reconocer, diagnosticar y recuperarse de errores	Los mensajes de error deben estar expresados en un lenguaje natural entendible a los usuarios (no en código o lenguaje máquina). Estos deben indicar de manera precisa el problema y sugerir una solución de forma constructiva.
10	Ayuda y documentación	Cualquier tipo de información debe ser fácil de buscar y estar centrado en la tarea del usuario. Las instrucciones deben consistir en una lista concreta no muy extensa de tareas a realizar.

Para llevar a cabo dicha prueba se ha establecido una lista de chequeo con un conjunto de requerimientos formulados en las 10 categorías. A continuación, se muestran los ejemplos para 2 de ellas:

Forma de uso Evaluación: El mismo se evalúa de 0 en caso de mal (cuando la respuesta al indicador sea "No") y 1 en caso que elemento revisado no presente errores (cuando la respuesta al indicador sea "Sí").

➤ *Visibilidad del sistema*

- Mantiene la homogeneidad de estilo con el resto de elementos de la plataforma.
- Se ubica en los lugares preestablecidos, sin romper con la composición estándar.
- Las etiquetas de los menús son descriptivas de cada una de las opciones.
- Favorecen la previsibilidad, usando términos que anticipen al usuario lo que se encontrará detrás.
- Tienen un tratamiento gráfico coherente según el tipo de etiqueta: color, tamaño y tipografía adecuados.

- Mantiene la alineación entre los campos para asegurar la armonía visual.

➤ *Diseño estético y minimalista*

- El orden de los elementos se ajusta a una línea visual que va de izquierda a derecha y de arriba a abajo.
- Los elementos se sitúan en la línea visual de forma jerárquica (cuanto más abarcan, antes se encuentran).
- Los elementos visuales (texto, tablas o gráficos) se adecuan a la paleta de colores establecida para la plataforma.
- Hay suficiente contraste entre el color del texto y el color de fondo, tanto en textos convencionales como dentro de tablas o diagramas.
- Se usan fondos de color blanco o tonos tenues (gris, crema, azul pastel) y nunca de colores vivos.
- Si se usan colores para representar datos u otras informaciones en diagramas, no se contrastan verdes sobre rojos o marrones, pues es la forma de daltonismo más habitual.

Luego de realizada la prueba apoyándose en los cuestionarios anteriores se pudo apreciar que, de un total de 12 indicadores de usabilidad, el módulo implementado cumple con 11 indicadores, cifra que representa el 95% de usabilidad para las funciones presentes. Para una mejor comprensión de los resultados obtenidos se han representado los mismos en la siguiente gráfica.

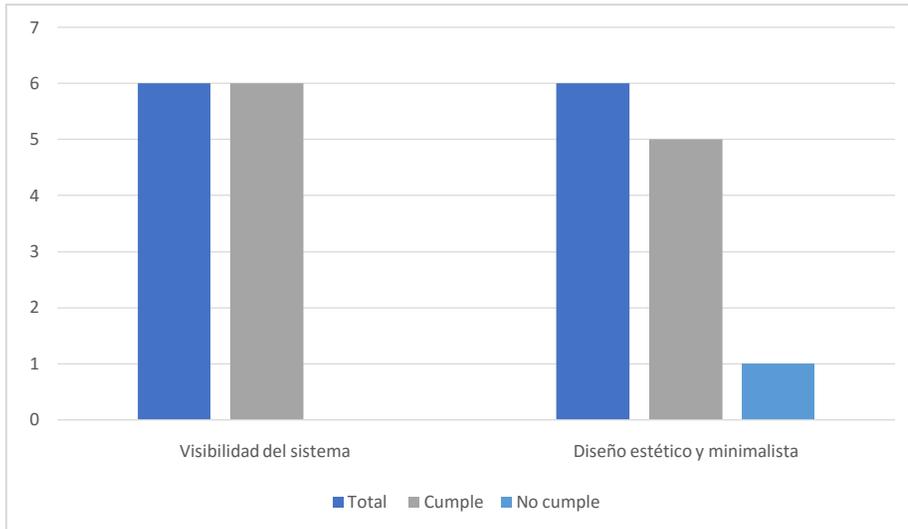


Ilustración 13 Cumplimiento de los indicadores de las categorías de usabilidad

Los resultados obtenidos fueron satisfactorios ya que los especialistas manifestaron que ~~el módulo el componente~~ presentaba una interfaz sencilla e intuitiva. Por último, los expertos involucrados en la prueba expresaron el estar satisfechos con el diseño del cuadro de mando.

Las pruebas de caja negra

funcionales son realizadas a la interfaz del software y aplicadas a las funcionalidades del mismo. ~~Como parte del proceso de dichas pruebas al módulo se diseñaron los casos de prueba en correspondencia con las descripciones de las historias de usuario~~ Se utilizan los DCP para, mediante la técnica de partición de equivalencia, elaborar los casos de prueba. En el Anexo 3 se muestra la DCP del requisito Mostrar cantidad de estudiantes.

En el proceso de aplicación de las pruebas se realizaron tres iteraciones utilizando el caso de prueba diseñado para detectar la mayor cantidad de errores en el funcionamiento del módulo.

Con formato: Fuente: 12 pto

Con formato: Fuente: (Predeterminado) Arial, 12 pto

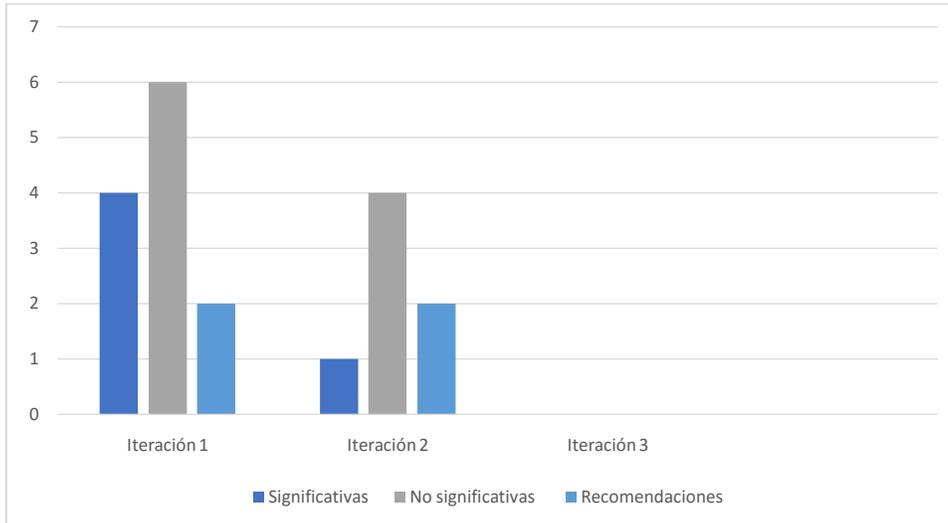


Ilustración 14 Gráfico de no conformidades detectadas al realizar las pruebas.

En la figura se evidencian las no conformidades detectadas al realizar las pruebas, estas fueron clasificadas atendiendo al nivel de afectación en el funcionamiento del módulo, en significativas, no significativas y recomendaciones. Al realizarse la primera iteración se detectaron 12 no conformidades, de las cuales 4 fueron significativas ya que afectaron el funcionamiento del componente, como no significativas se detectaron 3 errores ortográficos, 3 errores de mensajes y las 2 restante fueron recomendaciones para mejorar el cuadro de mando.

Concluida la corrección de los errores detectados en la primera iteración se le dio fin a esta y se inició una nueva, para detectar si existían otros errores. Una vez finalizada la segunda iteración se detectaron 7 no conformidades, 1 significativa, 3 errores ortográficos y 1 un error de mensaje en el caso de las no significativas y las restantes fueron recomendaciones para solucionar los errores. Al terminar de solucionar los errores detectados se finalizó la segunda iteración y se dio inicio a una tercera iteración para continuar comprobando si existían errores, al finalizar esta tercera iteración no se detectaron no conformidades.

Las pruebas de aceptación se caracterizan por la participación activa del usuario, que debe ejecutar los casos de prueba ayudado por miembros del equipo de pruebas. Están enfocadas

a demostrar que no se cumplen los requisitos, los criterios de aceptación o el contrato. Si no se consigue demostrar esto el cliente deberá aceptar el producto

En este nivel de pruebas se realizaron las pruebas funcionales donde el cliente pudo comprobar que la solución respondía a las necesidades acordadas. Una vez concluida la implementación de los componentes se detectaron un conjunto de no conformidades las cuales fueron clasificadas según su complejidad por lo que se definen de la siguiente forma:

Tabla 7 Clasificación de las NC

Clasificación de las NC	1ra Iteración	2da Iteración
Alta	3	0
Media	2	0
Baja	1	0
Total	6	0

Se consideran de clasificación Alta opciones que no funcionan en el sistema, de Media errores de tipo ortográficos y de clasificación Baja de redacción. Para un mejor entendimiento de los resultados, se muestra en el gráfico siguiente la cantidad de no conformidades detectadas por cada una de las iteraciones.

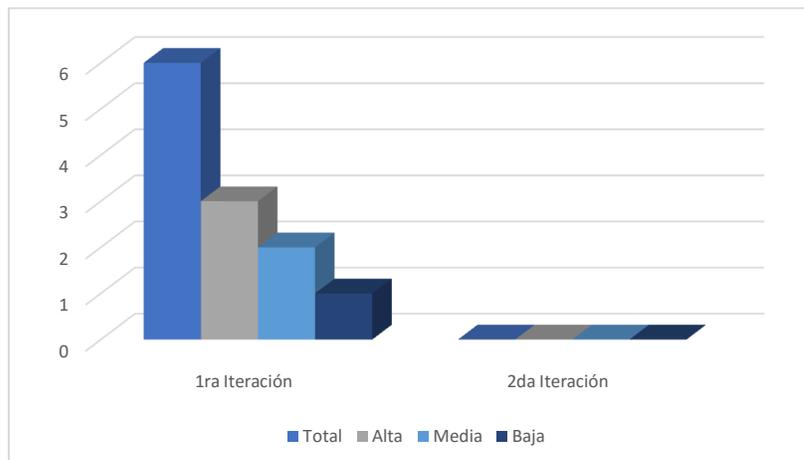


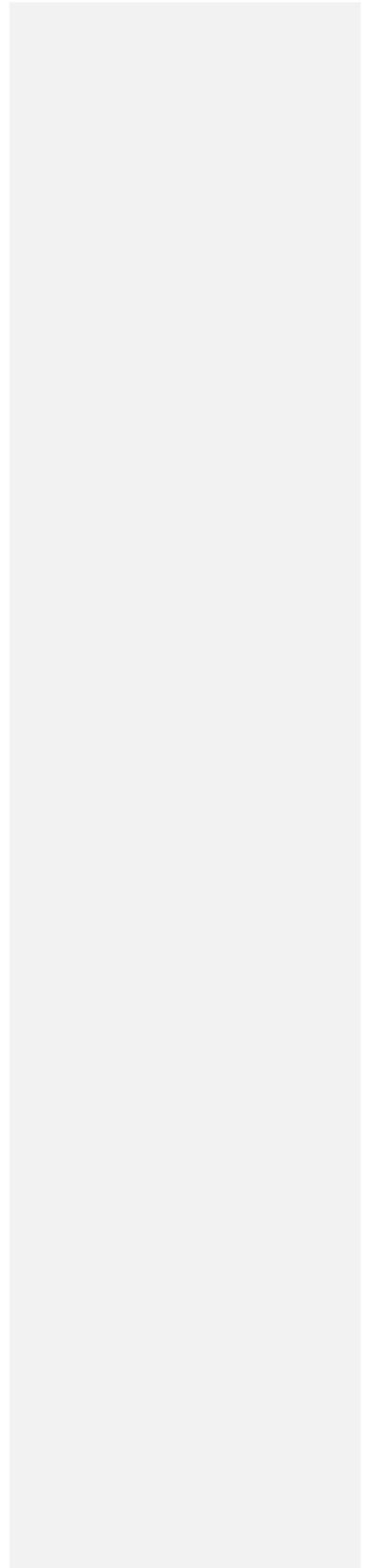
Ilustración 15 NC nivel de aceptación

Conclusiones del capítulo.

Como resultado del desarrollo del componente cuadro de mando para la plataforma XAUCE AKADEMOS para el MINED del centro FORTES se obtuvo un cuadro de mando que brinda a los usuarios la posibilidad de visualizar y monitorear la información, así como los datos asociados a estos. La solución posibilita una mejor gestión académica y de la información. Además de facilitar que las diferentes instituciones académicas puedan monitorear los resultados obtenidos y verificar el comportamiento de los principales KPI.

El componente ofrece una solución novedosa de gran utilidad para dar seguimiento y control de la "estrategia" que hace posible la alineación de los objetivos estratégicos con la gestión del día a día. Su empleo permite definir, claramente, el direccionamiento institucional, lo que posibilita analizar hacia dónde se dirigen las acciones diarias, en función de garantizar el cumplimiento de los objetivos a corto, mediano y largo plazo. Facilita la optimización de recursos y promueve el incremento del nivel de compromiso de cada uno de los centros para lograr sus objetivos.

|



CONCLUSIONES FINALES

Una vez finalizada la investigación y el desarrollo del presente trabajo se concluye que:

- El estudio de los conceptos asociados al objeto de estudio y de los sistemas similares permitió identificar los tipos de componentes apropiados para visualizar información de los indicadores gestionados en XAUCE AKADEMOS.
- La aplicación de la metodología de desarrollo AUP-UCI permitió que se obtuvieran los artefactos generados del proceso ingenieril, proporcionando comprensión a la solución desarrollada y favoreciendo la rigurosidad en su diseño.
- Con la implementación del módulo, se logró obtener un software que permite la representación de información referente a cada uno de los indicadores claves de AKADEMOS, a través del uso del cuadro de mando.
- La estrategia de pruebas definida y aplicada permitió identificar y resolver los errores en la implementación, garantizando el correcto funcionamiento del producto final. Se obtuvo así una aplicación que cumple con los requisitos definidos y que satisface las necesidades requeridas por el cliente.

BIBLIOGRAFÍA

- Adjust.** 2022. Adjust. [En línea] 2022. <https://www.adjust.com/glossary/dashboard/>.
- Altamirano, Manuel Jorge Espinoza.** 2017. Gale Academic OneFile. [En línea] Jan-June de 2017. <https://go.gale.com/ps/i.do?id=GALE%7CA629967436&sid=googleScholar&v=2.1&it=r&linkaccess=abs&issn=17282969&p=AONE&sw=w&userGroupName=anon~4921508b>. no. 39.
- Arias, Ángel y Durango, Alicia.** 2016. *Ingeniería y Arquitectura del Software: 2ª Edición*. s.l. : IT Campus Academy, 2016. ISBN: 978-1523365487.
- Báez, Olga Lucia Cárdenas.** 2016. *Automatización de casos de prueba para mejorar el proceso de calidad de software*. Bogotá : s.n., 2016.
- Chile, CNA-.** 2018. Comisión Nacional de Acreditación CNA-Chile. [En línea] 26 de septiembre de 2018. <https://www.cnachile.cl/noticias/Paginas/GESTI%C3%93N-ACAD%C3%89MICA.aspx#:~:text=La%20Gesti%C3%B3n%20Acad%C3%A9mica%20es%20aquel,las%20instituciones%20de%20educaci%C3%B3n%20superior..>
- CLASSLIFE EDUCATION, S.L.** 2022. CLASSLIFE . [En línea] 2022. <https://www.classlife.education/sistema-gestion-academica/>.
- Cronacademy.** 2022. Cronacademy. [En línea] 2022. <https://cronacademy.com/>.
- EDUCACIÓN 3.0.** 2022. Educaación3.0. [En línea] 2022. <https://www.educaciontrespuntocero.com/empresas/clickedu-analytics/>.
- Eriksson, Hans-Erik.** 2000. *Business Modeling with UML: Business Patterns at Work*. s.l. : John Wiley & Sons, 2000.
- Escalona, María y Koch, Nora.** 2002. *Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo*. Sevilla : s.n., 2002.
- Few, Stephen.** 2006. *Information Dashboard Design: The Effective Visual Communication of Data*. s.l. : O'Reilly Media, 2006. 1st Edición.
- Foote, Keith D.** 2017. DATAVERSITY. [En línea] 14 de September de 2017. <https://www.dataversity.net/brief-history-business-intelligence/#>.
- Fuentes Hernández, Julio César y Galeana Piedra, Selene Nallely.** 2013. *Sistema en línea para la evaluación docente en una institución educativa*. Distrito Federal, México : s.n., 2013.
- García Velázquez, Luis Ambrosio.** 2016. RECI Revista Iberoamericana de las Ciencias Computacionales e Informática. [En línea] julio de 2016. [Citado el: 23 de enero de 2019.] <http://reci.org.mx/index.php/reci/article/view/49/221>.
- , 2016. RECI Revista Iberoamericana de las Ciencias Computacionales e Informática. [En línea] Julio de 2016. [Citado el: 07 de Marzo de 2018.] <http://reci.org.mx/index.php/reci/article/view/49/221>.
- González, Roberto, Yll, Mario y Curriel, Lilian.** 2003. *Metodología de la Investigación Científica para las Ciencias Técnicas*. Matanzas : s.n., 2003.
- González, Zulma.** 2008. *Modelo de Diseño: Diseño y Patrones Arquitectónicos*. 2008.
- Hernández León, Rolando Alfredo y Coello González, Sayda.** Noviembre 2002. Cap5: Diseño Metodológico de la Investigación Científica. [aut. libro] Rolando Alfredo Hernández León y Sayda Coello González. "El Paradigma Cuantitativo de la Investigación Científica". La Habana : Editorial Universitaria, Noviembre 2002.
- Hiberus.** 2021. Hiberusblog. [En línea] 4 de marzo de 2021. <https://www.hiberus.com/crecemos-contigo/de-una-arquitectura-tradicional-a-microservicios/>.
- Kendall, K.** 2005. *Análisis y Diseño de Sistemas*. 2005.
- Klipfolio.** 2019. Klipfolio. [En línea] 2019. <https://www.klipfolio.com/resources/articles/what-is-a-key-performance-indicator>.
- Labs, Grafana.** 2022. Grafana. [En línea] 2022. <https://grafana.com/oss/grafana/?plcm=footer>.
- Lizardo, Maria Eugenia Arevalo.** 2010. [En línea] diciembre de 2010. <https://arevalomaria.wordpress.com/2010/12/02/introduccion-al-patron-de-arquitectura-por-capas/>.
- López, Oscar Andrés Peña.** 2015. *Formulación de propuesta para ejecución de pruebas funcionales en un sistema basado en PMI*. 2015.
- Maceo, Rafael Calzado.** 2015. *Sistema de Administración para el Sistema de Información Geográfica de la Universidad de las Ciencias Informáticas*. La Habana : s.n., 2015.

Maida, Esteban Gabriel y Pacienza, Julián. 2015. Metodologías de desarrollo de software: Tesis de Licenciatura en Sistemas y Computación. [En línea] 2015. [Citado el: 08 de Marzo de 2018.] <http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>.

Microsoft. 2022. Microsoft. [En línea] 2022. <https://learn.microsoft.com/es-es/azure/architecture/guide/architecture-styles/n-tier>.

Moreno, M. L. 2012. *Propuesta de arquitectura para la Línea de Productos de Software Aplicativos SIG.Geoinformática*. La Habana, Universidad de las Ciencias Informáticas : s.n., 2012.

Object Management Group. 2001. *Unified Modeling Language Specification*. 2001.

Olano, Jimmy. 2022. PandoraFMS. [En línea] 29 de septiembre de 2022. <https://pandorafms.com/blog/what-is-grafana/>.

Ortiz, Dany. 2022. Cyberclick. [En línea] 22 de septiembre de 2022. <https://www.cyberclick.es/numerical-blog/que-es-un-dashboard>.

PostgreSQL. 2018. *PostgreSQL-es*. [En línea] 2018. [Citado el: 25 de noviembre de 2018.] <http://www.postgresql.org/es/>.

POSTGRESQL. 2012. POSTGRESQL. [En línea] pgAdmin Introduction, 2012. [Citado el: 25 de noviembre de 2018.] <http://www.pgadmin.org/>.

Pressman, Roger S. 2010. *Ingeniería del Software, un enfoque práctico*. 2010.

Pressman, Roger S. 2010. *Ingeniería de Software Un enfoque Práctico*. 2010.

Pressman, Roger S. y Maxim, Bruce R. 2015. *Software engineering: a practitioner's approach, Eighth Edition*. New York : McGraw-Hill Education, 2015. ISBN 978-0-07-802212-8.

Pressman, Roger. 2010. *Ingeniería del Software. Un enfoque práctico*. Séptima Edición. s.l. : McGraw-Hill, 2010. ISBN: 978-607-15-0314-5.

Ramírez Martín y Rodríguez Donatien. 2009. *Sistema para la Identificación de Aguas en Pozos Petroleros (SIAPP)*. La Habana: Universidad de las Ciencias Informáticas : s.n., 2009.

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

Román, Ed. 2019. ER Gestión de Procesos. [En línea] 2019. <https://er-gestion.com.ar/mejora-continua/historia-de-los-key-performance-indicators-kpis-y-cuadro-de-mandos-integrales-o-equilibrados-cmi-o-balanced-scorecard-bsc/>.

Roncancio, Gabriel. 2020. Pensemos. [En línea] 2020. <https://gestion.pensemos.com/historia-cuadro-de-mando-integral-por-robert-kaplan>.

Rosales Morales, Y. M. C., Michael Eduardo y Trujillo Oliva, Adrian. 2013. *Extensión de la herramienta Visual Paradigm para la generación de clases de acceso a datos con Doctrine 2.0*. 2013.

Sommerville, Ian. 2016. *Software engineering: Tenth Edition*. s.l. : Pearson Education, 2016. 978-0-13-394303-0.

Tikait, Payal. 2022. SelectHub. [En línea] 28 de agosto de 2022. <https://www.selecthub.com/business-intelligence/business-intelligence-concepts/>.

TWIN, ALEXANDRA. 2022. Investopedia. [En línea] 17 de agosto de 2022. <https://www.investopedia.com/terms/k/kpi.asp>.

UML. 2016. UML. [En línea] 2016. [Citado el: 16 de Noviembre de 2017.] <http://www.uml.org/>.

Valdés Fernández, Yudeisy y García Prieto, Osmín. 2017. *Subsistema de planificación y gestión de canales virtuales para el sistema de gestión y transmisión de canales virtuales*. La Habana : s.n., 2017.

Vega Prieto, Roexcy, Rodríguez Luis, Zaylí y Justo Morell, Yaneisi Ofelia. 2015. Informática Jurídica. [En línea] 01 de Enero de 2015. [Citado el: 18 de Mayo de 2018.] <http://www.informatica-juridica.com/trabajos/procedimiento-para-realizar-pruebas-de-usabilidad/>.

Visual Paradigm International. 2010. *Analyst sends diagrams to ElaborView server*. [En línea] 2010. [Citado el: 25 de noviembre de 2018.] <http://www.visual-paradigm.com/product/vpuml/editions/standard.js>.

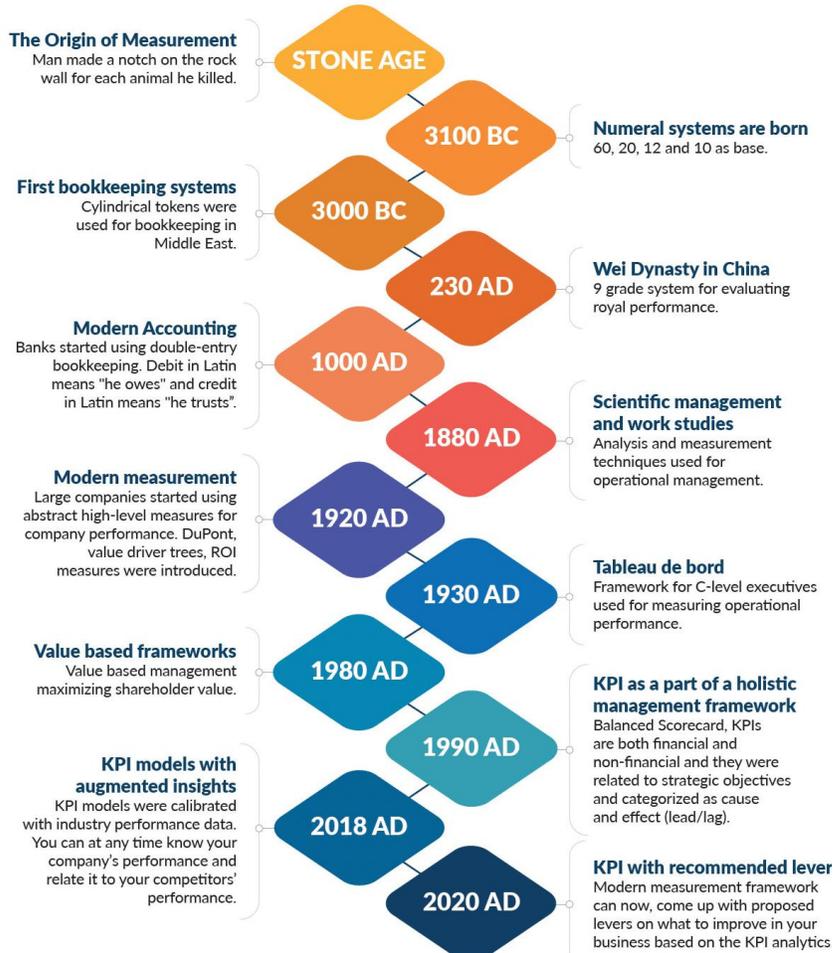
Young, Ralph. 2004. *The Requirements Engineering Handbook*. 2004.

Zea Ordóñez, Mariuxi Paola, Molina Ríos, Jimmy Rolando y Redrován Castillo, Fausto Fabián. 2017. *Administración de bases de datos con PostgreSQL*. Alicante : Área de Innovación y Desarrollo, S.L., 2017. ISSN: 978-84-946684-6-3.

ANEXOS

Anexo 1 – Histogramy the History of KPIs(Román, 2019)

THE HISTORY OF KPIs (KEY PERFORMANCE INDICATORS)



Anexo 2 Entrevista.

Para realizar la entrevista se seleccionó una muestra de los especialistas del proyecto XAUCE AKA-DEMOS para el MINED del centro FORTES.

Objetivo de la entrevista: "Obtención de los requisitos y la base tecnológica para la implementación de la solución".

Preguntas:

- ¿Qué es XAUCE AKADEMOS para el MINED?
- A su consideración ¿cuáles son los motivos que hacen necesaria la creación de un componente de cuadro de mando?
- ¿Qué herramientas se utilizan en el desarrollo de la plataforma y cuáles recomienda, además, para el desarrollo de dicho componente?
- ¿Qué funcionalidades y características considera que sean importantes para la implementación del componente?

Anexo 3 – Diagramas de Clase de Diseño de los componentes Control Docente, Trámite Docente, Plan de estudio y Titulación.

Con formato: Izquierda: 3
Derecha: 2,5 cm, Arriba: 2
Abajo: 2 cm, Ancho: 27,94
21,59 cm

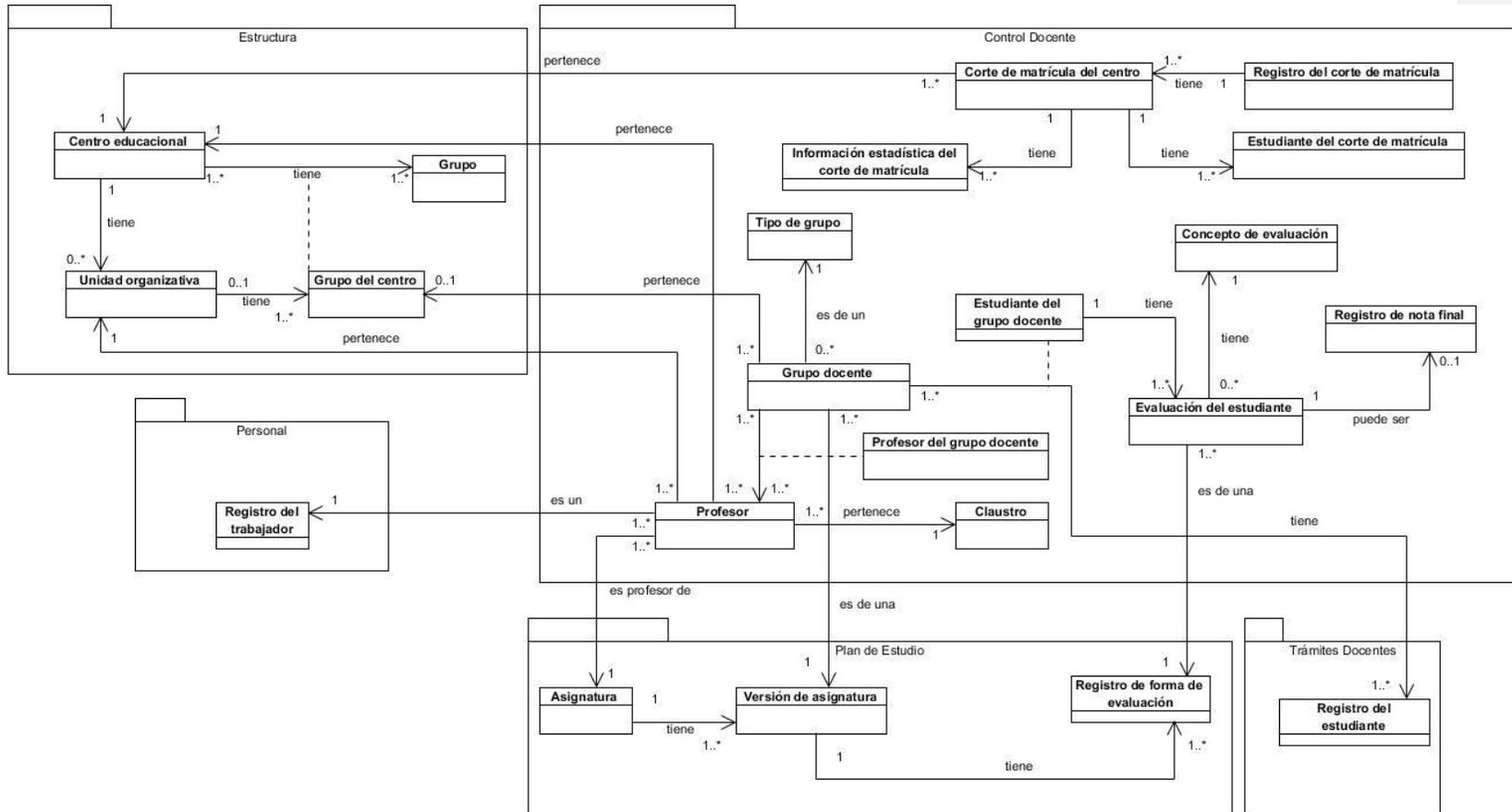


Ilustración 16 Diagrama de clases del dominio D-D Control Docente

Comentario [SNP4]: Diagrama de clases del dominio Control Docente con el resto

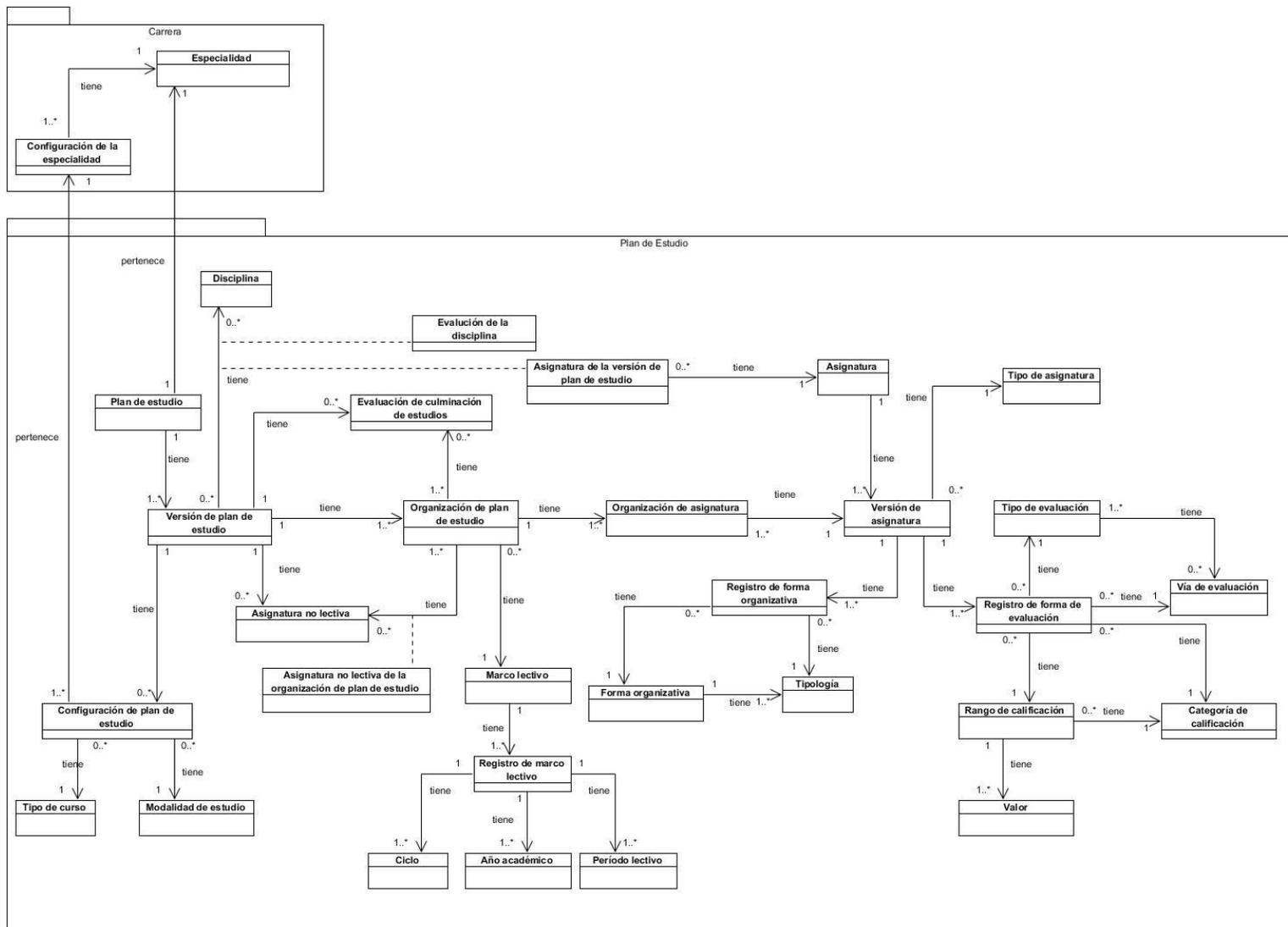


Ilustración 17 **Diagrama de clases del dominio D-D-Plan de estudio**

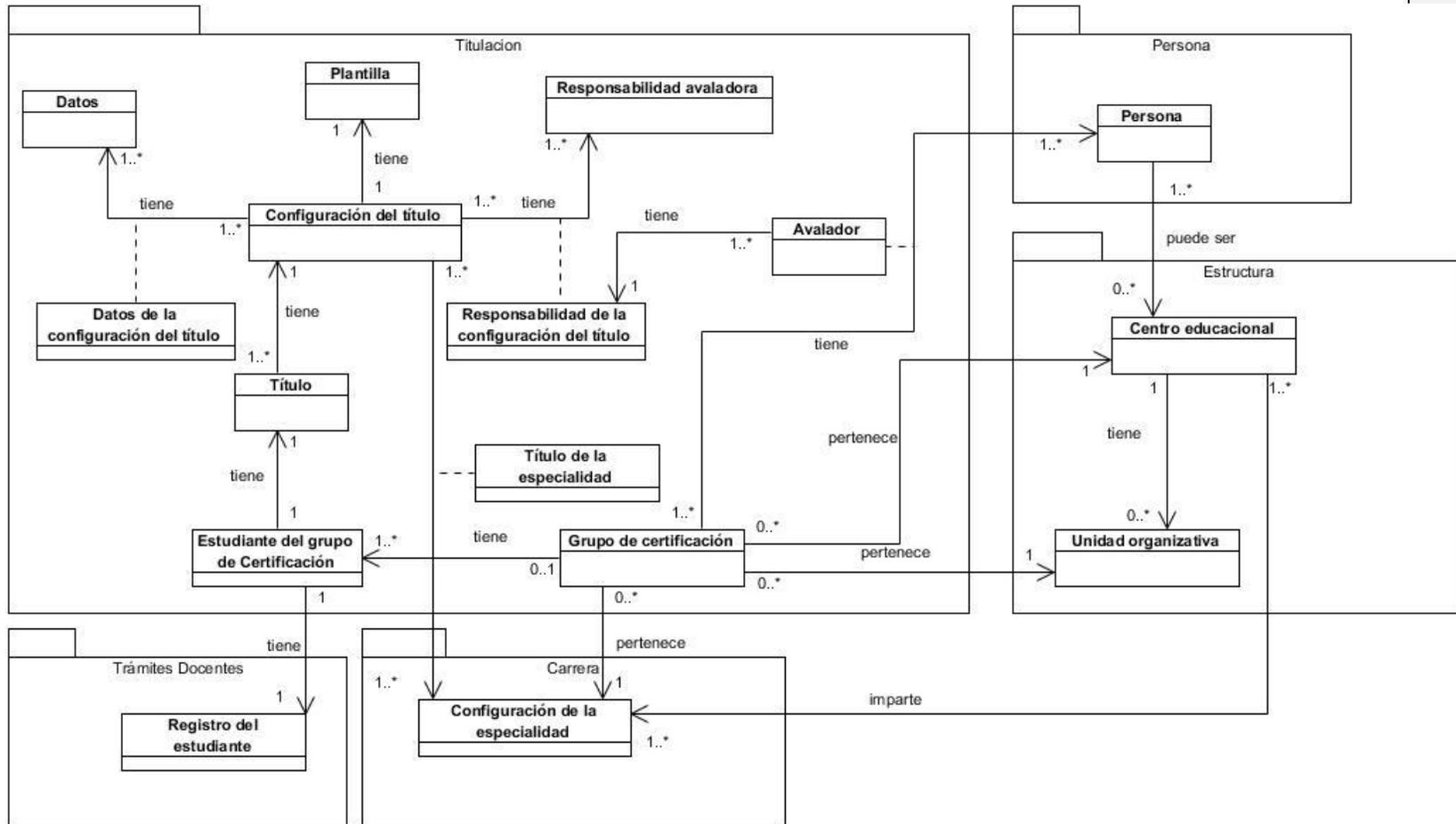


Ilustración 18 **Diagrama de clases del dominio D-D-Titulación**

Con formato: Izquierda

Tabla 8 Descripción de variables del Caso de Prueba correspondiente al requisito Mostrar la cantidad de estudiantes.

<u>No.</u>	<u>Nombre del Campo</u>	<u>Clasificación</u>	<u>Valor Nulo</u>	<u>Descripción</u>
<u>1</u>	<u>Filtro</u>	<u>Lista desplegable</u>	<u>No</u>	<u>Selección única de un listado de diferentes opciones para mostrar la cantidad de estudiantes.</u>
<u>2</u>	<u>Provincia</u>	<u>Campo de selección</u>	<u>No</u>	<u>Selección única que muestra la cantidad de estudiantes por provincia.</u>
<u>3</u>	<u>Municipio</u>	<u>Campo de selección</u>	<u>No</u>	<u>Selección única que muestra la cantidad de estudiantes por municipio.</u>
<u>4</u>	<u>Especialidad-modalidad</u>	<u>Campo de selección</u>	<u>No</u>	<u>Selección única que muestra la cantidad de estudiantes por especialidad-modalidad.</u>
<u>5</u>	<u>Estado docente</u>	<u>Campo de selección</u>	<u>No</u>	<u>Selección única que muestra la cantidad de estudiantes por estado docente.</u>
<u>6</u>	<u>Sexo</u>	<u>Campo de selección</u>	<u>No</u>	<u>Selección única que muestra la cantidad de estudiantes por sexo.</u>
<u>7</u>	<u>Raza</u>	<u>Campo de selección</u>	<u>No</u>	<u>Selección única que muestra la cantidad de estudiantes por raza.</u>
<u>8</u>	<u>Modalidad de estudio</u>	<u>Campo de selección</u>	<u>No</u>	<u>Selección única que muestra la cantidad de estudiantes por modalidad de estudio.</u>

Tabla con formato

Tabla 9DCP *Mostrar la cantidad de estudiantes.*

Escenario	Descripción	1	2	3	4	5	6	7	8	Respuesta del sistema	Flujo Central	
EC 1 <u>Mostrar la cantidad de estudiantes en el sistema.</u>	<u>El usuario visualiza la cantidad de estudiantes que se encuentran registrados en el sistema.</u>	NA	NA	NA	NA	NA	NA	NA	NA	<u>El sistema muestra el total de estudiantes. Y permite, además permite filtra dicha información.</u>	<u>Se muestra en el tablero.</u>	
		Vacío	vacío	vacío	vacío	vacío	vacío	vacío	vacío			
EC 2 <u>Filtrar por provincia.</u>	<u>El usuario visualiza la cantidad de estudiantes por provincia.</u>	V	NA	NA	NA	NA	NA	NA	NA	<u>El sistema muestra la cantidad de estudiantes por provincia</u>	<u>Tablero/Filtrar/Provincia</u>	
		X	Habana	vacío	vacío	vacío	vacío	vacío	vacío			
EC 3 <u>Filtrar por municipio.</u>	<u>El usuario visualiza la cantidad de estudiantes por municipio.</u>	NA	V	NA	NA	NA	NA	NA	NA	<u>El sistema muestra la cantidad de estudiantes por municipio</u>	<u>Tablero/Filtrar/Municipio.</u>	
		vacío	Florida	vacío	vacío	vacío	vacío	vacío	vacío			
EC 5 <u>Filtrar por estado docente.</u>	<u>El usuario visualiza la cantidad de estudiantes por estado docente.</u>	NA	NA	NA	NA	V	NA	NA	NA	<u>El sistema muestra la cantidad de estudiantes por estado docente.</u>	<u>Tablero/Filtrar/Estado docente.</u>	
		vacío	vacío	vacío	vacío	Matriculado	vacío	vacío	vacío			
EC 6 <u>Filtrar por sexo.</u>	<u>El usuario visualiza la cantidad de estudiantes por sexo.</u>	NA	NA	NA	NA	NA	V	NA	NA	<u>El sistema muestra la cantidad de estudiantes por sexo</u>	<u>Tablero/Filtrar/Sexo.</u>	
		vacío	vacío	vacío	vacío	vacío	F	vacío	vacío			
EC7 <u>Filtrar por raza.</u>	<u>El usuario visualiza la cantidad de estudiantes por raza.</u>	NA	NA	NA	NA	NA	NA	V	NA	<u>El sistema muestra la cantidad de estudiantes por raza</u>	<u>Tablero/Filtrar/Raza.</u>	
		vacío	vacío	vacío	vacío	vacío	vacío	Blanca	vacío			
EC8 <u>Filtrar por modalidad de estudio</u>	<u>El usuario visualiza la cantidad de estudiantes por modalidad de estudio.</u>	NA	NA	NA	NA	NA	NA	NA	NA	V	<u>El sistema muestra la cantidad de estudiantes por modalidad de estudio</u>	<u>Tablero/Filtrar/Modalidad de estudio.</u>
		vacío	vacío	vacío	vacío	vacío	vacío	vacío	vacío	Semi-presencial		

Con formato: Fuente: Sin C

Con formato: Justificado