

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



Facultad de Ciencias y Tecnologías Computacionales

**Tablero de Control para el Comité Central del Partido
Comunista de Cuba**

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

Autora: Elisabet De los Rios Díaz

Tutores: Ing. Reinier Fernández Coello

Ing. Aldis Joan Abreu Medina

La Habana, diciembre de 2023

Año 65 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo, Elisabet De los Rios Díaz, soy la autora del Trabajo de Diploma "**Tablero de Control para el Comité Central del Partido Comunista de Cuba**" y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración de autoría en La Habana, al _____ días del mes _____ del año 2023.

Elisabet De los Rios Díaz

Autora

Ing. Reinier Fernández Coello

Tutor

Ing. Aldis Joan Abreu Medina

Tutor

DEDICATORIA

Con profunda gratitud y amor sincero, dedico esta investigación al abnegado espíritu de mi querida madre, Cristina Díaz Gregorich. Su apoyo incondicional, su sabiduría perdurable y su aliento sin límites han sido cruciales. A mis queridos hijos, Jerson Culo de los Ríos y Asly Amalia Antonio de los Ríos, les ofrezco este trabajo como testimonio del poder transformador de los lazos familiares. Su radiante presencia y su confianza en mis capacidades me han servido como fuente perpetua de inspiración, impulsándome hacia adelante a través de los rigores intelectuales que caracterizan esta importante búsqueda académica.

AGRADECIMIENTOS

Quiero agradecer de manera muy especial a mi madre Cristina Díaz por el apoyo que me ha dado todos estos años, sin ella no hubiera sido posible que realizara este sueño.

También quiero agradecer a mis hijos por el tiempo que no les dediqué.

A mi querida amiga, Ivonne de la Caridad Collada Peña, por su apoyo incondicional.

A mi querida prima que es como la hermana que nunca tuve Neysis Hernández Díaz que siempre me guio y me incentivó a estudiar, a ser mejor persona y a no dejar de luchar por mis sueños.

A todos y cada de mis profesores por sus enseñanzas, a mi tutor Reinier Fernández quiero agradecerle por el apoyo, la paciencia y entrega con que guio mis pasos.

A mis amistades y a todas las personas que de una forma u otra estuvieron ahí para mí.

RESUMEN

Para las organizaciones es crucial la gestión de grandes cantidades de datos para tomar decisiones sobre su desempeño o cumplimiento laboral. En Cuba, ejemplo de ello es el Partido Comunista de Cuba, organización política fruto genuino de la Revolución, la cual lleva la gestión de varios procesos, entre ellos: política de Cuadro y vida interna. Con el fin de proveer un sistema capaz de procesar toda la información a gestionar en estos procesos, surge un proyecto de cooperación entre la Universidad de las Ciencias Informáticas, el Centro de Informatización Gobierno Empresa y el Partido Comunista de Cuba. A pesar de estos avances, toda la información y datos necesarios para realizar el análisis del comportamiento de algunos indicadores se realiza de forma manual. Esto provoca que se afecte el monitoreo de los datos asociados a los indicadores de interés por esta organización. Por tal motivo la presente investigación tiene como objetivo representar los datos de los procesos de vida interna y política de Cuadro en un tablero de control, de forma tal que contribuya a su monitoreo en tiempo real. De ahí que el progreso de la solución propuesta está guiado por la metodología Proceso Unificado Ágil, en su variación para la Universidad. Además, se utilizan tecnologías según la soberanía tecnológica a la que aspira el país. Una vez obtenida la solución, se validó a través de métricas y pruebas de software, corroborando que ésta cuenta con la calidad requerida para su posterior utilización por los funcionarios y estadísticos del Partido.

Palabras claves: indicadores claves del desempeño, monitoreo, tablero de control.

ABSTRACT

For organizations, it is crucial to manage large amounts of data to make decisions about their performance or labor accomplishment. In Cuba, an example of this is the Communist Party of Cuba. In this organization, several processes are managed, among them: the cadre policy and the internal life process. In order to provide a system capable of processing all the information to be managed in these processes, a cooperation project arose between the Universidad de las Ciencias Informáticas, the Centro de Informatización Gobierno Empresa and the Communist Party of Cuba. Despite these advances, all the information and data needed to analyze the behavior of some indicators is done manually. This affects the monitoring of the data associated with the indicators of interest for this organization. For this reason, the objective of this research is to develop a control panel for this organization, to contribute to the monitoring of data in real time. Hence, the development of the proposed solution is guided by the Agile Unified Process methodology, in its variation for the University. In addition, technologies are used according to the technological sovereignty to which the country aspires. Once the solution was obtained, it was validated through metrics and software tests, corroborating that it has the required quality for its subsequent use by the Party officials and statisticians.

KEYWORDS: *control panel, key performance indicators, monitoring.*

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICOS-METODOLÓGICOS.....	6
I.1. Conceptos asociados al problema	6
I.1.1. Indicadores claves de desempeño (<i>KPI</i>)	6
I.1.2. Tablero de control.....	7
I.2. Caracterización de los sistemas fuentes de información a la solución propuesta	9
I.3. Análisis de soluciones similares	10
I.4. Metodología de desarrollo de software	11
I.4.1. Proceso Unificado Ágil variación UCI	11
I.5. Herramientas y tecnologías.....	13
I.5.1 Herramienta y lenguaje de modelado.....	13
I.5.2. Sistema gestor y herramienta para la administración de bases de datos	14
I.5.3. Herramienta para la visualización y formato de datos métricos	15
Conclusiones del capítulo	19
CAPÍTULO II: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	20
II.1. Descripción de la propuesta de solución.....	20
II.2. Disciplina de requisitos.....	20
II.2.1. Especificación de los requisitos funcionales.....	20
II.2.2. Especificación de los requisitos no funcionales.....	23
II.2.3. Validación de los requisitos	24
II.2.4. Historias de usuario	25
II.3. Arquitectura del sistema	28
II.4. Modelo de base de datos	29
II.5. Diagrama de despliegue	30
II.6. Interfaces del sistema	31
Conclusiones del capítulo	33
CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	34
III.1. Estrategia de pruebas de software.....	34
III.2. Disciplina de Pruebas internas.....	34
III.2.1. Pruebas funcionales	34
III.3. Disciplina pruebas de aceptación.....	37
Conclusiones del capítulo	37
CONCLUSIONES GENERALES	38

RECOMENDACIONES.....	39
REFERENCIAS BIBLIOGRÁFICAS	40
ANEXOS.....	Error! Bookmark not defined.

ÍNDICE DE TABLAS

Tabla 1. Fases de la metodología AUP-UCI. [Tomado de: (Sánchez, 2015)]	11
Tabla 2. Resumen del estudio sobre las herramientas para la visualización de datos métricos	18
Tabla 3. Requisitos funcionales el TC a desarrollar.	21
Tabla 4. HU_4: Total de militantes	26
Tabla 5. HU_5: Cantidad de militantes según el tipo de militancia	26
Tabla 6. HU_17: Total de Comités primarios.	27
Tabla 7. DCP de la HU_Mostrar el total de militantes en el país	35
Tabla 8. DCP de la HU_Cantidad de militantes según el tipo de militancia	36
Tabla 9. DCP de la HU_Total de Comités primarios.	36

ÍNDICE DE FIGURAS

Figura 1. Arquitectura propuesta del tablero de control para el CCPCC	29
Figura 2. Fragmento del modelo de datos: XIGEC	30
Figura 3. Diagrama de despliegue de la solución propuesta	31
Figura 4. Interfaz de la solución. Parte 1.	32
Figura 5. Interfaz de la solución. Parte 2.	32
Figura 6. Interfaz de la solución. Parte 3.	33
Figura 7. Total de NC por iteración – Pruebas internas	36
Figura 8. Total de NC por iteración – Pruebas de aceptación	37

INTRODUCCIÓN

La gestión de la información y de grandes cantidades de datos es crucial para la sociedad actual por varias razones, entre ellas se destaca su papel en el monitoreo de un proceso específico. En la era digital, las empresas y organizaciones dependen de los datos para monitorear su desempeño y de ahí tomar decisiones informadas. Al gestionar grandes cantidades de datos, las empresas pueden obtener información sobre el comportamiento de los clientes, las tendencias del mercado y la evolución del sector. Esto les permite tomar mejores decisiones y mejorar sus operaciones. Al automatizar la entrada de datos, el análisis y la elaboración de informes, las organizaciones pueden ahorrar tiempo y recursos, y mejorar la precisión. Con acceso a grandes cantidades de datos, los investigadores pueden desarrollar nuevos conocimientos, probar hipótesis y contribuir al avance de dicha organización (Villegas Zamora, 2019).

A raíz de esto, la sociedad cubana actual transita desde hace algunos años por un proceso que se ha definido como informatización de la sociedad: uno de los tres pilares que respalda la gestión gubernamental. Las acciones realizadas, si bien aún no alcanzan la magnitud que demanda el desarrollo del país, han propiciado avances incuestionables en el gobierno y comercio electrónico (Puig Meneses , 2021). De ahí que las Tecnologías de la Información y las Comunicaciones (TIC) pueden facilitar el acceso a la información y a los recursos, haciendo más fácil que las personas encuentren la información que necesitan y tomen decisiones con conocimiento de causa. Esto puede conducir a una mayor eficiencia y productividad, así como a un acceso más equitativo a la información para todos los miembros de la sociedad.

Por tales motivos, en el año 2017, mediante un levantamiento de tecnologías y sistemas informáticos en la Estructura Auxiliar del Comité Central del Partido Comunista de Cuba (CCPCC), se obtienen como resultados significativos, la necesidad de agilizar la gestión de la información asociada a los procesos del funcionamiento interno de la organización política y en específico las estadísticas más consultadas. Esta actividad sirvió de referencia para establecer las bases conceptuales y las etapas de la estrategia de trabajo para materializar el Programa de Transformación Digital del Partido. De tal forma, en el año 2021, como parte de esta estrategia, se realizó el diagnóstico para la transformación digital (CCPCC 2022).

Entre las necesidades identificadas, se creó un proyecto de colaboración entre la Universidad de las Ciencias Informáticas (UCI) y el CCPCC con el objetivo de crear soluciones informáticas capaz de contribuir a la gestión los Cuadros profesionales, dando paso así al surgimiento del

Sistema para la Gestión de Cuadros (XIGEC). Esto está dado porque en la actualidad, llevar el control de los datos de los Cuadros profesionales del Partido Comunista de Cuba (PCC) y candidatos a los eventos, es la tarea en la que los funcionarios invierten la mayor parte del tiempo. Por otro lado, era necesario gestionar la información asociada a los procesos de la vida interna de la organización política y de sus militantes en general, así como las estructuras y organizaciones de base, los militantes asociados a cada núcleo y los procesos de crecimiento o baja en la organización. Por estas razones se concibió de igual forma el Sistema de Control de la Militancia (SICOM).

A pesar de estos avances, no se cuenta con un espacio donde se pueda visualizar de forma inmediata los indicadores claves que intervienen en los procesos de control de la militancia, y gestión de Cuadros profesionales. Estos son métricas que ayudan a analizar el rendimiento de una determinada acción o estrategia.

Para acceder a esta información, se debe acceder al módulo que emiten los reportes estadísticos en cada uno de estos sistemas, a partir de la información que se almacena en ellos. Dichos reportes no permiten la configuración y adecuación de estos a las necesidades de la dirección del país. Además, están limitados a un nivel operacional, ya que 24 de esto sólo están concebidos de formato *PDF*, lo que dificulta aún más el monitoreo de los datos. Todo esto imposibilitando que la información sea analizada a modo de resumen atendiendo a diferentes indicadores, tales como: cantidad, porciento y promedio. Debido a que existe gran cantidad de reportes resulta tedioso realizar la evaluación de dicha información, así como llevar un análisis estadístico de los datos para una posterior toma de decisiones.

Las privaciones anteriores traen consigo que en ocasiones los funcionarios y estadísticos del CCPCCC inviertan mucho tiempo en obtener un resumen del comportamiento de algunos de sus procesos de interés, lo que dificulta el análisis basado en una relación causa efecto. Como consecuencia no se permite establecer comparaciones en el tiempo ni con otros indicadores.

A partir de la problemática antes descrita se genera la necesidad de resolver el siguiente **problema de la investigación**: ¿Cómo contribuir al monitoreo de los datos de los procesos de la vida interna y política de Cuadro en el Comité Central de Partido Comunista de Cuba en tiempo real?

Una vez detectado el problema a resolver se plantea como **objeto de estudio**: herramientas para la visualización y formato de datos métricos, enmarcándose en el **campo de acción**: tablero de

control para el monitoreo de los datos métricos asociados a los procesos de vida interna y política de Cuadro del Comité Central del PCC.

Para dar cumplimiento al problema a resolver planteado anteriormente, se establece como **objetivo general**: representar los datos de los procesos de vida interna y política de Cuadro en un tablero de control para su monitoreo en tiempo real en el Comité Central del Partido Comunista de Cuba.

Por lo tanto, se plantea la siguiente **hipótesis**: la representación de los datos métricos asociados a los procesos de vida interna y política de Cuadro del Comité Central del Partido Comunista de Cuba en un tablero de control contribuirá al monitoreo de estos procesos en tiempo real.

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

1. Identificar los referentes teóricos en los que se sustenta la propuesta de solución sobre las herramientas para la visualización y formato de datos métricos.
2. Realizar la identificación de requisitos, análisis y diseño de la propuesta de solución como aproximación a la implementación.
3. Visualizar en un tablero de control la información de los principales indicadores de los procesos de vida interna y política de Cuadro del PCC.
4. Validar el correcto funcionamiento de la solución propuesta aplicando métricas y pruebas de software para garantizar la calidad de la misma.

Para dar cumplimiento a los objetivos específicos propuesto se definen las siguientes **tareas a cumplir** en el desarrollo de la investigación:

1. Revisión de la bibliografía para elaborar el marco teórico conceptual en lo referente a las herramientas para la visualización y formato de datos métricos.
2. Análisis de los sistemas similares para conocer aspectos regulares en el diseño del TC que se propone como solución al problema identificado.
3. Análisis de las herramientas informáticas y metodología de desarrollo de software asumida para el desarrollo de la solución propuesta.
4. Definición de los requisitos funcionales y no funcionales del TC.
5. Identificación de los tipos de gráficas a utilizar para mostrar el resultado del análisis de los indicadores de los procesos de vida interna y política de Cuadro.
6. Diseño de las consultas a implementar para obtener la información necesaria desde las bases de datos de los sistemas: XIGEC y SICOM.

7. Aplicación de métricas para evaluar cualitativamente la calidad de la especificación de los requisitos obtenidos.
8. Ejecución de pruebas funcionales para evaluar la calidad del tablero de control obtenido como resultado de la presente investigación.

Para el desarrollo del presente trabajo se emplean los **métodos de investigación** que se describen a continuación:

Métodos teóricos

- **Analítico-Sintético:** este método sirvió para la recopilación de información requerida durante la realización del estudio del estado del arte y para el desarrollo del trabajo mediante la revisión de documentos y artículos, de donde se extrajeron los elementos más significativos de acuerdo con los principales indicadores de desempeño en cada uno de los procesos.
- **Hipotético deductivo:** siguiendo las reglas deductivas se llega a nuevos conocimientos y pronósticos, que con posterioridad serán sometidos a comprobaciones prácticas; este método posibilitó obtener la hipótesis planteada en el trabajo.

Método empírico

- **Observación:** para percibir los problemas existentes en procesos de vida interna y política de Cuadro en el PCC. Este método ayudó a obtener una mejor visión del problema y el objetivo de esta investigación.

Con el desarrollo de la presente investigación se obtienen **beneficios** tanto para la alta dirección del país como para los estadísticos y funcionarios del CCPCC, ya que:

- Se contará con un espacio donde se puedan visualizar los principales indicadores que intervienen en el proceso de vida interna y política de Cuadro del CCPCC.
- Se podrá monitorear de manera efectiva su desempeño en tiempo real y, por consiguiente, tomar decisiones de manera más rápida y eficiente.

El desarrollo del documento está **estructurado** en tres capítulos, a continuación, se hace referencia al contenido expuesto en cada uno de ellos:

Capítulo I: Fundamentos y referentes teóricos-metodológicos: en este capítulo se hace referencia a los elementos teóricos en los que se basa la investigación y donde se incluye un estudio del estado del arte de los tableros de control para el monitoreo de datos, a partir de la información registrada en los sistemas de gestión desplegados en el CCPCC. También se

describe la metodología que guiará la investigación, así como las tecnologías utilizadas en el desarrollo de la solución propuesta.

Capítulo II: Diseño de la solución propuesta al problema científico: en este capítulo se describe la solución propuesta para llevar a cabo la representación de los datos de los procesos de vida interna y política de Cuadro en el CCPC en un tablero de control. De igual forma se detallan las principales características, así como la especificación de los requisitos funcionales (RF) a implementar y los no funcionales (RnF), encapsulando los RF software mediante historias de usuario. Además, se detalla la arquitectura de la solución propuesta, así como una muestra del modelo de datos de los sistemas fuente de información al tablero de control.

Capítulo III: Validación de la solución propuesta: en este capítulo se describe la validación de la propuesta de solución. Se proponen las pruebas de software a realizar una vez culminado el proceso de implementación, para asegurar que el tablero de control cumple con las especificaciones requeridas por el cliente.

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICOS-METODOLÓGICOS

En este capítulo se aborda todo lo relacionado con la fundamentación teórica que sustenta la presente investigación, se realiza un estudio del estado del arte del sobre los TC y se especifican algunos conceptos asociados al objeto de estudio y campo de acción. También se describe la metodología que guía la investigación, así como las principales tecnologías que se utilizan para la representación de los datos de los procesos de vida interna y política de Cuadro del CCPCC en un tablero de control.

I.1. Conceptos asociados al problema

En la presente investigación se muestran un conjunto de conceptos asociados al dominio del problema, para brindar al lector una mayor claridad sobre lo abordado y una visión general de los temas tratados a lo largo del trabajo investigativo.

I.1.1. Indicadores claves de desempeño (KPI)

Los indicadores de desempeño tienen como objetivos principales medir el nivel de servicio, realizar un diagnóstico de la situación, comunicar e informar sobre la situación y los objetivos, motivar a los equipos responsables del cumplimiento de los objetivos reflejados en el *KPI* y, en general, evaluar cualquier progreso de forma constante (Victor Coutinho, 2022). Lo que no se puede medir no se puede controlar; lo que no se puede controlar no se puede gestionar; lo que no se puede gestionar no se puede mejorar (A taxonomy for key performance indicators management, 2019). Los KPI son parte clave para el incremento de la competitividad, entendiéndola como una expresión cuantitativa del comportamiento o desempeño de toda una organización o en una de sus partes, cuya magnitud al ser comparada con algún nivel de referencia, podría señalar desviaciones sobre la cual se tomarán acciones correctivas o preventivas según sea el caso (Reliability.org, 2023).

Un indicador clave de rendimiento (KPI) es una medida verificable que indica el grado de cumplimiento de los objetivos empresariales clave por parte de una empresa. Las empresas utilizan los KPI para analizar su progreso hacia los objetivos (Martins, 2022). Algunos de los indicadores claves que intervienen en los procesos de vida interna y política de Cuadro son:

- En el proceso de política de Cuadros del PCC:
 - Cantidad de Cuadros profesionales por color de piel.
 - Cantidad de Cuadros profesionales por sexo.
 - Cuadros que no poseen nivel universitario y que no estudian.
 - Las plantillas cubiertas y las vacantes.

- Cuadros sin trabajar.
- En el proceso de vida interna de la militancia del PCC:
 - Militantes activos.
 - Traslados e incorporaciones realizadas.
 - Bajas en la organización.

I.1.2. Tablero de control

Un tablero de control, también conocido como panel de control o *dashboard*, es una herramienta visual que recopila y muestra información clave sobre el desempeño y los indicadores clave de rendimiento (KPI) de una organización en tiempo real (Peña Pérez, 2022). La elaboración de un tablero de control debe estar soportada en un sistema de información que garantice el éxito en su funcionamiento y análisis, su construcción debe atender a las necesidades de control de la organización, además, su formulación debe ser clara y sencilla; existen etapas para su construcción (Sánchez-Retiz, y otros, 2019):

- Evaluación de los procesos: se realiza un diagnóstico de los procesos organizacionales, de los insumos y factores utilizados en cada una de las etapas del proceso.
- Definición de área de análisis: se escogen de acuerdo a la estructura formalmente establecida por la organización, tales áreas pueden ser la financiera, técnica, comercial, administrativa, de producción y atención al usuario.
- Establecer indicadores: los indicadores servirán como medidas estratégicas para evaluar la gestión de un área en particular, o de toda la empresa, en general.
- Definición del formato del tablero de control: en esta etapa se procede a diseñar un formato propio de cada área, empresa o sector, para lo que es necesario tener en cuenta las necesidades individuales de la entidad.
- Desarrollo total: esta etapa corresponde al despliegue total de la herramienta, donde se recogen todas las experiencias y metodologías aplicadas. Esto implica la puesta en marcha de todos los elementos que garanticen los resultados esperados de mejoramiento y valor agregado para la entidad.

A continuación, se describen algunas razones por las cuales un tablero de control es importante para una organización:

1. Monitoreo del desempeño: Un tablero de control permite a una organización monitorear de manera efectiva su desempeño en tiempo real. Proporciona una visión general de los KPI importantes y ayuda a identificar rápidamente cualquier desviación o problema en el

rendimiento. Esto permite a los responsables de la toma de decisiones tomar medidas correctivas de manera oportuna.

2. Toma de decisiones informada: Al proporcionar datos e información actualizados, un tablero de control permite a los líderes y gerentes tomar decisiones más informadas. Proporciona una visión clara y objetiva del estado de la organización y ayuda a identificar áreas problemáticas o de mejora. Esto permite a los equipos dirigir sus esfuerzos hacia las áreas que más necesitan atención.
3. Comunicación efectiva: Un tablero de control bien diseñado y visualmente atractivo puede ser una herramienta poderosa para comunicar información clave de manera clara y concisa. Permite a los diferentes miembros de la organización comprender fácilmente los datos y las tendencias sin tener que analizar informes extensos. Esto facilita la comunicación y alinea a todos los miembros de la organización en torno a los objetivos y metas comunes.
4. Seguimiento de objetivos y metas: Un tablero de control permite rastrear el progreso hacia los objetivos y metas establecidos. Al mostrar los KPI relevantes en tiempo real, ayuda a mantener a todos los miembros del equipo enfocados en los resultados deseados. Esto fomenta la responsabilidad y el seguimiento constante del desempeño, lo que a su vez impulsa la productividad y el logro de metas.
5. Identificación de tendencias y patrones: Al mostrar datos históricos y tendencias a lo largo del tiempo, un tablero de control puede ayudar a identificar patrones y tendencias importantes. Esto permite a los analistas y tomadores de decisiones comprender mejor el rendimiento pasado y realizar predicciones o proyecciones más precisas para el futuro. La capacidad de identificar tendencias puede ser especialmente valiosa para la planificación estratégica y la toma de decisiones a largo plazo.

En resumen, un tablero de control es importante para una organización porque proporciona una visión clara del desempeño y los KPI de interés para la organización. De igual forma ayudan en la toma de decisiones, facilita la comunicación, permite el seguimiento de objetivos y metas, así como a identificar tendencias y patrones importantes. De ahí el objetivo de la presente investigación con el fin de proveerle al CCPCC un tablero de control que le permita monitorear los datos de los procesos de política de Cuadros y vida interna del PCC. Para el desarrollo de la solución propuesta se tendrá en cuenta las etapas que define (Sánchez-Retiz, y otros, 2019), partiendo de la evaluación de los procesos hasta llegar a la representación de los datos de los procesos de vida interna y política de Cuadro del CCPCC en un tablero de control.

I.2. Caracterización de los sistemas fuentes de información a la solución propuesta

Los conceptos, como la vida misma, evolucionan y conducen a perfeccionar y ampliar los procesos para obtener de ellos el máximo provecho. En ese camino, Cuba asume nuevos preceptos que la llevan hacia la transformación digital: un nuevo momento en el que se integran las tecnologías digitales en todos los ámbitos de la sociedad, donde el centro del hacer son las personas (Puig Meneses , 2021). Como se explicó a inicios de la investigación, como producto del proyecto de colaboración entre la UCI y el CCPCC, se llevó a cabo el desarrollo de dos soluciones informáticas con el fin de contribuir a esta la transformación digital en el país. A continuación, se describen ambos sistemas:

Sistema para la Gestión de Cuadros (XIGEC): este sistema surge debido a que llevar el control de los datos de los Cuadros profesionales del Partido Comunista de Cuba (PCC) y candidatos a los eventos, es la tarea en la que los funcionarios invierten la mayor parte del tiempo. Actualmente este sistema cuenta con 32 reportes, estos no se generar en formato Excel, solo en formato PDF. En este proceso de política de Cuadro, es de interés a los funcionarios evaluar indicadores respecto a:

- Promedio de edad de los Cuadros profesionales por territorio.
- Cuadros profesionales por sexo por territorio
- Cuadros profesionales por color de piel por territorio
- Cuadros que no poseen nivel escolar Universitario y que no estudian por territorio.
- Plantillas cubiertas y plazas vacantes por territorio.
- Reservas de Cuadro por territorio.
- Cuadros sin trabajar por territorio.
- Candidatos por sexo por territorio
- Candidatos por color de piel por territorio
- Desglose de candidatos por nivel escolar.

Para llegar al análisis de estos indicadores, se deben evaluar 24 de los 32 reportes y aplicar los filtros que provee dicho sistema, por lo que no es posible llegar a un resultado de forma inmediata de estos indicadores. Además, en algunas de las veces se deben convertir los reportes en formato Excel para tabular los datos y evaluar los indicadores. Esto trae consigo que el monitoreo de estos se torne complejo para los funcionarios que interviene en el proceso de política de Cuadro del PCC.

Sistema de Control de la Militancia (SICOM): surge debido a la necesidad de gestionar la información asociada a los procesos de la vida interna de la organización política y de sus militantes en general, así como las estructuras y organizaciones de base, los militantes asociados a cada núcleo y los procesos de crecimiento o baja en la organización. Actualmente este sistema cuenta con 124 reportes, en formato *Excel* y *PDF*. En este proceso es de interés a los estadísticos evaluar indicadores respecto a:

- Total de militantes del PCC y de la UJC.
- Traslados completados y pendientes.
- Traslados masivos por núcleos.
- Crecimientos de 90 días.
- Sanciones por tipo de sanción
- Cotizaciones atrasadas.
- Total de núcleos y núcleos mixtos.
- Total de comités primarios y comités de centro.

Para llegar al análisis de estos indicadores, se deben evaluar 33 de los 124 reportes y aplicar los filtros que provee dicho sistema, por lo que no es posible llegar a un resultado de forma inmediata de estos indicadores. Esto trae consigo que el monitoreo de estos se torne complejo para los estadísticos que interviene en el proceso de vida interna del PCC.

I.3. Análisis de soluciones similares

Una vez esclarecidos los referentes teóricos en lo que se sustenta la investigación, se realiza un estudio de sistemas homólogos, con el objetivo de definir características y su viabilidad en el entorno que se desarrolla la presente investigación. Teniendo en cuenta que la misma se enmarca en los procesos de vida interna y política de Cuadro del PCC, siendo esta organización política única de su tipo en el mundo, fruto de la Revolución Cubana, se define solo realizar un estudio de sistemas homólogos a nivel nacional.

Luego de una búsqueda en las diferentes fuentes, documentos y archivos, sobre el objeto de estudio y campo de acción a nivel nacional, se evidencia que, en Cuba no existe solución implantada en las instancias del CCPCC similar a la que se desea llevar a cabo. Además, luego de una entrevista realizada a los funcionarios y estadísticos del CCPCC involucrados en los procesos de vida interna y política de Cuadro, se confirma tal afirmación. Por ello la autora de la presente investigación afirma la necesidad de la visualización de los principales indicadores en

un tablero de control para el monitoreo de estos procesos, de forma tal que se pueda obtener datos métricos en tiempo real.

I.4. Metodología de desarrollo de software

Una metodología es un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación que permiten guiar a los desarrolladores de aplicaciones. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería (Pressman, y otros, 2015).

I.4.1. Proceso Unificado Ágil variación UCI

El Proceso Unificado Ágil es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. Se preocupa especialmente de la gestión de riesgos, propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. En la Universidad de la Ciencias Informáticas (UCI) se desarrolló una versión de esta metodología ágil (AUP-UCI) con el objetivo de crear un estándar en el proceso productivo de manera que se adapte al ciclo de vida de los proyectos en la UCI (Sánchez, 2015).

La Metodología AUP-UCI está formada por tres fases, (Inicio, Ejecución y Cierre) las cuales contienen las características de las cuatro fases (Inicio, Elaboración, Construcción y Transición) propuestas en AUP.

Tabla 1. Fases de la metodología AUP-UCI. [Tomado de: (Sánchez, 2015)]

Fases	Objetivos
Inicio	<ul style="list-style-type: none"> • Planeación del proyecto • Estudio inicial de la organización cliente. • Alcance del proyecto • Estimaciones de tiempo, esfuerzo y costo
Ejecución	<ul style="list-style-type: none"> • Ejecución de las actividades requeridas para desarrollar el software. • Ajuste de los planes del proyecto. • Modelado del negocio. • Levantamiento de requisitos.

	<ul style="list-style-type: none"> • Elaboración de la arquitectura y el diseño. Implementación y liberación del producto.
Cierre	<ul style="list-style-type: none"> • Análisis de los resultados y ejecución del proyecto. • Cierre del proyecto.

Escenarios de la metodología AUP-UCI:

No 1: aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta respectivamente, donde la atención se centra en cómo el usuario va a utilizar el sistema. Es necesario que se tenga claro por el proyecto que los Casos de uso del Negocio muestran como los procesos son llevados a cabo por personas y los activos de la organización (Sánchez, 2015).

No 2: aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información (Sánchez, 2015).

No 3: aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. Se debe tener presente que este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados (Sánchez, 2015).

No 4: aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una Historia de Usuario no debe poseer demasiada información (Sánchez, 2015).

Una vez evaluado el negocio a informatizar, se obtuvo un negocio muy bien definido y se concibió que la propuesta no modela negocio. Por otra parte, el cliente estará acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y

validarlos. Además, no se considera un proyecto muy extenso, por lo que se propone encapsular los requisitos a través de HU, utilizando así el **escenario 4** que define la metodología AUP-UCI.

I.5. Herramientas y tecnologías

Las tecnologías son programas o instrucciones que facilitan la realización de una tarea determinada, o sea, están diseñadas para realizar tareas y trabajos específicos en el computador (GCFGlobal.org, 2023). A continuación, se describen las utilizadas en la representación de los datos de los procesos de vida interna y política de Cuadro del CCPCC en un tablero de control.

I.5.1 Herramienta y lenguaje de modelado

Las herramientas de Ingeniería de Software Asistida por Ordenador (*CASE* por sus siglas en inglés: *Computer Aided Software Engineering*) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan al proceso de desarrollo del software en tareas como: realizar un diseño del proyecto, cálculo de costos, implementación de parte del código con el diseño dado, compilación automática, documentación o detección de errores entre otras (Mendez, 2018).

Visual Paradigm for UML v8.0: es una herramienta CASE que soporta el desarrollo de software desde el análisis y diseño orientado a objeto, construcción, pruebas y despliegue. Ayuda a los equipos de desarrollo de softwares a capturar los requisitos correctos y transformarlos en diseños precisos, lo que ayuda a los desarrolladores a crear el software adecuado según los requisitos. Además, permite la importación y exportación de archivos de Lenguaje de Mercado Extensible (XML del inglés *Extensible Markup Language*) de diferentes versiones. Presenta licencia gratuita y comercial. Es multiplataforma, fácil de instalar y actualizar, compatible entre ediciones (Visual Paradigm.org, 2017).

Esta versión posee múltiples ventajas, entre las que se encuentra dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Además, el equipo de desarrollo posee experiencia en su empleo, lo que posibilita reducir el tiempo en el proceso de desarrollo del software ya que no hay necesidad de capacitar a los desarrolladores. Por último, la Universidad cuenta con una licencia académica para su uso.

Lenguaje Unificado de Modelado v2.0: el Lenguaje Unificado de Modelado™ (UML® del inglés *Unified Modeling Language*) prescribe un conjunto de notaciones y diagramas estándares para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos

diagramas y símbolos significan. Posibilitando así visualizar, especificar y documentar los artefactos o toda información que se obtiene o modifica durante un proceso de desarrollo de software. Además, se puede utilizar para modelar distintos tipos de sistemas de software, hardware y organizaciones del mundo real (Larman, 2002).

Según las características antes descritas se decide utilizar para la propuesta de solución Lenguaje Unificado de Modelado en su versión 2.0. ya que es la versión que traer por defecto la herramienta CASE seleccionada. Además, mejora el desarrollo de los proyectos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

I.5.2. Sistema gestor y herramienta para la administración de bases de datos

Un sistema gestor de base de datos (SGBD) es un conjunto de programas invisibles para el usuario final con el que se administra y gestiona la información que incluye una base de datos. Los gestores de datos o gestores de base de datos permiten administrar todo acceso a la base de datos, pues tienen el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones. Entre sus funciones se encuentra permitir a los usuarios de negocio almacenar la información, modificar datos y acceder a los activos de conocimiento de la organización. Asimismo, el gestor de base de datos también se ocupa de realizar consultas y hacer análisis para generar informes (Beynon-Davies 2018). A continuación, se describe el sistema gestor de base de datos y la herramienta para la administración de la misma.

PostgreSQL v12.4: es un sistema de bases de datos relacional de alta disponibilidad. Es capaz de funcionar de manera estable en el servidor y, por tanto, resulta robusto, además es consistente y tolerable a fallos. Es compatible con el modelo relacional pues asegura siempre su integridad referencial. Algunas de las principales características son (Lardiere, 2019):

- Alta concurrencia: es capaz de atender a muchos clientes al mismo tiempo y entregar la misma información de sus tablas, sin bloqueos
- Soporte para múltiples tipos de datos de manera nativa: ofrece los tipos de datos habituales en los sistemas gestores, pero además muchos otros que no están disponibles en otros competidores, como direcciones IP¹, direcciones MAC², Arrays³, números decimales con precisión configurable, figuras geométricas.

¹ Número que identifica a cualquier aparato electrónico que se conecte a internet

² Control de Acceso a Medios, identificador de 48 bits que corresponde de forma única a una tarjeta o dispositivo de la red

³ Manera de poder guardar datos de un mismo tipo o clase

- Soporte a *triggers*⁴: permite definir eventos y generar acciones cuando estos se disparan.
- Trabajo con vistas: pueden consultar los datos de manera diferente al modo en el que se almacenan.
- Objeto-relacional: permite trabajar con sus datos como si fueran objetos y ofrece mecanismos de orientación a objetos, como herencia de tablas.
- Soporte para bases de datos distribuidas: donde el trabajo con transacciones asegura que estas tendrán éxito cuando han podido realizarse en todos los sistemas involucrados.
- Soporte para gran cantidad de lenguajes: PostgreSQL es capaz de trabajar con funciones internas, que se ejecutan en el servidor, escritas en diversos lenguajes como C, C++, Java, PHP o Python.

PgAdmin IV v6.3: es una herramienta indispensable para gestionar y administrar base de datos en PostgreSQL. El software tiene la apariencia de una aplicación de escritorio sea cual sea el entorno de tiempo de ejecución (escritorio o web), y mejora enormemente respecto a pgAdmin III con elementos de interfaz de usuario actualizados, opciones de despliegue multiusuario, web, paneles y un diseño más moderno. La herramienta pgAdmin IV en su versión 6.3 incluye varias correcciones de errores y nuevas funciones tales como (PostgreSQL 2021):

- Se agregó soporte para la autenticación de dos factores para mejorar la seguridad. La autenticación de dos factores (2FA) es una capa adicional de seguridad que se utiliza al iniciar sesión en sitios web o aplicaciones. Con 2FA, debe iniciar sesión con su nombre de usuario y contraseña y proporcionar otra forma de autenticación que solo el usuario conoce o tiene acceso.
- Se agregó soporte para deshabilitar el descubrimiento automático de los servidores de la base de datos.

I.5.3. Herramienta para la visualización y formato de datos métricos

Para dar cumplimiento al objetivo fundamental de la presente investigación, se hace necesario utilizar una herramienta para la visualización y formato de datos métricos, por lo que a continuación se analizarán las existentes de acuerdo con sus características.

CloudMonix: es una herramienta esencial para obtener información avanzada sobre el rendimiento del sistema, notificaciones inmediatas de mal funcionamiento a través de integraciones, escalado automático de recursos y autocuración. Entre sus características se tiene que (ProgSoft.net, 2023):

⁴ Objeto que se asocia con tablas y se almacena en una base de datos

- Obtiene todos sus sistemas y recursos en un solo lugar con alimentación de datos en vivo.
- Establece métricas y rastrea indicadores clave de rendimiento con control granular.
- Mantiene el control de los problemas de producción al recibir notificaciones inmediatas a través de integraciones de terceros llave en mano y *APF*⁵ (interfaz del programa de aplicación) robusta.
- Con las acciones de *PowerShell*⁶ automatiza la ejecución de scripts para garantizar un servicio ininterrumpido.
- Automatiza los procedimientos de recuperación mediante secuencias de comandos de respuestas comunes a problemas de producción: auto-recuperación.
- Cuenta con recursos de cálculo y escala automática basados en métricas del entorno monitoreado.

Cyclotron: es una plataforma para crear y alojar paneles de control, independiente de cualquier fuente de datos y basada en navegador para construir tableros. Tiene un editor de paneles incorporado y aloja los paneles directamente. Entre sus características se tiene que (ProgSoft.net, 2023):

- No requiere programación, utiliza un enfoque declarativo para la configuración que incluye cientos de propiedades para la personalización.
- Los paneles se definen declarativamente como un documento *JSON*⁷, que contiene todas las propiedades necesarias para representar el panel.

Grafana: es un sistema de visualización interactiva y dinámica basado en software libre, específicamente con licencia Apache 2.0, ideada por Torkel Ödegaard y creada en enero de 2014. Grafana está escrita en Lenguaje *Go* y *Node.js LTS* y con una fuerte Interfaz de Programación de Aplicaciones (*API*). Es compatible con más de 30 fuentes de datos/bases de datos de código abierto y comerciales, como *MySQL*, *PostgreSQL*, *Graphite*, *Elasticsearch*, *Open TSDB* (Grafana Labs, 2023). Se trata de un software multiplataforma y extensible mediante *plugins* en el que los usuarios pueden construir de forma personalizada su panel de visualización de datos, así como compartir los mismos de manera sencilla. Entre sus características se encuentran (Benito, 2021):

⁵ Conjunto de reglas que permite que diferentes programas se comuniquen entre sí

⁶ Interfaz de consola con posibilidad de escritura y unión de comandos por medio de instrucciones

⁷ Formato de texto sencillo para el intercambio de datos

- Visualización de múltiples tipos de gráficas con multitud de opciones con las que se pueden enriquecer y extender.
- Creación de paneles de control dinámicos y reutilizables con la posibilidad de compartirlos.
- Utilización de diversas y múltiples fuentes de datos de las que se pueden obtener métricas personalizadas, así como filtrar datos y realizar anotaciones en tiempo real.
- Definición de alertas y notificaciones.

Kibana: es una aplicación gratuita y abierta que funciona con capacidades de visualización de datos y de búsqueda para los datos indexados en *Elasticsearch*⁸, utilizándolo como motor de búsqueda para sus análisis. Actúa como la interfaz de usuario para monitorear, gestionar y asegurar un *cluster*⁹ del *Elastic Stack*¹⁰ y como concentrador centralizado de las soluciones integradas desarrolladas en *Elastic Stack*. Entre sus principales características se tiene que (ProgSoft.net, 2023):

- Monitorea, administra y asegura una instancia de *Elastic Stack* a través de una interfaz web.
- Centraliza el acceso para soluciones integradas desarrolladas en *Elastic Stack* para aplicaciones de visualización, seguridad y búsqueda empresarial.

Prometheus: es un conjunto de herramientas de monitoreo y alerta de sistemas de código abierto independiente. Entre sus principales características se tiene que (ProgSoft.net, 2023):

- Presenta un modelo de datos multidimensional (series de tiempo identificadas por nombre de métrica y pares clave / valor).
- Tiene un lenguaje de consulta flexible para aprovechar esta dimensionalidad.
- No tiene dependencia del almacenamiento distribuido; los nodos de un solo servidor son autónomos.
- Múltiples modos de gráficos y soporte de tablero.
- Los objetivos se descubren mediante el descubrimiento de servicios o la configuración estática.

⁸ Servidor de búsqueda

⁹ Varios servidores trabajando juntos como uno

¹⁰ Conjunto de herramientas que permiten recoger datos de cualquier tipo de fuente y en cualquier formato para realizar búsquedas, análisis y visualización de los datos en tiempo real

Tabla 2: Resumen del estudio sobre las herramientas para la visualización de datos métricos

	Rendimiento	Visualización y edición	Fuentes de datos compatibles	Alarmas y seguimiento
Grafana	No necesita equipos con altas prestaciones	Posee funciones relacionadas con la visualización de métricas de series de tiempo e interfaz amigable.	Permite múltiples fuentes de datos.	Posee un sistema de alarmas y notificaciones. Cuenta con filtro de datos y plantillas de tablero.
Kibana	Necesita equipos con altas prestaciones.	Fácil de usar e interfaz amigable.	Permite múltiples fuentes de datos.	Permite configurar alarmas y notificaciones.
Prometheus	Necesita equipos con altas prestaciones.	Depende de las plantillas de la consola para la visualización.	Se encuentra limitado en cuanto a la compatibilidad con fuentes de datos.	Permite configurar alarmas y notificaciones.
CloudMonix	No necesita equipos con altas prestaciones	Fácil de usar e interfaz amigable.	Se encuentra limitado en cuanto a la compatibilidad con fuentes de datos.	No permite configurar alarmas y notificaciones.
Cyclotron	No necesita equipos con altas prestaciones	Fácil de usar e interfaz amigable.	Permite múltiples fuentes de datos.	No permite configurar alarmas y notificaciones.

Fuente: Elaboración propia

Luego del estudio realizado sobre las herramientas existentes para la visualización y formato de datos métricos, la autora de la presente investigación opta por la utilización de **Grafana** en su **versión 7.5**. La selección se basa por las máximas opciones que posee dicha herramienta para

dividir los datos de una manera comprensible; con su característica de soportar múltiples bases de datos. Por otro lado, es una herramienta con interfaz fácil de entender y operar. Además, se tuvo en consideración que, de todas estas herramientas, es la herramienta con la que la autora tiene conocimientos previos, por lo que no es necesario dedicar mucho tiempo en capacitación de la misma.

Grafana v7.5: Grafana en esta nueva versión ha rediseñado el editor de paneles para que haya una experiencia consistente en todo Grafana. Estas actualizaciones también facilitan la personalización de los paneles y el *dashboard* proporcionado. Un nuevo panel de la tabla que soporta el desplazamiento horizontal y el cambio de tamaño de las columnas. Los usuarios tienen la posibilidad de colapsar o expandir, reordenar, ocultar y renombrar las columnas en el editor para que puedan ver exactamente en qué necesitan enfocarse. Este nuevo panel también soporta nuevos modos de visualización de celdas, como mostrar un indicador de barras dentro de una celda. Un nuevo modo de diseño automático que facilita el cambio de tamaño de los paneles para una variedad de formatos, incluyendo tablas, datos de series no temporales y gráficos. La capacidad de anular la zona horaria utilizada para mostrar los valores de fecha y hora en un *Dashboard* (Grafana Labs, 2023).

Conclusiones del capítulo

Luego de haber realizado el establecimiento de los referentes teóricos-metodológicos en los que se sustenta la investigación se tiene que, la conceptualización de los principales conceptos asociados al objeto de estudio, permitieron una mejor comprensión de la misma. Además, la caracterización de los sistemas fuentes al TC que se propone como solución al problema que da paso a la presente investigación, se identificó como la información registrada incide en los indicadores y su contribución en el proceso como tal. Por otra parte, con el análisis y la investigación realizada sobre posibles sistemas homólogos se demostró de manera clara y detallada la necesidad de visualizar en un TC los principales indicadores de los procesos de vida interna y política de cuadro, que permita el monitoreo de los datos en tiempo real. Por último, la caracterización de la metodología de desarrollo y de las tecnologías a utilizar, permitió establecer las etapas a seguir en la investigación, así como las tecnologías a tener en cuenta en el desarrollo de la solución propuesta.

CAPÍTULO II: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se plantea la propuesta de solución al problema principal de la presente investigación. De ello parte el levantamiento, especificación y validación de los requisitos funcionales y no funcionales, para la representación de los datos de los procesos de vida interna y política de Cuadro del CCPCC en un tablero de control, así como la descripción de las historias de usuarios obtenidas. Por último, se desarrolla la disciplina de Análisis y diseño donde se describe la arquitectura y el modelo de datos fuente al tablero de control a desarrollar.

II.1. Descripción de la propuesta de solución

La propuesta de solución de la presente investigación responde a los procesos identificados, partiendo de que está enfocada en visualizar los indicadores claves que intervienen en los procesos de vida interna y política de Cuadro de manera resumida, permitiendo así el monitoreo de datos métricos de interés para el PCC. Para darle respuesta a la situación problemática del presente trabajo se propone la representación de los datos métricos de los principales indicadores de los procesos de vida interna y política de Cuadro en un TC.

II.2. Disciplina de requisitos

La ingeniería de requisitos ayuda a los ingenieros de software a entender el problema en cuya solución trabajarán. Incluyen el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, que es lo que el cliente quiere y cómo interactuarán los usuarios finales con él software (Santos, 2016).

Los requisitos se clasifican en dos tipos; los requisitos funcionales y los requisitos no funcionales. Los requisitos funcionales describen lo que el sistema debe hacer. Mientras que los requisitos no funcionales son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento (Pressman, y otros, 2015).

II.2.1. Especificación de los requisitos funcionales

Los requerimientos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir, define qué es lo que el sistema debe hacer, así como las funciones que será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas, para producir salidas (Pressman, y otros, 2015).

Tabla 3. Requisitos funcionales el TC a desarrollar.

No	Nombre	Descripción	Prioridad
RF 1.	Mostrar la cantidad total de usuarios conectados.	Permite visualizar la cantidad de usuarios conectados en el sistema.	Alta
RF 2.	Mostrar la cantidad de usuarios activos.	Permite visualizar la cantidad de usuarios activos en el sistema.	Alta
RF 3.	Mostrar la cantidad de usuarios inactivos.	Permite visualizar la cantidad de usuarios inactivos en el sistema.	Alta
RF 4.	Mostrar total de militantes.	Permite visualizar la cantidad total de militantes registrados en el sistema.	Alta
RF 5.	Mostrar la cantidad de militantes por tipo de militancia.	Permite visualizar la cantidad de militantes del PCC y de la UJC en el sistema.	Alta
RF 6.	Mostrar el total de traslados.	Permite visualizar el total de traslados de militantes hacia otros núcleos.	Alta
RF 7.	Mostrar los traslados pendientes.	Permite visualizar la cantidad de traslados pendientes.	Alta
RF 8.	Mostrar los traslados completados.	Permite visualizar la cantidad de traslados que fueron completados.	Alta
RF 9.	Mostrar el total de traslados masivos por núcleos.	Permite visualizar el total de traslados masivos por núcleos.	Alta
RF 10.	Mostrar el total de crecimientos 90 días.	Permite visualizar el total de crecimientos de 90 días de la organización, a partir de la fecha de crecimiento.	Alta
RF 11.	Mostrar el total de sanciones.	Permite la visualización del total de sanciones.	Alta
RF 12.	Mostrar el total de sanciones internas.	Permite la visualización de la cantidad de sanciones de carácter interno.	Alta
RF 13.	Mostrar el total de sanciones externas.	Permite la visualización de la cantidad de sanciones de carácter externo.	Alta
RF 14.	Mostrar el total de cotizaciones atrasadas.	Permite la visualización del total de cotizaciones atrasadas.	Alta

RF 15.	Mostrar el total de núcleos.	Permite la visualización del total de núcleos del PCC en el país.	Alta
RF 16.	Mostrar cantidad de núcleos mixtos.	Permite la visualización de la cantidad de núcleos mixtos.	Alta
RF 17.	Mostrar la cantidad de Comités primarios.	Permite la visualización de la cantidad de Comités primarios.	Alta
RF 18.	Mostrar la cantidad de Comités de centro.	Permite la visualización de la cantidad de Comités de centros.	Alta
RF 19.	Mostrar el total de bajas en la organización.	Permite la visualización del total de bajas en la organización.	Alta
RF 20.	Mostrar promedio de edad de los Cuadros.	Permite la visualización del promedio de edad de los Cuadros profesionales del Partido.	Alta
RF 21.	Mostrar el total de Cuadros profesionales por sexo.	Permite la visualización del total de Cuadros profesionales del Partido por sexo.	Alta
RF 22.	Mostrar el total de Cuadros profesionales por color de piel.	Permite la visualización del total de Cuadros profesionales del Partido por color de piel.	Alta
RF 23.	Mostrar el total de Cuadros que no poseen nivel escolar universitario y que no estudian.	Permite la visualización del total de Cuadros profesionales del Partido que no poseen nivel escolar universitario y que no estudian.	Alta
RF 24.	Mostrar el total de plantillas cubiertas.	Permite la visualización del total plantillas cubiertas.	Alta
RF 25.	Mostrar el total de plazas vacantes.	Permite la visualización del total de plazas vacantes.	Alta
RF 26.	Mostrar total de reservas de Cuadro.	Permite la visualización del total de reservas de Cuadros para el organismo.	Alta
RF 27.	Mostrar el total de Cuadros sin trabajar.	Permite la visualización del total de Cuadros profesionales del Partido sin trabajar.	Alta

Fuente: Elaboración propia

II.2.2. Especificación de los requisitos no funcionales

Los requerimientos no funcionales (RnF) son propiedades o cualidades que el producto debe tener, debe pensarse en propiedades que hacen al producto atractivo, usable, rápido o confiable. Son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este, como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento (Pressman, y otros, 2015).

Usabilidad

- RnF1: el tablero de control debe poder ser usado por personas que tenga conocimientos básicos de sistemas informáticos.
- RnF2: el tablero de control debe proporcionar un acceso fácil y rápido a los gráficos por la persona encargada.
- RnF3: el tablero de control debe mostrar la información de forma lógica y correctamente estructurada según los procesos e indicadores a mostrar.

Seguridad

- RnF4: el TC debe garantizar la protección de los datos almacenados, se aplicará en la autenticación de administrador la encriptación de la contraseña.
- RnF5: el TC debe establecer mecanismos de control y verificación para los procesos susceptibles de fraude.

Eficiencia

- RnF6: el TC debe tener tiempos de respuesta no mayor de $300\text{ms} \pm 100\text{ms}$ para el 60% de las peticiones de consulta.

Soporte

- RnF7: el TC contará, antes de su puesta en marcha, con un período de pruebas.

Portabilidad

- RnF8: el TC debe ser fácil de instalar y configurar.

Fiabilidad

- RnF9: el TC debe tener una disponibilidad del 99.9% de las veces que se intente acceder al mismo.
- RnF10: el sistema deberá estar disponible las 24 horas de los 7 días a la semana.

II.2.3. Validación de los requisitos

La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesite o el cliente desea, su objetivo es asegurar que el software está de acuerdo con su especificación. Se comprueba que el sistema cumple con los requisitos funcionales y no funcionales que se han especificado utilizando las siguientes técnicas de validación de requisitos (Sommerville, 2016):

- **Construcción de prototipos:** se realiza un modelo ejecutable utilizando el componente para que los clientes puedan experimentar con él y determinar si cumple sus necesidades. En la presente investigación, se definió un prototipo para cada dato métricos a mostrar en el TC utilizando los gráficos tipos de la herramienta Grafana, y se evaluó con el cliente hasta quedar finalmente aceptados, estos se pueden consultar en el documento “*Historias de usuario. Tablero de Control para el Comité Central del Partido Comunista de Cuba*” generado por la autora de la presente investigación.
- **Revisiones formales de los requisitos:** se realizaron revisiones formales de cada requisito, por parte del cliente y el equipo de desarrollo, validando que la interpretación de cada una de las descripciones no sea ambigua, ni presenten omisiones o errores.

Mediante estas revisiones y con el objetivo de medir la calidad de la especificación de los requisitos de software se aplica la métrica Calidad de la especificación (CE). El empleo de esta métrica permite obtener un alto nivel de entendimiento y precisión de los requisitos, para llegar a ello, se debe primeramente calcular el total de requisitos de software, como se muestra a continuación:

- **Nr:** total de requisitos de software.
- **Nf:** cantidad de requisitos funcionales.
- **Nnf:** cantidad de requisitos no funcionales.

$$\mathbf{Nr = Nf + Nnf}$$

Sustituyendo los valores en la ecuación, se obtiene:

$$\mathbf{Nr = 27 + 10}$$

$$\mathbf{Nr = 37}$$

Finalmente, para calcular la calidad de la Especificidad de los Requisitos (ER), en este caso se evaluó la ausencia de ambigüedad en los mismos, para ello se realiza la siguiente operación:

- **Nui:** número de requisitos para los cuales todos los revisores tuvieron interpretaciones idénticas.

$$\mathbf{Q1 = Nui / Nr}$$

Para sustituir los valores de las variables en la ecuación se tuvo en cuenta que, de los requisitos especificados para la representación de los datos de los procesos de vida interna y política de Cuadro del CCPCC en un tablero de control, uno de ellos causó contradicción (RF9) en su interpretación. Por tanto, la variable Q1 obtiene el siguiente valor:

$$Q1 = 36/37$$

$$Q1 = 0.97$$

Es importante aclarar que mientras más cerca de 1 está el valor de Q1, menor es la ambigüedad. Teniendo en cuenta el resultado anterior, igual a 0.97, se concluye que el 97% de los requisitos es entendible. El requisito identificado como ambiguo fue modificado y validado para garantizar su correcta comprensión, llegando al resultado ideal donde el 100% de los requisitos tuvieron una única interpretación por el equipo de revisores.

II.2.4. Historias de usuario

Las historias de usuario (HU) se usan, en el contexto de la ingeniería de requisitos ágil, como una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal. Las historias de usuario son una herramienta que agiliza la administración de requisitos, reduciendo la cantidad de documentos formales y tiempo necesarios (Alexander Menzinsky, 2018).

Las HU son la técnica que utilizan las metodologías ágiles (SCRUM, XP, AUP) para especificar los requisitos de software, estas deben ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a las tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra HU. Las estimaciones de esfuerzo, asociado a la implementación de las historias, las establecen los programadores, utilizando como medida el punto. Un punto, equivale a una semana ideal de programación (5 días laborables). Las historias, generalmente, valen de 1 a 3 puntos. Cada HU recoge al menos los siguientes aspectos (Perurena, 2017):

- Número: posee el número asignado a la HU.
- Nombre del requisito: atributo que contiene el nombre del requisito
- Programador: el programador del sistema
- Prioridad: evidencia el nivel de prioridad de la HU en el negocio.
- Tiempo real: el tiempo que demoró el equipo de desarrollo en realizar la HU.
- Descripción: posee una breve descripción de lo que realizará la HU
- Observaciones: algunas aclaraciones que es importante señalar acerca de la HU.
- Prototipo de interfaz: ilustración de la HU.

A continuación, se describen tres Historias de Usuario de prioridad Alta presentes en la representación de los datos de los procesos de vida interna y política de Cuadro del CCPCC en un tablero de control. El resto se encuentran en el documento “*Historias de usuario. Tablero de Control para el Comité Central del Partido Comunista de Cuba*”, elaborado por la autora de la presente investigación, el cual relaciona el 100% de las HU obtenidas.

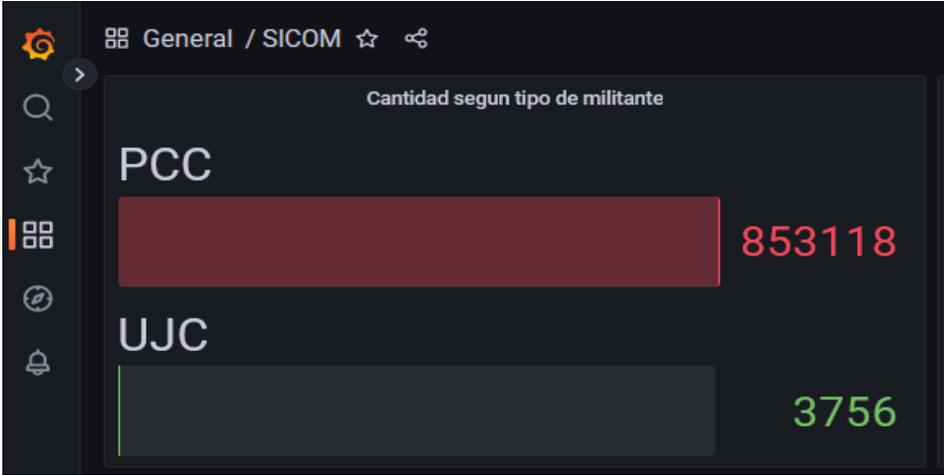
Tabla 4. HU_4: Total de militantes

Historias de Usuario	
Número: 04	Nombre de la HU: Total de militantes
Programador: Elisabet de los Rios Diaz	
Prioridad: Alta	Tiempo estimado: 8 horas
Riesgo en desarrollo: medio	Tiempo real: 10 horas
Descripción: permite visualizar la cantidad de militantes que hay en el país	
Observaciones:	
Prototipo de interfaz:	
 <p>The screenshot shows a dark-themed dashboard interface. At the top, there is a navigation bar with a gear icon, a search icon, and the text 'General / SICOM'. Below this, a large circular gauge is displayed with the title 'Total de Militantes' and the number '856874' in a large, bright green font. A vertical sidebar on the left contains several icons: a magnifying glass, a star, a grid, a refresh symbol, and a bell.</p>	

Fuente: Elaboración propia

Tabla 5. HU_5: Cantidad de militantes según el tipo de militancia

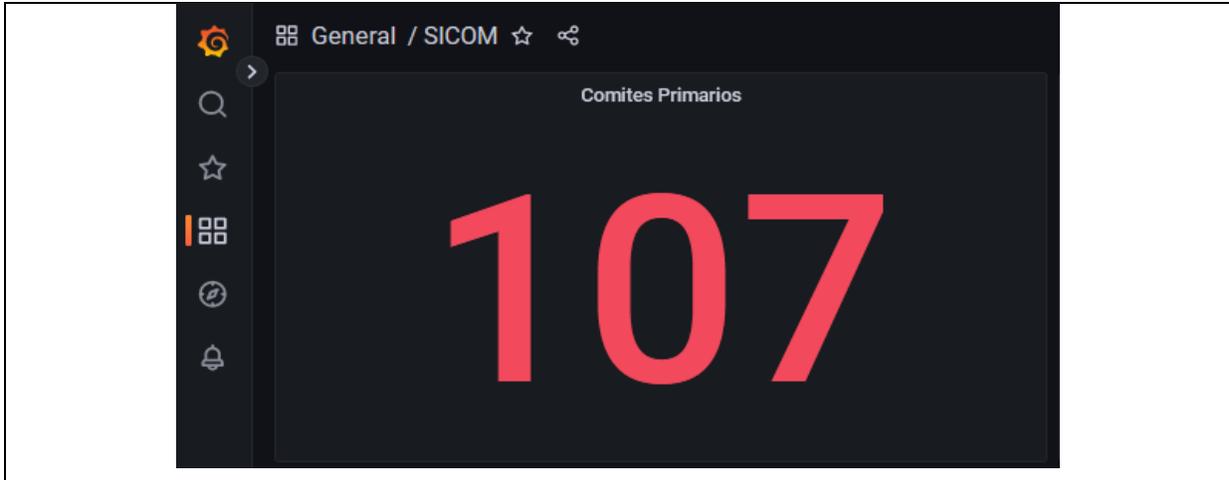
Historias de Usuario

Número: 05	Nombre de la HU: Cantidad según el tipo de militancia	
Programador: Elisabet De los Rios Diaz		
Prioridad: Alta	Tiempo estimado: 8 horas	
Riesgo en desarrollo: medio	Tiempo real: 12 horas	
Descripción: permite visualizar la cantidad total de militantes del PCC y de la UJC.		
Observaciones:		
Prototipo de interfaz:		
		

Fuente: Elaboración propia

Tabla 6. HU_17: Total de Comités primarios.

Historias de Usuario		
Número: 17	Nombre de la HU: Total de comités primarios	
Programador: Elisabet De los Rios Diaz		
Prioridad: Alta	Tiempo estimado: 8 horas	
Riesgo en desarrollo: alto	Tiempo real: 8 horas	
Descripción: permite visualizar la cantidad total de Comité primarios en el país.		
Prototipo de interfaz:		



Fuente: Elaboración propia

II.3. Arquitectura del sistema

La arquitectura de software es la etapa en el proceso de construcción del software que constituye el enlace crucial entre el diseño y la ingeniería de requerimientos, ya que identifica los principales componentes estructurales en un sistema y la relación entre ellos. Su salida consiste en un modelo que describe la forma en que se organiza el sistema como un conjunto de componentes en comunicación (Sommerville, 2016). Por la característica de la solución propuesta se propone la utilización de una arquitectura cliente servidor.

Arquitectura cliente-servidor: es uno de los estilos arquitectónicos distribuidos más conocidos, el cual está compuesto por dos componentes, el proveedor y el consumidor. El proveedor es un servidor que brinda una serie de servicios o recursos los cuales son consumido por el Cliente. Por eso existe un servidor y múltiples clientes que se conectan al servidor para recuperar todos los recursos necesarios para funcionar, en este sentido, el cliente solo es una capa para representar los datos y se detonan acciones para modificar el estado del servidor, mientras que el servidor es el que hace todo el trabajo pesado (Cloud Native Glossary Authors, 2023).

La idea central de separar al cliente del servidor radica en la idea de centralizar la información y la separación de responsabilidades. Por una parte, el servidor será la única entidad que tendrá acceso a los datos y los servirá solo a los clientes del cual el confía, y de esta forma, se protege la información y la lógica detrás del procesamiento de los datos. Por otro lado, el cliente suele ser instalado en computadoras con bajos recursos, pues desde allí no se procesa información, simplemente actúa como un visor de los datos y delega las operaciones pesadas al servidor

(Cloud Native Glossary Authors, 2023). A continuación, se describe cómo incide la solución propuesta en esta arquitectura:

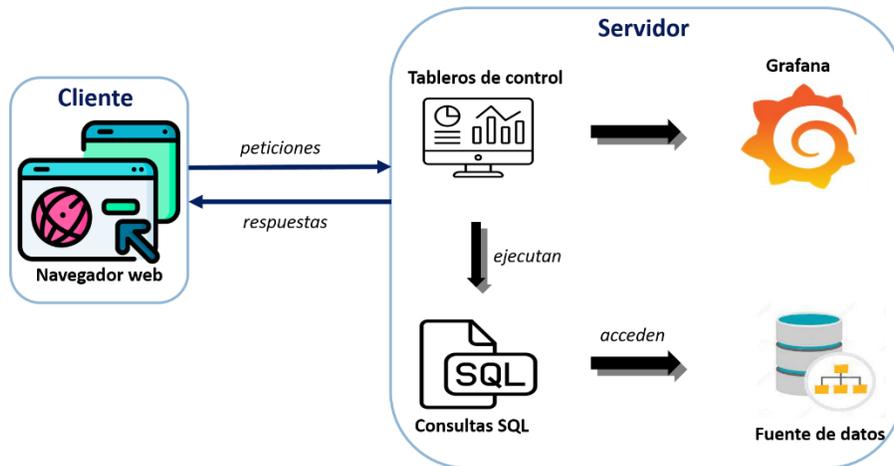


Figura 1. Arquitectura propuesta del tablero de control para el CCPCC

Fuente: Elaboración propia

Del lado del cliente estarán las computadoras que tendrán instalados los navegadores web a utilizar para realizar las peticiones de los datos métricos a través de los gráficos diseñados. Por su parte el servidor, procesará las peticiones realizadas por el lado cliente, ejecutando las consultas SQL, las cuales accederán a la fuente de datos, dígame las bases de datos de los sistemas XIGEC y SICOM. Una vez que ya se tenga la información a mostrar en el gráfico, se envía una respuesta del lado del servidor hacia el lado cliente para mostrar la información solicitada.

II.4. Modelo de base de datos

El modelo de datos se utiliza para describir la estructura lógica y física de la información persistente gestionada por el sistema. Se utiliza para definir la correlación entre las clases de diseño persistentes y las estructuras de datos persistentes, y para definir las estructuras de datos persistentes (IntelDig, 2018). En la figura que se muestra a continuación se relaciona una muestra de las estructuras de datos persistente del XIGEC, relacionado con la política de Cuadro, elaborado por el arquitecto de base de datos de dicho proyecto. Esto como fuente proveedora de información al TC que se propone como solución al problema que da paso a la presente investigación:

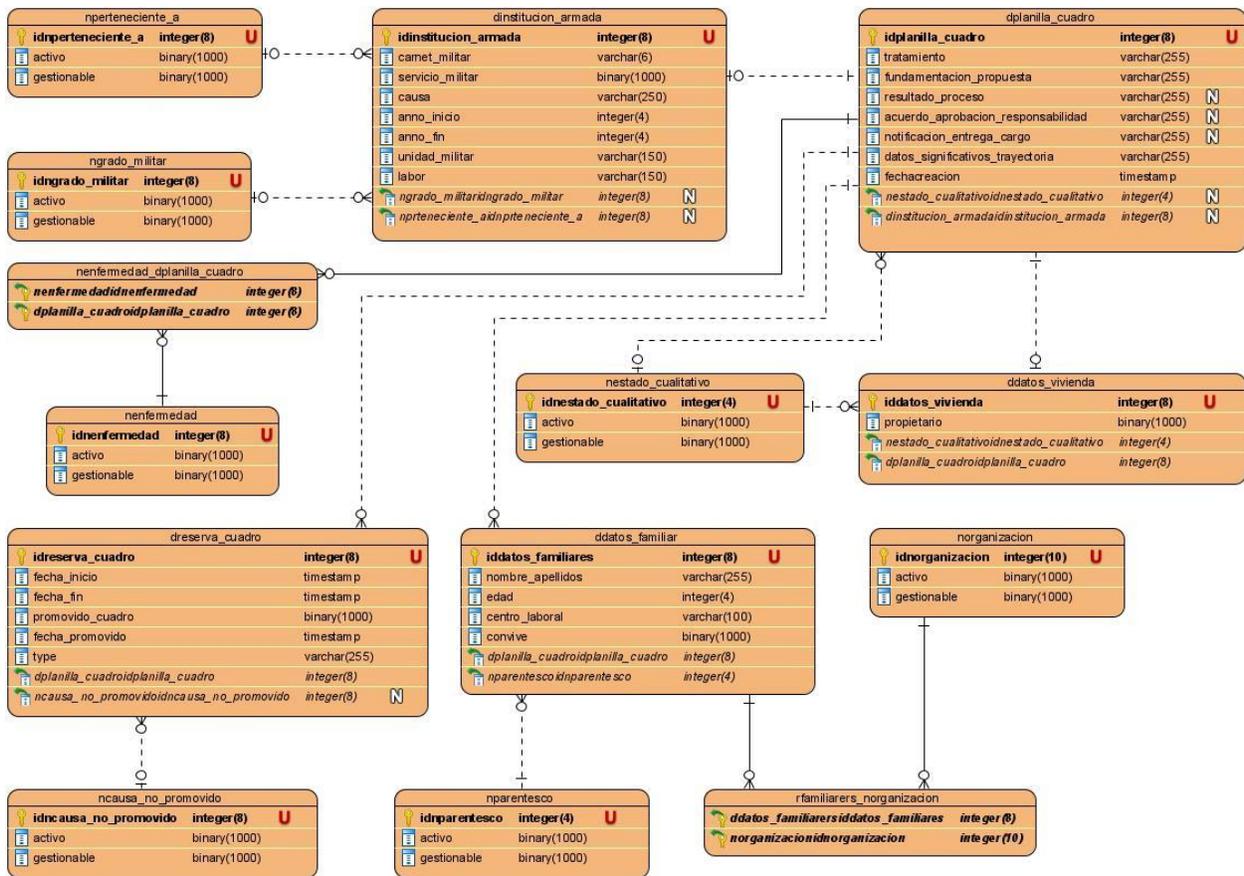


Figura 2. Fragmento del modelo de datos: XIGEC

Fuente: Modificado de (Gómez Perdomo, 2022)

El modelo de datos en general cuenta con tres tablas fundamentales para almacenar la información de los Cuadros profesionales del PCC. Las cuales son: *dplanilla_persona* y *dplanilla_cuadro*, *dreserva_cuadro*, que contienen los datos esenciales de las personas, de los Cuadros profesionales y de las reservas de Cuadro respectivamente. También se encuentran el resto de las tablas relacionadas con *dplanilla_cuadro* como parte del modelo de base de datos del sistema en general, encargadas también de gestionar la información de interés para el CCPC a relacionar en la planilla de los Cuadros profesionales del PCC.

II.5. Diagrama de despliegue

El diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos), o sea, muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación (Larman, 2002).

A continuación, se modela la arquitectura de la solución propuesta, una vez puesta en ejecución en el entorno del cliente.

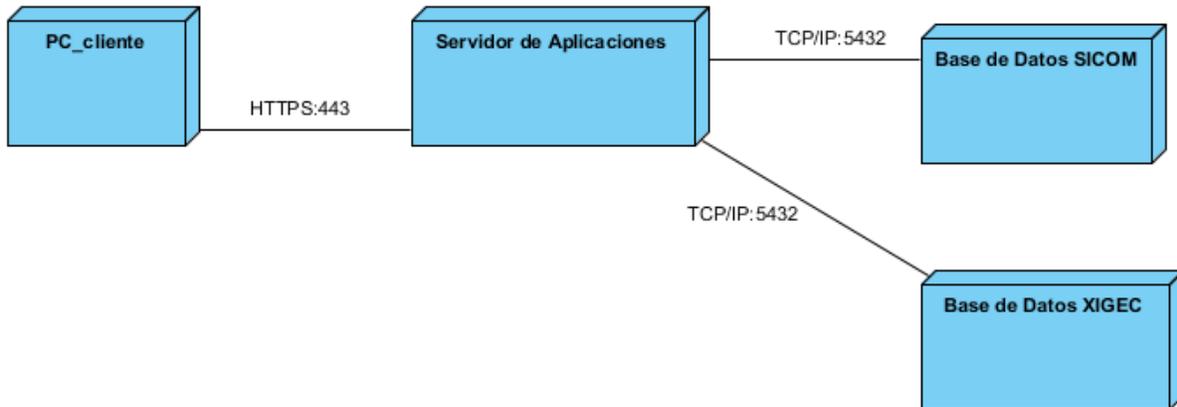


Figura 3. Diagrama de despliegue de la solución propuesta

Fuente: Elaboración propia

La solución se despliega en cuatro nodos físicos. Primeramente, se encuentra la PC_Cliente que tiene instalado el navegador web, ya sea desde una computadora o laptop, con el que se comunica mediante un protocolo *HTTPS:443* con el servidor de aplicación: Grafana, el cual tendrá alojado todos los gráficos que engloban la solución propuesta. El mismo ejecuta las consultas SQL, por lo que mantiene una conexión con los servidores de base de datos desarrollados en *PostgreSQL*, donde se encuentra almacenada la información gestionada en las soluciones: XIGEC y SICOM. Estas bases de datos son necesarias para el funcionamiento del tablero de control ya que sirven de fuente de información, pero no son parte de los resultados desarrollados en este trabajo.

II.6. Interfaces del sistema

La interfaz de un sistema es la dinámica física y lógica de interconexión entre dos aparatos o sistemas independientes, o bien entre un sistema informático y un usuario (Scolari, 2021). A continuación, se muestran cómo debería quedar las interfaces del tablero de control una vez implementado.

En la siguiente figura se muestra la interfaz de los gráficos correspondiente a la Cantidad de sanciones externas, Traslados pendientes, Cotizaciones atrasadas, Total de bajas de la organización, Total de traslados por núcleos y cantidad de militantes según el tipo de militancia (PCC o UJC).

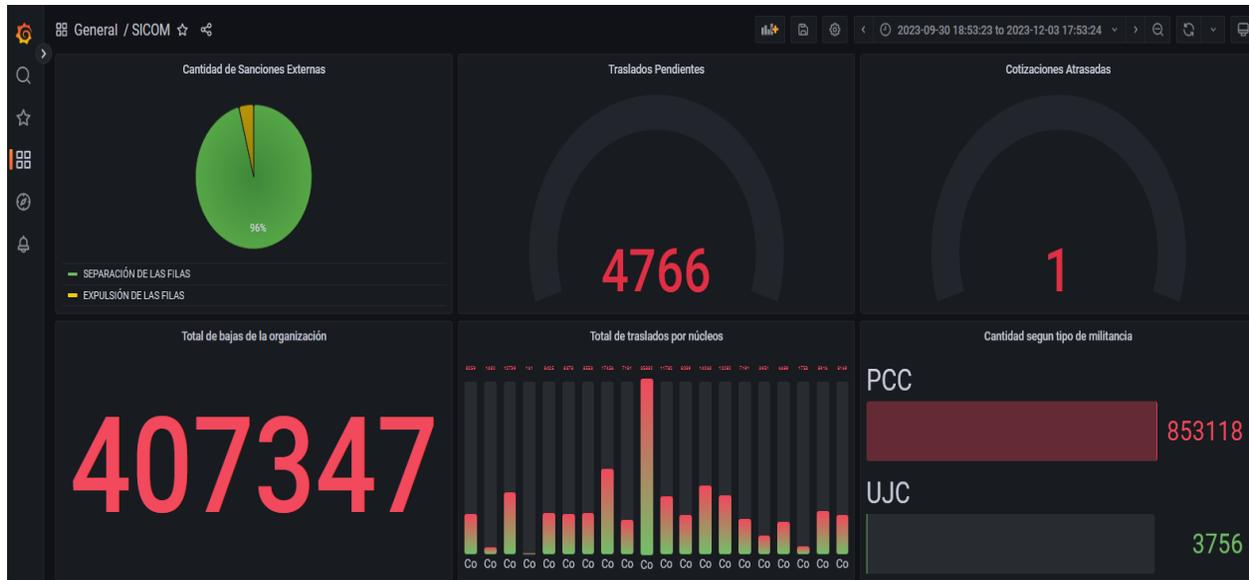


Figura 4. Interfaz de la solución. Parte 1.
Fuente: Elaboración propia

En la siguiente figura se muestra una imagen del gráfico que da cumplimiento al desarrollo del RF16, a través del cual se visualiza el total de núcleos mixtos (donde militan militantes del PCC y de la UJC).



Figura 5. Interfaz de la solución. Parte 2.
Fuente: Elaboración propia

En la siguiente figura se muestra la interfaz correspondiente a los gráficos obtenidos de los RF, RF1, RF2, RF3, RF9, RF10, RF13, RF15, RF16, RF1 y RF18, permitiendo la visualización de la total de usuarios conectado, usuarios activos, usuarios inactivos, total de traslados por núcleos, crecimiento en 90 días, sanciones externas, núcleos mixtos, comités primarios y comité de centro.

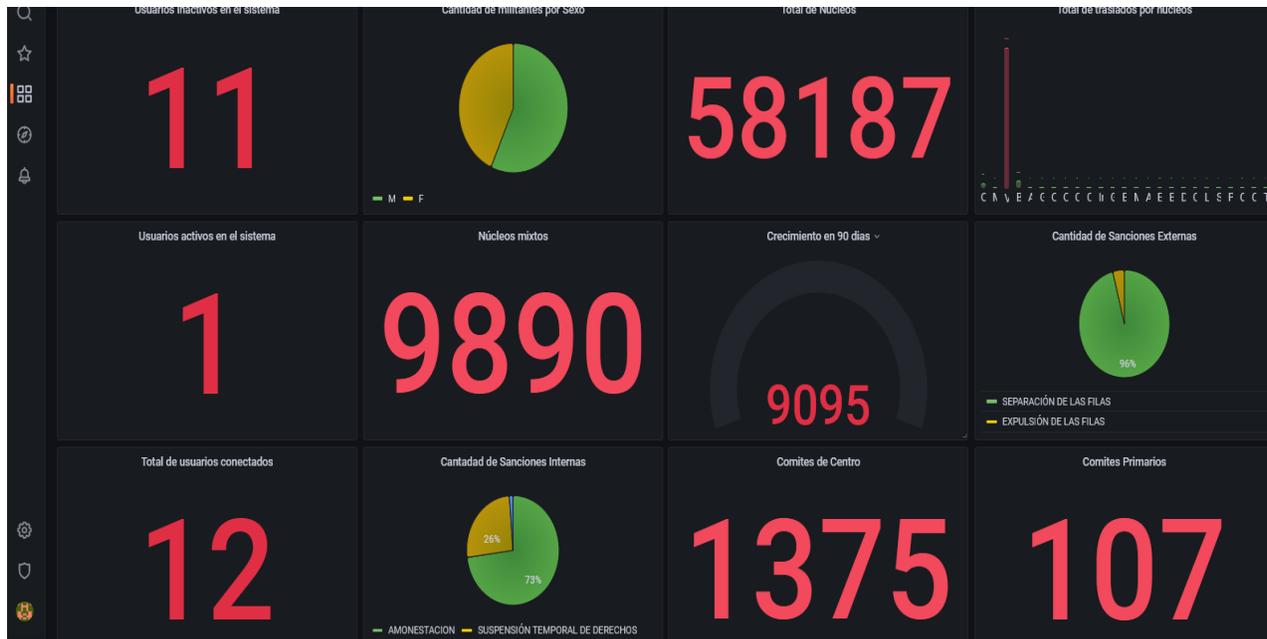


Figura 6. Interfaz de la solución. Parte 3.
Fuente: Elaboración propia

Conclusiones del capítulo

Luego de haber realizado descripción de la solución propuesta, a partir de la metodología de desarrollo AUP-UCI, se tiene que la disciplina de Requisitos permitió la identificación y descripción de todos los requisitos funcionales y no funcionales a incorporar en la solución. Por otra parte, la confección de las historias de usuario permitió un mejor entendimiento y estructuración de la solución. Finalmente, con el cumplimiento de la disciplina de Análisis y diseño se obtiene una visión general del sistema, es decir, una propuesta de cómo quedaría la solución propuesta y la interacción entre cada uno de los elementos que la componen.

CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se presentan los resultados una vez aplicados las pruebas de software, una vez implementado el tablero de control para el CCPCC. Esto con el objetivo de verificar el correcto funcionamiento del mismo, en correspondencia a la metodología AUP en su variación UCI, y así comprobar que la solución propuesta cumple con todas las especificaciones planteadas por el cliente.

III.1. Estrategia de pruebas de software

Las pruebas de software son elementos críticos para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación (Pressman, y otros, 2015). En el caso de la presente investigación se decide establecer una estrategia de pruebas con el objetivo de determinar cómo se van a llevar a cabo este proceso. Es por ello que para validar la propuesta de solución se establece una estrategia a partir de las disciplinas establecidas por la metodología AUP en su variación para la UCI y de los niveles de pruebas que plantea (Pressman, y otros, 2015) en el libro “*Software Engineering: A practitioner’s approach*”. En dicha estrategia se decide aplicar la disciplina de Pruebas Internas y de Aceptación a nivel de sistema, con la aplicación de sus respectivos métodos y técnicas. A continuación, se describen los resultados obtenidos una vez ejecutada dicha estrategia de pruebas.

III.2. Disciplina de Pruebas internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible componentes de prueba ejecutables para automatizar las pruebas (Rodríguez Sánchez, 2015). A continuación, se describen los resultados.

III.2.1. Pruebas funcionales

Las pruebas funcionales son pruebas diseñadas tomando como referencia las especificaciones funcionales de un componente o sistema (lo que vamos a testear, el software o una parte de él). Se realizan para comprobar si el software cumple las funciones esperadas (Pressman, 2010). Para llevar a cabo estas pruebas se aplicó el método de Caja negra que a continuación se describe.

Método de Caja negra

El método de caja negra se enfoca en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca, en vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente a los métodos de caja blanca (Pressman, y otros, 2015). Para llevar a cabo el método de Caja negra se utilizan las técnicas de Partición de equivalencia que a continuación se describe.

Partición de equivalencia: es un método de prueba que divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Un caso de prueba ideal descubre de primera mano una clase de errores (por ejemplo, procesamiento incorrecto de todos los datos carácter) que de otro modo podrían requerir la ejecución de muchos casos de prueba antes de observar el error general (Pressman, y otros, 2015).

El método se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse. Una clase de equivalencia representa un conjunto de estados válidos y no válidos para las condiciones de entrada. Para aplicar esta técnica, se deben primeramente realizar el Diseños de casos prueba (DCP) con el objetivo de obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del software. Su comportamiento se evalúa verificando que los resultados en el TC corresponden con la información almacenada en la base de datos y comprobada una vez generados los reportes.

Para evaluar la calidad de la propuesta se propone utilizar un total de 2 casos de prueba correspondientes a cada uno de los procesos. A raíz de esta evaluación se detectan posibles no conformidades según su comportamiento en la solución propuesta. A continuación, se muestra un fragmento del DCP_Tablero de SICOM:

Tabla 7. DCP de la HU_Mostrar el total de militantes en el país

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 4.1 Campos correctos	Al acceder a este gráfico se muestra el total de militantes en el país	El sistema muestra la cantidad total de los militantes del país.	General/ SICOM

Fuente: Elaboración propia

Tabla 8. DCP de la HU_Cantidad de militantes según el tipo de militancia

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 5.1 Campos correctos	Al acceder a este gráfico se muestra el total militantes según el tipo de militancia.	El sistema muestra la cantidad total de los militantes según el tipo de militancia.	General/ SICOM

Fuente: Elaboración propia

Tabla 9. DCP de la HU_Total de Comités primarios.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 17.1 Campos correctos	Al acceder a este gráfico se muestra el total de Comités primarios.	El sistema muestra la cantidad total de Comités primarios.	General/ SICOM

Fuente: Elaboración propia

Para aplicar el método de Caja negra a la solución, se efectuaron un total de 3 iteraciones para poder alcanzar resultados satisfactorios, atendiendo al correcto comportamiento del sistema ante diferentes situaciones. A continuación, se muestra un resumen de las No Conformidades (NC) en cada iteración de prueba:

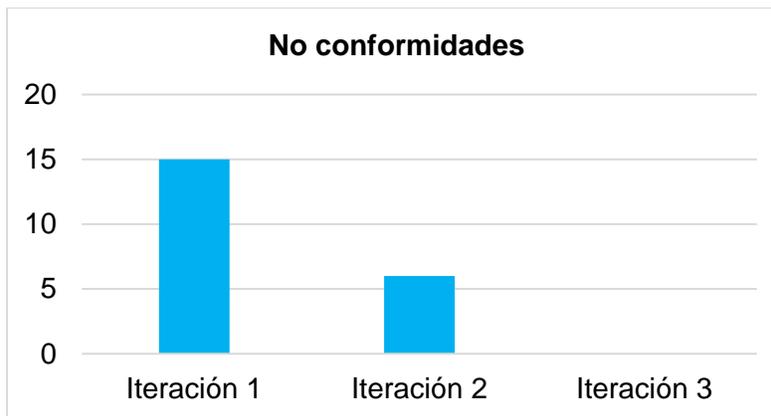


Figura 7. Total de NC por iteración – Pruebas internas

Fuente: Elaboración propia

Como se muestra en la **Figura 5**, como resultado de la estrategia de pruebas llevada a cabo, en la 1ra iteración se detectaron 15 NC, y en una 2da iteración 6 NC, todas clasificadas en: Errores al mostrar los datos en la aplicación, esto estaba dado porque no se consideraron correctamente las condiciones en las consultas a la base de datos.

III.3. Disciplina pruebas de aceptación

Son las pruebas finales antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por los usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido (Matos, 2017). Para la aplicación de esta disciplina de prueba, se utilizaron los mismos DCP utilizados en la disciplina de Pruebas Internas, lo que esta vez verificados por el usuario final. A continuación, se muestran los resultados obtenidos:

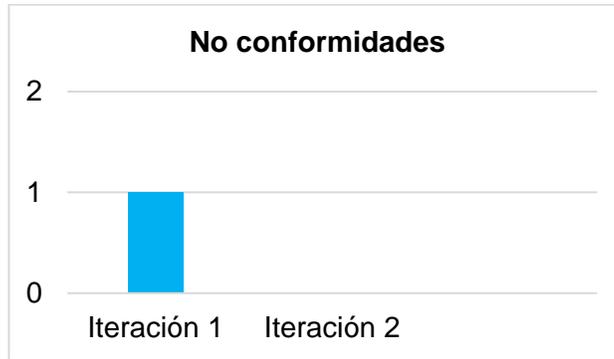


Figura 8. Total de NC por iteración – Pruebas de aceptación
Fuente: Elaboración propia

Como se muestra en la **Figura 6**, como resultado de las pruebas de aceptación llevadas a cabo en el entorno y por el propio usuario, en la 1ra iteración sólo se detectó 1 NC, la cual estuvo concebida a raíz de una mala interpretación de la columna *mes_informado* de la tabla *cotización_cotización* de la base de datos de SICOM. Luego de la aclaración se modificó la consulta y se volvió a probar la solución propuesta, donde en una 2da interacción no se detectaron NC. Luego de estos resultados se demuestra que el tablero de control para el CCPCC cumple con las necesidades requeridas. Además, ratifica que se encuentra listo para ser puesto en producción para su utilización por los usuarios finales, y de esta forma contribuir al monitoreo de los datos de los procesos de vida interna y política de Cuadro del PCC.

Conclusiones del capítulo

Luego de haber realizado la validación de la solución propuesta, se tiene que, la ejecución de la disciplina de Pruebas Internas permitió comprobar que las funciones internas y externas son operativas, donde se produce un resultado correcto, a través de los datos métricos que se muestran en el TC. Por su parte la ejecución de pruebas en la disciplina de Pruebas de Aceptación validó que la solución propuesta cumple con las necesidades solicitadas y que se encuentra lista para su utilización por los usuarios finales.

CONCLUSIONES GENERALES

Con la realización de este trabajo de diploma se representan en un tablero de control los principales indicadores que intervienen en los procesos de vida interna y política de Cuadros en el Comité Central del Partido Comunista de Cuba. Al llegar a este resultado se puede concluir lo siguiente:

1. El establecimiento de los referentes teóricos sobre las herramientas para la visualización y formato de datos métricos facilitó una mejor comprensión para la posterior representación de los datos asociados a los procesos de vida interna y política de Cuadros del CCPCC.
2. La identificación de requisitos, así como el análisis y diseño del tablero de control para el Comité Central del Partido Comunista de Cuba, permitió obtener una mayor comprensión y la especificación en detalle de lo que debe hacer y cómo debe implementarse los diferentes elementos de la solución propuesta.
3. La visualización de la información de los principales indicadores de los procesos de vida interna y política de Cuadro en el CCPCC a través de un TC contribuyó al monitoreo de los datos de estos procesos en tiempo real.
4. La validación de la solución propuesta aplicando métricas y pruebas de software permitió comprobar que las funciones internas y externas son operativas, donde se produce un resultado correcto, y que cumple con las necesidades solicitadas por los usuarios finales.

RECOMENDACIONES

Luego de finalizar la presente investigación se recomienda integrar los indicadores de interés asociados al proceso de atención a la población en la solución propuesta, según los intereses del Comité Central del Partido Comunista de Cuba.

REFERENCIAS BIBLIOGRÁFICAS

Benito, Carlos Hidalgo de. 2021. *Monitorización de un entorno empresarial con Prometheus-Grafana.* 2021.

Cloud Native Glossary Authors. 2023. Cloud Native Glossary Authors. *Arquitectura Cliente-Servidor.* [En línea] Documentation Distributed under CC BY 4.0, 2023. <https://glossary.cncf.io/es/client-server-architecture/>.

Domínguez, Eladio, y otros. 2019. *A taxonomy for key performance indicators management.* [ed.] ScienceDirect. 1, s.l. : Computer Standards & Interfaces, 2019, Vol. 64.

GCFGlobal.org. 2023. GCFGlobal.org. *¿Qué es un programa o aplicación?* [En línea] 2023. <https://edu.gcfglobal.org/es/informatica-basica/que-es-un-programa-o-aplicacion/1/>.

Gómez Perdomo, Yosvany. 2022. *Modelo de diseño del Sistema para la Gestión de Cuadros XIGEC.* La Habana : Universidad de las Ciencias Informáticas, 2022.

Grafana Labs. 2023. Grafana Labs. *What's new in Grafana v7.5.* [En línea] 2023. <https://grafana.com/docs/grafana/latest/whatsnew/whats-new-in-v7-5/>.

IntelDig. 2018. Tecnologías-Información. *Modelo de datos.* [En línea] 2018. <https://www.tecnologias-informacion.com/modeladodatos.html>.

Lardiere, Sebastien. 2019. *PotgreSQL: Administracion y explotacion de sus bases de datos.* 2019.

Larman, Craig. 2002. *UML y Patrones: Introducción al análisis y diseño orientado a objetos.* 2ra. Mexico : Prentice Hall Hispanoamerica, S.A, 2002. 84-205-3438--2.

Martins, Julia. 2022. ASANA. *Qué es un KPI, para qué sirve y cómo utilizarlo en tu proyecto.* [En línea] 2022. <https://asana.com/es/resources/key-performance-indicator-kpi>.

Matos, José Manuel Prado. 2017. *Proceso de pruebas de aceptación de software.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2017. Tesis de Maestría.

Mendez. 2018. uniciencista.gfrodriguez.online. *uniciencista.gfrodriguez.online.* [En línea] 10 de febrero de 2018. <http://uniciencista.gfrodriguez.online/2018/02/herramientas-case-principales-usos.html>.

Menzinsky, Alexander; López, Gertrudis; Palacio, Juan y Sobrino, Miguel Ángel. 2018. scrummanager.com. [En línea] agosto de 2018. [Citado el: 23 de mayo de 2023.] https://www.scrummanager.com/files/scrum_manager_historias_usuario.pdf.

Peña Pérez, David. 2022. *Aplicación sobre Google Cloud Platform que permite la creación de Dashboard para facilitar el seguimiento de los alumnos.* Ingeniería Telemática y Electrónica, Universidad Politécnica de Madrid. Madrid : Biblioteca de la Universidad Politécnica de Madrid., 2022. Proyecto Fin de Carrera / Trabajo Fin de Grado.

Perurena, Richard Anca. 2017. *Aplicacion web para monitoreo y control de una camara Pan Tilt basada en servomotores.* La Habana : s.n., 2017.

Pressman, Roger S. y Maxim, Bruce R. 2015. *Software Engineering: A practitioner's approach.* Nueva York : McGraw-Hill Education, 2015. Vol. 8th. ISBN: 978-0-07-802212-8.

ProgSoft.net. 2023. CloudMonix Alternativas y software similar - ProgSoft.net. *CloudMonix Alternativas y software similar - ProgSoft.net.* [En línea] 2023. <https://progsoft.net/es/software/cloudmonix>.

Puig Meneses , Yaima. 2021. *De la informatización de la sociedad a la transformación digital en Cuba.* La Habana : Palacio de La Revolución, 2021.

Reliability.org. 2023. *Los Indicadores de Gestion.* s.l. : International Maintenance Conference 2023, 2023.

Sánchez, Tamara Rodríguez. 2015. *Metodología de desarrollo para la actividad productiva de la UCI.* 2015.

Sánchez-Retiz, Carlos Alberto y Rodríguez-Bello, Luz Angélica. 2019. *Toma de decisiones en empresas pequeñas que combinan varias actividades económicas. Construcción de un tablero de control.* Bogotá : s.n., 2019. ISSN 2145-4558.

Santos, Eylin Campillo. 2016. *Generación de productos mediante sistema de apoyo en la disciplina de requisitos.* La Habana : s.n., 2016.

Scolari, Carlos A. 2021. *Las leyes de la interfaz: diseño, ecología, evolución, tecnología.* s.l. : Gedisa, 2021.

Sommerville, Ian. 2016. *Software Engineering*. Tenth edition . England : Pearson Education Limited, 2016. ISBN 10: 1-292-09613-6.

Victor Coutinho. 2022. *Indicadores claves de desempeño y su aplicación en la gerencia estratégica de las empresas de salud*. San Pedro, Lourdes de Montes de Oca : Scielo, 2022. On-line version ISSN 1659-0775, Print version ISSN 1659-0775.

Villegas Zamora, Diego Alonso. 2019. *La importancia de la estadística aplicada para la toma de decisiones en Marketing*. Cochabamba - Bolivia : UNIVERSIDAD DEL VALLE, 2019. ISSN 2521-2737.

Visual Paradigm.org. 2017. Visual Paradigm Web site. [En línea] 2017. <https://www.visual-paradigm.com/>.