

**MEDICIÓN DE LA CALIDAD DEL SISTEMA DE INFORMACION
GEORREFERENCIADO DEL OBSERVATORIO DEL DELITO DE PASTO -
SIGEODEP UTILIZANDO LA NORMA ISO/IEC 9126**

**EDGAR GILDARDO CAICEDO BRAVO
JOSE LUIS GARCIA PARRA
JOSE LUIS MARTINEZ DIAZ**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA DE SISTEMAS
SAN JUAN DE PASTO
2014**

**MEDICIÓN DE LA CALIDAD DEL SISTEMA DE INFORMACION
GEORREFERENCIADO DEL OBSERVATORIO DEL DELITO DE PASTO -
SIGEODEP UTILIZANDO LA NORMA ISO/IEC 9126**

**EDGAR GILDARDO CAICEDO BRAVO
JOSE LUIS GARCIA PARRA
JOSE LUIS MARTINEZ DIAZ**

**Trabajo de grado presentado como requisito parcial para optar al título de
Ingeniero de Sistemas**

**Director:
Ing. MSc. ALEXANDER BARON SALAZAR**

**Co – Director:
Ing. MSc. GONZALO HERNÁNDEZ GARZON**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA DE SISTEMAS
SAN JUAN DE PASTO
2014**

NOTA DE RESPONSABILIDAD

Las ideas y conclusiones aportadas en este Trabajo de Grado son Responsabilidad de los autores.

Artículo 1 del Acuerdo No. 324 de octubre 11 de 1966, emanado del honorable Concejo Directivo de la Universidad de Nariño.

“La Universidad de Nariño no se hace responsable de las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”.

Artículo 13, Acuerdo N. 005 de 2010 emanado del Honorable Consejo Académico.

Nota de Aceptación:

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

San Juan de Pasto, Noviembre de 2014

AGRADECIMIENTOS

Los autores del presente trabajo agradecemos a nuestro asesor, el magister Alexander Barón Salazar, por sus consejos, aportes y guías. A todos los profesores de La Universidad de Nariño por darnos tan valiosa formación no solo como profesionales sino también como personas. Al grupo de investigación GRIAS, por darnos la oportunidad de trabajar con ellos en el proyecto SIGEODEP y porque de alguna manera colaboraron en la formación de éste trabajo. Al personal de la oficina del Observatorio del delito de Pasto por su colaboración en las diferentes pruebas ejecutadas.

A Dios, a nuestros padres por su acompañamiento y apoyo durante nuestra formación y la elaboración de este proyecto, a nuestros amigos y compañeros que nos acompañaron durante todo el proceso educativo.

RESUMEN

La calidad de software se ha convertido en un paso muy importante en el proceso de desarrollo de cualquier software, debido a que se puede garantizar la calidad del producto al momento de ser entregado al usuario final; la norma ISO /IEC 9126, es una herramienta que permite determinar dicha calidad, por medio de una serie de características, sub-características, métricas y atributos establecidos, que clasifican al software para su posterior evaluación.

Este trabajo muestra la experiencia y resultados obtenidos al momento de someter a un proceso pruebas y posteriormente determinar la calidad del software SIGEODEP, un proyecto realizado por el grupo de investigación GRIAS, en convenio con la Alcaldía de Pasto y financiado con COLCIENCIAS; el cual soporta el proceso de recolección, integración y procesamiento inicial de los datos recolectados por el Observatorio del Delito referente a las lesiones de causa externa fatal y no fatal que son recopiladas por instituciones de salud, justicia y prevención de la ciudad de Pasto.

ABSTRACT

Software quality has become a very important step in any software development process, because of it can guarantee the quality of the product immediately to be delivered to final user; ISO/IEC 9126, is a tool that allows to determine the quality, through a series of characteristics, sub-characteristics, metrics and settled attributes, which are classified the software for further evaluation.

This work show the experience and results obtained of undergo a process of testing and subsequently determine the quality of the software SIGEODEP, project undertaken by GRIAS research group, in agreement with the Mayor's Office of Pasto and funded by COLCIENCIAS; which supports the process of collection, integration and processing initial data collected by the observatory on crime, relating to injuries from fatal external cause and non-fatal which are compiled by health institutions, justice and prevention of the city of Pasto.

CONTENIDO

Pág.

INTRODUCCIÓN	15
1. MARCO TEÓRICO.....	21
1.1 INTRODUCCIÓN AL CAPITULO	21
1.2 CALIDAD DE SOFTWARE	21
1.2.1 Aplicación de la calidad en software	21
1.3 PRUEBAS DE SOFTWARE	33
1.3.1 Tipos de pruebas:	35
1.4 NORMA ISO/IEC 9126: EVALUACION DE PRODUCTOS DE SOFTWARE	39
1.4.1 Partes de la norma.	40
1.4.2 Metodología propuesta para la evaluación de calidad de productos de software en uso:	43
2. SISTEMA DE INFORMACIÓN GEORREFERENCIADO DEL OBSERVATORIO DEL DELITO DE PASTO.....	56
2.1 INTRODUCCIÓN	56
2.2 PLATAFORMA.....	56
2.3 MARCAS REGISTRADAS DE TECNOLOGIA SOFTWARE.....	56
2.4 FUNCIONALIDADES PRINCIPALES.....	59
2.5 DIAGRAMA GENERAL DE CASOS DE USO	60
2.6 MODELO ESTRUCTURAL DEL SOFTWARE	61
2.7 ARQUITECTURA.....	62
2.8 ÁREA DE TRABAJO SIGEODEP	62
2.9 MÓDULO DE CARGA MASIVA	63
2.10 MÓDULO DE CARGA POR FORMULARIO	65
2.11 MÓDULO CRUCE DE INDICADORES	66

2.12	MÓDULO MAPAS TEMÁTICOS	67
3.	RESULTADOS DEL PROCESO	69
3.1	METODOLOGÍA DE EVALUACIÓN.....	70
3.2	EVALUACIÓN MÓDULO: MÓDULO DE CARGA MASIVA.....	74
3.3	EVALUACIÓN MÓDULO: MÓDULO CARGA POR FORMULARIO.....	80
3.4	EVALUACIÓN MÓDULO: CRUCE DE INDICADORES	86
3.5	EVALUACIÓN MÓDULO: MÓDULO MAPAS TEMÁTICOS	92
4.	CONCLUSIONES.....	101
5.	RECOMENDACIONES	102
	TRABAJOS FUTUROS.....	103
	BIBLIOGRAFÍA.....	104

LISTA DE CUADROS

Pág.

Cuadro 1: Composición de la norma ISO/IEC 9126.....	40
Cuadro 2: Calidad interna y externa de la norma ISO/IEC 9126	42
Cuadro 3: Calidad en uso de la norma ISO/IEC 9126	42
Cuadro 4: Característica Usabilidad	44
Cuadro 5: Característica Mantenibilidad	46
Cuadro 6: Característica Eficiencia.....	48
Cuadro 7: Característica Funcionalidad	50
Cuadro 8: Característica Confiabilidad	52
Cuadro 9: Característica portabilidad.....	54

LISTA DE TABLAS

	Pág.
Tabla 1: Evolución de concepto de calidad total	23
Tabla 2: Concepto de calidad a través del tiempo	23
Tabla 3: Modelo de Furps (1987).....	31
Tabla 4: Costo de un error en las diferentes etapas del proceso.....	32
Tabla 5: Usabilidad	45
Tabla 6: Mantenibilidad	47
Tabla 7: Eficiencia.....	48
Tabla 8: Funcionalidad.....	50
Tabla 9: Confiabilidad	52
Tabla 10: Portabilidad	54
Tabla 11: Niveles de calificación.....	73
Tabla 12: Ponderación general	74
Tabla 13: Ponderación características del módulo carga masiva	75
Tabla 14: Resultados por características	76
Tabla 15: Resultado general módulo de carga masiva	78
Tabla 16: Ponderación de características del módulo carga por formulario	80
Tabla 17: Resultados por Características	82
Tabla 18: Resultado general módulo carga por formulario	83
Tabla 19: Ponderación de características del módulo cruce de indicadores	86
Tabla 20: Resultados por características	88
Tabla 21: Resultado general módulo cruce de indicadores	89
Tabla 22: Ponderación de características del módulo mapas temáticos	92
Tabla 23: Resultados por características	94
Tabla 24: Resultado general módulo mapas temáticos	95
Tabla 25: Consolidado producto software SIGODEP	97

LISTA DE FIGURAS

	Pág.
Ilustración 1: Metodología planteada para el proceso de medición de calidad ..	20
Ilustración 2: Dinámica de componentes de calidad del producto software	27
Ilustración 3: Propuesta del modelo de calidad de McCall (1977)	28
Ilustración 4: Modelo de Boehm (1978)	30
Ilustración 5: Modelo de calidad ISO/IEC 9126	32
Ilustración 6: Representación de pruebas de caja blanca y caja negra	38
Ilustración 7: Cómo actúa la calidad en el ciclo de vida del software	41
Ilustración 8: Diagrama de Módulos.....	59
Ilustración 9: Diagrama general de casos de uso	60
Ilustración 10: Modelo estructural del software SIGEODEP	61
Ilustración 11: Arquitectura del software SIGEODEP	62
Ilustración 12: Interfaz principal SIGEODEP.....	63
Ilustración 13: Diagrama de casos de uso módulo de carga masiva	64
Ilustración 14: Interfaz módulo de carga masiva.....	64
Ilustración 15: Diagrama de casos de uso módulo de carga por formulario	65
Ilustración 16: Interfaz módulo carga por formulario	66
Ilustración 17: Diagrama de casos de uso módulo cruce de indicadores	66
Ilustración 18: Interfaz módulo cruce de indicadores.....	67
Ilustración 19: Diagrama de casos de uso módulo de mapas temáticos	67
Ilustración 20: Interfaz Módulo de Mapas Temáticos.....	68
Ilustración 21: Comparación de resultados del módulo carga masiva	78
Ilustración 22: Comparación de resultados del módulo de carga por formulario .	84
Ilustración 23: Comparación de calificaciones del módulo de cruce de indicadores.	90
Ilustración 24: Comparación de resultados del módulo de mapas temáticos.	96
Ilustración 25: Consolidado producto SIGEODEP	98

GLOSARIO

- **BACKLOG:** es un documento de alto nivel para todo el proyecto. Contiene descripciones genéricas de todos los requisitos, funcionalidades deseables, etc.
- **CICLO DE VIDA DE SOFTWARE:** consta de los procedimientos: Definición de objetivos, análisis de requisitos, diseño general, diseño en detalle, programación, prueba de unidad, integración, documentación, implementación y mantenimiento.
- **COLCIENCIAS:** es el Departamento Administrativo de Ciencia, Tecnología e Innovación de Colombia.
- **DESARROLLADOR:** Se refiere al personal encargado de realizar la programación de cualquier software, en este caso del software SIGEODEP.
- **ESTÁNDARES DE CALIDAD:** es estándar que reúne los requisitos mínimos en busca de la excelencia dentro de una organización institucional.
- **GRIAS:** Grupo de investigación de la universidad de Nariño, encargado del desarrollo del software SIGEODEP.
- **GUI:** Interfaz gráfica de usuario.
- **HERRAMIENTAS CASE:** Ingeniería de software asistida por computadora, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software
- **IEEE:** es la mayor asociación profesional del mundo dedicada al avance de la innovación tecnológica y excelencia en beneficio de la humanidad.
- **ISO:** (Organización Internacional de Normalización) es una organización de membresía no gubernamental independiente y el mayor desarrollador mundial de las Normas Internacionales voluntarias.
- **MINERIA DE DATOS:** proceso de detectar la información procesable de los conjuntos grandes de datos.
- **REGISTROS LOGS:** archivos usados para registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre para una aplicación

- RETROALIMENTACIÓN: es un proceso por el cual se toman datos o información de la salida de un sistema y se redirige a la entrada, con objetivo de controlar y mejorar su comportamiento y/o rendimiento.
- SIGEODEP: Software que será sometido a medición de calidad.
- SPRINT: es el período en el cual se lleva a cabo el trabajo en sí.

INTRODUCCIÓN

La gestión de calidad a nivel mundial cada vez cobra más importancia, porque es la manera como desarrolladores e industria de software pueden garantizar que un producto llega a manos del cliente con los niveles de calidad deseados. Por este motivo organizaciones internacionales de estandarización como la ISO (International Organization for Standardization) y la IEEE (Institute of Electrical and Electronics Engineers) permanentemente formulan nuevas normas y evolucionan las existentes, como estrategia para garantizar la calidad del producto software. Existen altas exigencias por parte de los clientes al momento de invertir en el software que adquieren, hoy en día el software acapara gran parte de las actividades diarias de las personas, por estos motivos, el software que llega a la mano de los usuarios debe tener garantizado un nivel de calidad alto ya que se considera como un factor clave de éxito de las organizaciones.

La calidad se convierte en un importante punto diferenciador, además de aumentar la satisfacción general del cliente, disminuir costos y optimizar los recursos. Los productos o servicios que ostentan certificados de calidad son preferidos por los compradores porque transmiten seguridad y confianza. La creciente preocupación por la calidad en la industria del software tiene como objetivo principal el desarrollo sistemático de productos y servicios de mejor calidad y el cumplimiento de las necesidades y expectativas de los clientes.

Las pruebas de software tienen como objetivo proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada, es una actividad más en el proceso de control de calidad. Las pruebas son un conjunto de actividades dentro del desarrollo de software, estas pueden ser implementadas en cualquier momento de dicho proceso de desarrollo, existen distintos modelos de desarrollo de software, así como modelos de pruebas.

Para la evaluación de un software es necesario ligarse a estándares internacionales de evaluación de la calidad, la Norma ISO/IEC 9126 es un estándar internacional dedicada a esta labor, fue publicada en el año de 1991 bajo el nombre en inglés: "Information technology-Software product evaluation-Quality characteristics and guidelines for their use". Dentro de la comunidad es una de las que más reconocimiento tiene ya que ha sido desarrollada a través de un proceso de investigación realizado durante los últimos 30 años, además que puede ser aplicado a cualquier tipo de software.

El grupo de investigación GRIAS (Grupo de Investigación Aplicada en Sistemas) del Departamento de Sistemas de la Universidad de Nariño en el marco del proyecto: *Detección de Patrones Delictivos con Técnicas de Minería de Datos en*

el Observatorio del Delito del Municipio de Pasto, desarrollado en convenio con la Alcaldía de Pasto y financiado por Colciencias (Departamento Administrativo de Ciencia, Tecnología e Innovación), han desarrollado el software SIGEODEP (Sistema de información Georreferenciado del Observatorio del Delito de Pasto). SIGEODEP soporta el proceso de recolección, integración y procesamiento inicial de los datos recolectados por el Observatorio del Delito referente a las lesiones de causa externa fatal y no fatal que son recopiladas por instituciones de salud, justicia y prevención de la ciudad de Pasto. Por tal motivo, como el equipo de desarrollo desconocía los niveles de calidad del software SIGEODEP, fue sometido a un proceso de medición que permitió determinar con exactitud sus niveles de calidad, ya que las únicas pruebas a las que había sido sometido el software, eran pruebas de laboratorio que no presentan indicadores precisos de la calidad del producto.

El proceso de evaluación fue desarrollado con la colaboración del equipo de desarrollo y con el grupo de usuarios del Observatorio, se definieron las ponderaciones establecidas por la Norma y se desarrolló el proceso el cual determino que el software tiene unos niveles de calidad buenos y con un plan de mejora alcanzara la excelencia en su calidad.

El documento está estructurado de la siguiente manera: en el primer capítulo se presenta inicialmente los elementos de identificación de la investigación, posteriormente se presenta el marco teórico que fundamenta científicamente nuestro trabajo de grado, el tercer capítulo es sobre el software evaluado SIGEODEP donde se tratan aspectos técnicos del mismo, el cuarto capítulo presenta el proceso de evaluación del software con sus resultados y respectivas recomendaciones, y finalmente las conclusiones de la experiencia de esta evaluación y referencias.

INTRODUCCIÓN A LA INVESTIGACIÓN

Introducción al capítulo: En este capítulo se presenta una idea general de los elementos que guían el trabajo de grado estos elementos son; el tema del trabajo de grado, en la línea de investigación que fue inscrito, su alcance, la descripción del problema, los objetivos, la justificación, los antecedentes y la metodología utilizada para el desarrollo del proceso de evaluación.

TEMA

Título: Medición de la Calidad del Sistema de Información Georreferenciado del Observatorio del Delito de Pasto - SIGEODEP utilizando la norma ISO/IEC 9126.

Línea de investigación: Este trabajo de grado se inscribe en la línea de investigación: Software y manejo de Información.

Alcance y delimitación: En este trabajo de investigación se evaluarán aspectos funcionales, de calidad y de información del software SIGEODEP utilizando la norma ISO/IEC 9126, se emplearán como parámetros de medición características, sub-características y atributos que integran la estructura de evaluación de la norma, se construirá un plan de pruebas que guiará el proceso de medición, finalmente los resultados obtenidos serán consolidados en un informe técnico.

Modalidad: El presente trabajo de grado corresponde a la modalidad de trabajo de investigación.

DESCRIPCIÓN DEL PROBLEMA

Planteamiento del problema: La calidad de software actualmente es un tema de alta preocupación por parte de la comunidad académica, científica e industrial de la ingeniería del software porque se requiere cada vez más de software altamente confiable, el software que es desarrollado y puesto en marcha en los diferentes tipos de empresas, sin ser previamente medido, no será tan productivo ni competitivo y mucho menos confiable, como un software que si ha sido sometido a un proceso de medición de calidad.

El software SIGEODEP desarrollado por el grupo de investigación GRIAS, es uno de los productos del proyecto: Detección de Patrones Delictivos con Técnicas de Minería de Datos en el Observatorio del Delito del Municipio de Pasto, realizado por la Universidad de Nariño en convenio con la alcaldía de Pasto y financiado por Colciencias.

El equipo de desarrollo desconoce los niveles de calidad del software SIGEODEP, esto se debe a que únicamente ha sido sometido a pruebas de laboratorio que no presentan indicadores precisos de la calidad del producto, en tal sentido, es conveniente someterlo a un proceso de medición que permita determinar con exactitud sus niveles de calidad.

La ausencia de actividades de medición de calidad del producto software puede resultar costosas tareas de soporte a usuarios insatisfechos, así como altos costos de mantenimiento, la insatisfacción del cliente trae como consecuencia pérdida de imagen o prestigio de la entidad desarrolladora de software, así como una baja de la credibilidad, lo cual genera una alteración de la normalidad de trabajo del usuario final.

Formulación del problema:

¿Cuáles son los niveles de calidad del software SIGEODEP–Sistema de Información Georreferenciado del Observatorio del Delito de Pasto?

Sistematización del problema:

- ¿Cuáles son los aspectos relevantes del software SIGEODEP que deben ser sometidos a un proceso de medición de calidad?
- ¿Cuál es la estructura de medición basada en la norma ISO/IEC 9126 que se aplicará en el proceso de medición de calidad del software SIGEODEP?
- ¿De qué manera se pueden estructurar y definir las pruebas a realizar para medir la calidad del software SIGEODEP?
- ¿Cómo se pueden obtener los valores de calidad asociados la estructura de evaluación de la norma ISO/IEC 9126 aplicada a los aspectos funcionales, de calidad y de información del software SIGEODEP?
- ¿Cómo se pueden presentar los resultados del proceso de medición de calidad del software SIGEODEP?

OBJETIVOS

Objetivo general: Determinar los niveles de calidad del software SIGEODEP – Sistema de Información Georreferenciado del Observatorio del Delito de Pasto utilizando la norma ISO/IEC 9126.

Objetivos específicos:

- Definir los aspectos funcionales, de calidad y de información del software SIGEODEP que serán objeto de medición de la calidad.
- Definir las características, sub-características y atributos de la norma ISO/IEC 9126 que serán empeladas en el proceso de medición de calidad aplicado al software SIGEODEP.
- Construir el plan de pruebas de acuerdo con los aspectos funcionales, de calidad y de información del software SIGEODEP aplicando la estructura de medición de la norma ISO/IEC 9126.
- Ejecutar las pruebas configuradas en el plan para obtener los valores asociados a la estructura de evaluación de la norma ISO/IEC 9126 aplicada al software SIGEODEP.
- Elaborar con base en los resultados del proceso de medición, el reporte de pruebas y el informe de calidad del software SIGEODEP.

JUSTIFICACIÓN

Existen altas exigencias por parte de los clientes al momento de invertir en el software que adquieren. Hoy en día el software acapara gran parte de las actividades diarias de las personas y pueden llegar a manejar desde grandes sumas de dinero hasta vidas de personas. Desde que nos levantamos hasta que nos acostamos estamos en continua relación con aplicaciones software, en tal sentido el software se convierte en un producto que determina en buena parte el nivel de calidad de vida de las personas y factor clave de éxito de las organizaciones.

Para el equipo de desarrollo y para los usuarios es muy importante conocer con exactitud los niveles de calidad de un producto y/o proceso software. Con base en los resultados obtenidos en un proceso de medición de calidad, las organizaciones toman decisiones sobre las estrategias de puesta en operación del producto.

Actualmente el equipo de desarrollo desconoce los niveles de calidad del software SIGEODEP, por lo cual es necesario realizar el proceso de medición de calidad propuesto en este proyecto, de no hacerlo, está expuesto al hallazgo de errores por parte del cliente, hecho que seguramente generará desconfianza en el producto y el consecuente inconformismo de clientes y usuarios. Una eventual falla crítica podría traer como consecuencia pérdidas de tiempo, esfuerzo de trabajo y de recursos.

La medición de calidad del software SIGEODEP permite identificar oportunidades de mejora del producto que de aprovecharse, evita pérdidas para el cliente en tiempo de operación.

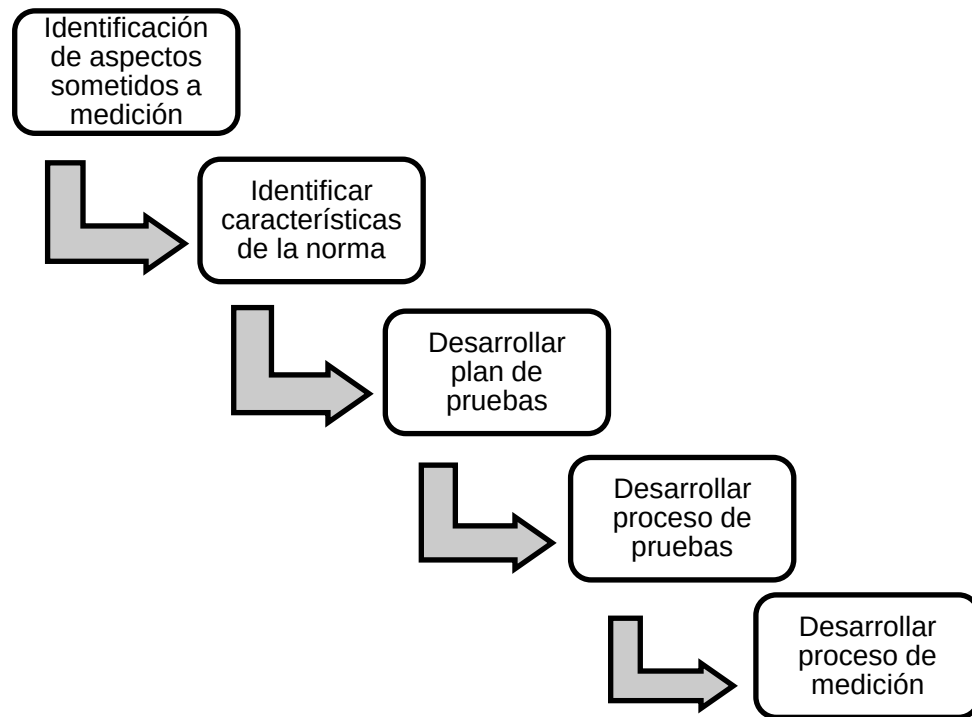
Los aspectos descritos evidencian la necesidad de realizar la medición de la calidad del software SIGEODEP utilizando la norma ISO/IEC 9126 para determinar sus niveles de calidad., de tal manera que el equipo de desarrollo identifique oportunidades de mejora de la calidad del software evaluado.

Como un aporte a la solución de la problemática descrita, este trabajo de grado busca desarrollar el proceso de medición del software SIGEODEP utilizando la norma ISO/IEC 9126 para determinar sus niveles de calidad, de tal manera que el equipo de desarrollo identifique oportunidades de mejora de la calidad del software evaluado.

METODOLOGÍA

Para desarrollar el proceso de medición de la calidad al software SIGEODEP se seguirá la siguiente metodología:

Ilustración 1: Metodología planteada para el proceso de medición de calidad



Primero se identificarán los aspectos funcionales, de calidad y de información del software SIGEODEP factibles de ser sometidos a un proceso de medición de calidad.

Segundo se identificarán las características sub-características y atributos de la norma ISO/IEC 9126 que serán empeladas en el proceso de medición de calidad aplicado al software SIGEODEP.

Tercero se desarrollará el plan de pruebas de acuerdo con los aspectos funcionales, de calidad y de información identificados en el sistema SIGEODEP aplicando la norma ISO/IEC 9126.

Cuarto se desarrollará el proceso de pruebas al software SIGEODEP. Y por último se desarrollará el proceso de medición de calidad.

Cada una de las actividades presentadas en la Ilustración 1 Metodología Planteada para el Proceso de Medición de Calidad, se ha discriminado en tareas para guiar de manera más detallada el desarrollo del proyecto.

1. MARCO TEÓRICO

1.1 INTRODUCCIÓN AL CAPITULO

En este capítulo se describe brevemente los fundamentos teóricos que soportan la investigación, estos fundamentos le dan soporte científico al trabajo de grado, estos elementos son: la calidad de software donde se presentan los diferentes modelos de evaluación que existen, posteriormente las pruebas de software y los diferentes tipos de pruebas y finalmente la Norma ISO/IEC 9126 y su estructura de evaluación.

1.2 CALIDAD DE SOFTWARE

En la actualidad, a nivel mundial la industria del software se ha convertido en un enorme gigante, el cual crece y se desarrolla a ritmo agitado. Todas las empresas quieren producir aplicaciones informáticas con alta calidad, en el menor tiempo posible y a costos mínimos. Aumentando así la competitividad entre ellas debido a que los clientes son cada vez más exigentes. La calidad del software juega un papel importante dentro del desarrollo de aplicaciones informáticas influyendo positivamente en la decisión de un cliente a la hora de escoger el producto que necesita.

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

1.2.1 Aplicación de la calidad en software. La garantía de calidad se inicia con las acciones que se realizan durante la planificación como el conjunto de procedimientos, técnicas y herramientas durante el ciclo de vida, las actividades de auditoría tales como revisiones técnicas o inspecciones, optimizando los criterios de corrección, completitud y las funciones de información de la gestión, más orientadas a la documentación y al desarrollo de pruebas.

En el desarrollo de software, el control de la calidad es realizado por el mismo desarrollador, que dispone de poco tiempo, cuando lo tiene. En otros casos, está ligado a las etapas del ciclo de vida del desarrollo del software o bajo un conjunto de rígidos patrones (listas de chequeo) cuya revisión suele consumir demasiado tiempo, por eso el proceso de medición de calidad casi no se lleva a cabo al

desarrollar software esta es una mala decisión ya que cuando el producto esté en funcionamiento los errores pueden salir más costosos.

La calidad total depende de la calidad con la cual se lleve a cabo todo el proceso - las cinco etapas del ciclo de vida de un desarrollo de software (análisis, diseño, implementación, pruebas e implantación)- y cada subproceso o fase del proyecto, adicionalmente, es indispensable contar con los siguientes componentes: claridad, involucración, planeamiento, estándares, entrenamiento, experiencia, controles, documentación, soporte y finalización. Es importante resaltar que en el proceso de mejora continua se realizan actividades técnicas (aplicación de la ingeniería del software), organizativas (aplicación de modelos de proceso o planes) y humanas (formación del personal y motivación).

En la actualidad, la construcción de software presenta dificultades tales como insuficiencia en la especificación de requisitos, diseño poco profundo, mala gestión de la configuración, poca flexibilidad para la incorporación de cambios, prolongado tiempo de duración y aumento en los costos. Si bien es preciso aclarar que “la demanda de software y la complejidad del producto en sí, parecen crecer a mayor velocidad que las metodologías, el personal capacitado y las herramientas para automatizar la producción. A pesar de la incorporación de herramientas CASE (del inglés Computer Assisted Software Engineering o Ingeniería de Software Asistido por Computadora), la producción de software continúa siendo una actividad con alta participación de recursos humanos, cien por cien intelectual y en cierto sentido, sin insumos ni materias primas”¹.

Calidad total: La calidad se ha convertido hoy en día en un objetivo primordial para todo tipo de empresas. Inicialmente, las organizaciones se enfocaron en el control de calidad utilizando técnicas de inspección para la producción con el fin de detectar defectos en los productos y las actividades. Más adelante nació el concepto de aseguramiento de la calidad como estrategia que permite ofrecer un nivel continuo de calidad en los productos o servicios, llegando con la mejora continua de los procesos y productos, involucrando a todos los miembros de la organización y cuyo centro es la satisfacción del cliente interno y externo. (Ver tabla 1)

¹ Calidad Total en Informática: Mario Iván Cruz

Tabla 1: Evolución de concepto de calidad total

	CONCEPTO DE CALIDAD	FINALIDAD
Control de Calidad	Técnicas de inspección en producción para evitar la salida de bienes defectuosos.	• Satisfacer las necesidades técnicas del producto.
Aseguramiento de la Calidad	Sistemas y procedimientos de la organización para evitar que se produzcan bienes defectuosos.	• Satisfacer al cliente. • Prevenir errores. • Reducir costes. • Ser competitivo.
Calidad Total	Teoría de la administración empresarial centrada en la permanente satisfacción de las expectativas del cliente.	• Satisfacer tanto al cliente externo como interno. • Ser altamente competitivo. • Mejora Continua.

Calidad en el ámbito computacional: El término calidad en la ingeniería de software se ha desarrollado desde hace ya varias décadas y continúa siendo un tema vigente hoy en día debido al continuo crecimiento de los sistemas de información. En la tabla 2, se pueden observar diferentes apreciaciones sobre el concepto de calidad a través del tiempo.

Tabla 2: Concepto de calidad a través del tiempo

ETAPA	CONCEPTO DE CALIDAD	FINALIDAD
Artesanal	Hacer las cosas bien independientemente del costo o esfuerzo necesario para ello.	❖ Satisfacer al cliente. ❖ Satisfacer al artesano, por el trabajo bien hecho. ❖ Crear un producto único.
Revolución Industrial	Hacer muchas cosas sin importar si son de calidad. (Se identifica Producción con Calidad).	❖ Satisfacer una gran demanda de bienes. ❖ Obtener beneficios.
Segunda Guerra Mundial	Asegurar la eficacia del armamento sin importar el costo, con la mayor y más rápida producción. (Eficacia + Plazo = Calidad).	❖ Garantizar la disponibilidad de un armamento eficaz en la cantidad y el momento preciso.

ETAPA	CONCEPTO DE CALIDAD	FINALIDAD
Postguerra (Japón)	Hacer las cosas bien desde la primera vez.	❖ Minimizar costes mediante la Calidad. ❖ Satisfacer al cliente. ❖ Ser competitivo.
Postguerra (Resto del mundo)	Producir. Cuanto más, mejor.	❖ Satisfacer la gran demanda de bienes causada por la guerra.

El concepto de calidad se ha convertido en un término intuitivo, utilizado popular y profesionalmente, lo que le otorga una faceta multidimensional, por lo tanto, puede hablarse de calidad desde diversos puntos de vista: el de las personas, el de los recursos, de los procesos, de los productos, de las organizaciones, entre otros. En el presente trabajo se enfatizará el punto de vista del producto, y específicamente, del producto de software en uso.

Calidad del producto software: Se habla todo el tiempo de problemas relacionados con la calidad del software pero no se tiene una definición precisa de lo que ésta significa. Sin una definición clara, concisa y medible de lo que es la calidad del software, no se puede tomar buenas decisiones de negocio respecto del uso de los recursos, ni en qué áreas mejorar la calidad, ni que herramientas y técnicas utilizar para mejorar la calidad.

Con relación a la calidad del software, se han planteado algunas definiciones. Algunas de ellas son:

Según Pressman² la calidad del software se define como:

“La concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”.

El estándar de la IEEE para la calidad de software³ define el término calidad del software como:

² PRESSMAN S, Roger. Ingeniería de Software Un enfoque práctico. México: McGraw Hill, 1993.

³ IEEE STD 729 – 1983. Glossary of Software Engineering Terminology.

- La totalidad de rasgos y características de un producto de software que se refieren a su habilidad para satisfacer necesidades específicas.
- El grado en el cual el software posee una combinación deseada de atributos.
- El grado en el cual un usuario o cliente percibe que el software cumple con sus expectativas.
- Las características del software que determinan el grado en el cual dicho software en uso cumple con las expectativas del cliente.

Desde el punto de vista del cliente o usuario Watts Humphrey dice:

“El foco principal de cualquier definición de calidad de software debería ser las necesidades del cliente. Entonces las preguntas claves son ¿Quiénes son los usuarios?, ¿Qué es importante para ellos? Y ¿Cómo sus prioridades se relacionan con la manera en que se construye, empaqueta y se da soporte al producto?”. Más allá de cómo se define la calidad del software, para que la definición tenga sentido esta debe ser medible. Para poder controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición, ya que, como bien plantea Tom De Marco, "no se puede controlar lo que no se puede medir".

Para poder identificar los costos y beneficios del software se definieron los atributos de calidad. La intención es separar el software en atributos que puedan ser medidos o cuantificados (en términos de costo beneficio). Ejemplos de estos atributos son confiabilidad, adaptabilidad, usabilidad y funcionalidad.

La calidad del software no debe observarse como un elemento aislado sino como una elaboración que se construye durante todo el ciclo de vida del software. La calidad se encuentra presente en cada etapa del ciclo de vida siempre y cuando en cada una de ellas se satisfagan rigurosamente las especificaciones establecidas. Por lo tanto, la calidad del producto software está determinada tanto por el grado de cumplimiento de los requisitos del usuario, como por la profundidad de las pruebas y la cobertura de la documentación. Así, la calidad del producto está influenciada por la calidad de los recursos (personas, herramientas, etc.), de los procesos (procesos de desarrollo bien definidos, organizados, documentados, etc.), y en general, por la calidad de todo elemento participante en el ciclo de vida del software.

Algunos estudios relacionados con el tema, plantean la posibilidad de evaluar la calidad de los productos de software diferenciando calidad interna y calidad externa.

Se entiende por calidad interna la totalidad de características del software producidas desde una perspectiva interna. Dichas características permanecen generalmente constantes a lo largo del ciclo de vida del software, aunque éste sea rediseñado y programado nuevamente. La calidad interna se mide utilizando métricas internas que evalúan aspectos del producto sin abarcar su comportamiento ni su entorno. Los aspectos relacionados con la calidad interna son derivados del producto en sí mismo y no de su comportamiento en un sistema específico, por lo tanto, la calidad interna no se aplica a objetos ejecutables, sino que se evalúa en las etapas tempranas del ciclo de vida del producto, es decir, durante el análisis, diseño y codificación.

La calidad interna, solo es perceptible a los profesionales del grupo de Ingeniería de Software:

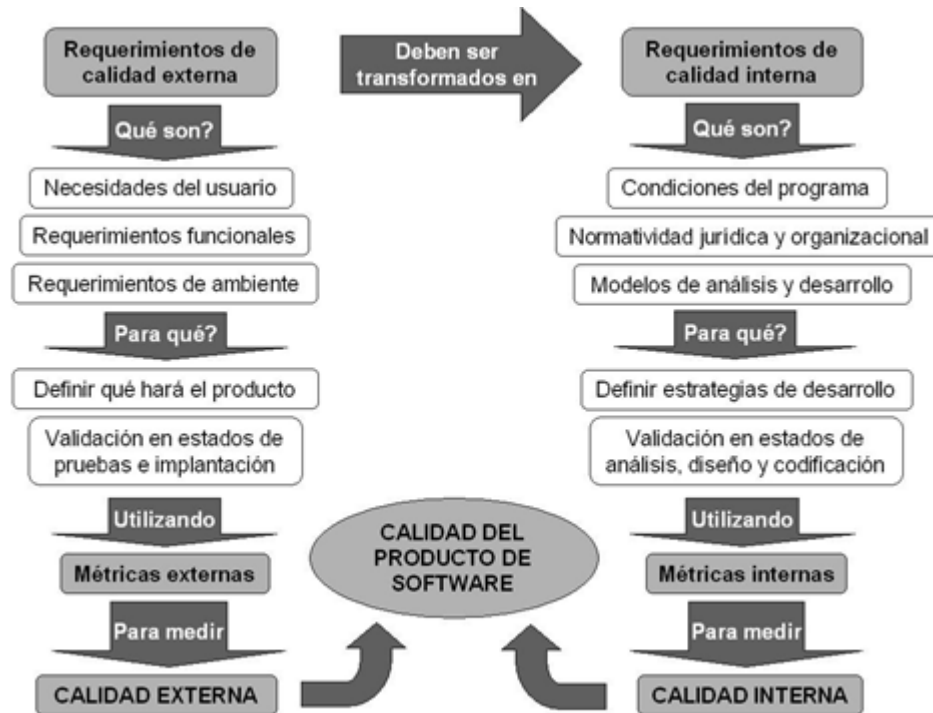
- Buenas prácticas de desarrollo y mantenimiento del software, explícitamente documentadas.
- Satisfacción usuario de mantenimiento.

La calidad externa se refiere a la totalidad de características del software producidas desde una perspectiva externa, las cuales se evalúan cuando el producto se encuentra en ejecución, al realizar pruebas en un ambiente simulado utilizando datos de prueba. La calidad externa se evalúa utilizando métricas externas, las cuales tienen en cuenta el comportamiento del producto en un entorno determinado y se aplican al producto en etapas finales del ciclo de vida del software como: integración, pruebas, implementación y mantenimiento. (Ver ilustración 1)

La calidad externa, puede ser detectada por el usuario:

- Concordancia con los requisitos explícitamente definidos.
- Satisfacción del usuario final.

Ilustración 2: Dinámica de componentes de calidad del producto software



Los conceptos de calidad aportaron en este trabajo de grado para medir la calidad interna como externa del software SIGODEP, a partir de ellos se identificaron las pruebas que se realizaron tanto de caja blanca como de caja negra para un mejor análisis de las funcionalidades que presenta el software.

Modelos de calidad de software: Los modelos de calidad de software surgieron a mediados de los años 70's, conformando una base importante para las evaluaciones de los productos de software más reconocidas que se utilizan actualmente. Los modelos permiten evaluar la calidad de un producto de software definiendo objetivos específicos de calidad. Uno de los enfoques más utilizados para modelos de calidad de software consiste en descomponer jerárquicamente la calidad en características, sub-características, atributos, ítems, entre otros, que pueden ser utilizados como puntos clave en la observación de los aspectos relevantes del producto a evaluar. Cada modelo utiliza un conjunto diferente de ítems a observar, según el aspecto que se desea evaluar.

Algunos de los modelos más utilizados son:

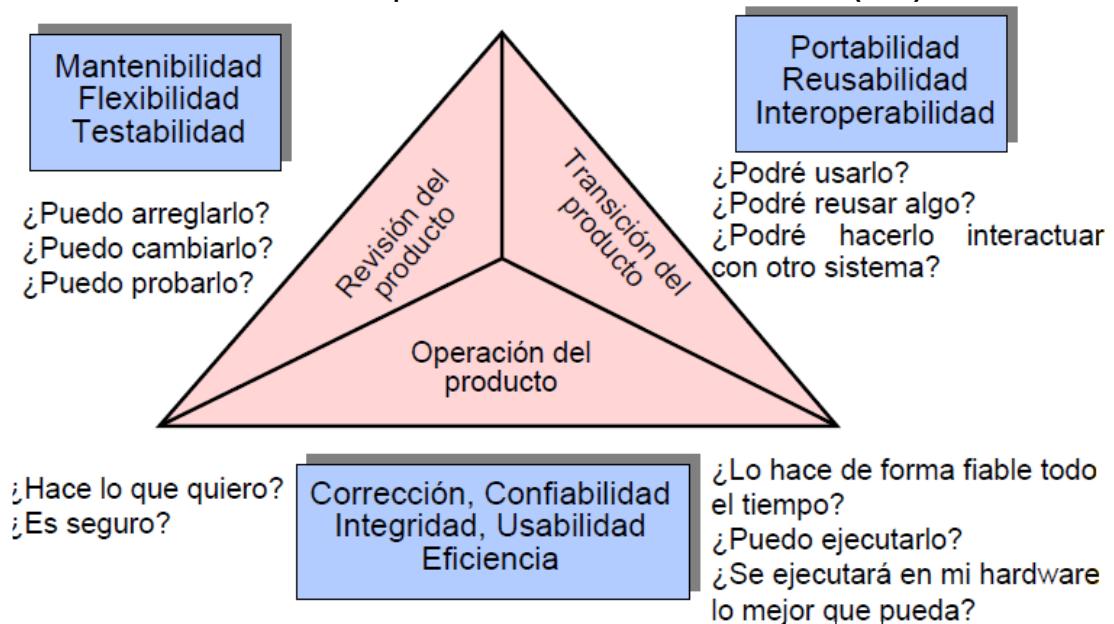
- Modelo de McCall.
- Modelo de Gilb.
- Modelo de Boehm.

- Modelo FURPS+ (Hewlett Packard).
- Modelo propuesto por la norma ISO/IEC 9126.

a. Modelo de McCall

El modelo de McCall, también llamado modelo factores/criterios/métricas, se basa en tres aspectos fundamentales, orientados a la visión del usuario: la revisión del producto, la transición del producto y la operación del producto, cada uno de los cuales enmarca un conjunto determinado de características y busca dar respuesta a una serie de preguntas con el fin de identificar el grado de calidad del producto y en qué medida satisface las necesidades del usuario. (Ver ilustración 2)

Ilustración 3: Propuesta del modelo de calidad de McCall (1977)



b. Modelo de Gilb

El modelo de Gilb en 1988, presenta como aspecto fundamental la definición de los atributos de calidad que realmente interesan al usuario y el nivel de calidad que debe tener cada uno de ellos para satisfacerlo, ya que no tiene sentido exigir calidad en un producto, si no se cuenta con esta base.

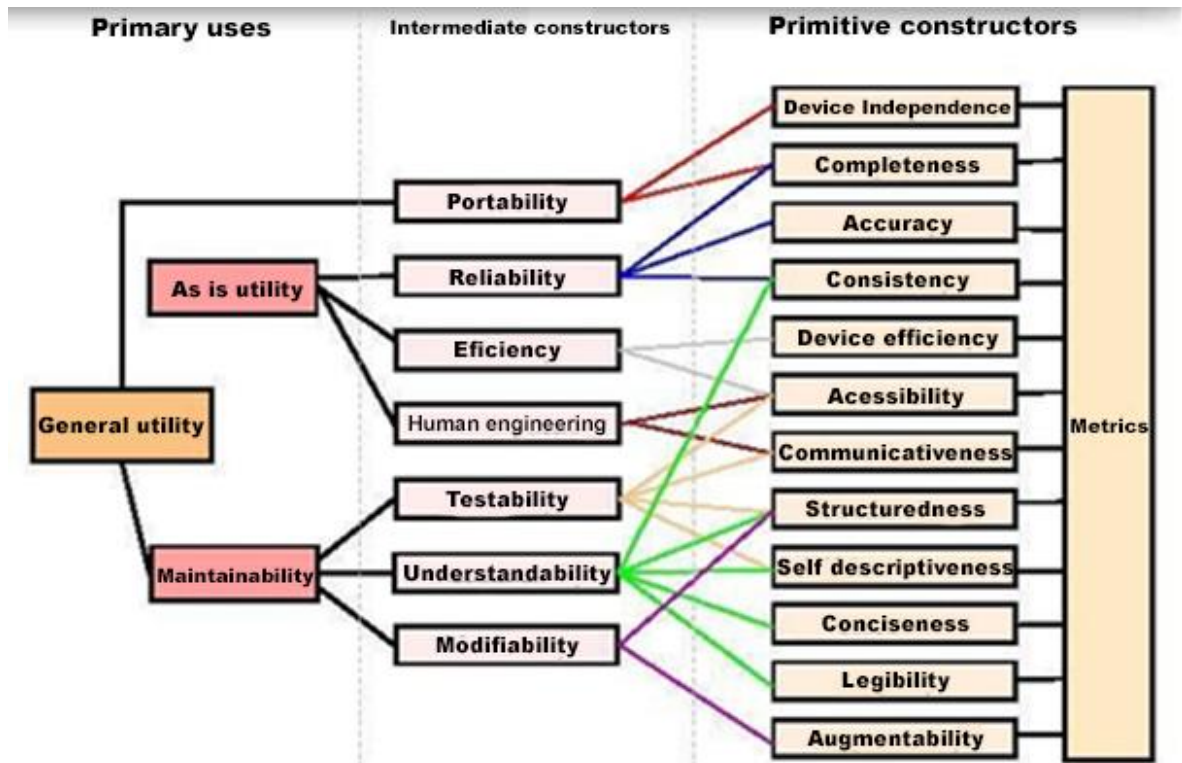
Gilb propone características como la corrección, la integridad, la facilidad de mantenimiento y la facilidad de uso, como base para proporcionar indicadores útiles para los equipos de trabajo y sugiere las definiciones, puntos de vista y medidas para cada uno de las siguientes características:

- **Corrección:** Grado en el que el software lleva a cabo su función requerida. Si un programa no opera correctamente, no dará valor agregado a sus usuarios.
- **Facilidad de mantenimiento:** Posibilidad de corregir un programa si se encuentra un error, adaptarlo si cambia su entorno, o mejorarlo si el cliente desea un cambio.
- **Integridad:** Habilidad de un sistema para resistir ataques, tanto accidentales como intencionados, contra su seguridad, a nivel de cualquiera de los tres principales componentes del software: programas, datos y documentos. Para medir la integridad, Gilb sugiere la utilización de otros dos atributos como base: amenaza y seguridad.
- **Amenaza:** es la probabilidad (que se puede estimar o deducir de la evidencia empírica) de que un ataque de cualquier tipo ocurra en un tiempo determinado.
- **Seguridad:** es la probabilidad de que se pueda repeler un determinado ataque.
- **Facilidad de uso:** Es un intento por cuantificar “lo amigable que puede ser el producto con el usuario”.

c. Modelo de Boehm

El modelo de calidad expuesto por Boehm en 1978, propone la categorización de algunas características de calidad del software en tres niveles: usos principales, componentes intermedios y componentes primitivos, dentro de los cuales se agrupan diferentes características, dependiendo del nivel al que pertenezcan y a su grado de especialidad, de tal manera que, al pasar de nivel, la cantidad de características es mayor, buscando abarcar el concepto de calidad de software en su totalidad y finalizando en la aplicación de métricas con el fin de obtener indicadores y resultados claros, correspondientes al nivel de calidad de un producto de software. (Ver ilustración 4)

Ilustración 4: Modelo de Boehm (1978)



d. Modelo de Furps

Este modelo, basado en el modelo de la Norma ISO/IEC 9126, busca clasificar los diferentes tipos de requisitos o restricciones en categorías (restricciones de diseño, requisitos de Implementación, requisitos de interface, requisitos físicos), de tal manera que se pueda observar con facilidad qué aspectos se deben tener en cuenta para cumplir con dichos requisitos y asegurar la calidad de un producto.

Bajo el acrónimo FURPS+, por sus siglas en inglés, se definen las siguientes características: (Ver tabla 3)

Tabla 3: Modelo de Furps (1987)

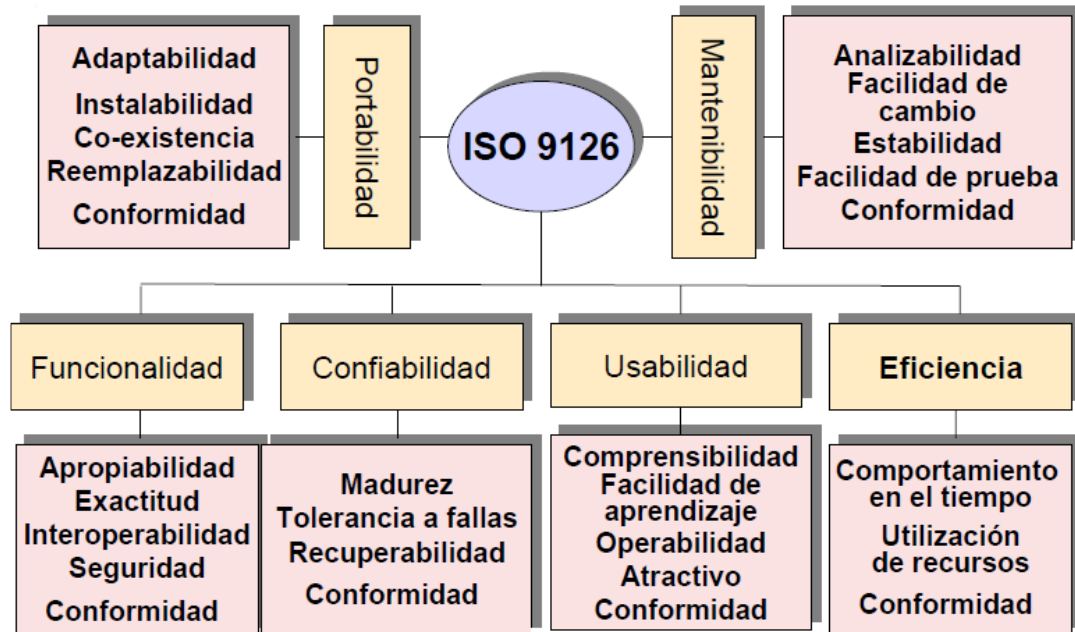
Sigla	Tipo de Requerimiento		Descripción
F	Functional	Funcional	Características, capacidades y algunos aspectos de seguridad.
U	Usability	Facilidad de Uso	Factores Humanos (interacción), ayuda, documentación.
R	Reliability	Fiabilidad	Frecuencia de fallos, capacidad de recuperación de un fallo y grado de previsión.
P	Performance	Rendimiento	Tiempos de respuesta, productividad, precisión, disponibilidad, uso de los recursos.
S	Supportability	Soporte	Adaptabilidad, facilidad de mantenimiento, internacionalización, facilidad de configuración.
+	Plus	Implementación	Limitación de recursos, lenguajes y herramientas hardware.
		Interfaz	Restricciones impuestas para la interacción con sistemas externos (no es GUI).
		Operaciones	Gestión del sistema, pautas administrativas, puesta en marcha.
		Empaquetamiento	Forma de distribución.
		Legales	Licencia, derechos de autor, etc.

e. Norma ISO/IEC 9126: Modelo de calidad de producto software

La norma internacional ISO/IEC 9126 proporciona las guías de un modelo de calidad para un producto de software. Esta norma sigue el esquema propuesto por

los modelos anteriores (McCall, Boehm, etc.), además, retoma algunas de las características planteadas por dichos modelos, pero puntualiza seis características que considera fundamentales en la evaluación de calidad de un producto de software, a saber: la funcionalidad, la confiabilidad, la usabilidad, la eficiencia, la mantenibilidad y la portabilidad. (Ver ilustración 5)

Ilustración 5: Modelo de calidad ISO/IEC 9126



En la siguiente tabla 4, Pressman muestra el costo relativo de corregir un error en las distintas etapas del ciclo de vida.

Tabla 4: Costo de un error en las diferentes etapas del proceso

Fase	Requisitos	Diseño	Código	Prueba (desarrollo)	Prueba (sistema)	Explotación, Producción
Multiplicador Coste	1	3-6	10	15-40	30-70	40-1000

Como se puede observar en la Tabla 4, el costo relativo de corregir un error en producción es cientos de veces más que en la etapa de requisitos. El 50% del costo de corrección de defectos en las últimas fases se debe a defectos introducidos en las primeras.

Entonces uno de los propósitos que guían el aseguramiento de la calidad es que es mucho menos costoso corregir los problemas en sus fases iniciales que esperar hasta que un problema se manifieste a través de las quejas del usuario.

1.3 PRUEBAS DE SOFTWARE

Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo. A cada uno corresponde un nivel distinto de involucramiento en las actividades de desarrollo.

Concretamente la prueba de software se puede definir como una actividad en la cual un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificadas (configuración de la prueba), registrándose los resultados obtenidos. Seguidamente se realiza un proceso de evaluación en el que los resultados obtenidos se comparan con los resultados esperados para localizar fallos en el software. Estos fallos conducen a un proceso de depuración en el que es necesario identificar la falta asociada con cada fallo y corregirla, pudiendo dar lugar a una nueva prueba, para obtener un cierto nivel de confianza en el software probado.

Las pruebas de software son bastantes importantes en el ciclo de vida del producto software ya que ayudan a identificar posibles inconvenientes que se van teniendo en cada etapa del desarrollo.

Características y fases de la prueba: El objetivo de las pruebas no es asegurar la ausencia de defectos en un software, únicamente puede demostrar que existen defectos en el software. Nuestro objetivo es pues, diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Para ser más eficaces (es decir, con más alta probabilidad de encontrar errores), las pruebas deberían ser realizadas por un equipo independiente al que realizó el software. El ingeniero de software que creó el sistema no es el más adecuado para llevar a cabo las pruebas de dicho software, ya que inconscientemente tratará de demostrar que el software funciona, y no que no lo hace, por lo que la prueba puede tener menos éxito.

Una prueba de software, comparando los resultados obtenidos con los esperados. A continuación se presentan algunas características de una buena prueba:

- Una buena prueba ha de tener una alta probabilidad de encontrar un fallo. Para alcanzar este objetivo el responsable de la prueba debe entender el software e intentar desarrollar una imagen mental de cómo podría fallar.

- Una buena prueba debe centrarse en dos objetivos:
 - Probar si el software no hace lo que debe hacer.
 - Probar si el software hace lo que no debe hacer.
- Una buena prueba no debe ser redundante. El tiempo y los recursos son limitados, así que todas las pruebas deberían tener un propósito diferente.
- Una buena prueba debería ser la “mejor de la cosecha”. Esto es, se debería emplear la prueba que tenga la más alta probabilidad de descubrir una clase entera de errores.
- Una buena prueba no debería ser ni demasiado sencilla ni demasiado compleja, pero si se quieren combinar varias pruebas a la vez se pueden enmascarar errores, por lo que en general, cada prueba debería realizarse separadamente.

Aquí se muestra cuáles son las tareas a realizar para probar un software:

- a. Diseño de las pruebas: esto es, identificación de la técnica o técnicas de pruebas que se utilizarán para probar el software. Distintas técnicas de prueba ejercitan diferentes criterios como guía para realizar las pruebas. Seguidamente se podrá ver algunas de estas técnicas.
- b. Generación de los casos de prueba: los casos de prueba representan los datos que se utilizarán como entrada para ejecutar el software a probar. Más concretamente los casos de prueba determinan un conjunto de entradas, condiciones de ejecución y resultados esperados para un objetivo particular. Por lo tanto, durante la tarea de generación de casos de prueba, se han de confeccionar los distintos casos de prueba según la técnica o técnicas identificadas previamente. La generación de cada caso de prueba debe ir acompañada del resultado que ha de producir el software al ejecutar dicho caso.
- c. Definición de los procedimientos de la prueba: esto es, especificación de cómo se va a llevar a cabo el proceso, quién lo va a realizar, cuándo, donde, etc.
- d. Ejecución de la prueba: aplicando los casos de prueba generados previamente e identificando los posibles fallos producidos al comparar los resultados esperados con los resultados obtenidos.
- e. Realización de un informe de la prueba: con el resultado de la ejecución de las pruebas, qué casos de prueba pasaron satisfactoriamente, cuáles no, y qué fallos se detectaron. Tras estas tareas es necesario realizar un proceso de depuración de las faltas asociadas a los fallos identificados.

1.3.1 Tipos de pruebas:

a. Pruebas unitarias

Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente del sistema una vez que ha sido codificado. La prueba de unidad es la primera fase de las pruebas dinámicas y se realizan sobre cada módulo del software de manera independiente. El objetivo es comprobar que el módulo, entendido como una unidad funcional de un programa independiente, está correctamente codificado. En estas pruebas cada módulo será probado por separado y lo hará, generalmente, la persona que lo creó.

b. Pruebas de integración

Aun cuando los módulos de un programa funcionen bien por separado es necesario probarlos conjuntamente: un módulo puede tener un efecto adverso o inadvertido sobre otro módulo; las subfunciones, cuando se combinan, pueden no producir la función principal deseada; la imprecisión aceptada individualmente puede crecer hasta niveles inaceptables al combinar los módulos; los datos pueden perderse o malinterpretarse entre interfaces, etc.

Por lo tanto, es necesario probar el software ensamblando todos los módulos probados previamente. Éste es el objetivo de las pruebas de integración.

c. Pruebas de validación

Las pruebas de validación empiezan tras la culminación de la prueba de integración, cuando se han ejercitado los componentes individuales. Se ha terminado de ensamblar el software como paquete y se han descubierto y corregido los errores de interfaz.

La validación se define de una forma simple en que se alcanza cuando el software funciona de tal manera que satisface las expectativas razonables del cliente (especificación de requisitos-criterios de validación).

d. Pruebas de sistema

Las pruebas de sistema buscan diferencias entre la solución desarrollada y los requerimientos, enfocándose en la identificación de los errores que se puedan generar entre la especificación funcional y el diseño del sistema, así como, el negocio objeto de la aplicación.

Este tipo de pruebas tiene como propósito ejercitar profundamente el sistema para verificar que se han integrado adecuadamente todos los elementos del sistema.

(hardware, otro software, etc.) y que realizan las funciones adecuadas. Concretamente se debe comprobar que:

- Se cumplen los requisitos funcionales establecidos.
- El funcionamiento y rendimiento de las interfaces hardware, software y de usuario.
- La adecuación de la documentación de usuario.
- Rendimiento y respuesta en condiciones límite y de sobrecarga.

e. Pruebas de interoperabilidad

En esta prueba se valida que el sistema se comuniquen de manera exitosa con los sistemas externos con que se requiera, de acuerdo a los requerimientos no funcionales. Estos sistemas pueden ser sistemas propios de cada entidad, servicios de estampado de tiempo, servicios para la validación de certificados digitales, servicios publicados en el tramitador transaccional, servicios de envío de SMS, servicios de envío de correo electrónico.

f. Pruebas de regresión

En esta prueba se valida que el sistema mantenga su correcta funcionalidad después de la incorporación de un ajuste, corrección o nuevo requerimiento. Es una prueba funcional y técnica que valida que el sistema siga funcionando perfectamente después de que las correcciones sean aplicadas.

La regresión consiste en la repetición selectiva de pruebas para detectar fallos introducidos durante la modificación de un sistema o componente de un sistema. Se efectuarán para comprobar que los cambios no han originado efectos adversos no intencionados o que se siguen cumpliendo los requisitos especificados.

En las pruebas de regresión hay que:

- Probar íntegramente los módulos que se han cambiado.
- Decidir las pruebas a efectuar para los módulos que no han cambiado y que han sido afectados por los cambios producidos.

Este tipo de pruebas ha de realizarse, tanto durante el desarrollo cuando se produzcan cambios en el software, como durante el mantenimiento.

g. Pruebas funcionales

La prueba funcional es un proceso para procurar encontrar discrepancias entre el software desarrollado y la especificación funcional. La prueba funcional normalmente es una actividad de caja negra. Esta prueba permite validar:

- Los procesos y reglas de negocio establecidas.
- Que se cumplan los requerimientos funcionales establecidos.

En esta prueba se validan los Casos de Uso que fueron aprobados por el cliente, y a partir de ellos se diseñan y ejecutan los set de pruebas correspondientes. Se deben elaborar los casos de pruebas necesarios que permitan asegurar el funcionamiento de todos los flujos normales y alternos de dichos casos de uso.

h. Pruebas de seguridad

Estas pruebas tienen dos enfoques:

- Pruebas de seguridad de la aplicación; donde se verifica que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido.
- Pruebas de seguridad del sistema; donde se verifica que solo los actores con acceso al sistema y a la aplicación están habilitados para accederla.

i. Pruebas de aceptación

El objetivo de las pruebas de aceptación es validar que la solución desarrollada cumpla con el funcionamiento esperado y permitir al usuario de dicho sistema determine su aceptación, desde el punto de vista de su funcionalidad y de su rendimiento. Estas pruebas son realizadas por el cliente, donde comprueba que el sistema cumple con lo definido y se obtiene la conformidad del cliente. Esta prueba se realiza mediante el proceso de validación de caja negra.

Estas pruebas se caracterizan por:

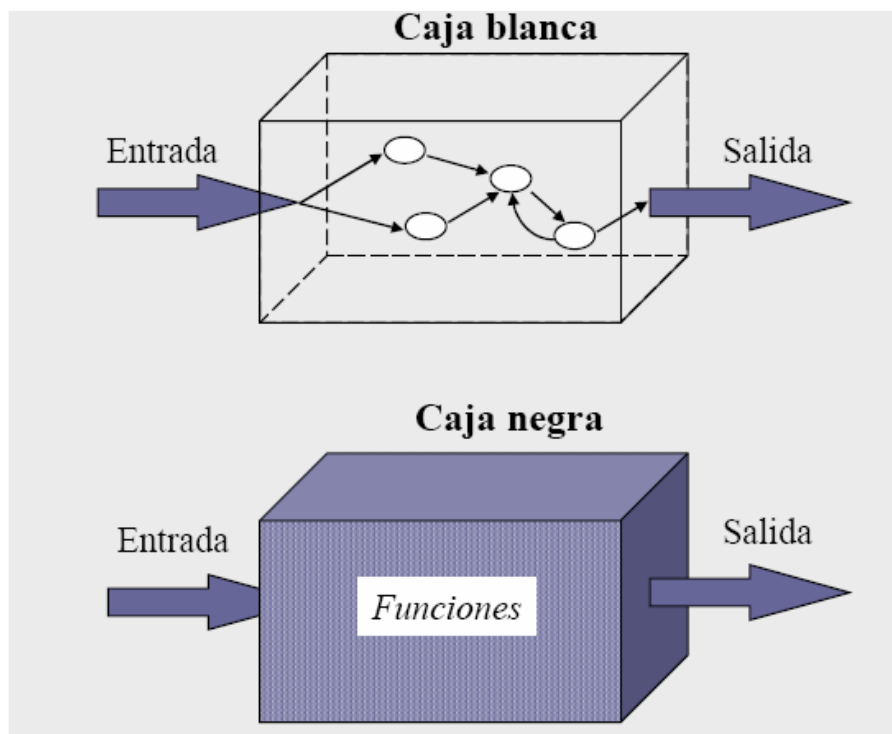
- Participación activa del usuario, que debe ejecutar los casos de prueba ayudado por miembros del equipo de pruebas.
- Están enfocadas a probar los requisitos de usuario, o mejor dicho a demostrar que no se cumplen los requisitos, los criterios de aceptación o el contrato. Si no se consigue demostrar esto el cliente deberá aceptar el producto.
- Corresponden a la fase final del proceso de desarrollo de software.

j. Técnicas de pruebas de software

Como se ha indicado anteriormente, las técnicas de evaluación dinámica o prueba proporcionan distintos criterios para generar casos de prueba que provoquen fallos en los programas. Estas técnicas se agrupan en:

- Técnicas de caja blanca o estructural, que se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.
- Técnicas de caja negra o funcionales, que realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar. (Ver ilustración 6)

Ilustración 6: Representación de pruebas de caja blanca y caja negra



k. Pruebas de caja blanca o estructural

A este tipo de técnicas se le conoce también como Técnicas de Caja Transparente o de Cristal. Este método se centra en cómo diseñar los casos de prueba

atendiendo al comportamiento interno y la estructura del programa. Se examina así la lógica interna del programa sin considerar los aspectos de rendimiento.

El objetivo de la técnica es diseñar casos de prueba para que se ejecuten, al menos una vez, todas las sentencias del programa, y todas las condiciones tanto en su vertiente verdadera como falsa.

I. Pruebas de caja negra o funcional.

También conocidas como Pruebas de Comportamiento, estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una “Caja Negra” cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas. No obstante, como el estudio de todas las posibles entradas y salidas de un programa sería impracticable se selecciona un conjunto de ellas sobre las que se realizan las pruebas. Para seleccionar el conjunto de entradas y salidas sobre las que trabajar, hay que tener en cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en nuestro sistema, y como consecuencia producen una serie de salidas que revelan la presencia de defectos. Entonces, dado que la prueba exhaustiva es imposible, el objetivo final es pues, encontrar una serie de datos de entrada cuya probabilidad de pertenecer al conjunto de entradas que causan dicho comportamiento erróneo sea lo más alto posible.

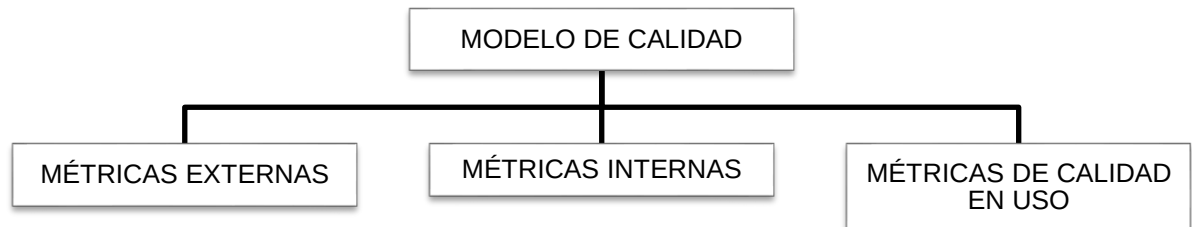
A partir de los diferente tipos de pruebas se definieron las pruebas que sirvieron para medir el correcto funcionamiento y el cumplimiento de los requisitos planteados al inicio del proyecto, al ejecutarlas, los resultados dan un concepto del grado de funcionalidad del software para su posterior análisis y así notificar la calidad del mismo.

1.4 NORMA ISO/IEC 9126: EVALUACION DE PRODUCTOS DE SOFTWARE

Por su estructura evaluativa, la norma facilita la evaluación del producto software desde diferentes criterios asociados a la ingeniería de requisitos, adquisición de datos e información, desarrollo, uso, evaluación, mantenimiento, soporte y auditoria, entre otros. Los objetivos de la norma son varios, algunos ejemplos de ellos serían el validar el proceso de definición de requisitos utilizado en el desarrollo del software, diseño y pruebas internas, garantizar la calidad del producto, definir los criterios de aceptación de un producto software ya terminado, etc. (Ver cuadro 1)

1.4.1 Partes de la norma. La norma ISO/IEC 9126 se compone por cuatro partes que se relacionan como se muestra a continuación:

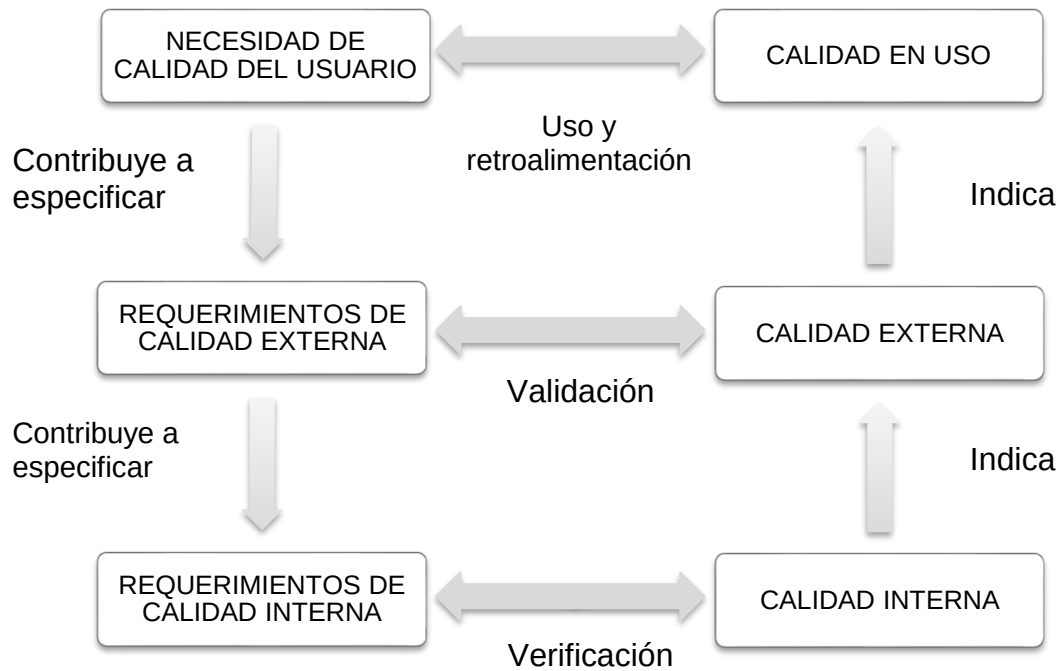
Cuadro 1: Composición de la norma ISO/IEC 9126



Según este modelo, la calidad se descompone en cuatro partes: modelo de calidad, calidad interna, calidad externa y calidad en uso, de esta forma están cubiertas las necesidades del usuario, desarrolladores y del mismo producto software.

La siguiente grafica muestra cómo actúa la calidad en el ciclo de vida del software y permite explicar que por medio de las necesidades de calidad que tiene el usuario para con el software, se debe facilitar la definición de los requerimientos de calidad externa y por medio de estos los requerimientos de calidad interna. (Ver ilustración 7)

Ilustración 7: Cómo actúa la calidad en el ciclo de vida del software



Las métricas externas, hacen referencia a las etapas finales en su proceso de desarrollo del producto software, es decir, integración, pruebas, implementación y mantenimientos, es la parte observable del producto software, cuando el código es ejecutado en un entorno requerido, estas métricas establecen una escala cuantitativa.

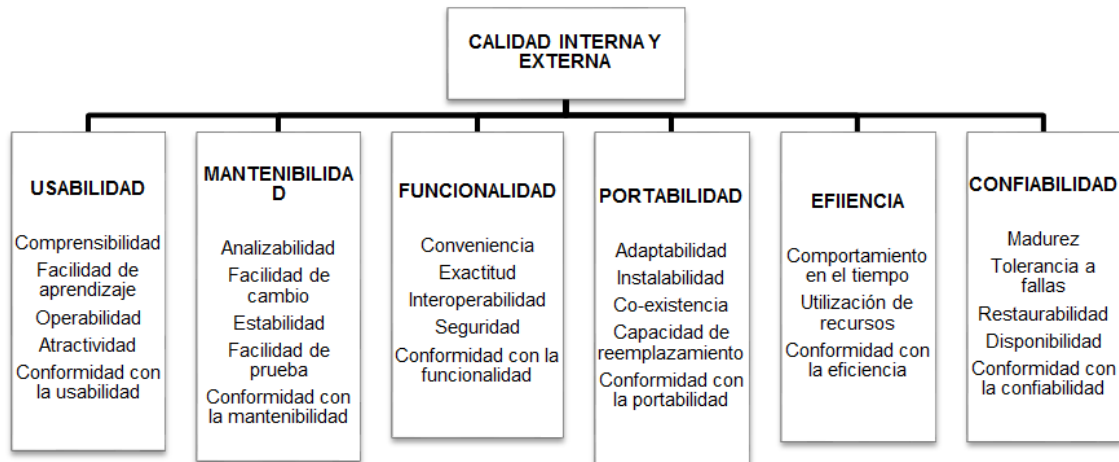
Las métricas internas, hacen referencia a las etapas iniciales del producto software en su proceso de desarrollo, es decir, durante el análisis, diseño y codificación, en estas métricas se incluyen los diferentes documentos generados durante el proceso de desarrollo, entre los cuales se encuentran: especificación de requerimientos, la arquitectura, metodología, código fuente etc. Estas métricas establecen una escala cuantitativa.

Las métricas de calidad en uso, hacen referencia a la interrelación entre el usuario y el producto software desde el punto de vista de la eficiencia y la satisfacción, es decir, la capacidad que tiene el software para facilitar al usuario realizar sus respectivas actividades con eficacia, seguridad, productividad y completa satisfacción.

La norma ISO/IEC 9126 establece que las métricas internas y externas tienen su modelo calidad, que está conformado por seis características: mantenibilidad,

usabilidad, funcionalidad, confiabilidad, portabilidad y eficiencia; cada una de las cuales posee sus respectivas sub-características, lo que permite profundizar en la evaluación de la calidad del producto software. (Ver cuadro 2)

Cuadro 2: Calidad interna y externa de la norma ISO/IEC 9126



La calidad en uso también tiene su modelo de calidad que está conformado por cuatro características: eficacia, seguridad, productividad y satisfacción. (Ver cuadro 3)

Cuadro 3: Calidad en uso de la norma ISO/IEC 9126



Efectividad: Se refiere a la capacidad del producto de software para brindar la posibilidad al usuario de alcanzar metas u objetivos especificados con exactitud y completitud en un contexto de uso particular.

Productividad: Se refiere a la capacidad del producto de software para permitir a los usuarios demandar una cantidad apropiada de recursos en relación a la efectividad alcanzada en un contexto de uso específico.

Seguridad: Se refiere a la capacidad del producto software para alcanzar niveles aceptables de riesgo en un contexto especial. Se habla de riesgo a nivel de daños o perjuicios a las persona, al negocio, al software, a las propiedades o al medio ambiente.

Satisfacción: Se refiere a la capacidad del producto de software para satisfacer a los usuarios en un marco de trabajo particular, entendiendo por satisfacción como una respuesta de los usuarios a la interacción con el producto incluyendo las actitudes producidas frente a su uso.

Para la evaluación, especificación o requerimientos de la calidad de cualquier producto software utilizando la norma, es necesario cumplir los siguientes requisitos:

- Utilizar las seis características y sub-características establecidas.
- Describir la jerarquía de atributos propios para la evaluación.
- Proveer un cuadro sinóptico o mapa conceptual que permita visualizar las características y sub-características empleadas en la evaluación.
- Si la especificación o el requerimiento contiene métricas comparativas, se debe aclarar cómo dichas métricas cumplen con las propiedades definidas por la norma.

1.4.2 Metodología propuesta para la evaluación de calidad de productos de software en uso:

a. Usabilidad

Capacidad del producto de software para ser atractivo, comprendido, aprendido y utilizado por el usuario cuando es usado bajo condiciones específicas; La usabilidad de un sistema pretende determinar qué tan fácil es para un usuario, independiente del tipo de usuario que sea, enfrentarse al producto de software. (Ver cuadro 4) (Ver tabla 5)

Cuadro 4: Característica Usabilidad

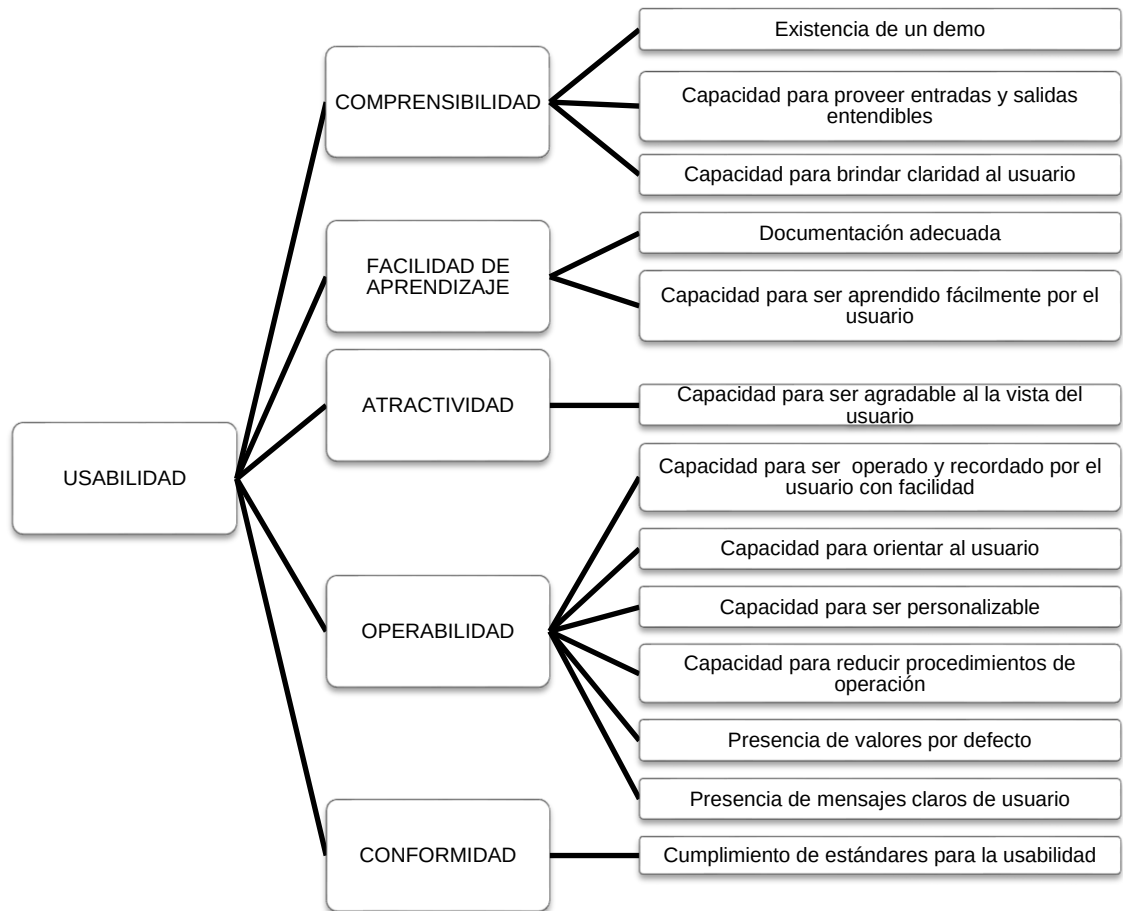


Tabla 5: Usabilidad

USABILIDAD		
SUB-CARACTERÍSTICA	ATRIBUTO	
Comprensibilidad: Capacidad de un producto de software para permitirle al usuario conocer cómo puede utilizar el sistema, qué tareas puede llevar a cabo, cuáles son sus condiciones de uso y cuáles de ellas son necesarias para que pueda desenvolverse de una manera adecuada cuando se encuentre navegando en el mismo.	Existencia de un demo	Evalúa la capacidad del producto de software para brindar una presentación preliminar al usuario.
	Capacidad para proveer entradas y salidas entendibles	Evalúa la capacidad del software para brindar claridad al usuario en cuanto a la especificación de los valores, archivos o caracteres a ingresar y a los resultados que se espera a partir de estos al realizar alguna operación.
	Capacidad para brindar claridad al usuario	Evalúa la capacidad del software para brindar herramientas al usuario que eviten confusión con respecto a los mensajes que se muestran y a las funcionalidades y operaciones que puede llevar a cabo en determinado instante de tiempo.
Facilidad de aprendizaje: Capacidad del software para permitir al usuario aprender su aplicación y funcionamiento, refiriéndose al tiempo y esfuerzo requerido y para alcanzar un grado de aprehensión de las utilidades, funcionalidades y diseño del mismo.	Capacidad para ser aprendido fácilmente por el usuario	Evalúa la facilidad que brinda el producto de software al usuario, para captar y comprender su funcionamiento en el menor tiempo posible.
	Documentación adecuada	Evalúa la disponibilidad de la documentación necesaria para la operación de un producto de software, en la cual se debe expresar la forma en la que el usuario y el software deben interactuar.
Atractividad: Capacidad del producto software para ser llamativo y agradable al usuario	Capacidad para ser agradable a la vista del usuario	Evalúa la capacidad del producto de software para hacer que, cuando el usuario interactúe con él, se encuentre a gusto y atraído para utilizar el sistema.
Operabilidad: Capacidad del software para permitirle al usuario su operación y control	Capacidad para ser operado y recordado por el usuario con facilidad:	Evalúa la capacidad del producto de software para proveer herramientas al usuario que le permitan recordar sus funcionalidades con facilidad.
	Capacidad para orientar al usuario	Evalúa la capacidad del software para guiar interactivamente a todo tipo de usuario en la operación del sistema.
	Capacidad para ser personalizable	Capacidad del software para permitir la personalización de las operaciones que un usuario realiza en el sistema.
	Capacidad para reducir procedimientos de operación	Evalúa la capacidad del producto de software para permitir la optimización de algunas operaciones.
	Presencia de valores por defecto	Evalúa la capacidad del producto de software para brindar al usuario la posibilidad de realizar operaciones de una manera fácil y rápida.

Continuación tabla 5.

	Presencia de mensajes claros de usuario	Evalúa la capacidad del producto de software para brindar claridad en los mensajes ofrecidos al usuario.
Conformidad en la usabilidad: Capacidad del producto de software para adherirse a los estándares, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.	Cumplimiento de estándares para la usabilidad	Evalúa la capacidad del software para cumplir con los estándares relacionados con la usabilidad establecidos previamente.

b. Mantenibilidad

Capacidad del producto de software para ser corregido, comprendido a nivel de código y mejorado, de tal manera que las modificaciones sean acordes con los cambios del entorno, los requisitos y las especificaciones funcionales del mismo. (Ver cuadro 5) (Ver tabla 6)

Cuadro 5: Característica Mantenibilidad

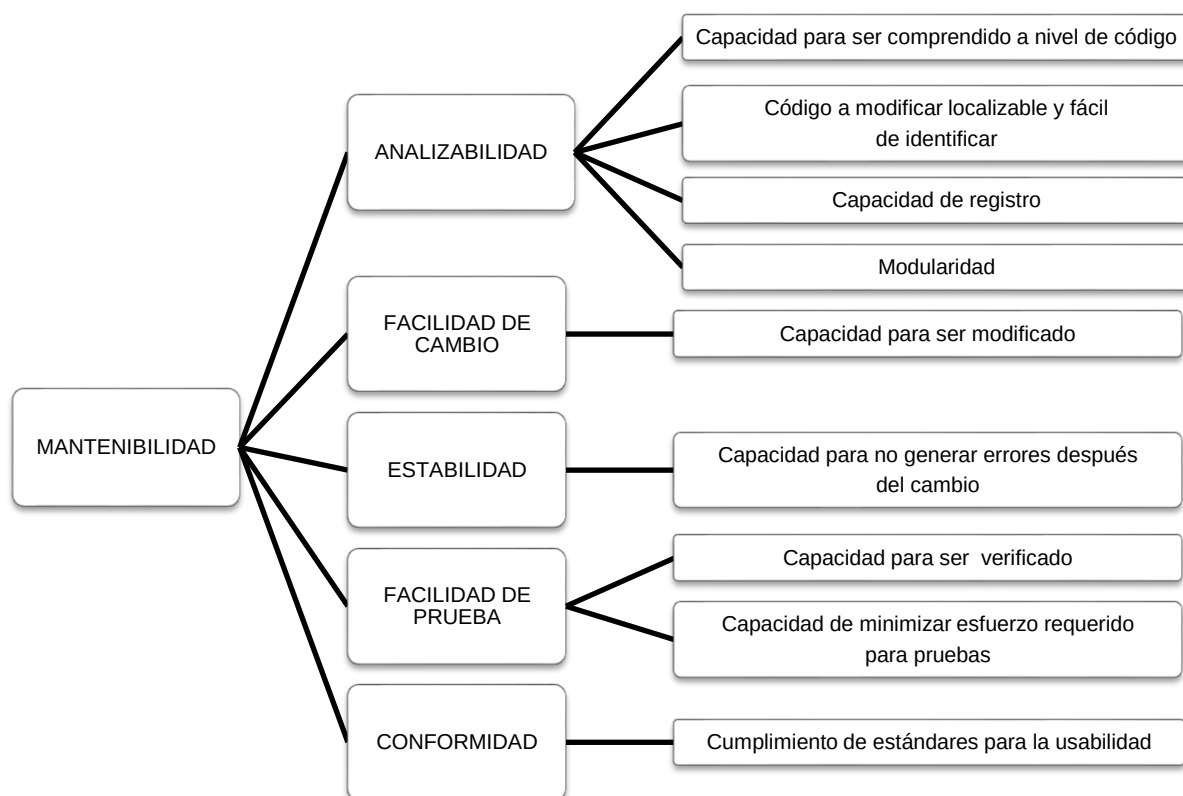


Tabla 6: Mantenibilidad

MANTENIBILIDAD		
SUB-CARACTERÍSTICA	ATRIBUTO	
Analizabilidad: Capacidad del producto de software para ser diagnosticado por deficiencias o causas de fallas en el software y/o facilitar la localización de las partes que se deben modificar.	Capacidad para ser comprendido a nivel de código	Evalúa la capacidad del producto de software para brindar al personal del mantenimiento la posibilidad de comprender fácilmente la estructura, diseño, variables y condiciones que se utilizan internamente.
	Código a modificar localizable y fácil de identificar	Evalúa la capacidad del software para brindar al personal de mantenimiento, facilidad para localizar e identificar en él las porciones de código a modificar.
	Capacidad de registro	Evalúa la capacidad del producto de software para guardar registros (logs) correspondientes al estado en el que se encuentra el sistema.
	Modularidad	Evalúa la estructura del código implementado en el producto.
Facilidad de Cambio: Capacidad del producto de software para permitir la implementación de cambios, en la codificación, diseño y/o documentación de la aplicación en un instante de tiempo.	Capacidad para ser modificado fácilmente	Evalúa la capacidad del producto de software para brindarle al personal de mantenimiento, mecanismos, guías y ayudas que aporten ideas sobre el comportamiento del sistema.
Estabilidad: Capacidad del producto de software para evitar que se presenten efectos no esperados debido a modificaciones en el aplicativo.	Capacidad para no generar errores después del cambio	Evalúa la capacidad del software para asegurar que los errores generados después de realizar cambios en el sistema sean mínimos.
Facilidad de prueba: Capacidad del producto de software para permitir validar las modificaciones realizadas en un momento determinado de tiempo.	Capacidad para ser verificado	Evalúa la capacidad del software para permitir que la verificación de las modificaciones realizadas se realice fácil y rápidamente.
	Capacidad de minimizar el esfuerzo requerido para pruebas	Evalúa la capacidad del software para brindar, a las personas encargadas de verificar las modificaciones, la posibilidad de que el esfuerzo y tiempo requeridos para realizar las pruebas sea mínimo.
Conformidad en la mantenibilidad: Capacidad del producto de software para adherirse a los estándares relacionados con la mantenibilidad.	Cumplimiento de estándares para la mantenibilidad	Evalúa la capacidad del software de cumplir los estándares relacionados con la Mantenibilidad, de una manera adecuada.

c. Eficiencia

Capacidad del producto de software para proporcionar un desempeño apropiado, en relación con la cantidad de recurso utilizado, bajo condiciones establecidas en determinado momento del tiempo. (Ver cuadro 6) (Ver tabla 7)

Cuadro 6: Característica Eficiencia

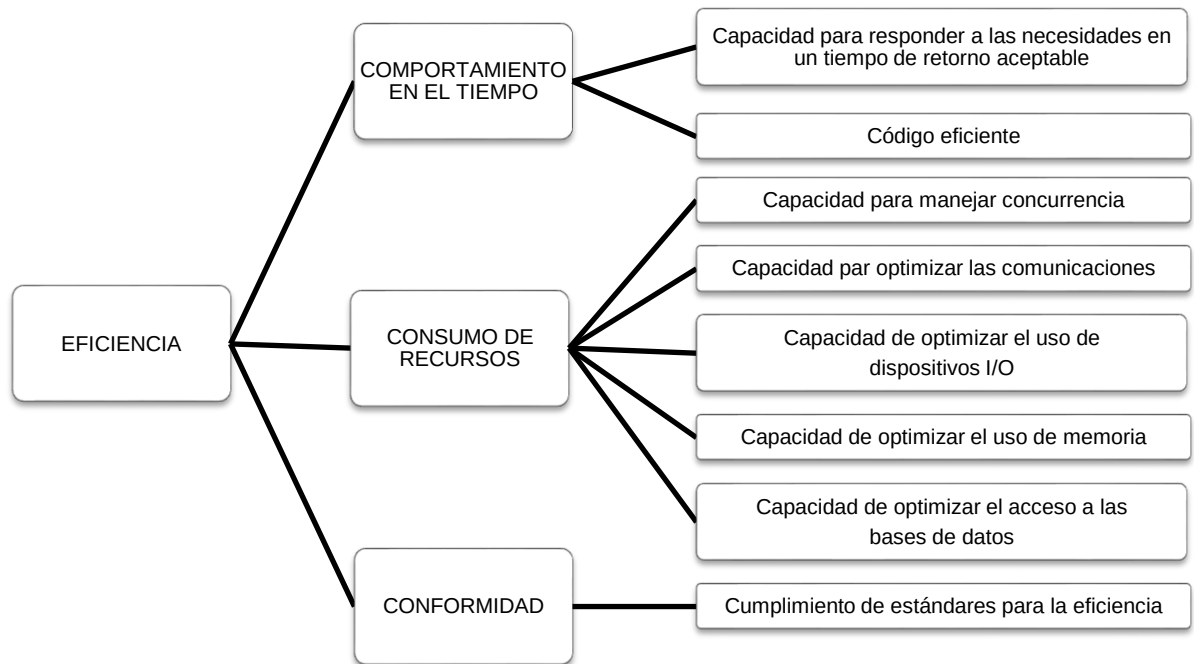


Tabla 7: Eficiencia

EFICIENCIA		
SUB-CARACTERÍSTICA	ATRIBUTO	
Comportamiento en el tiempo: Capacidad del producto de software para proporcionar adecuados tiempos de respuesta y procesamiento, y buenas tasas de eficiencia en el desempeño de su función, bajo condiciones establecidas.	Código eficiente	Evalúa la capacidad del producto de software para adherirse a los principios establecidos en el análisis de algoritmos.
	Capacidad para responder a las necesidades en un tiempo de retorno aceptable:	Evalúa la capacidad de un producto de software para responder ágilmente a las operaciones realizadas en el sistema.

SUB-CARACTERÍSTICA	ATRIBUTO	
Consumo de recursos: Capacidad del producto de software para utilizar una apropiada cantidad y tipos de recursos cuando el software desempeña su función bajo condiciones establecidas.	Capacidad para optimizar las comunicaciones:	Evalúa la capacidad del producto de software para aprovechar el esquema de comunicaciones establecido.
	Capacidad de optimizar el uso de memoria	Evalúa la capacidad del producto de software para optimizar el uso de la memoria disponible.
	Capacidad de optimizar el uso de dispositivos I/O	Evalúa la capacidad del producto de software para utilizar adecuadamente los mecanismos de entrada y salida disponibles.
	Capacidad de optimizar el acceso a las bases de datos	Evalúa la capacidad del producto de software para acceder a las bases de datos, para determinar si lo está haciendo de la forma más eficiente posible.
Conformidad en la eficiencia: Evalúa la capacidad del producto de software para cumplir con los estándares establecidos en la organización, relacionados con la eficiencia.	Cumplimiento de estándares para la eficiencia	Verifica si existen estándares definidos y rango de aceptación que permitan reconocer cuándo el producto cuenta con tiempos de respuesta y consumo de recursos aceptables

d. Funcionalidad

Capacidad del producto de software para suministrar un conjunto de funciones que satisfagan las necesidades implícitas o explícitas de los usuarios, al ser utilizado bajo condiciones específicas. (Ver cuadro 7) (Ver tabla 8)

Cuadro 7: Característica Funcionalidad

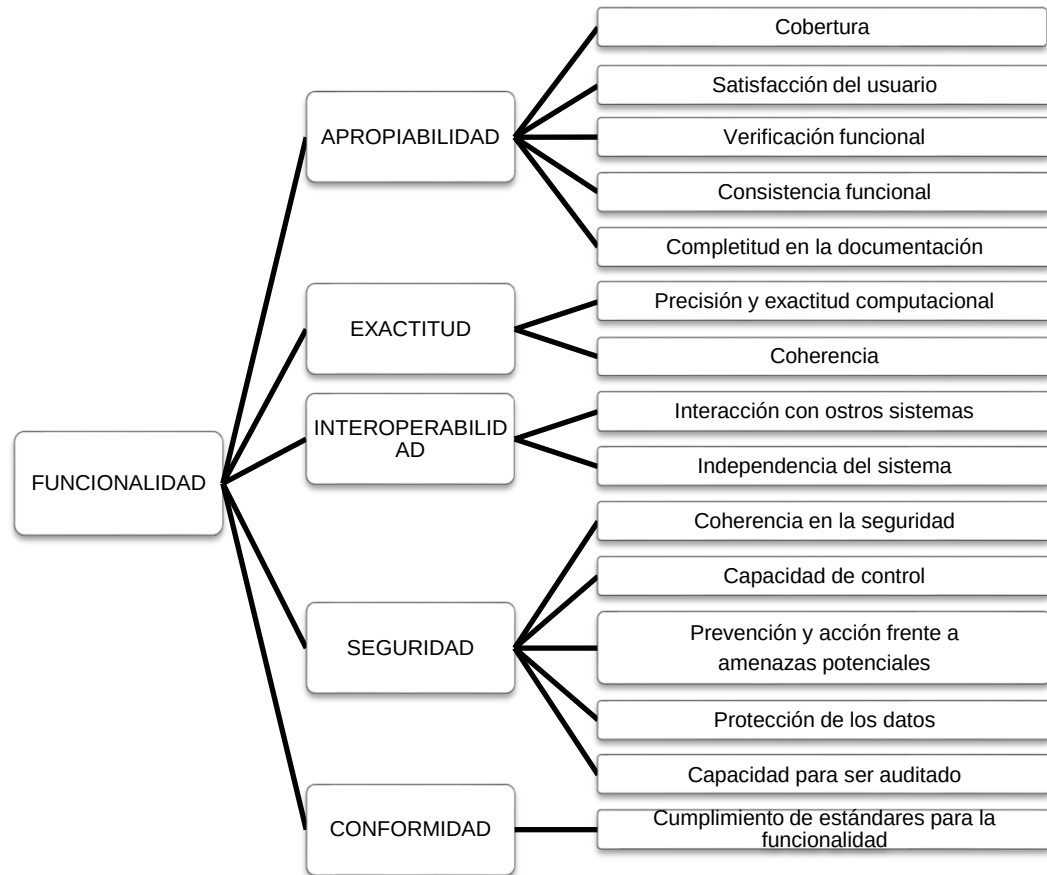


Tabla 8: Funcionalidad

FUNCIONALIDAD		
SUB-CARACTERÍSTICA	ATRIBUTO	
Apropiabilidad: Capacidad del software para suministrar un conjunto apropiado de funciones que cumplan tareas específicas y objetivos del usuario.	Cobertura	Evalúa la completitud del producto en su funcionalidad y en su documentación.
	Satisfacción del usuario	Evalúa el nivel de satisfacción del usuario frente al producto final entregado.
	Verificación funcional	Permite conocer si las funcionalidades implementadas han sido verificadas adecuadamente, antes de hacer la entrega formal del producto al usuario final.

SUB-CARACTERÍSTICA	ATRIBUTO	
	Consistencia funcional	Evalúa la integridad y la coherencia existente entre las diversas funcionalidades implementadas en el producto.
	Compleitud en la documentación	Evalúa si existe una documentación completa como parte del producto final.
Exactitud: Capacidad del software para proveer los resultados correctos y los efectos pactados con un adecuado grado de precisión	Precisión y exactitud computacional	Evalúa la aproximación de los resultados al valor verdadero que se supone representa.
	Coherencia	Evalúa la consistencia en los resultados arrojados por el producto, para varias ejecuciones diferentes bajo las mismas condiciones iniciales.
Interoperabilidad: Capacidad del software para interactuar con uno o más sistemas específicos	Interacción con otros sistemas	Evalúa la existencia de interacción del sistema con otros aplicativos en el mismo ambiente y además, que dicha interacción se realice correctamente.
	Independencia del sistema	Evalúa el grado de dependencia del producto de software con relación a otros sistemas y a recursos externos.
Seguridad: Capacidad del software para proteger los datos y la información, con el fin de que personas no autorizadas nunca puedan ingresar al sistema, leer o modificar los datos, y que a las personas que sí estén autorizadas no se les niegue el acceso al sistema ni a la información que necesiten.	Coherencia en la seguridad	Evalúa si el producto de software es consistente con la seguridad de la organización, durante todo el ciclo de vida.
	Capacidad de control	Evalúa la capacidad del producto para controlar el acceso a la información, en equilibrio con los niveles de riesgo existentes.
	Prevención y acción frente a amenazas potenciales	Evalúa la capacidad del sistema para prevenir, detectar y reaccionar ante las amenazas existentes, ya sea por posibles ataques o por accidentes.
	Protección de los datos	Evalúa si el sistema es capaz de manejar los datos en forma cifrada, utilizando un método seguro.
	Capacidad para ser auditado	Indica si el sistema tiene la capacidad, para que se le realice un seguimiento detallado con el fin de detectar si su operación y administración son correctas.
Conformidad en la funcionalidad: Capacidad del software para ajustarse a los estándares, convenciones y regulaciones relacionadas con la funcionalidad	Cumplimiento de estándares para la funcionalidad	Verifica que exista una buena documentación desde la definición de requerimientos, de tal forma que se pueda realizar un análisis posterior que permita verificar cuales de los requerimientos pactados se cumplieron y cuáles no.

e. Confiabilidad

Capacidad del producto de software para conservar su nivel de desempeño bajo condiciones específicas, durante un determinado período de tiempo.

El producto será confiable siempre y cuando el usuario pueda tener siempre la certeza de que su información estará protegida ante cualquier tipo de falla, garantizando que los datos estarán disponibles en el momento en el que se requieran. (Ver cuadro 8) (Ver tabla 9)

Cuadro 8: Característica Confiabilidad

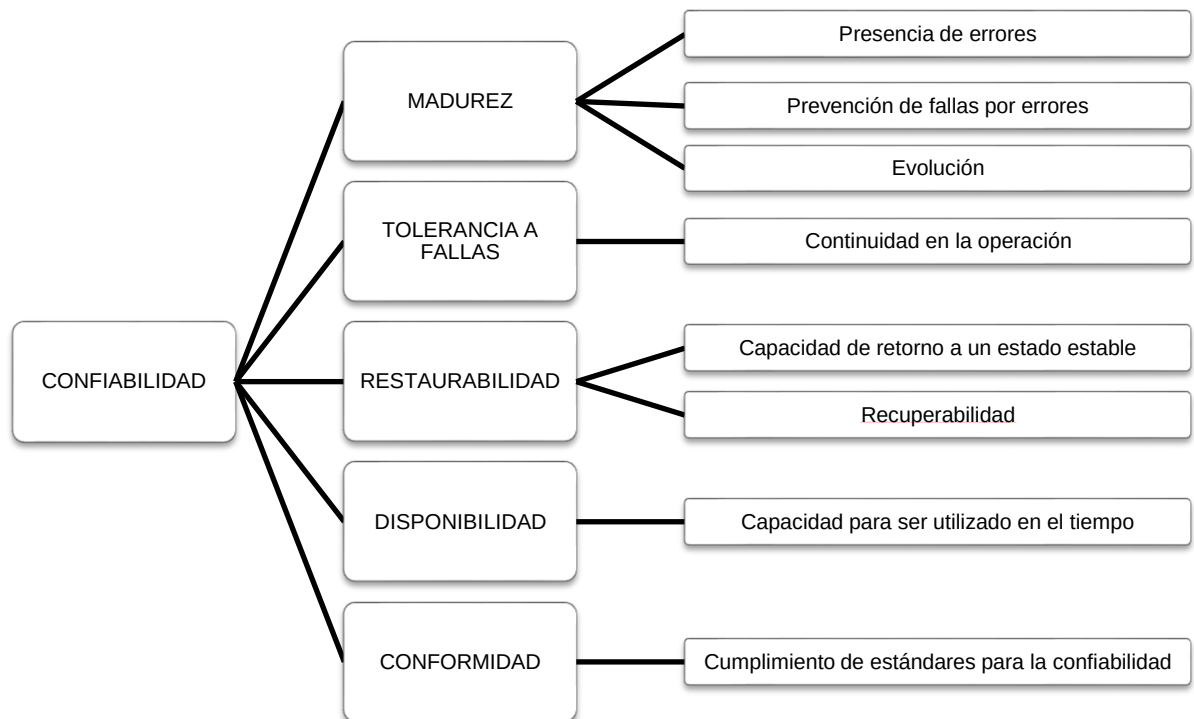


Tabla 9: Confiabilidad

CONFIABILIDAD		
SUB-CARACTERÍSTICA	ATRIBUTO	
Madurez: Capacidad del software para evitar fallas ocasionadas por defectos en el programa	Presencia de errores	Evalúa la existencia de errores en el producto.
	Prevención de fallas por errores	Evalúa la capacidad de prevenir fallas ocasionadas por errores internos del programa.

SUB-CARACTERÍSTICA	ATRIBUTO	
	Evolución	Evalúa la transformación gradual que ha presentado el producto.
Tolerancia a fallas: Capacidad del software para conservar un determinado nivel de desempeño frente a defectos en el software	Continuidad en la operación	Evalúa la capacidad del aplicativo para seguir ejecutando sus funcionalidades, aún en casos de fallas internas o externas.
Restaurabilidad: Capacidad del software para restablecer un determinado nivel de desempeño en caso de una falla y recuperar los datos directamente afectados, de una manera consistente	Capacidad de retorno a un estado estable	Evalúa la capacidad del producto para regresar a un estado estable después de la ocurrencia de una falla, interna o externa.
	Recuperabilidad	Evalúa la capacidad del software para recuperarse de los daños producidos durante un fallo: devolver las acciones que corrompieron la funcionalidad y recuperar los datos directamente afectados
Disponibilidad: Capacidad del software para encontrarse en un estado apto para desempeñar una función requerida en un momento de tiempo determinado, bajo ciertas condiciones de uso	Capacidad para ser utilizado en el tiempo	Evalúa el porcentaje de tiempo en el que el sistema está realmente apto para ser utilizado, en relación con el tiempo total en el que debe estar listo para ser utilizado.
Conformidad en la confiabilidad: Capacidad del software para ajustarse a los estándares, convenciones y regulaciones relacionadas con el manejo de errores de software.	Cumplimiento de estándares para la confiabilidad	Exige que exista una buena documentación relacionada, tanto con el estudio de errores, como con las funcionalidades y planes de acción que permitirán llevar un buen de los mismos.

f. Portabilidad

Capacidad del producto de software para ser transferido de un ambiente a otro. Se evalúa en tres ámbitos diferentes: (Ver cuadro 9) (Ver tabla 10)

- Ambiente organizacional
- Ambiente hardware
- Ambiente software

Cuadro 9: Característica portabilidad

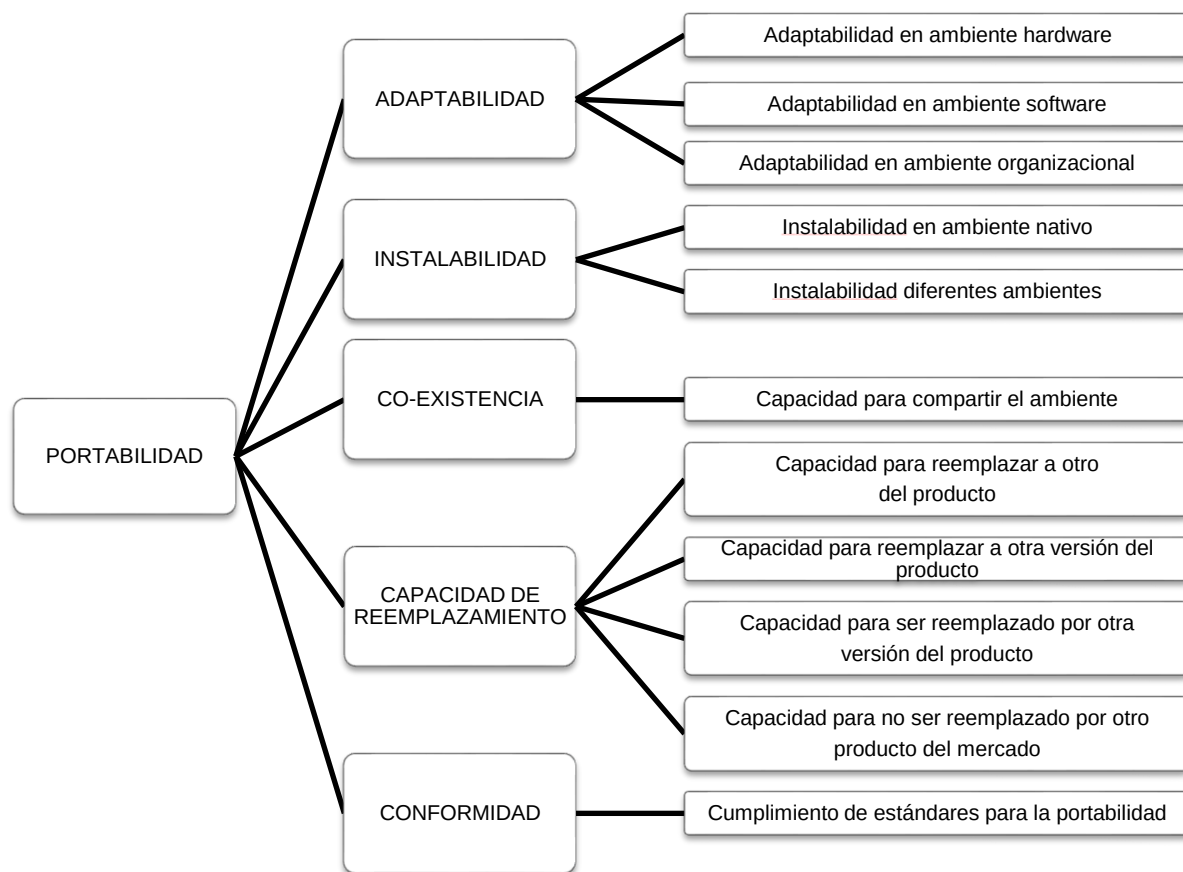


Tabla 10: Portabilidad

PORTABILIDAD		
SUB CARACTERÍSTICA	ATRIBUTO	
Adaptabilidad: Capacidad del software para acoplarse a diferentes ambientes, sin necesidad de aplicar acciones adicionales a las consideradas inicialmente en el diseño del producto	Adaptabilidad en ambiente hardware	Evalúa la capacidad del producto para adaptarse a diferentes ambientes tipo hardware.
	Adaptabilidad en ambiente software	Evalúa la capacidad del producto para adaptarse a diferentes ambientes tipo software.
	Adaptabilidad en ambiente organizacional	Evalúa la capacidad del producto para adaptarse a diferentes ambientes dentro de la organización.

SUB CARACTERÍSTICA	ATRIBUTO	
Instalabilidad: Capacidad del software para ser instalado en un ambiente determinado	Instalabilidad en ambiente nativo	Evalúa la capacidad del producto para ser instalado exitosamente en el ambiente propio.
	Instalabilidad en diferentes ambientes	Evalúa la capacidad del producto para ser instalado exitosamente en diferentes ambientes (hardware y software).
Co-Existencia: Capacidad del software para convivir correctamente con otros sistemas en un ambiente común, compartiendo recursos	Capacidad para compartir el ambiente	Evalúa la capacidad del producto para funcionar exitosamente en un ambiente donde existen más aplicativos en ejecución simultáneamente.
Capacidad de reemplazamiento: Capacidad del software para ser utilizado en lugar de otro software que ejecute la misma funcionalidad, en el mismo ambiente	Capacidad para reemplazar a otro producto	Evalúa la capacidad del producto de software para ser utilizado en lugar de otro producto similar en el mercado.
	Capacidad para reemplazar a otra versión del producto	Evalúa la porción del producto que ha sido reutilizada a partir de otras versiones del mismo.
	Capacidad para ser reemplazado por otra versión del producto	Evalúa la capacidad del producto de software para ser reemplazado por una nueva versión del mismo.
	Capacidad para no ser reemplazado por otro producto del mercado	Evalúa la capacidad del producto de software para no ser reemplazado por otro producto del mercado.
Conformidad en la portabilidad: Capacidad del producto de software para ajustarse a los estándares, convenciones y regulaciones relacionadas con la portabilidad	Cumplimiento de estándares para la portabilidad	Exige que exista una buena documentación relacionada, tanto con el comportamiento del sistema en los diferentes ambientes como con la forma de transferir el aplicativo.

La documentación de la norma ISO/IEC, utilizada en esta investigación es la base del proceso de calificación del software SIGEODEP, en ella se encuentran definidas las características, sub – características y atributos que se deben evaluar en un producto software; la identificación de cada uno de estos aspectos permiten determinar cuáles son aplicables y cuales no al software SIGEODEP.

2. SISTEMA DE INFORMACIÓN GEORREFERENCIADO DEL OBSERVATORIO DEL DELITO DE PASTO

2.1 INTRODUCCIÓN

El grupo de investigación GRIAS (Grupo de Investigación Aplicada en Sistemas) del Departamento de Sistemas de la Universidad de Nariño en el marco del proyecto Detección de Patrones Delictivos con Técnicas de Minería de Datos en el Observatorio del Delito del Municipio de Pasto, han desarrollado el software SIGEODEP (Sistema Geo Referenciado del Observatorio del Delito de Pasto).

SIGEODEP soporta el proceso de recolección, integración y procesamiento inicial de los datos recolectados por el Observatorio del Delito referente a las lesiones de causa externa fatal y no fatal que son recopiladas por instituciones de salud, justicia y prevención de la ciudad de Pasto. El sistema gestiona 6 formularios que capturan información sobre eventos de Homicidios, Suicidios, Muertes en Accidentes de Tránsito y Muertes accidentales, así como los eventos de Violencia Intrafamiliar, Lesiones no Intencionales. El sistema permite procesar dicha información para la generación de tablas cruzadas y mapas temáticos que soportan el análisis de datos por parte de los funcionarios del Observatorio.

2.2 PLATAFORMA

SIGEODEP es una aplicación desarrollada en java bajo la plataforma NetBeans 7.2 y la tecnología JSF (JavaServer Faces) y utiliza una base de datos PostgreSQL.

2.3 MARCAS REGISTRADAS DE TECNOLOGIA SOFTWARE

NetBeans: es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma

y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

Ambos productos son de código abierto y gratuito para uso tanto comercial como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL) v1.0 and the GNU General Public License (GPL) v2.⁴

JSF (JavaServer Faces): La tecnología JavaServer Faces incluye:

Un conjunto de APIs para representar componentes de interfaz de usuario y la gestión de su estado, la gestión de eventos y la validación de entrada, la definición de la página de navegación, y el apoyo a la internacionalización y accesibilidad.

Un JavaServer Pages (JSP) de la biblioteca de etiquetas personalizadas para expresar una interfaz JavaServer Faces dentro de una página JSP.

Diseñado para ser flexible, la tecnología JavaServer Faces aprovecha los conceptos de interfaz de usuario y la web, es estándar sin limitar a los desarrolladores a un idioma determinado, protocolo o dispositivo cliente. Las clases de componentes de interfaz de usuario que se incluyen con la tecnología JavaServer Faces encapsulan la funcionalidad de los componentes, no la presentación específica del cliente, permitiendo así JavaServer Faces componentes de interfaz de usuario que se presta a diversos dispositivos cliente. Mediante la combinación de la funcionalidad de los componentes de interfaz de usuario con procesadores personalizados, que definen los atributos de renderizado para un componente específico de interfaz de usuario, los desarrolladores pueden construir etiquetas personalizadas a un dispositivo cliente en particular. Como ventaja, la tecnología JavaServer Faces proporciona un procesador personalizado y una biblioteca de etiquetas JSP personalizada para la prestación a un cliente de HTML, lo que permite a los desarrolladores de aplicaciones Java Platform, Enterprise Edition (Java EE) utilizar la tecnología JavaServer Faces en sus aplicaciones.

La facilidad de uso es el objetivo principal, la arquitectura JavaServer Faces define claramente una separación entre la lógica de aplicación y la presentación, al tiempo que facilita la conexión de la capa de presentación para el código de la aplicación. Este diseño permite que cada miembro de un equipo de desarrollo de aplicaciones web pueda centrarse en su pieza del proceso de desarrollo, y también proporciona un modelo de programación sencillo para unir las piezas.

⁴ Disponible en internet: https://netbeans.org/index_es.html [citado agosto de 2014]

Desarrollado a través del Java Community Process bajo JSR - 314, la tecnología JavaServer Faces establece el estándar para la construcción de interfaces de usuario del lado del servidor. Con las aportaciones del grupo de expertos, el JavaServer Faces APIs están siendo diseñado para que puedan ser aprovechados por las herramientas que harán que el desarrollo de aplicaciones web, sea más fácil. Varios vendedores de herramientas respetados eran miembros del grupo de expertos JSR-314, que desarrolló la especificación JavaServer Faces 1.0. Estos proveedores se han comprometido a apoyar la tecnología JavaServer Faces en sus herramientas, fomentando así la adopción de la tecnología JavaServer Faces estándar.⁵

PostgreSQL: es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos 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. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.⁶

Algunas características son:

- Es una base de datos 100% ACID
- Integridad referencial
- Tablespace
- Nested transactions (savepoints)
- Replicación asincrónica/sincrónica / Streaming replication - Hot Standby
- Two-phase commit
- PITR - point in time recovery
- Copias de seguridad en caliente (Online/hot backups)
- Unicode
- Juegos de caracteres internacionales
- Regionalización por columna
- Multi-Version Concurrency Control (MVCC)
- Múltiples métodos de autenticación
- Acceso encriptado via SSL
- Actualización in-situ integrada (pg_upgrade)
- SE-postgres
- Completa documentación
- Licencia BSD

⁵ Disponible en internet: <http://www.oracle.com/technetwork/java/javaee/jaserverfaces-139869.html> [citado agosto de 2014]

⁶ Disponible en internet: http://www.postgresql.org/es/sobre_postgresql#caracteristicas [citado agosto de 2014]

- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.
- Funciones/procedimientos almacenados (stored procedures) en numerosos lenguajes de programación, entre otros PL/pgSQL (similar al PL/SQL de Oracle), PL/Perl, PL/Python y PL/Tcl
- Bloques anónimos de código de procedimientos (sentencias DO)
- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido, ...)
- APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt y muchos otros.
- SQL92, SQL99, SQL2003, SQL2008
- Restricciones de unicidad postergables (deferrable constraints)

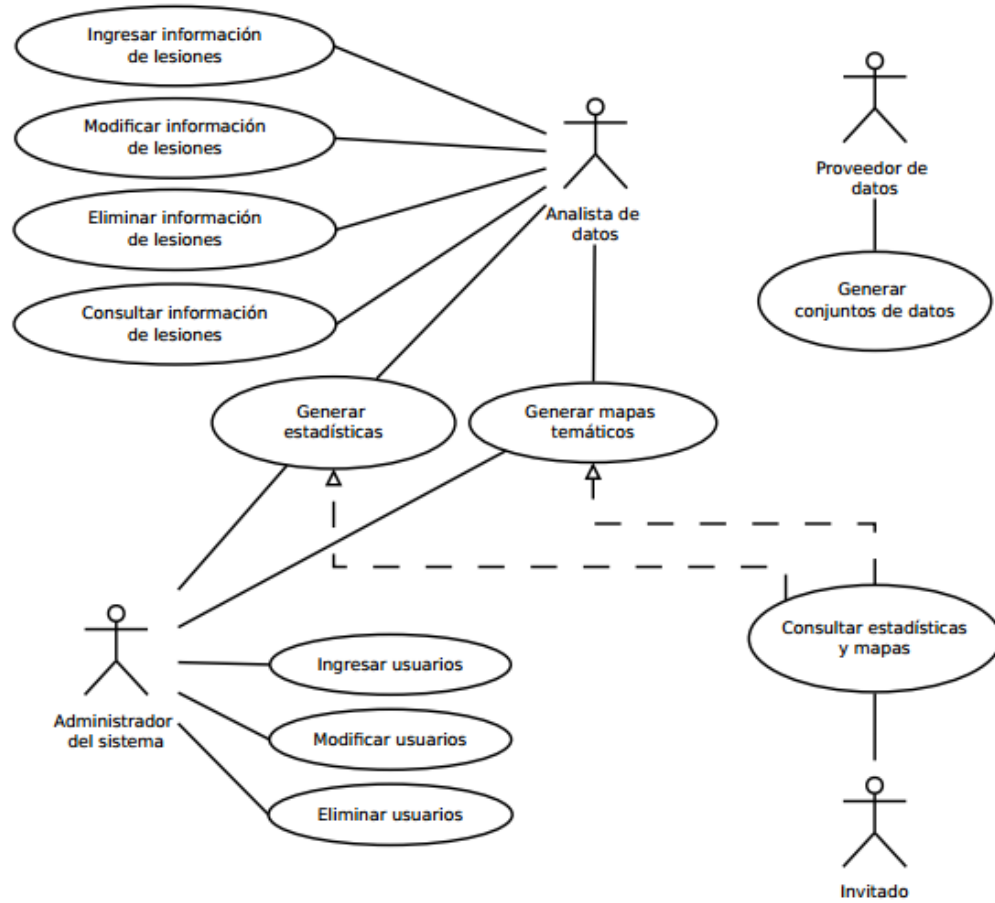
2.4 FUNCIONALIDADES PRINCIPALES

Ilustración 8: Diagrama de Módulos



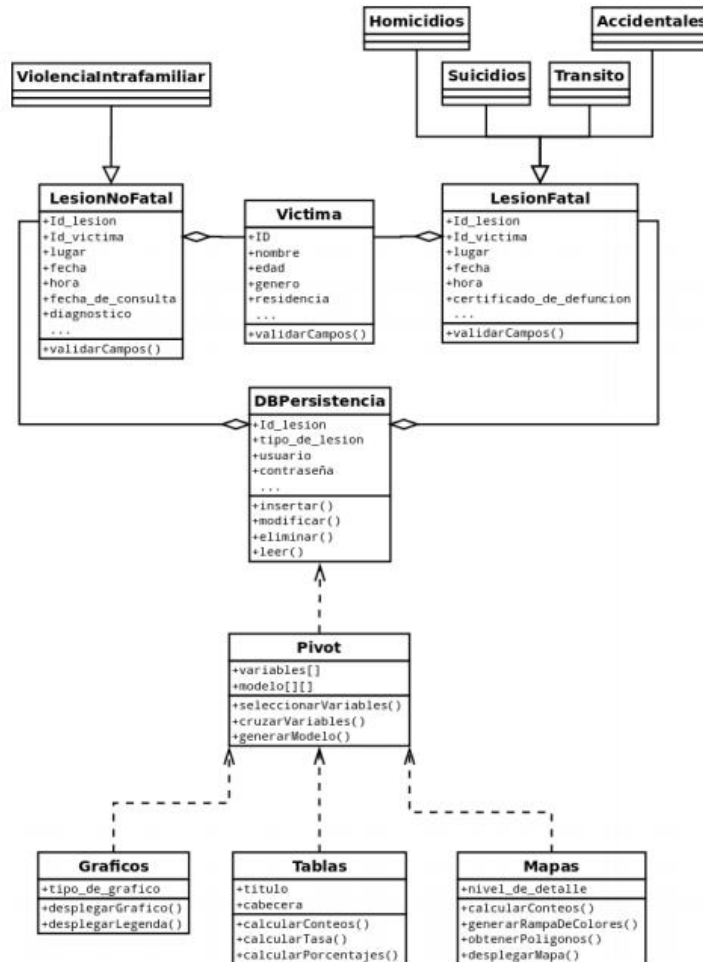
2.5 DIAGRAMA GENERAL DE CASOS DE USO

Ilustración 9: Diagrama general de casos de uso



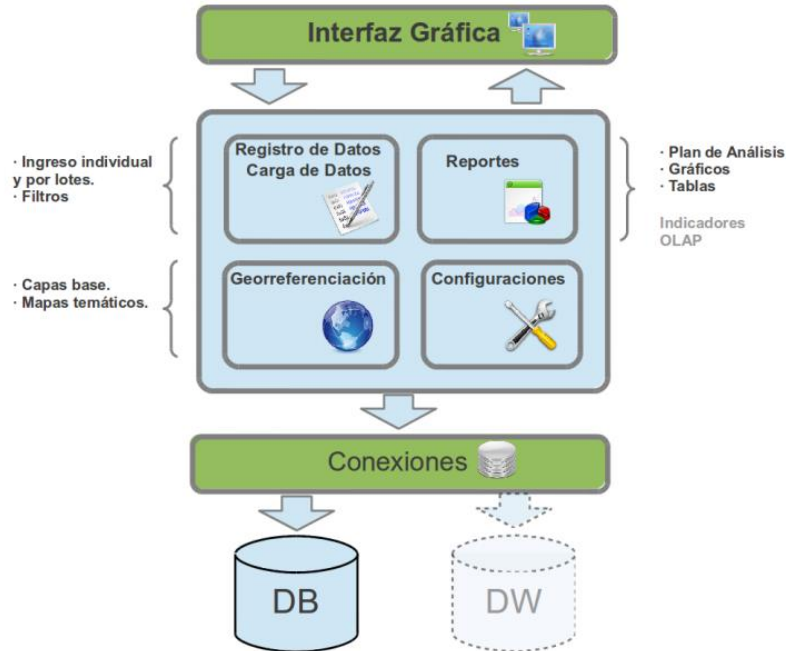
2.6 MODELO ESTRUCTURAL DEL SOFTWARE

Ilustración 10: Modelo estructural del software SIGODEP



2.7 ARQUITECTURA

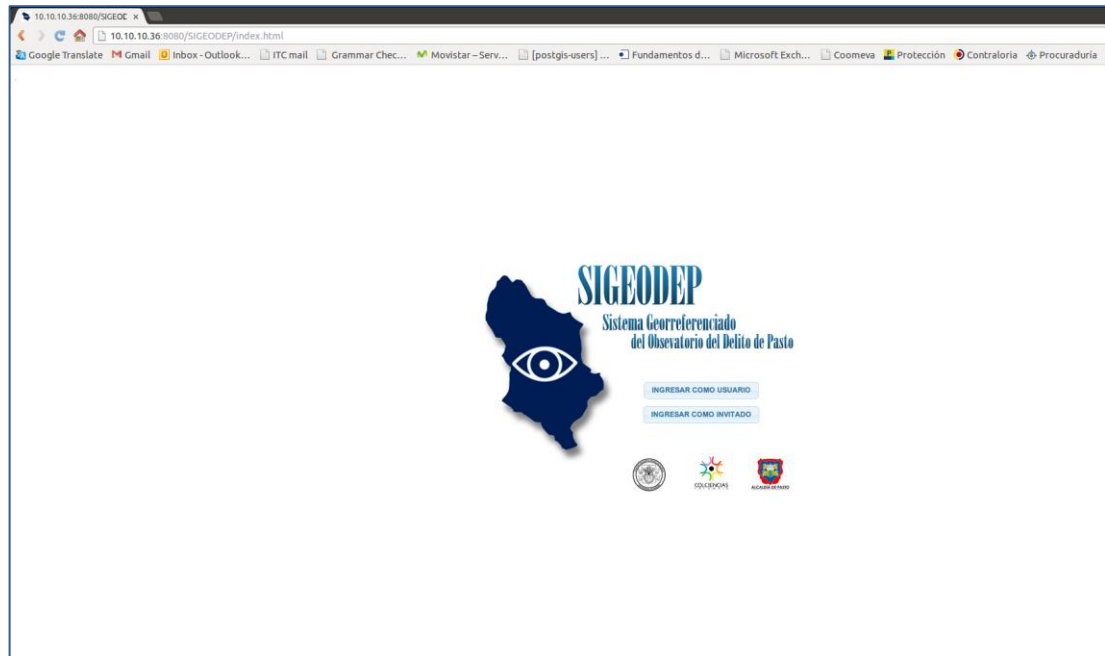
Ilustración 11: Arquitectura del software SIGEODEP



2.8 ÁREA DE TRABAJO SIGEODEP

En términos generales, el sistema provee una interface gráfica de usuario bastante amigable y sencilla de administrar, la primera pantalla es para que el usuario se registre y posteriormente tenga acceso dependiendo de sus permisos, de manera estándar presenta un menú vertical que permite al usuario recorrer las funcionalidades del sistema, un conjunto de pestañas que permiten el acceso a las funcionalidades macro. (Ver ilustración 12)

Ilustración 12: Interfaz principal SIGODEP



2.9 MÓDULO DE CARGA MASIVA

El módulo de carga masiva permite capturar datos desde archivos planos enviados desde distintos proveedores (Medicina legal, Fiscalía, Hospital Departamental, etc.) que usualmente poseen diferentes formatos. Los archivos planos se corresponden a la información solicitada por una de las fichas gestionadas por el Observatorio del Delito. El sistema soporta la extracción, transformación y carga de los datos provenientes de otras fuentes al formato utilizado por el Observatorio para su posterior ingreso en la base de datos. (Ver ilustración 13-14)

Ilustración 13: Diagrama de casos de uso módulo de carga masiva

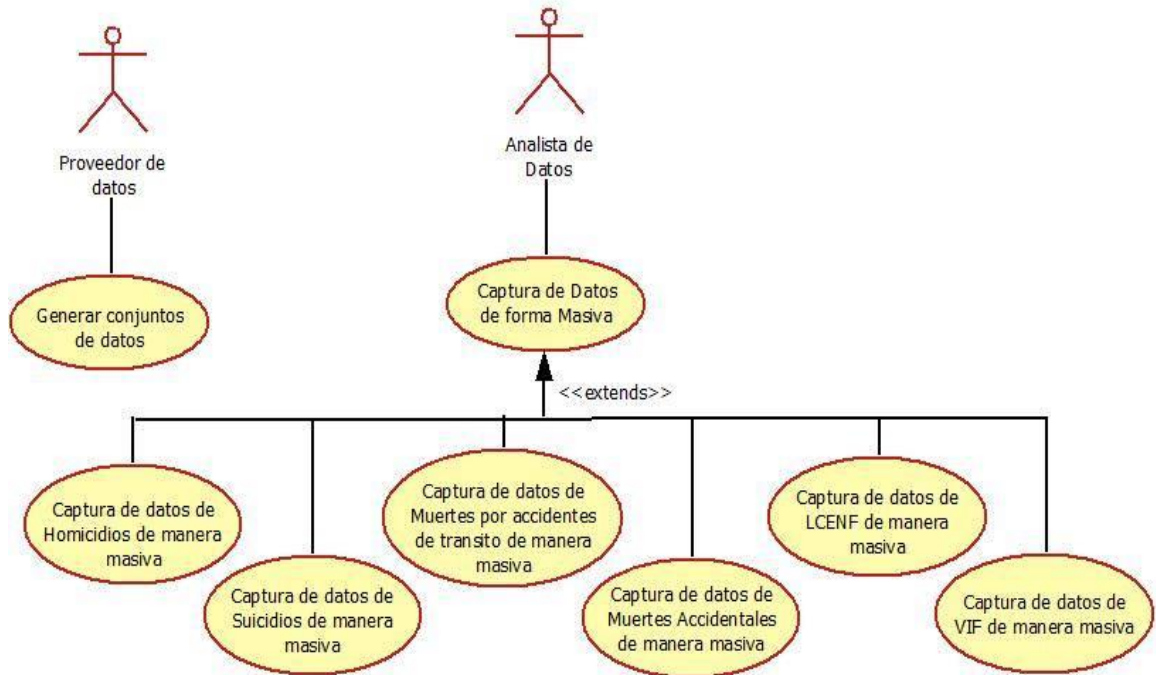


Ilustración 14: Interfaz módulo de carga masiva

2.10 MÓDULO DE CARGA POR FORMULARIO

El módulo de carga por formulario permite ingresar toda la información que se encuentra en las Fichas SCC y los archivos CHK manejados por las diferentes entidades que maneja el Observatorio del Delito. El sistema soporta la extracción, transformación y carga de los datos provenientes de otras fuentes al formato utilizado por el Observatorio para su posterior ingreso en la base de datos. (Ver ilustración 15-16)

Ilustración 15: Diagrama de casos de uso módulo de carga por formulario

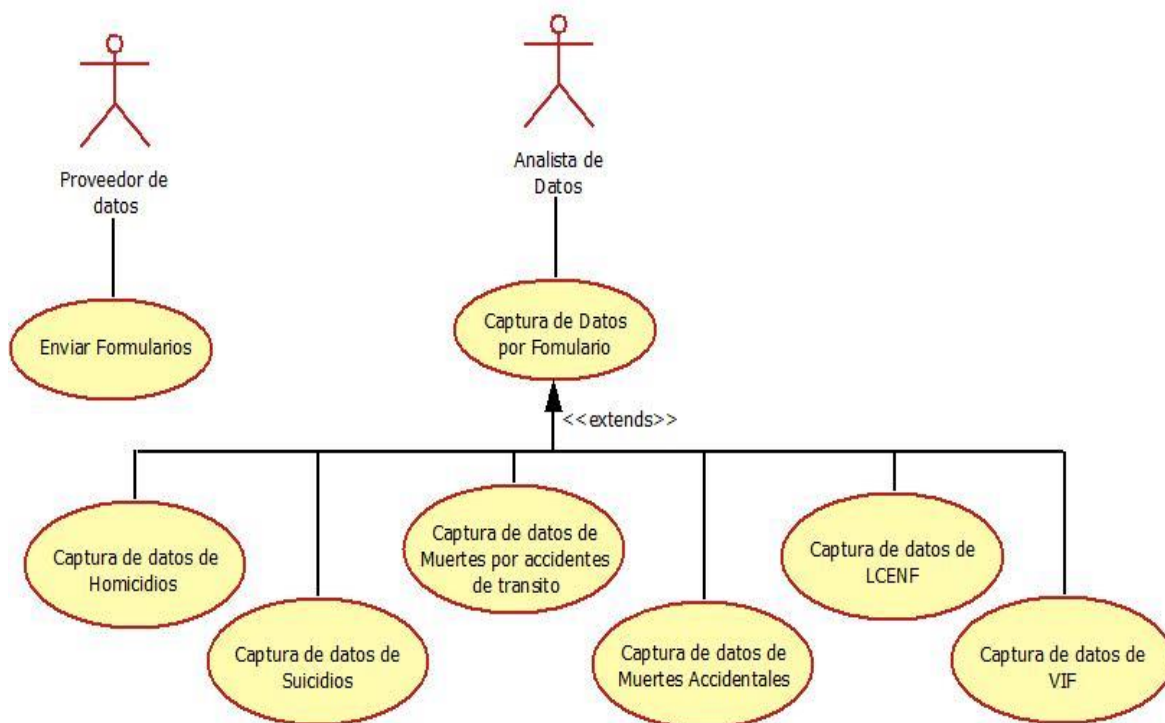


Ilustración 16: Interfaz módulo carga por formulario

2.11 MÓDULO CRUCE DE INDICADORES

El módulo de cruce de indicadores permite realizar consultas en la base de datos, cruzando diferentes indicadores y mostrando la información de manera estadística en el tipo de gráfico que el usuario elija. Los gráficos presentados corresponden a la información solicitada por el usuario y gestionadas por el Observatorio del Delito. (Ver ilustración 17-18)

Ilustración 17: Diagrama de casos de uso módulo cruce de indicadores

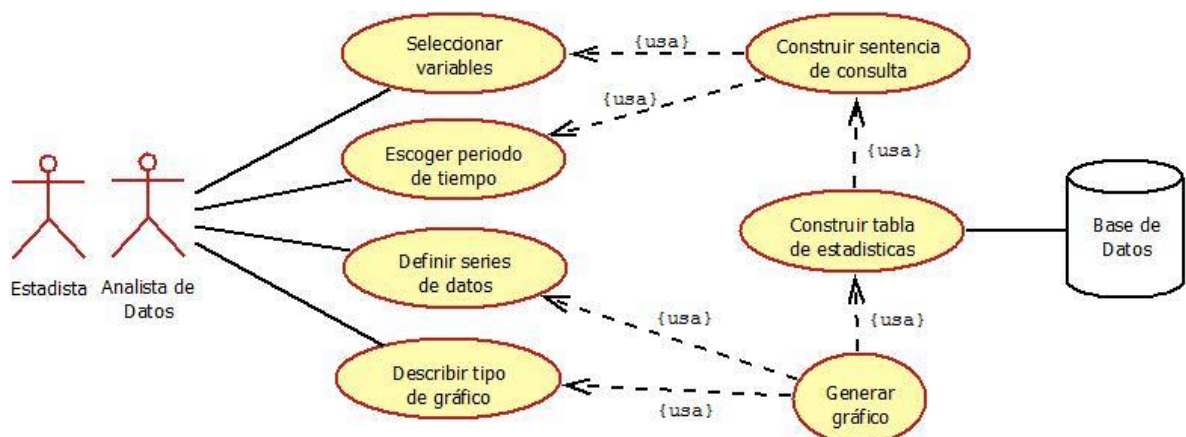
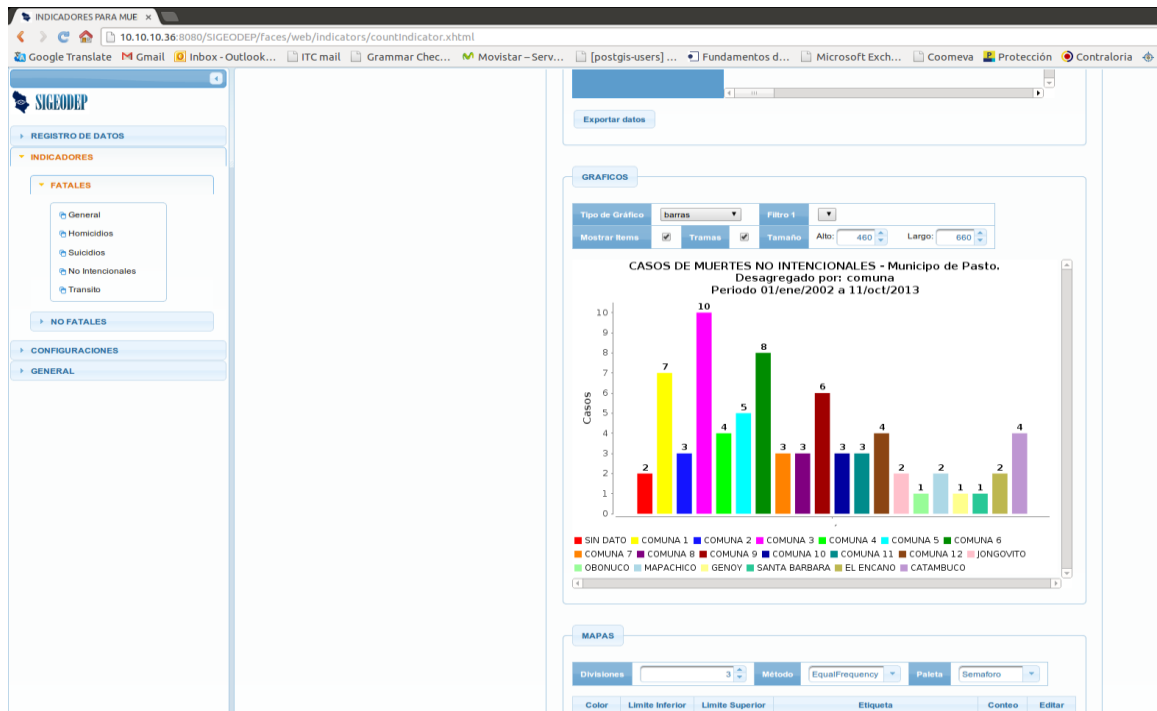


Ilustración 18: Interfaz módulo cruce de indicadores



2.12 MÓDULO MAPAS TEMÁTICOS

El módulo de mapas temáticos permite mostrar la información consultada en forma gráfica utilizando el mapa del municipio de Pasto, en el mapa se muestran las diferentes variables que el usuario escoja clasificando la información con diferentes colores. Los gráficos presentados corresponden a la información solicitada por el usuario gestionadas por el Observatorio del Delito. (Ver ilustración 19-20)

Ilustración 19: Diagrama de casos de uso módulo de mapas temáticos

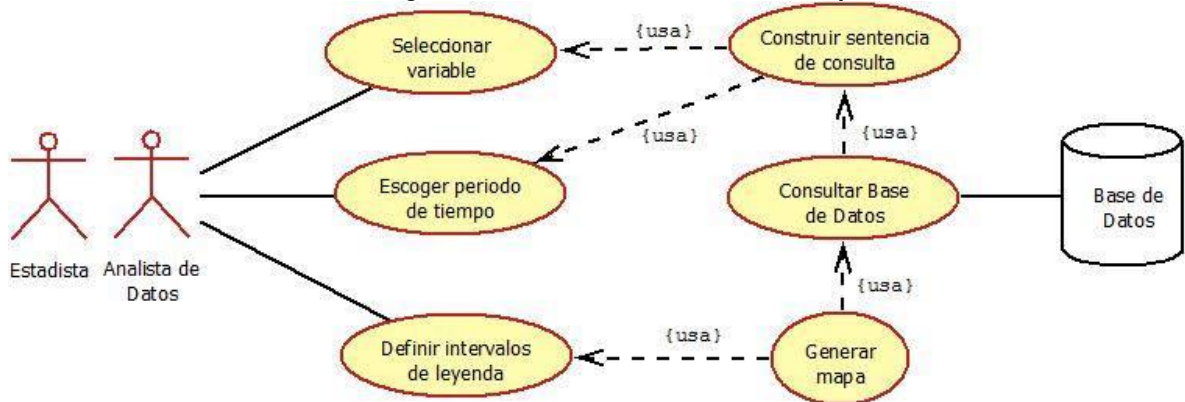
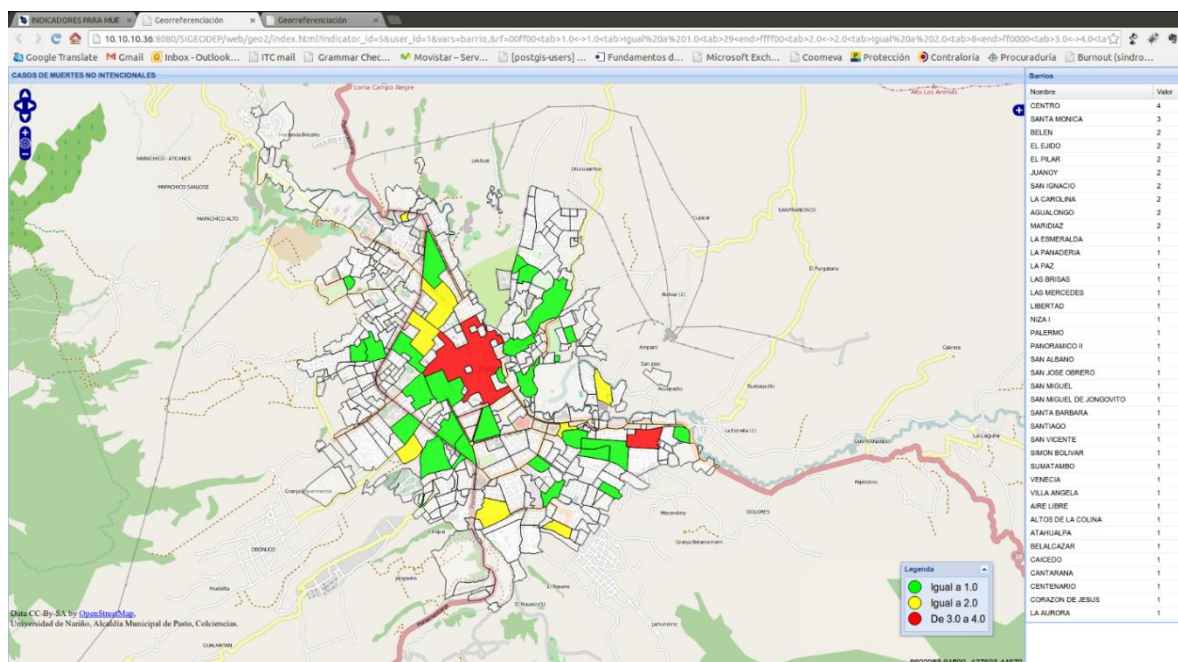


Ilustración 20: Interfaz Módulo de Mapas Temáticos



3. RESULTADOS DEL PROCESO

Introducción: El Observatorio del delito de Pasto, tiene como función principal llevar un seguimiento a los actos delictivos como a los accidentes que ocurren en la ciudad teniendo en cuenta muchas variables para su posterior análisis, este seguimiento se lleva de manera manual y por eso se gasta más tiempo en la recolección de datos que en su análisis.

Para soportar la información que se requiere almacenar en el proceso de prestación del servicio, el Observatorio del Delito cuenta con la aplicación SIGEODEP Sistema geo referenciado del observatorio del delito de Pasto. La disponibilidad y confiabilidad de la información que almacena el sistema permite a los involucrados en el proceso productivo cumplir efectivamente sus funciones.

Con el presente trabajo se pretende iniciar la aplicación de las métricas de software para evaluar la calidad del sistema SIGEODEP.

La metodología a utilizar consiste inicialmente en determinar las características y sub-características de calidad que aplican en la evaluación del producto, las que posteriormente son medidas mediante métricas que se explican en el desarrollo del trabajo. Finalmente, se generan las conclusiones y recomendaciones producto de la evaluación.

Información técnica del producto software:

Descripción general SIGEODEP:

SIGEODEP - El objetivo del sistema es detectar patrones delictivos utilizando técnicas de minería de datos a partir de la información que recibe el Observatorio del Delito del municipio de Pasto.

SIGEODEP - contar con información descriptiva, predictiva y oportuna que les permita diseñar y establecer políticas, estrategias, planes de seguridad y prevenciones efectivas, fundamentales en la lucha contra el delito, por lo tanto el observatorio del delito de Pasto busca:

- Contar con un sistema de información transaccional eficiente, con información confiable y oportuna que sirva de base para los diferentes sistemas de soporte a las decisiones.
- Contar con un sistema de información geo referenciado que identifique los lugares donde ocurrieron los hechos delictivos.

- Contar con un mercado de datos donde se almacenen los datos históricos de una manera resumida y consistente.
- Contar con sistemas inteligentes que descubran patrones de comportamiento de victimarios en relación con sus víctimas, establezcan los perfiles, lugares y ocurrencias conjuntas de delitos.

3.1 METODOLOGÍA DE EVALUACIÓN

Introducción a la evaluación: La presente evaluación tiene como fin determinar la calidad del producto software seleccionado para dar un aporte de sugerencias y mejoras a realizar al producto.

Se seleccionó el software SIGODEP, por ser el software que fue desarrollado por el grupo de investigación GRIAS quienes desconocen el nivel de calidad del sistema, teniendo acceso al código fuente, al código objeto y a la documentación existente.

El modelo seleccionado para realizar la evaluación es el propuesto por la Norma ISO/IEC 9126, que propone las características de usabilidad, mantenibilidad, confiabilidad, portabilidad, funcionalidad y eficiencia.

El trabajo de evaluación se realizó específicamente analizando el funcionamiento del sistema y la documentación.

Acercamiento inicial: El acercamiento inicial se desarrolló por medio de una entrevista aplicada al personal que desarrollo el sistema, este proceso de recolección de información se llevó a cabo para:

- Conocer el proceso que soporta el software.
- Adquirir un mayor conocimiento de las particularidades funcionales y no funcionales del software a evaluar.
- Identificar elementos funcionales que posteriormente permitan definir las características, sub-características y atributos que debe ser evaluados así como las métricas que deben ser aplicadas.

A primera vista y a partir de la información recolectada en la entrevista, se identificaron las siguientes anomalías:

- En el proceso de ingreso de datos se necesita una buena capacitación por parte del equipo que desarrollo el software ya que se necesita información

específica para esta actividad.

- En el momento de realizar la relación de variables se debe conocer muy bien como las variables están definidas en la base de datos porque cada organización maneja sus variables y se deben relacionar con las que están definidas.
- Al momento de registrar un dato y si este está mal borra, toda la información de la caja de texto se borra y solo informa que está mal la información escrita.

Justificación de las características seleccionadas:

Usabilidad: Esta característica se entiende como la capacidad de un producto de software para permitirle al usuario conocer cómo puede utilizar el sistema, qué tareas puede llevar a cabo, cuáles son sus condiciones de uso y cuáles de ellas son necesarias para que pueda desenvolverse de una manera adecuada cuando se encuentra navegando en el mismo.

Específicamente en el planteamiento del problema, el producto evaluado presenta una dificultad a la hora de ingresar los datos ya sea de manera individual o desde un archivo. En el producto evaluado se necesita de una amplia capacitación que permita conocer cómo es la apariencia y funcionalidad del software al momento de su utilización.

La facilidad de uso integra las sub-características: comprensibilidad, facilidad de aprendizaje, operabilidad, atraktividad, conformidad con la usabilidad.

Mantenibilidad: Se entiende por mantenibilidad, la capacidad del producto software para ser modificado y a la facilidad que presenta para ser corregido, comprendido, adaptado y mejorado de acuerdo con los cambios presentados en el ambiente y en los requisitos y especificaciones funcionales del mismo.

Para el grupo de desarrollo es importante conocer si el software desarrollado es mantenible, ya que las por las condiciones particulares del proceso que soporta se somete a cambios frecuentes, porque como ya se ha mencionado anteriormente las diferentes organizaciones cambian sus formatos de ingreso de datos, igualmente es importante establecer la medida de la facilidad con que los errores detectados en el software pueden ser corregidos, esta medida será base para la toma de decisiones respecto de la continuidad del producto software.

La mantenibilidad integra las características: Analizabilidad, Facilidad de Cambio, Estabilidad, Facilidad de Prueba y Conformidad con la mantenibilidad.

Confiabilidad: Capacidad del producto de software para conservar su nivel de

desempeño bajo condiciones específicas, durante un determinado período de tiempo.

Esta característica principal es incluida en el proceso de medición del producto software debido a que es importante para el equipo de desarrollo como para el observatorio del delito:

Conocer la capacidad del sistema para conservar su nivel de desempeño bajo condiciones específicas durante un determinado periodo de tiempo, para establecer la calidad de la información administrada por el producto software.

Que el producto sea confiable siempre y cuando el usuario pueda tener siempre la certeza de que la información está protegida ante cualquier tipo de falla garantizando que su disponibilidad en el que se requiera.

Esta característica se integra en cuatro sub-características que abarcan los aspectos relevantes a nivel de confiabilidad de un producto software: Madurez, Tolerancia a fallas, Restaurabilidad, Disponibilidad y Conformidad en la confiabilidad.

Portabilidad: El producto de software debe tener la capacidad de ser instalado en un abanico amplio de entornos, con la posibilidad de ser transferido de uno a otro.

Esta característica evalúa la capacidad del software para acoplarse a diferentes ambientes, sin necesidad de aplicar acciones adicionales a las consideradas inicialmente en el diseño del producto. Como es una aplicación web solo se necesita tener conexión a internet para su funcionamiento independientemente del sistema operativo.

La portabilidad incluye las sub-características: Coexistencia, Capacidad de reemplazo y Adaptabilidad.

Funcionalidad: Se entiende la funcionalidad como la capacidad del producto software para satisfacer unas necesidades implícitas y explícitas del usuario, al ser utilizado bajo condiciones específicas.

Aquí se evaluará el cumplimiento de los requerimientos especificados en la ingeniería de requisitos como también la exactitud de los resultados obtenidos. Esta información es básica para la toma de decisiones respecto de la continuidad del producto software y/o respecto a su desempeño en las condiciones particulares en que se desarrolla el proceso que el producto software soporta.

La funcionalidad integra las sub-características: Conformidad en la funcionalidad, Exactitud, Interoperabilidad y Seguridad.

Eficiencia: La eficiencia se refiere a la capacidad del producto software para proporcionar un desempeño apropiado, en relación con la cantidad de recurso utilizado, bajo condiciones establecidas en determinado momento del tiempo.

Esta característica se seleccionó para evaluar la eficiencia del sistema dependiendo del archivo y los usuarios conectados al tiempo, como es su comportamiento en el tiempo al igual que el consumo de los recursos y los estándares establecidos. También se evaluará si el equipo de desarrollo tomó en cuenta los estándares para la codificación.

La eficiencia integra las sub-características: Comportamiento en el tiempo, Consumo de recursos y Conformidad en la eficiencia.

Niveles de calificación: Las calificaciones tanto para las características como para las Sub-características se realizarán en un rango de valores que va de 1 a 100 y el resultado de la calificación del producto se evaluará como se describe a continuación: (Ver tabla 11)

Tabla 11: Niveles de calificación

RANGOS DE CALIFICACIÓN	RESULTADO DE LA EVALUACIÓN
90 – 100	Excelente
80 – 89	Bueno
70 – 79	Aceptable
50 – 69	Deficiente
0 – 49	Malo

El producto evaluado como malo, si aún no está en uso no se recomienda su liberación y si está en uso se recomienda su reemplazo, presenta fallas de calidad que hacen que deba pensarse en rediseño del producto de tal forma que se puedan implementar prácticas que garanticen su calidad.

El producto evaluado como deficiente si aún no ha sido liberado requiere mejoras significativas para poder ser liberado y si está en uso requiere que se tomen medidas inmediatas para su mejora o dependiendo de la situación que se deje de utilizar el producto hasta tanto no se haya efectuado un plan de mejora.

El producto evaluado como aceptable puede ser liberado; sin embargo requiere un plan de acción para fortalecer los aspectos críticos.

El producto evaluado como Bueno puede ser liberado y requiere fortalecer algunas Sub-características.

El producto evaluado como excelente puede ser liberado, si aún no está en uso, si está en uso debe tender a seguir conservando las buenas prácticas que hasta el

momento ha llevado y a mirar los posibles aspectos que ayuden a su sostenibilidad en el tiempo.

Ponderación general: De acuerdo con la información obtenida a partir de la reunión con el equipo de desarrollo, los usuarios y el equipo evaluador, a cada una de las características se les dio una un peso en la evaluación, tal como se muestra a continuación: (Ver tabla 12)

Tabla 12: Ponderación general

CARACTERÍSTICA	PONDERACIÓN
Usabilidad	0.24
Mantenibilidad	0.20
Confiabilidad	0.14
Portabilidad	0.08
Funcionalidad	0.20
Eficiencia	0.14

3.2 EVALUACIÓN MÓDULO: MÓDULO DE CARGA MASIVA

Descripción: Este módulo tiene la funcionalidad de recolectar datos a partir de archivos que llegan por parte de instituciones vinculadas al Observatorio del Delito, este módulo almacena esta información para su posterior análisis. A continuación se presenta el diagrama de casos de uso de este módulo.

Ponderación: De acuerdo con la información obtenida a partir de la reunión con el equipo de desarrollo, los usuarios y el equipo evaluador, a cada una de las características se les dio una un peso en la evaluación, tal como se muestra a continuación: (Ver tabla 13)

Tabla 13: Ponderación características del módulo carga masiva

CARACTERISTICA	SUB-CARACTERÍSTICAS	PONDERACIÓN
USABILIDAD	Comprensibilidad	0,15
	Facilidad de Aprendizaje	0,15
	Operabilidad	0,30
	Atractividad	0,10
	Conformidad con la Usabilidad	0,30
MANTENIBILIDAD	Analizabilidad	0,25
	Facilidad de Cambio	0,35
	Estabilidad	0,25
	Facilidad de prueba	0,10
	Conformidad con la Mantenibilidad	0,05
CONFIABILIDAD	Madurez	0,20
	Tolerancia a fallas	0,25
	Restaurabilidad	0,30
	Disponibilidad	0,20
	Conformidad con la Confiabilidad	0,05
PORTABILIDAD	Adaptabilidad	0,50
	Instalabilidad	0,00**
	Coexistencia	0,00**
	Capacidad de Reemplazo	0,50
	Conformidad con la Portabilidad	0,00**
FUNCIONALIDAD	Apropiabilidad	0,20
	Exactitud	0,30
	Interoperabilidad	0,10
	Conformidad en la funcionalidad	0,10
	Seguridad	0,30
EFICIENCIA	Comportamiento en el tiempo	0,35
	Consumo de recursos	0,35
	Conformidad en la eficiencia	0,30

**Las sub-características que no se evaluaron se les dio una ponderación de cero (0).

Listas de chequeo y estrategias de evaluación: Como paso inicial de la evaluación, se elaboraron listas de chequeo por cada sub-característica a evaluar y las estrategias de evaluación, posterior a esto se realizó el proceso de clasificación de las preguntas de las listas por medio de la herramienta epidat, la

cual a través del alfa de Cronbach, determina si las preguntas son consistentes. Las listas de chequeo por sub-característica y las estrategias de evaluación fueron consignadas en los anexos: Anexo Lista de chequeo Módulo carga masiva y Anexo Reporte de pruebas módulo carga masiva. (Ver tabla 14)

Métricas: Las métricas utilizadas fueron en su mayoría basadas en la evaluación del cumplimiento de mejores prácticas que se plantearon teniendo como base las listas de chequeo iniciales, las estrategias de evaluación y los aspectos que el producto debía considerar acorde con las sub-características a evaluar. El total de las métricas fueron consignadas en el Anexo Lista de chequeo Módulo carga masiva.

Evaluación: La evaluación se desarrolló utilizando como parámetro una plantilla consignada en la evaluación módulo carga masiva, la cual se presenta a continuación.

Resultados por características:

Tabla 14: Resultados por características

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
USABILIDAD	21,30	21,55	Comprensibilidad	82,86	12,43	84,07	12,61
			Facilidad de Aprendizaje	95,84	14,38	96,34	14,45
			Operabilidad	87,17	26,15	88,33	26,50
			Atractividad	87,78	8,78	92,50	9,25
			Conformidad con la Usabilidad	90,00	27,00	90,00	27,00
					88,73		89,81

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
MANTENIBILIDAD	16,59	16,76	Analizabilidad	86,94	21,74	90,28	22,57
			Facilidad de Cambio	90,00	31,50	90,00	31,50
			Estabilidad	83,33	20,83	83,33	20,83
			Facilidad de Prueba	52,50	5,25	52,50	5,25
			Conformidad con la Mantenibilidad	72,50	3,63	72,50	3,63
					82,94		83,78

Característica	Ponderación	Calificación ponderada	Sub-características	Ponderación	Calificación	Calificación ponderada
CONFIABILIDAD*	0,14	12,37	Madurez	0,20	97,50	19,50
			Tolerancia a fallas	0,25	73,33	18,33
			Restaurabilidad	0,30	85,00	25,50
			Disponibilidad	0,20	100,00	20,00
			Conformidad con la Confiabilidad	0,05	100,00	5,00
				1,00		88,33

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
PORTABILIDAD	7,33	7,39	Adaptabilidad	88,00	44,00	88,00	44,00
			Capacidad de Reemplazo	95,24	47,62	96,67	48,34
					91,62		92,34

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
FUNCIONALIDAD	17,5	18,34	Apropiabilidad	93,33	18,67	93,33	18,67
			Exactitud	100,00	30,00	100,00	30,00
			Interoperabilidad	100,00	10,00	100,00	10,00
			Seguridad	62,81	18,84	76,81	23,04
			Conformidad con la funcionalidad	100,00	10,00	100,00	10,00
					87,51		91,71

Característica	Ponderación	Calificación ponderada	Sub-características	Ponderación	Calificación	Calificación ponderada
EFICIENCIA*	0,14	13,16	Comportamiento en el tiempo	0,35	100,00	35,00
			Consumo de recursos	0,35	82,86	29,00
			Conformidad en la eficiencia	0,30	100,00	30,00
				1,00		94,00

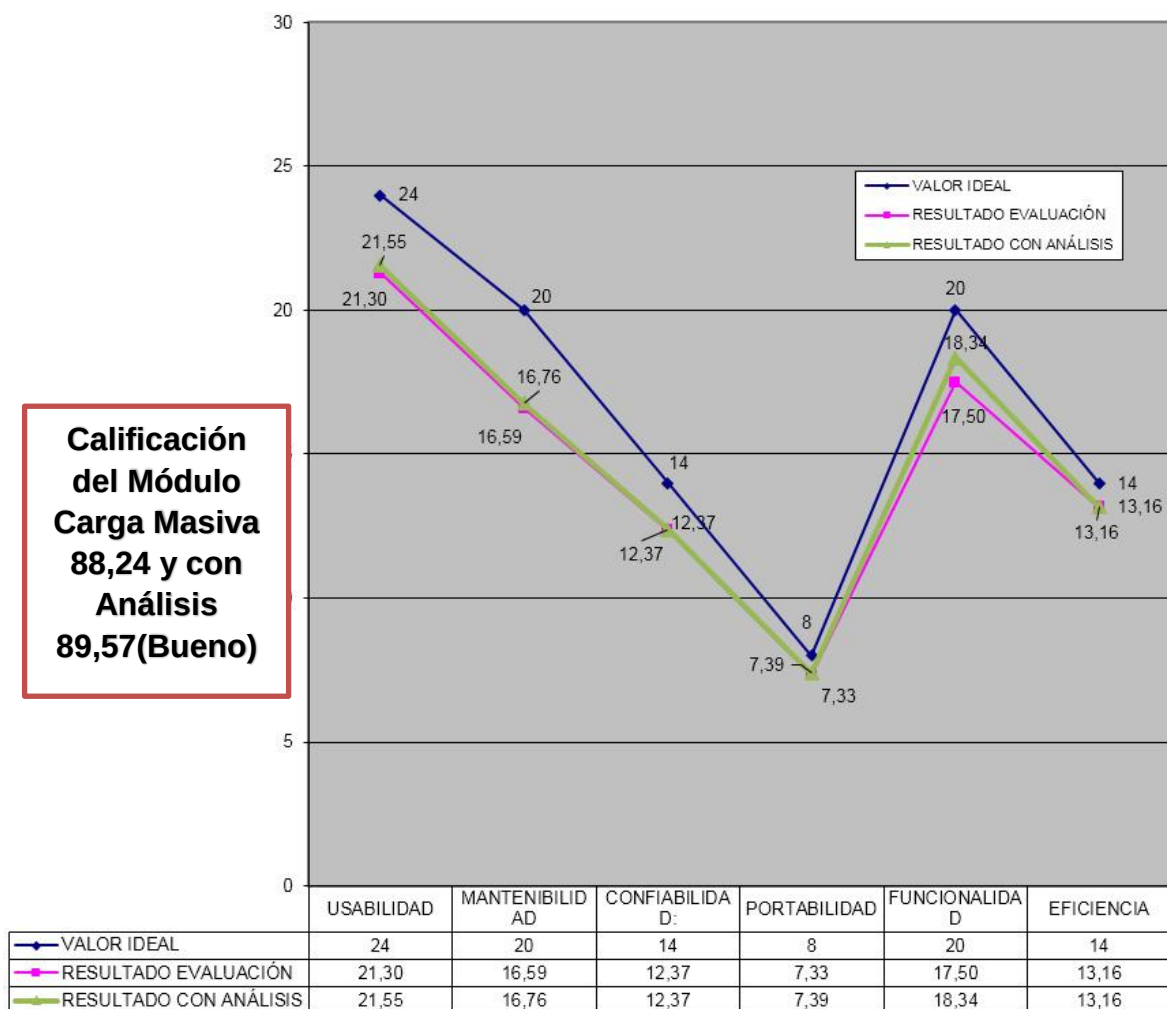
* Estas características después del análisis no tuvieron ningún cambio.

Resultado General:

Tabla 15: Resultado general módulo de carga masiva

Características	Ponderación	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
Usabilidad	0,24	88,73	21,30	89,81	21,55
Mantenibilidad	0,20	82,94	16,59	83,78	16,76
Confiabilidad	0,14	88,33	12,37	88,33	12,37
Portabilidad	0,08	91,62	7,33	92,34	7,39
Funcionalidad	0,20	87,51	17,50	91,71	18,34
Eficiencia	0,14	94,00	13,16	94,00	13,16
Total	1,00		88,24		89,57

Ilustración 21: Comparación de resultados del módulo carga masiva



Análisis de resultados del módulo de carga masiva:

- El resultado de la evaluación permite calificar al producto con un grado de calidad Bueno (89,00) sin embargo requiere la implementación de un plan que le permita alcanzar la excelencia en calidad.
- La usabilidad del producto (88,73) es Bueno; obteniendo la calificación más baja en comprensibilidad esto sugiere que la capacitación debe ser exhaustiva para su perfecto desenvolvimiento en todas las funcionalidades del módulo.
- La mantenibilidad del producto (82,94) es Buena, presentando nivel de evaluación deficiente en la facilidad de prueba.
- La confiabilidad del producto (88,33) es Buena, obteniendo la calificación más baja en tolerancia a fallas (40,00), lo que sugiere un punto de atención en esta sub-característica.
- La portabilidad del producto (91,62) es Excelente, el software es de fácil instalación por lo que únicamente se tienen en cuenta dos ítem para la calificación.
- La funcionalidad del producto (87,51) con base en los niveles de calificación puede evaluarse como Bueno, presentando un nivel de calificación malo en Seguridad ya que el usuario no tiene completa seguridad de que la información es subida a la base de datos sin sufrir ninguna alteración.
- La eficiencia del producto (94,00) es Excelente, siendo notorio que el módulo realiza lo que el usuario solicitó inicialmente.

Recomendaciones al módulo de carga masiva:

- En lo referente a mantenibilidad del módulo de carga masiva se sugiere implementar mejores prácticas en la sub-característica Estabilidad, la adecuada documentación que permita hacer un seguimiento a los errores, en capacidad para ser verificado que las modificaciones realizadas sean posibles hacerlas en corto tiempo y respectivo proceso de verificación no tenga tanta dificultad, y la analizabilidad del código sugiere que el código del módulo se encuentre completamente documentado.
- En lo referente a la confiabilidad en el aspecto de continuidad en la operación se recomienda mejorar lo relacionado a salvaguardar la información ingresada y analizada cuando haya un problema de red o se quiera cerrar la ventana actual en la cual se está manipulando la información.

- Respecto a la funcionalidad del módulo de carga masiva se sugiere implementar las mejores prácticas relacionadas con la seguridad, tales como actualización de claves de acceso con cierta frecuencia, aislamiento físicamente del sistema y de comprobación de que los datos no han sido modificados durante la transferencia.

3.3 EVALUACIÓN MÓDULO: MÓDULO CARGA POR FORMULARIO

Descripción: El módulo de carga por formulario tiene como principal funcionalidad el ingreso de información de todas las lesiones fatales y no fatales que maneja el Observatorio del Delito para su ingreso a la base de datos y posterior análisis por parte de los funcionarios. A continuación se presenta el diagrama de casos de uso de este módulo. (Ver tabla 16)

Ponderación: De acuerdo con la información obtenida a partir de la reunión con el equipo de desarrollo, los usuarios y el equipo evaluador, a cada una de las características se les dio una un peso en la evaluación, tal como se muestra a continuación:

Tabla 16: Ponderación de características del módulo carga por formulario

CARACTERISTICA	SUB-CARACTERÍSTICAS	PONDERACIÓN
USABILIDAD	Comprensibilidad	0,15
	Facilidad de Aprendizaje	0,15
	Operabilidad	0,30
	Atractividad	0,10
	Conformidad con la Usabilidad	0,30
MANTENIBILIDAD	Analizabilidad	0,25
	Facilidad de Cambio	0,35
	Estabilidad	0,25
	Facilidad de Prueba	0,10
	Conformidad con la mantenibilidad	0,05
CONFIABILIDAD	Madurez	0,20
	Tolerancia a fallas	0,25
	Restaurabilidad	0,30
	Disponibilidad	0,20
	Conformidad con la Confiabilidad	0,05

CARACTERISTICA	SUB-CARACTERÍSTICAS	PONDERACIÓN
PORTABILIDAD	Adaptabilidad	0,50
	Capacidad de Reemplazo	0,50
FUNCIONALIDAD	Apropiabilidad	0,20
	Exactitud	0,30
	Interoperabilidad	0,10
	seguridad	0,30
	Conformidad con la funcionalidad	0,10
EFICIENCIA	Comportamiento en el tiempo	0,35
	Consumo de recursos	0,35
	Conformidad en la eficiencia	0,30

Listas de chequeo y estrategias de evaluación: Como paso inicial de la evaluación, se elaboraron listas de chequeo por cada sub-característica a evaluar y las estrategias de evaluación, posterior a esto se realizó el proceso de clasificación de las preguntas de las listas por medio de la herramienta epidat, la cual a través del alfa de Cronbach, determina si las preguntas son consistentes. Las listas de chequeo por sub-característica y las estrategias de evaluación fueron consignadas en los anexos: Anexo Lista de chequeo módulo carga por formulario y Anexo Reporte de pruebas módulo carga por formulario. (Ver tabla 17)

Métricas: Las métricas utilizadas fueron en su mayoría basadas en la evaluación del cumplimiento de mejores prácticas que se plantearon teniendo como base las listas de chequeo iniciales, las estrategias de evaluación y los aspectos que el producto debía considerar acorde con las sub-características a evaluar.

El total de las métricas fueron consignadas en el Anexo Lista de chequeo módulo carga masiva.

Evaluación: La evaluación se desarrolló utilizando como parámetro una plantilla consignada en la evaluación módulo carga por formulario, la cual se presenta a continuación.

Tabla 17: Resultados por Características

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
USABILIDAD	21,22	21,17	Comprensibilidad	85,43	12,81	85,43	12,81
			Facilidad de Aprendizaje	95,84	14,38	95,84	14,38
			Operabilidad	83,67	25,10	83,08	24,92
			Atractividad	91,11	9,11	91,11	9,11
			Conformidad con la Usabilidad	90,00	27,00	90,00	27,00
					88,40		88,22

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
MANTENIBILIDAD	16,98	17,39	Analizabilidad	85,69	21,42	89,03	22,26
			Facilidad de Cambio	90,00	31,50	90,00	31,50
			Estabilidad	83,33	20,83	83,33	20,83
			Facilidad de prueba	75,00	7,50	87,50	8,75
			Conformidad con la mantenibilidad	72,50	3,63	72,50	3,63
					84,88		86,97

Característica	Ponderación	Calificación ponderada	Sub-características	Ponderación	Calificación	Calificación ponderada
CONFIABILIDAD*	0,14	12,32	Madurez	0,20	95,70	19,14
			Tolerancia a fallas	0,25	73,33	18,33
			Restaurabilidad	0,30	85,00	25,50
			Disponibilidad	0,20	100,00	20,00
			Conformidad con la Confiabilidad	0,05	100,00	5,00
				1,00		87,98

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
PORTABILIDAD	6,71	6,75	Adaptabilidad	81,78	40,89	82,76	41,38
			Capacidad de Reemplazo	85,95	42,98	85,95	42,98
					83,87		84,36

Característica	Ponderación	Calificación ponderada	Sub-características	Ponderación	Calificación	Calificación ponderada
FUNCIONALIDAD*	0,20	18,45	Apropiabilidad	0,20	91,67	18,33
			Exactitud	0,30	100,00	30,00
			Interoperabilidad	0,10	100,00	10,00
			Conformidad en la funcionalidad	0,10	100,00	10,00
			Seguridad	0,30	79,72	23,92
				1,00		92,25

Característica	Ponderación	Calificación ponderada	Sub-características	Ponderación	Calificación	Calificación ponderada
EFICIENCIA*	0,14	13,34	Comportamiento en el tiempo	0,35	95,00	33,25
			Consumo de recursos	0,35	91,43	32,00
			Conformidad en la eficiencia	0,30	100,00	30,00
				1,00		95,25

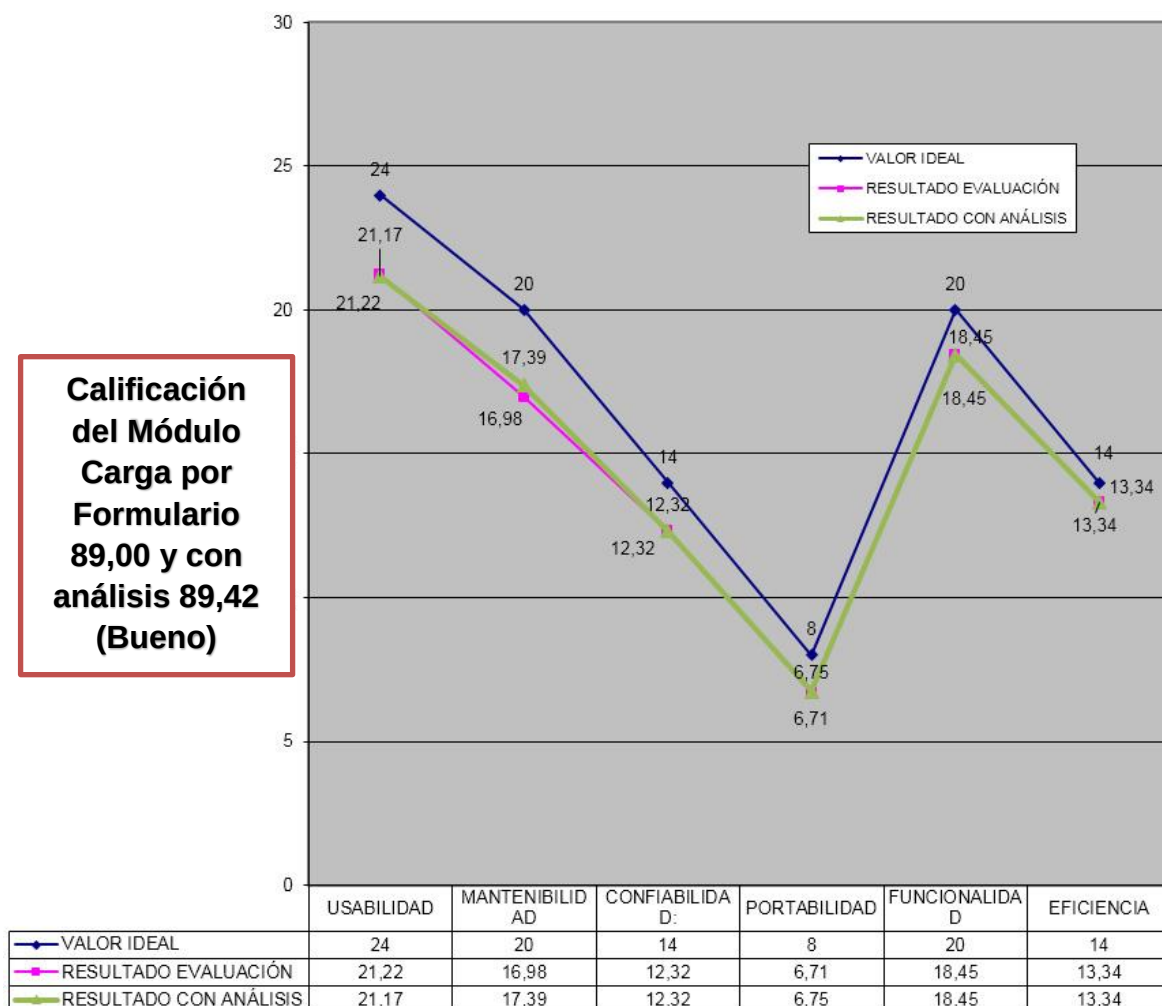
* Estas características después del análisis no tuvieron ningún cambio.

Resultado General:

Tabla 14: Resultado general módulo carga por formulario

Característica	Ponderación	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
Usabilidad	0,24	88,40	21,22	88,22	21,17
Mantenibilidad	0,20	84,88	16,98	86,97	17,39
Confiabilidad	0,14	87,97	12,32	87,97	12,32
Portabilidad	0,08	83,87	6,71	84,36	6,75
Funcionalidad	0,20	92,25	18,45	92,25	18,45
Eficiencia	0,14	92,25	13,34	92,25	13,34
Total	1,00		89,00		89,42

Ilustración 22: Comparación de resultados del módulo de carga por formulario



Análisis de resultados del módulo de carga por formulario:

- El resultado de la evaluación permite calificar al módulo carga por formulario con un grado de calidad Bueno (89,00) sin embargo requiere la implementación de un plan de mejora que le permita alcanzar un mejor estándar de calidad.
- La usabilidad del módulo carga por formulario (88,40) es Buena; sin embargo obtuvo una calificación buena en lo referente a la comprensibilidad (85,43) y la Operabilidad (83,67) lo cual bajo su buen promedio para llegar a una calificación de excelencia.

- La mantenibilidad del módulo carga por formulario (84,88) es Buena, presentando nivel de evaluación aceptable en la facilidad de prueba y conformidad con la mantenibilidad.
- La confiabilidad del módulo carga por formulario (87,97) es Buena, sin embargo obtuvo una calificación mala en lo referente a la tolerancia a fallas (40).
- La portabilidad del módulo carga por formulario (83,87) es Buena, sin embargo debido a que es módulo web su calificación puede ser mejor.
- La funcionalidad del módulo carga por formulario (92,25) con base en los niveles de calificación puede evaluarse como Excelente, presentando un nivel de calificación muy bueno respecto a su funcionamiento.
- La eficiencia del módulo carga por formulario (95,25) es Excelente, presentando un nivel de calificación muy alto.

Recomendaciones al módulo de carga por formulario:

- Respecto a la funcionalidad del módulo carga por formulario se sugiere implementar las mejores prácticas relacionadas con la seguridad, tales como actualización de claves de acceso con cierta frecuencia , aislamiento de donde se encuentra físicamente aislado el sistema y de comprobación de que los datos no han sido modificados durante la transferencia.
- En lo referente a mantenibilidad del módulo carga por formulario se sugiere implementar mejores prácticas relacionadas con la facilidad de cambio como que sus funcionalidades a la hora del ingreso de datos sean suficientes, en la sub-característica estabilidad la adecuada documentación que permita hacer un seguimiento a los errores, en capacidad para ser verificado que las modificaciones realizadas sean posibles hacerlas en corto tiempo y respectivo proceso de verificación no tenga tanta dificultad, la analizabilidad del código que el código del módulo se encuentre completamente documentado
- En lo referente a la eficiencia en el aspecto de la capacidad para optimizar el acceso a la base de datos se recomienda mejorar el aspecto relacionado con el control periódico de la base de datos
- En lo referente a la confiabilidad en el aspecto de continuidad en la operación se recomienda mejorar lo relacionado a salvaguardar la información ingresada manualmente cuando haya un problema de red o se quiera cerrar la ventana actual en la cual se está diligenciando el formulario.

3.4 EVALUACIÓN MÓDULO: CRUCE DE INDICADORES

Descripción: El módulo de cruce de indicadores tiene como principal funcionalidad el cruce de variables para mostrar al usuario de manera estadística y en diferentes tipos de gráficos los diferentes resultados a partir de los cruces, Los gráficos presentados corresponden a la información solicitada por el usuario gestionadas por el Observatorio del Delito. A continuación se presenta el diagrama de casos de uso del módulo. (Ver tabla 19)

Ponderación: De acuerdo con la información obtenida a partir de la reunión con el equipo de desarrollo, los usuarios y el equipo evaluador, a cada una de las características se les dio una un peso en la evaluación, tal como se muestra a continuación:

Tabla 19: Ponderación de características del módulo cruce de indicadores

CARACTERISTICA	SUB-CARACTERÍSTICAS	PONDERACIÓN
USABILIDAD	Comprensibilidad	0,15
	Facilidad de Aprendizaje	0,15
	Operabilidad	0,30
	Atractividad	0,10
	Conformidad con la Usabilidad	0,30
MANTENIBILIDAD	Analizabilidad	0,25
	Facilidad de Cambio	0,35
	Estabilidad	0,25
	Facilidad de prueba	0,10
	Conformidad con la Mantenibilidad	0,05
CONFIABILIDAD	Madurez	0,20
	Tolerancia a fallas	0,25
	Restaurabilidad	0,30
	Disponibilidad	0,20
	Conformidad con la Confiabilidad	0,05
PORTABILIDAD	Adaptabilidad	0,50
	Instalabilidad	0,00**
	Coexistencia	0,00**
	Capacidad de Reemplazo	0,50
	Conformidad con la Portabilidad	0,00**

CARACTERISTICA	SUB-CARACTERÍSTICAS	PONDERACIÓN
FUNCIONALIDAD	Apropiabilidad	0,20
	Exactitud	0,30
	Interoperabilidad	0,10
	Conformidad en la funcionalidad	0,10
	Seguridad	0,30
EFICIENCIA	Comportamiento en el tiempo	0,35
	Consumo de recursos	0,35
	Conformidad en la eficiencia	0,30

**Las sub-características que no se evaluaron se les dio una ponderación de cero (0).

Listas de chequeo y estrategias de evaluación: Como paso inicial de la evaluación, se elaboraron listas de chequeo por cada sub-característica a evaluar y las estrategias de evaluación, posterior a esto se realizó el proceso de clasificación de las preguntas de las listas por medio de la herramienta epidat, la cual a través del alfa de Cronbach, determina si las preguntas son consistentes. Las listas de chequeo por sub-característica y las estrategias de evaluación fueron consignadas en los anexos: Lista de chequeo módulo cruce de indicadores y Anexo Reporte de pruebas módulo cruce de indicadores. (Ver tabla 20)

Métricas: Las métricas utilizadas fueron en su mayoría basadas en la evaluación del cumplimiento de mejores prácticas que se plantearon teniendo como base las listas de chequeo iniciales, las estrategias de evaluación y los aspectos que el producto debía considerar acorde con las sub-características a evaluar. El total de las métricas fueron consignadas en Anexo Lista de chequeo módulo cruce de indicadores.

Evaluación: La evaluación se desarrolló utilizando como parámetro una plantilla consignada en la calificación módulo cruce de indicadores, la cual se presenta a continuación.

Tabla 20: Resultados por características

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
USABILIDAD	21,10	21,58	Comprensibilidad	84,34	12,65	84,34	12,65
			Facilidad de Aprendizaje	92,34	13,85	92,34	13,85
			Operabilidad	86,00	25,80	92,00	27,60
			Atractividad	86,22	8,62	88,00	8,80
			Conformidad con la Usabilidad	90,00	27,00	90,00	27,00
					87,92		89,90

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
MANTENIBILIDAD	16,56	16,62	Analizabilidad:	84,03	21,01	87,36	21,84
			Facilidad de Cambio	84,00	29,40	84,00	29,40
			Estabilidad	85,00	21,25	85,00	21,25
			Facilidad de Prueba	75,00	7,50	70,00	7,00
			Conformidad con la Mantenibilidad	72,50	3,63	75,50	3,63
					82,78		83,12

Característica	Ponderación	Calificación ponderada	Sub-características	Ponderación	Calificación	Calificación ponderada
CONFIABILIDAD*	0,14	11,88	Madurez	0,20	94,17	18,83
			Tolerancia a fallas	0,25	60,00	15,00
			Restaurabilidad	0,30	86,67	26,00
			Disponibilidad	0,20	100,00	20,00
			Conformidad con la Confiabilidad	0,05	100,00	5,00
				1,00		84,84

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
PORTABILIDAD	7,16	7,25	Adaptabilidad	85,11	42,56	86,31	43,16
			Capacidad de Reemplazo	93,81	46,91	95,00	47,50
					89,46		90,66

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
FUNCIONALIDAD	18,52	18,50	Apropiabilidad	93,00	18,60	92,44	18,49
			Exactitud	100,00	30,00	100,00	30,00
			Interoperabilidad	100,00	10,00	10,00	10,00
			Seguridad	80,01	24,00	80,01	24,00
			Conformidad con la funcionalidad	100,00	10,00	100,00	10,00
					92,60		92,49

Característica	Ponderación	Calificación ponderada	Sub-características	Ponderación	Calificación	Calificación ponderada
EFICIENCIA*	0,14	13,67	Comportamiento en el tiempo	0,35	93,33	32,67
			Consumo de recursos	0,35	100,00	35,00
			Conformidad en la eficiencia	0,30	100,00	30,00
				1,00		97,67

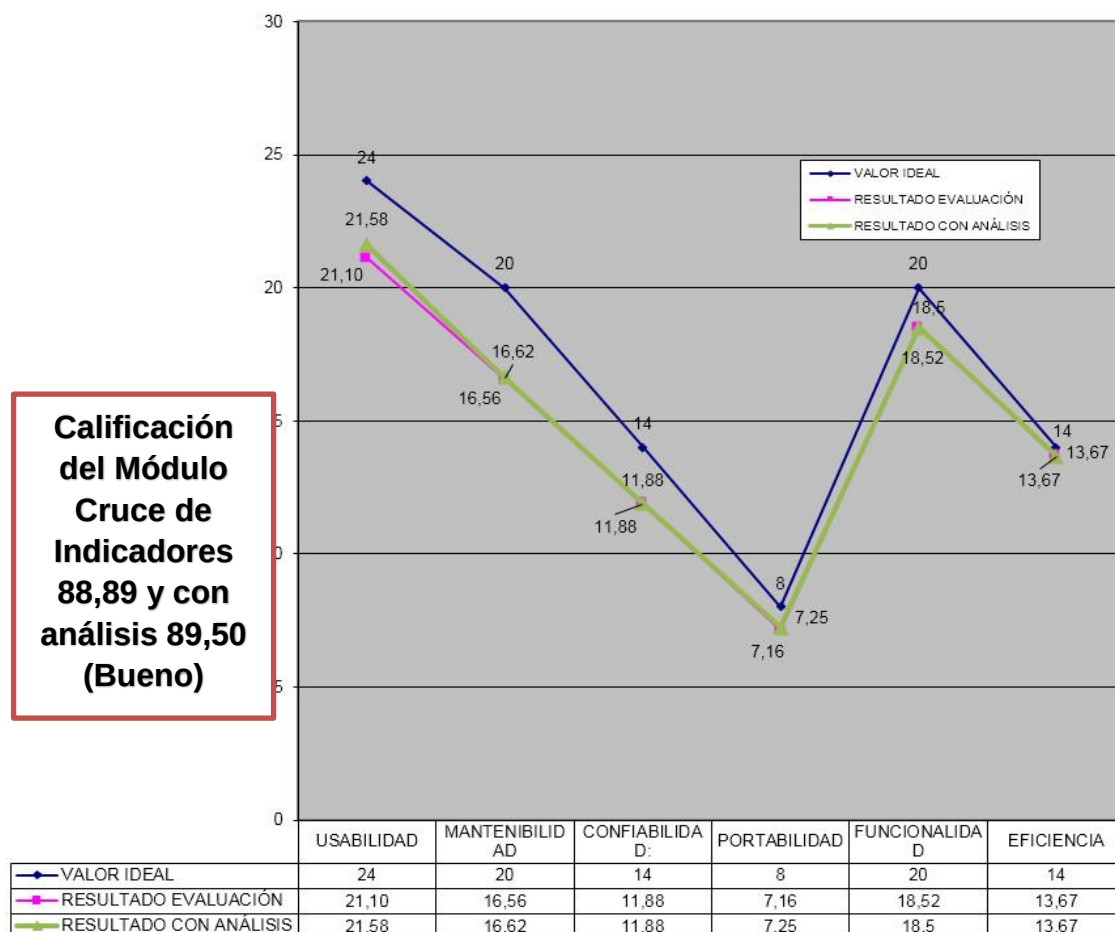
* Estas características después del análisis no tuvieron ningún cambio.

Resultado General:

Tabla 21: Resultado general módulo cruce de indicadores

Características	Ponderación	Calificación	Calificación Ponderada	Calificación con análisis	Calificación Ponderada con análisis
Usabilidad	0,24	87,92	21,10	89,90	21,58
Mantenibilidad	0,20	82,78	16,56	83,12	16,62
Confiabilidad	0,14	84,84	11,88	84,84	11,88
Portabilidad	0,08	89,46	7,16	90,66	7,25
Funcionalidad	0,20	92,60	18,52	92,49	18,50
Eficiencia	0,14	97,67	13,67	97,67	13,67
Total	1,00		88,89		89,50

Ilustración 23: Comparación de calificaciones del módulo de cruce de indicadores



Análisis de resultados del módulo cruce de indicadores:

- El resultado de la evaluación permite calificar al módulo cruce de indicadores con un grado de calidad Bueno (88,89) sin embargo requiere la implementación de un plan de mejora que le permita alcanzar la excelencia de calidad al módulo.
- La usabilidad del producto (87,92) es Buena; su calificación más baja es en comprensibilidad lo cual indica que es un poco complicado comprender todas las funcionalidades del módulo, en general los usuarios calificaron al módulo como bueno esto quiere decir que el módulo hace lo que los usuarios necesitan.

- La mantenibilidad del producto (82,78) es Buena, presentando un nivel de evaluación aceptable en la facilidad de prueba y en la conformidad con la mantenibilidad.
- La confiabilidad del producto (84,84) es Buena, sin embargo está en el límite la Tolerancia a fallas, que es bastante importante para el usuario ya que es una aplicación web, esto sugiere un punto de atención en esta sub-característica.
- La portabilidad del módulo carga por formulario (83,87) es Buena, sin embargo debido a que es módulo web su calificación puede ser mejor.
- La funcionalidad del producto (92,60) con base en los niveles de calificación puede evaluarse como Excelente, indicando que las funcionalidades que posee el software son bastante útiles para el usuario, y cumpliendo con lo que inicialmente solicitaron.
- La eficiencia del producto (97,67) es Excelente, siendo notorio que el Consumo de recursos (100,0) tanto en hardware como en software es excelente por parte del aplicativo.

Recomendaciones al módulo cruce de indicadores:

- En la usabilidad respecto al módulo se recomienda implementar una serie de opciones que permitan la Operabilidad por parte del usuario de una forma más fácil y rápida, como también la diferenciación de los campos para la fácil navegación de los usuarios.
- En lo referente a mantenibilidad del producto se sugiere implementar un registro de las modificaciones para llevar un control total sobre los cambios realizados y los errores encontrados, así como también se recomienda implementar un registro sobre las acciones que realizan los usuarios en el módulo todo esto para que se facilite la rastreabilidad en el sistema.
- Igualmente, respecto a la confiabilidad del producto se sugiere implementar una función que al momento de que el sistema ocurra una falla el usuario pueda continuar desde un punto de restauración que tenga definido el sistema, así también se recomienda contar con un método para que el usuario tenga total seguridad de que la información no va a ser alterada en ningún momento.
- Respecto a la funcionalidad del producto se sugiere implementar las mejores prácticas relacionadas con la seguridad, tales como exigencia de actualización de claves de acceso periódicamente, desactivación de claves de acceso para usuarios ausentes, así como también se recomienda que los servidores se

reubiquen en un lugar aislado y que el servidor este en diferente máquina que la base de datos.

- En la eficiencia se obtuvo una calificación casi perfecta lo que representa que el módulo es muy eficiente para lo que el usuario lo necesita solo se recomienda bajar el tiempo en el procesamiento de datos aunque esto es relativo a la tarea que se esté desempeñando.

3.5 EVALUACIÓN MÓDULO: MÓDULO MAPAS TEMÁTICOS

Descripción: El módulo de mapas temáticos tiene como principal funcionalidad mostrar la información consultada en forma gráfica utilizando el mapa del municipio de Pasto, en el mapa se muestran las diferentes variables que el usuario escoja clasificando la información con diferentes colores. A continuación se muestra el diagrama de casos de uso de este módulo. (Ver tabla 22)

Ponderación: De acuerdo con la información obtenida a partir de la reunión con el equipo de desarrollo, los usuarios y el equipo evaluador, a cada una de las características se les dio un un peso en la evaluación, tal como se muestra a continuación:

Tabla 22: Ponderación de características del módulo mapas temáticos

CARACTERISTICA	SUB-CARACTERÍSTICAS	PONDERACIÓN
USABILIDAD	Comprensibilidad	0,15
	Facilidad de Aprendizaje	0,15
	Operabilidad	0,30
	Atractividad	0,10
	Conformidad con la Usabilidad:	0,30
MANTENIBILIDAD	Analizabilidad	0,25
	Facilidad de Cambio	0,35
	Estabilidad	0,25
	Facilidad de prueba	0,10
	Conformidad con la mantenibilidad	0,05
CONFIABILIDAD	Madurez	0,27
	Tolerancia a fallas	0,32
	Restaurabilidad	0,00**
	Disponibilidad	0,27
	Conformidad con la Confiabilidad	0,14

CARACTERISTICA	SUB-CARACTERÍSTICAS	PONDERACIÓN
PORTABILIDAD	Adaptabilidad	0,50
	Instalabilidad	0,00**
	Coexistencia	0,00**
	Capacidad de Reemplazo	0,50
	Conformidad con la Portabilidad	0,00**
FUNCIONALIDAD	Apropiabilidad	0,20
	Exactitud	0,30
	Interoperabilidad	0,10
	Conformidad en la funcionalidad	0,10
	Seguridad	0,30
EFICIENCIA	Comportamiento en el tiempo	0,35
	Consumo de recursos	0,35
	Conformidad en la eficiencia	0,30

**Las sub-características que no se evaluaron se les dio una ponderación de cero (0).

Listas de chequeo y estrategias de evaluación: Como paso inicial de la evaluación, se elaboraron listas de chequeo por cada sub-característica a evaluar y las estrategias de evaluación, posterior a esto se realizó el proceso de clasificación de las preguntas de las listas por medio de la herramienta epidat, la cual a través del alfa de Cronbach, determina si las preguntas son consistentes. Las listas de chequeo por sub-característica y las estrategias de evaluación fueron consignadas en los anexos: Anexo Lista de chequeo módulo mapas temáticos y Anexo Reporte de pruebas módulo mapas temáticos. (Ver tabla 23)

Métricas: Las métricas utilizadas fueron en su mayoría basadas en la evaluación del cumplimiento de mejores prácticas que se plantearon teniendo como base las listas de chequeo iniciales, las estrategias de evaluación y los aspectos que el producto debía considerar acorde con las sub-características a evaluar. El total de las métricas fueron consignadas en el Anexo Lista de chequeo módulo mapas temáticos.

Evaluación: La evaluación se desarrolló utilizando como parámetro una plantilla consignada en la calificación módulo mapas temáticos, la cual se presenta a continuación.

Tabla 23: Resultados por características

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
USABILIDAD	21,60	21,59	Comprensibilidad	85,43	12,81	85,43	12,81
			Facilidad de Aprendizaje	92,34	13,85	92,34	13,85
			Operabilidad	87,57	26,27	87,40	26,22
			Atractividad	82,67	8,27	82,67	8,27
			Conformidad con la Usabilidad	96,00	28,80	96,00	28,80
					90,00		89,95

Característica	Ponderación	Calificación ponderada	Sub-características	Ponderación	Calificación	Calificación ponderada
MANTENIBILIDAD *	0,20	16,61	Analizabilidad	0,25	86,00	21,50
			Facilidad de Cambio	0,35	90,00	31,50
			Estabilidad	0,25	81,67	20,42
			Facilidad de Prueba	0,10	62,71	6,27
			Conformidad con la Mantenibilidad	0,05	67,50	3,38
				1,00		83,06

Característica	Ponderación	Calificación ponderada	Sub-características	Ponderación	Calificación	Calificación ponderada
CONFIABILIDAD *	0,14	12,18	Madurez	0,27	99,17	26,78
			Tolerancia a fallas	0,32	60,00	19,20
			Restaurabilidad	0,00	0,00	0,00
			Disponibilidad	0,27	100,00	27,00
			Conformidad con la Confiabilidad	0,14	100,00	14,00
				1,00		86,98

Característica	Calificación ponderada	Calificación ponderada con análisis	Sub-características	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
PORTABILIDAD	7,0504	7,0528	Adaptabilidad	82,45	41,23	83,82	41,91
			Capacidad de Reemplazo	93,81	46,91	92,50	46,25
					88,13		88,16

Característica	Ponderación	Calificación ponderada	Sub-características	Ponderación	Calificación	Calificación ponderada
FUNCIONALIDAD*	0,20	18,97	Apropiabilidad	0,20	90,67	18,13
			Exactitud	0,30	100,00	30,00
			Interoperabilidad	0,10	100,00	10,00
			Seguridad	0,30	89,04	26,71
			Conformidad con la funcionalidad	0,10	100,00	10,00
				1,00		94,85

Característica	Ponderación	Calificación ponderada	Sub-características	Ponderación	Calificación	Calificación ponderada
EFICIENCIA *	0,14	13,76	Comportamiento en el tiempo	0,35	95,00	33,25
			Consumo de recursos	0,35	100,00	35,00
			Conformidad en la eficiencia	0,30	100,00	30,00
				1,00		98,25

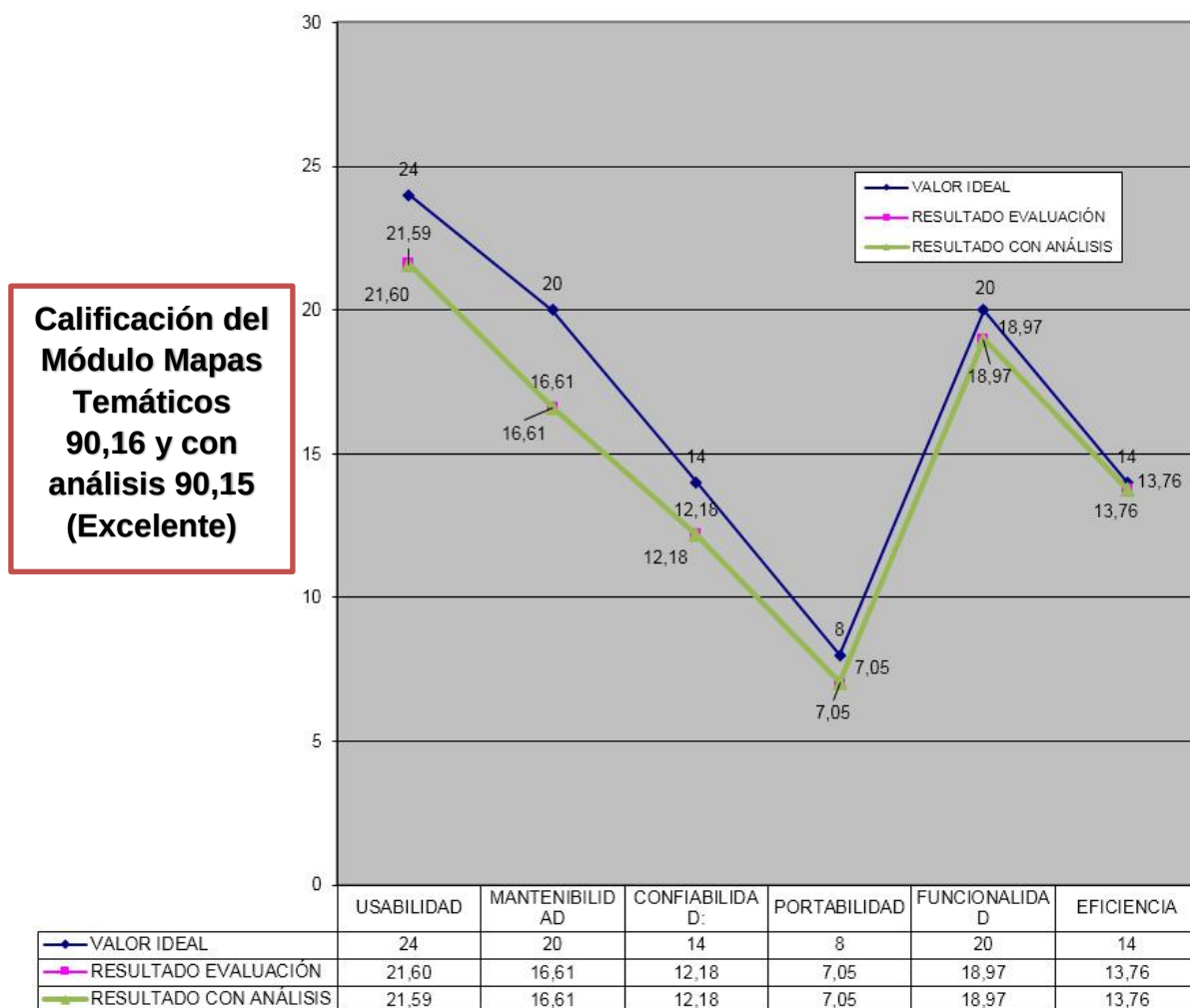
* Estas características después del análisis no tuvieron ningún cambio.

Resultado General:

Tabla 24: Resultado general módulo mapas temáticos

Característica	Ponderación	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
Usabilidad	0,24	90,00	21,60	89,95	21,59
Mantenibilidad	0,20	83,06	16,61	83,06	16,61
Confiabilidad	0,14	86,98	12,18	86,98	12,18
Portabilidad	0,08	88,13	7,05	88,16	7,05
Funcionalidad	0,20	94,85	18,97	94,85	18,97
Eficiencia	0,14	98,25	13,76	98,25	13,76
Total	1,00		90,16		90,15

Ilustración 24: Comparación de resultados del módulo de mapas temáticos



Fuente: (Esta Investigación)

Análisis de resultados del módulo mapas temáticos

- El resultado de la evaluación permite calificar al producto con un grado de calidad Excelente (90,16), lo que determina que el módulo satisface todas los requerimientos que él usuario solicitó.
- La usabilidad del producto (90,0) es Excelente; el módulo es fácil de utilizar, de fácil aprendizaje y con una interfaz gráfica agradable al usuario.
- La mantenibilidad del producto (83,06) es Bueno; a pesar de ésta calificación, sería bueno revisar: Facilidad de prueba (62,71) y Conformidad con la mantenibilidad (67,50), para que la calificación sea Excelente.

- La confiabilidad del producto (86,98) es Bueno, para que el módulo tenga una calificación de Excelente sería recomendable revisar la Sub-característica Tolerancia a fallas (60,00), que fue que saco menor calificación.
- La portabilidad del módulo carga por formulario (88,13) es Buena, sin embargo debido a que es módulo web su calificación puede ser mejor.
- La funcionalidad del producto (94,85) con base en los niveles de calificación puede evaluarse como Excelente, lo que indica que el módulo cumple todas las funciones que el usuario necesita.
- La eficiencia del producto (98,25) es Excelente, el módulo desempeña un comportamiento apropiado en cuanto a consumo de recursos y comportamiento en el tiempo.

Recomendaciones la módulo mapas temáticos:

- En lo referente a mantenibilidad del producto se sugiere implementar mejores prácticas relacionadas con la facilidad de prueba, como mejorar el proceso de verificación de las modificaciones realizadas al código.
- Igualmente, respecto a la mantenibilidad del producto se sugiere implementar mejores prácticas relacionadas con la conformidad con la mantenibilidad, tales como la estandarización de la documentación de mantenimiento y la implementación de procedimientos para el seguimiento de errores.
- Respecto a la confiabilidad del producto se sugiere implementar las mejores prácticas relacionadas con la tolerancia a fallas, ya que el módulo no posee acciones para ejecutarse cuando ocurran errores impredecibles (problemas en la red, eléctricos, etc.) y los de mensajes que ayuden al usuario a identificar el posible error no son los adecuados. (Ver tabla 25) (Ver ilustración 26)

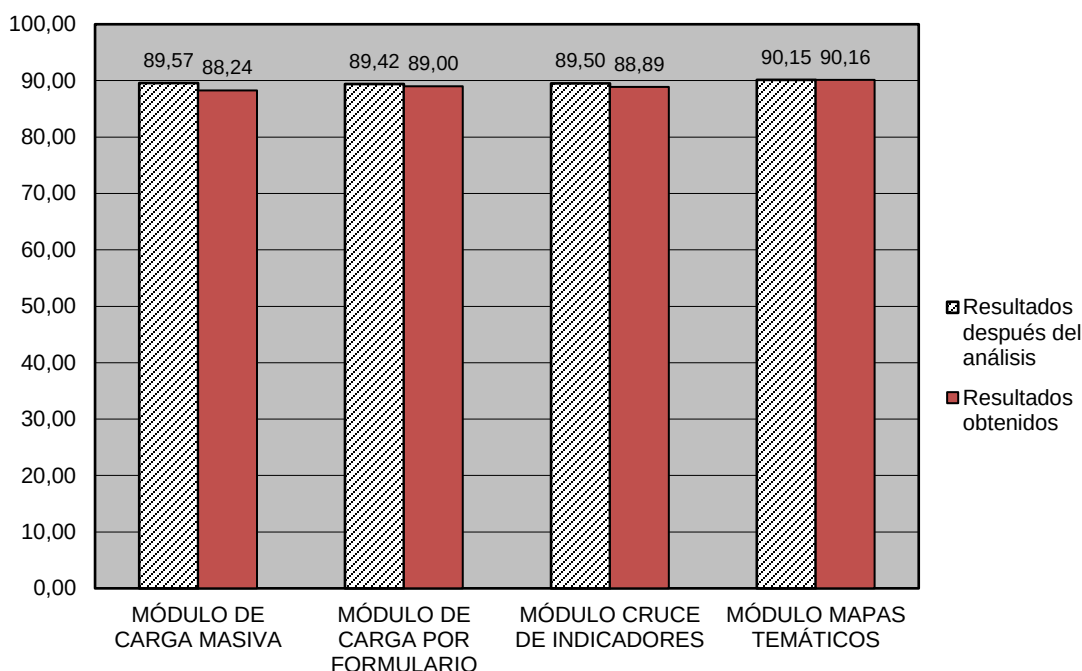
Consolidado producto SIGEODEP:

Tabla 25: Consolidado producto software SIGEODEP

Módulos	Calificación	Calificación ponderada	Calificación con análisis	Calificación ponderada con análisis
Carga Masiva	88,24	22,06	89,57	22,39
Carga por Formulario	89,00	22,25	89,42	22,36
Cruce de Indicadores	88,89	22,22	89,50	22,38
Mapas Temáticos	90,16	22,54	90,15	22,54
TOTAL		89,07		89,67

La ponderación de los módulos obtenido es de 25%, es decir, cada módulo tiene el mismo grado de importancia, debido a que cada módulo es indispensable para realizar las diferentes funciones de la organización.

Ilustración 25: Consolidado producto SIGODEP



Análisis de resultados de la evaluación del software:

- El proceso de evaluación de software propuesto en la norma ISO 9126 tiene un lugar importante en el ciclo de vida del software, debido a esto es necesario darle mayor difusión a este proceso y considerarlo como un concepto esencial al desarrollar software y aplicarlo para mejorar la calidad de software.
- Los niveles de calidad de un producto software deberían ser tenidos mucho más en cuenta por los desarrolladores porque este proporciona seguridad en los procesos que desarrolla el sistema ya que dan a conocer si el software está realizando lo que el usuario necesita.
- Realizar un proceso de evaluación de la calidad de software a un producto en desarrollo es una práctica que permite el establecimiento de acciones preventivas a fallas que al liberar el producto puedan presentarse.
- En búsqueda de altos niveles de calidad del producto software, La medición del grado de calidad de un producto software solo es útil si se acompaña de un proceso de mejoramiento derivado de los resultados de la evaluación.

- Los resultados de un proceso de calidad son el insumo INGENIERIL más importante para la toma de decisiones respecto a la continuidad de un producto software.
- El software garantiza que las necesidades del usuario son satisfechas en su totalidad y de manera ágil, posee una interfaz gráfica agradable, de fácil comprensión y uso, por lo que se acopla a la organización y a su forma de trabajo de la mejor manera, aumentando la eficiencia del trabajo realizado.
- Se garantiza que el software podrá ser utilizado por un largo lapso de tiempo con las características que tiene actualmente y puede ser mejorado por otros desarrolladores sin mayor dificultad; posee algunas deficiencias en cuanto a las acciones que debe tomar ante una posible falla ya que éstas están designadas a lo que pueda hacer el navegador.
- El software después del proceso de evaluación de calidad arroja que son muy pocas las dificultades que presenta, se puede observar que la mayoría de lo que se propone evaluar en las características fueron tenidas en cuenta en el proceso de desarrollo del software.
- Cada módulo cumple con su objetivo dándole al usuario una forma más fácil de almacenar los datos para su posterior análisis que el tiempo que se necesitaba con el anterior método para el registro de datos.

Recomendaciones generales del software:

- Aunque el software es agradable y cumple con las necesidades del usuario, se recomienda que la información sobre la conexión del sistema con la base de datos se encuentre ubicada en la interfaz, ya que ésta es importante para que el usuario sepa cómo desenvolverse ante una posible desconexión.
- En cuanto a la protección de la información, se recomienda una verificación periódica a la base de datos por parte de los administradores del sistema para que se controle que la base de datos se encuentra sin errores y si está trabajando de la forma más eficiente posible; también se recomienda que el lugar donde esté ubicado el servidor sea un lugar protegido de posibles robos y accidentes, como incendios inundaciones o fallas eléctricas.
- Una recomendación importante es que a la hora de desarrollar un nuevo software se trate de implementar la evaluación de la calidad desde el primer momento del desarrollo esto evitará que al momento de salir el producto a funcionamiento tenga errores que se pudieron evitar durante este ciclo.

- Cada vez que cualquier funcionalidad del software presente un cambio se recomienda llevar una documentación sobre los mismos para saber las nuevas funcionalidades adquiridas como también la corrección de errores que ha sufrido.

4. CONCLUSIONES

Durante el proceso de medición de calidad se puede concluir, que este proceso es fundamental a la hora de aplicar buenas prácticas en el desarrollo de software, puesto que. la medición de calidad ratifica el buen o mal ciclo de desarrollo del software desde la ingeniería de requisitos hasta la prueba y entrega al usuario final.

En búsqueda de altos niveles de calidad de un producto software, la medición del grado de calidad solo es útil si se acompaña de un proceso de mejoramiento derivado de los resultados de la evaluación.

El proceso de medición de calidad estándar fue apoyado por coeficientes estadísticos los cuales permitieron tener una mayor coherencia en las listas de chequeo y sus ítems correspondientes, para tener más precisión a la hora de evaluar las características, sub-características y atributos estipulados en la norma ISO/IEC 9126.

Realizar un proceso de evaluación de la calidad de software a un producto en desarrollo, es una práctica que permite el establecimiento de acciones preventivas, a fallas que se pueden presentar al momento de liberar el producto.

Los resultados obtenidos en cualquier proceso de medición de calidad son sumamente importantes a la hora de tomar decisiones respecto al proceso de desarrollo como también la continuidad del producto software.

El proceso de evaluación de software propuesto en la norma ISO/IEC 9126 tiene un lugar importante en el ciclo de vida del software, debido a esto es necesario darle mayor difusión a este proceso y considerarlo como un concepto esencial al desarrollar software y aplicarlo para mejorar la calidad de software.

Finalizado el proceso de medición de calidad de software a SIGEODEP, de acuerdo a los resultados arrojados durante el proceso, se puede concluir que el software cuenta con niveles de calidad superiores, los cuales ratifican el buen proceso de ingeniería de software que se llevó a cabo al realizarlo, lo cual le garantiza una buena experiencia al usuario final.

Desde la perspectiva de la experiencia adquirida con este proceso de medición de calidad de software, se puede decir que es una excelente práctica antes de iniciar la vida laboral, puesto que en este proceso se aplicaron muchos conocimientos adquiridos durante el desarrollo de la carrera de ingeniería de sistemas.

5. RECOMENDACIONES

Exigir que los diferentes tipos de software adquiridos estén bajo certificación en alguna norma de calidad; desde el punto de vista educativo, enseñar en las universidades los pasos para el desarrollo de software con mayor profundidad, así el proceso de calidad de software se convertiría en una disciplina que los ingenieros de sistemas e ingenieros de software aplicarán a sus trabajos futuros.

Realizar el proceso de calificación de un determinado software sean ingenieros calificados en gestión de calidad o estudiantes con algún tipo de educación en gestión de calidad.

Utilizar herramientas estadísticas que permitan garantizar que las pruebas y preguntas utilizadas sean acordes y evalúen lo que se debe evaluar, es decir que se pueda determinar el nivel de concordancia y consistencia de las diferentes pruebas y preguntas utilizadas.

Verificar que el proceso de calidad se realice a medida que se desarrolla el software, para así poder corregir a tiempo los errores encontrados y garantizar una buena calidad del producto.

Aplicar herramientas y/o técnicas estadísticas para realizar ponderaciones más subjetivas tanto a los módulos de cada software como a las diferentes características de la norma a utilizar, ya que ninguna norma posee guías que oriente este proceso de ponderación.

TRABAJOS FUTUROS

Sistematizar el proceso de evaluación mediante la construcción de una herramienta software, debido a las dificultades obtenidas en este trabajo que se encontraron al utilizar herramientas ofimáticas como lo son Microsoft Excel y Microsoft Word, programas, que a pesar de que sirven para cumplir con los objetivos, no son lo suficientemente eficientes.

Investigar más a fondo las normas ISO/IEC 14598 e ISO/IEC 25000, ya que se podrían adecuar de mejor manera al software evaluado o también se podría utilizar algunas de sus características para combinarlas entre sí para obtener un mejor proceso de evaluación.

En nuestra universidad y en varias empresas municipales, se desarrolla software y no se les aplica el proceso de gestión de calidad, con base en este proyecto, estudiantes de ingeniería de sistemas, podrían realizar la respectiva evaluación a estos productos software.

Realizar la evaluación del software SIGEODEP con una norma que se compare a la norma ISO/IEC 9126 y comparar resultados, para verificar que el software ha sido evaluado exitosamente.

BIBLIOGRAFÍA

CRUZ, M. I. Calidad de Software. Calidad Enfocada al desarrollo de software. ITESCAM. s.f.

F. C.; JARAMILLO, A. M., & L, Adriana. Plan de Calidad para la evaluacion de productos software en uso con base en la norma ISO/IEC 9126. Universidad Eafit. Medellin: Universidad Eafit, 2003.

FERNÁNDEZ CARRASCO, O. M., D. G., & A. B. Un enfoque actual sobre la calidad del software. Bogotá: ACIMED, 1995.

GARCIA, P. Calidad y Mantenimiento en el Desarrollo de Software. Madrid: RA MA, 2002.

IEEE. SWEBOK Software Engineering Body of Knowledge, 2004.

ISO/IEC, I. S. Norma ISO/IEC 9126 Software engineering -- Product quality, 2003.

IT-MENTOR. Pruebas de Software. Obtenido de materias.fi.uba.ar/7548/PruebasSoftware.pdf

PANTALEO, G. Calidad en el Desarrollo de Software. Bogotá: MARCOMBO, 2012.

SOCIETY, I. C. IEEE Standard for Information Technology – System and Software Life Cycle Processes. New York :s n., 2010.