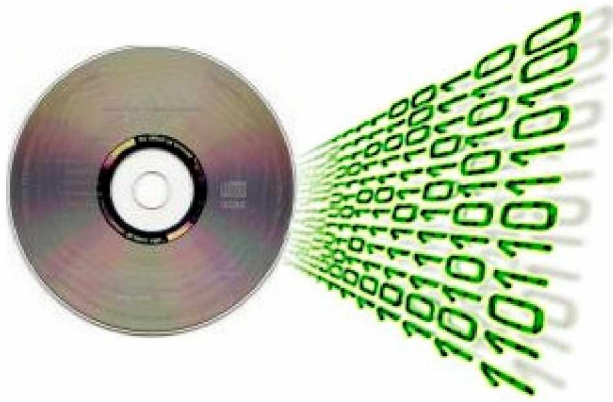


# ***GUÍA TÉCNICA PARA EVALUACIÓN DE SOFTWARE***



**CARLOS ALBERTO LARGO GARCIA  
ERLEDY MARIN MAZO**

# **GUIA TECNICA PARA EVALUACION DE SOFTWARE**

**CARLOS ALBERTO LARGO GARCIA  
ERLEDY MARIN MAZO**

@ Reservados todos los derechos – primera edición 2005

Prohibida su reproducción total o parcial de esta obra, por cualquier medio sin permiso escrito.

## CONTENIDO

CAPITULO	PAG.
INTRODUCCIÓN.	1
OBJETIVOS	2
1. CONCEPTOS BÁSICOS	3
2. NORMA ISO /IEC 9126	4
3. MODELO DE CALIDAD	6
3.1 Modelo de calidad externa e interna	6
3.2 calidad de uso	7
4. MODELO DE CALIDAD EXTERNA E INTERNA Y CALIDAD EN USO	9
4.1 Funcionalidad	10
4.2 Confiabilidad	11
4.3 Usabilidad	13
4.4 Eficiencia	15
4.5 Capacidad de mantenimiento	16
4.6 Portabilidad	17
4.7 Calidad en uso	19
5. MÉTRICAS	23
5.1 conceptos básicos de métricas	23

6. PROCESO DE EVALUACIÓN DE SOFTWARE	24
6.1 estado del software	24
6.2 Identificar el tipo de software	24
6.3 Perfiles de Evaluadores	24
6.4 Especificar objetivos	25
6.5 Aplicar el modelo de calidad	25
6.6 Criterios de evaluación	26
6.7 Seleccionar métricas	26
6.8 Establecer criterios	27
6.9 Tomar medidas	28
6.7 Resultados	28
6.9 Documentación	28
6.10 Seguimiento	28
7. PRACTICA	29
GLOSARIO	
BIBLIOGRAFÍA	

# INTRODUCCIÓN

El avance informático actual es muy alto comparado con lo se tenía en los años 90, al hablar de desarrollo de software se hace más notable, en el hecho por ejemplo de pasar de una programación de código línea a línea, a un método de programación gráfico orientado a objetos donde el desarrollo es mas rápido y atractivo para el cliente.

Pero sin embargo con estas ventajas que se tiene con las nuevas herramientas de desarrollo de software se olvida la calidad del producto que es entregado, no es solamente una calidad gráfica, o la calidad de velocidad en la respuesta, hay que tener en cuenta otras cualidades, para buscar una integralidad al afirmar que el software es de calidad.

Los desarrolladores del software, opinan que el sus productos son los mejores del mercado, pero se han preguntado que opina el cliente?.

El tener un documento que explique los requerimientos para evaluar el software ayuda al desarrollo, compra o auditoría de cualquier aplicación informática del mercado, teniendo en cuenta que hoy en día es muy importante para las empresas privadas o públicas la inversión en este tipo de producto, los cuales verifican la calidad a la hora de entrar a producción, donde se detectan las falencias, reportando allí pérdidas.

Esta guía presenta indicadores de calidad de un software; al momento de la entrega, basados en los estándares de calidad sugeridos la norma ISO/IEC 9126; de la ISO (Organización Internacional de Normalización) y la IEC (Comisión Electrotécnica Internacional).

## **OBJETIVOS**

- Sensibilizar a los lectores con los conceptos necesarios para la evaluación de software
- Orientar a estudiantes y Profesionales del campo informático en los estándares mínimos de calidad para recibir o entregar un desarrollo comercial o libre
- Brindar a las empresas o clientes compradores del software una herramienta para tener en cuenta en sus departamentos de sistemas en el momento que desean adquirir una aplicación informática.
- Aumentar y mantener la calidad en el desarrollo de las aplicaciones realizadas dentro de la dirección de sistemas de una empresa.
- Sensibilizar a los lectores la importancia de la certificación de calidad software.
- Colaborar en el desarrollo de software cumpliendo con los requerimientos del cliente y entregar un producto de calidad.
- Aportar un modelo o instrumento para la evaluación de aplicaciones informáticas en empresas de auditorías.

# 1. CONCEPTOS BASICOS

## QUE ES LA CALIDAD?

*Conjunto de propiedades y características de un producto o servicio, que le confieren aptitud para satisfacer unas necesidades explícitas o implícitas (ISO 8402)*

## QUE ES LA CALIDAD DEL SOFTWARE?

*La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario". (IEEE, Std. 610-1990).*

## QUE ES ISO 9000?

Un Conjunto de normas internacionales que definen requisitos para los llamados sistemas de gestión de calidad. ISO 9000 tiene reconocimiento internacional y se aplica a todas las industrias, incluso al desarrollo de tecnologías de la información.

Los requisitos de la norma están en el documento conocido como ISO 9001:2000.



## 2. RESUMEN NORMA ISO/IEC 9126

Esta norma Internacional fue publicado en 1992, la cual es usada para la evaluación de la calidad de software, llamado *“Information technology – Software product evaluation – Quality characteristics and guidelines for their use”*; o también conocido como ISO 9126 (o ISO/IEC 9126).

Este estándar describe 6 características generales. y son definidas transcribiéndolas de su fuente original así:

**Funcionalidad**, y que textualmente la define: *“A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied set of users”*

**Confiabilidad**, y que textualmente la define: *“A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time”*

**Usabilidad**, y que textualmente la define: *“A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.”*

**Eficiencia**, y que textualmente la define: *“A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions”*

**Mantenibilidad**, y que textualmente la define: *“A set of attributes that bear on the effort needed to make specified modifications”*

**Portabilidad**, y que textualmente la define: *“A set of attributes that bear on the ability of software to be transferred from one environment to another”*

Con estas bases se explica al lector de forma sencilla, para que pueda aplicar los procesos para tener en cuenta a la hora de evaluar un software, los temas que contiene la guía son los siguientes:

Modelo de Calidad

Indicadores de Calidad

Métricas

Proceso de evaluación

Practica Instrumento de evaluación

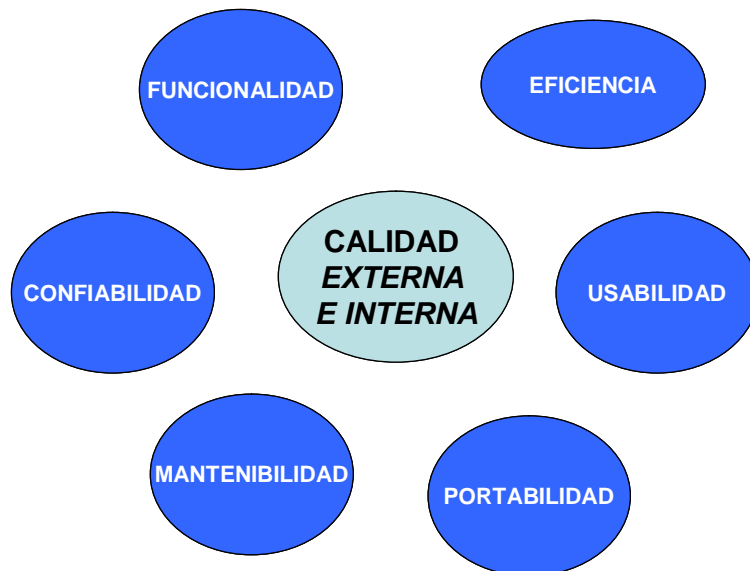
### 3. MODELO DE CALIDAD

La ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría de software.

Los modelos de calidad para el software se describen así:

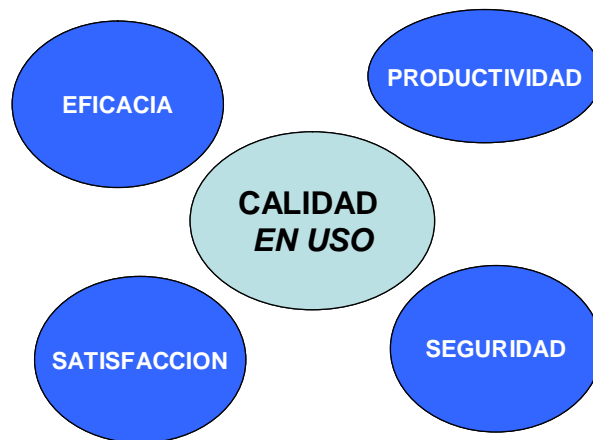
#### 3.1 Calidad interna y externa

Especifica 6 características para calidad interna y externa, las cuales, están subdivididas. Estas divisiones se manifiestan externamente cuando el software es usado como parte de un sistema Informático, y son el resultado de atributos internos de software.



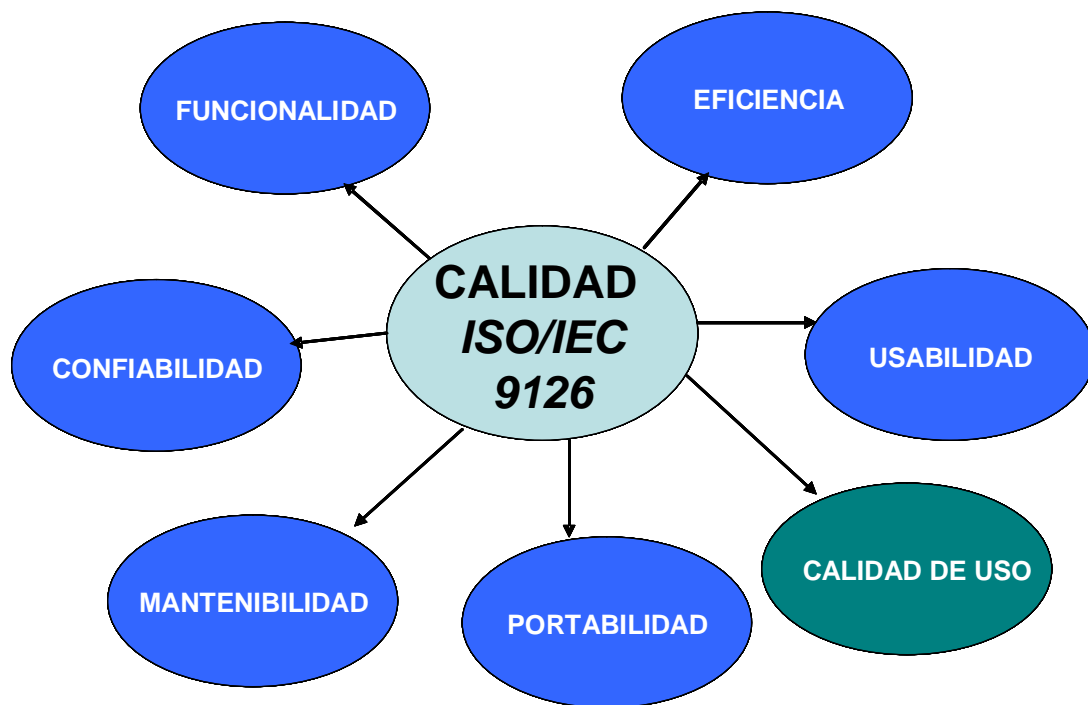
### 3.2 calidad en uso

Calidad en uso es el efecto combinado para el usuario final de las 6 características de la calidad interna y externa del software. Especifica 4 características para la calidad en uso.



Al unir la calidad interna y externa con la calidad en uso se define un modelo de evaluación mas completo, se puede pensar que la usabilidad del modelo de calidad externa e interna pueda ser igual al modelo de calidad en uso, pero no, la usabilidad es la forma como los profesionales interpretan o asimilan la funcionabilidad del software y la calidad en uso se puede asumir como la forma que lo asimila o maneja el usuario final.

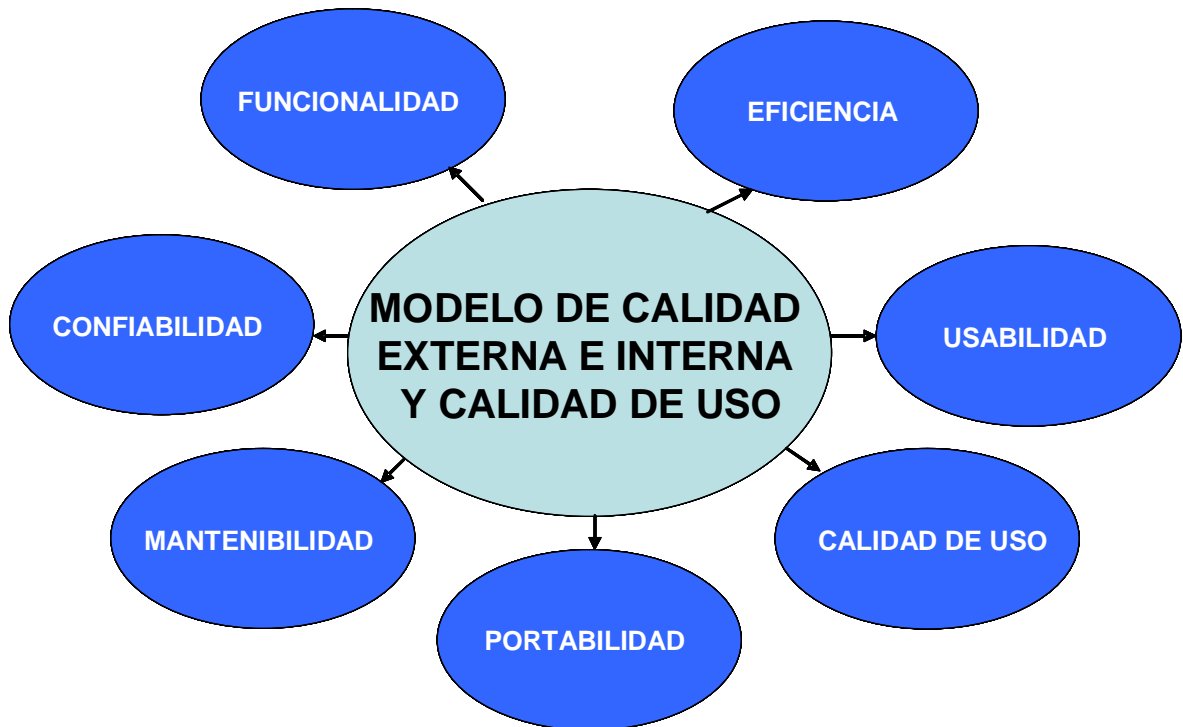
Si se unen los dos modelos, podríamos definir que los seis indicadores del primer modelo tienen sus atributos y el modelo de calidad en uso sus 4 indicadores pasarían a hacer sus atributos, mirándolo gráficamente quedaría así:



La calidad del software se evalúa teniendo en cuenta la etapa del desarrollo, se deben fijar las metas de la calidad tanto para el software final como para desarrollos incompletos y tener en cuenta que es imposible que las metas y criterios sean iguales para un software pequeño y un gran software empresarial.

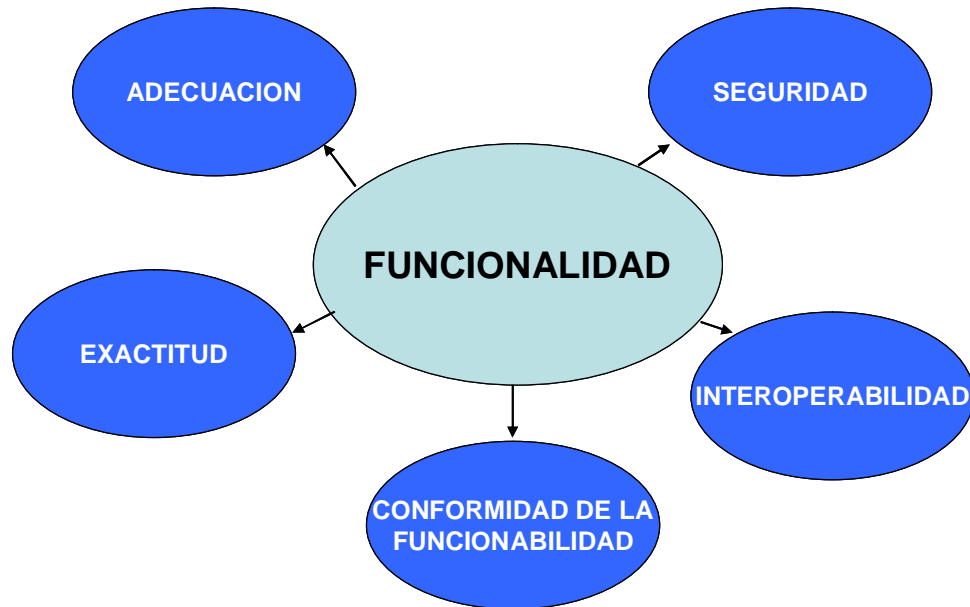
## 4. MODELO DE CALIDAD EXTERNA E INTERNA Y CALIDAD EN USO

Se establecen categorías para las cualidades de la calidad externa e interna y calidad en uso del software, teniendo en cuenta estos 7 indicadores (funcionalidad, confiabilidad, utilidad, eficiencia, capacidad de mantenimiento, portabilidad y calidad en uso), que se subdividen a su vez en en varios indicadores; estas se pueden medir por métrica interna o externa.



Las definiciones se dan para cada característica y subcaracterística de calidad del software que influye en la calidad. Para cada característica y subcaracterística, la capacidad del software es determinada por un conjunto de atributos internos que pueden ser medidos. Las características y sub características se pueden medir externamente por la capacidad del sistema que contiene el software.

## 4.1 Funcionalidad



Funcionalidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas.

Ejemplo, lo que hace el software para satisfacer las necesidades sin tener en cuenta de cómo lo hace y cuando.

La funcionalidad se divide en 5 criterios:

### **Adecuación**

La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.

### **Exactitud**

La capacidad del software para hacer procesos y entregar los resultados

solicitados con precisión o de forma esperada.

### **Interoperabilidad**

La capacidad del software de interactuar con uno o más sistemas específicos.

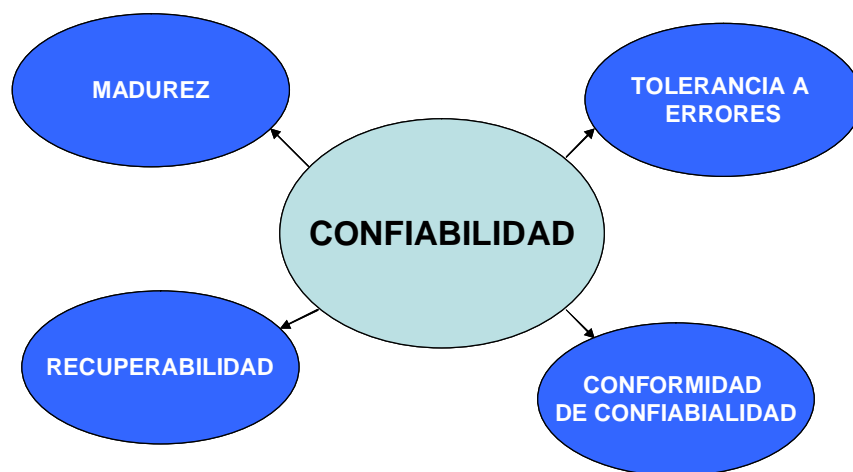
### **Seguridad**

La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados

### **Conformidad de la funcionalidad**

La capacidad del software de cumplir los estándares referentes a la funcionalidad.

## **4.2 Confiabilidad**





La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizado en condiciones específicas.

La definición de confiabilidad en la **ISO/IEC 2382-14:1997** es "la habilidad de la unidad funcional de realizar una función requerida...".

En este caso a la confiabilidad se amplía a sostener un nivel especificado de funcionamiento y no una función requerida.

La confiabilidad se divide en 4 criterios:

### **Madurez**

La capacidad que tiene el software para evitar fallas cuando encuentra errores.

Ejemplo, la forma como el software advierte al usuario cuando realiza operaciones en la unidad de diskett vacía, o cuando no encuentra espacio suficiente el disco duro donde está almacenando los datos.

### **Tolerancia a errores**

La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.

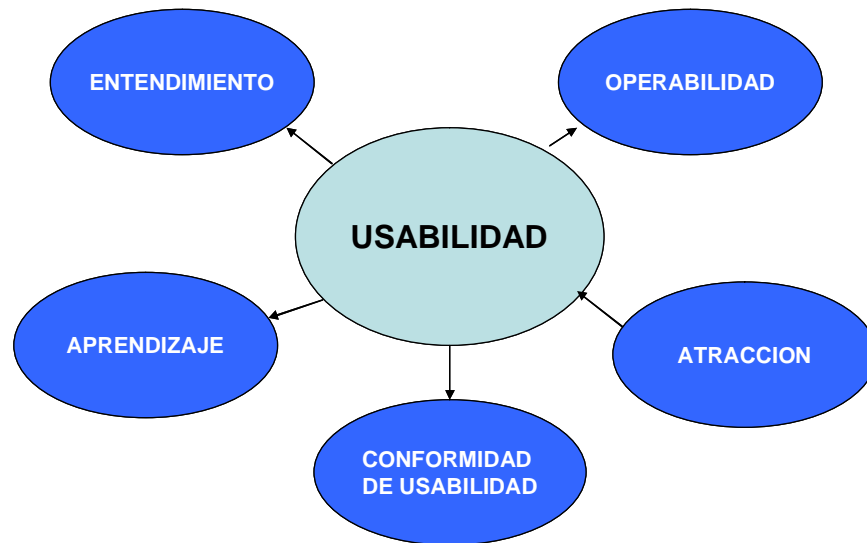
### **Recuperabilidad**

La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.

### **Conformidad de la fiabilidad**

La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

### 4.3 Usabilidad



La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva.

Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad.

La usabilidad esta determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido.

#### Entendimiento

La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación.

En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.

### **Aprendizaje**

La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.

### **Operabilidad**

La manera como el software permite al usuario operarlo y controlarlo.

### **Atracción**

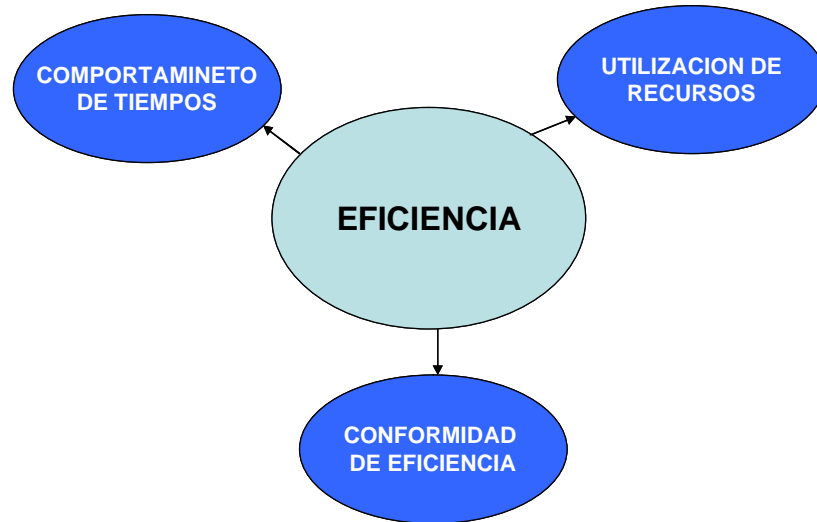
La presentación del software debe ser atractivo al usuario.

Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.

### **Conformidad de uso**

La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

## 4.4 Eficiencia



La eficiencia del software es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas.

Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo, entre otros.

### **Comportamiento de tiempos**

Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas.

Ejemplo, ejecutar el procedimiento mas complejo del software y esperar su tiempo de respuesta, realizar la misma función pero con mas cantidad de registros.

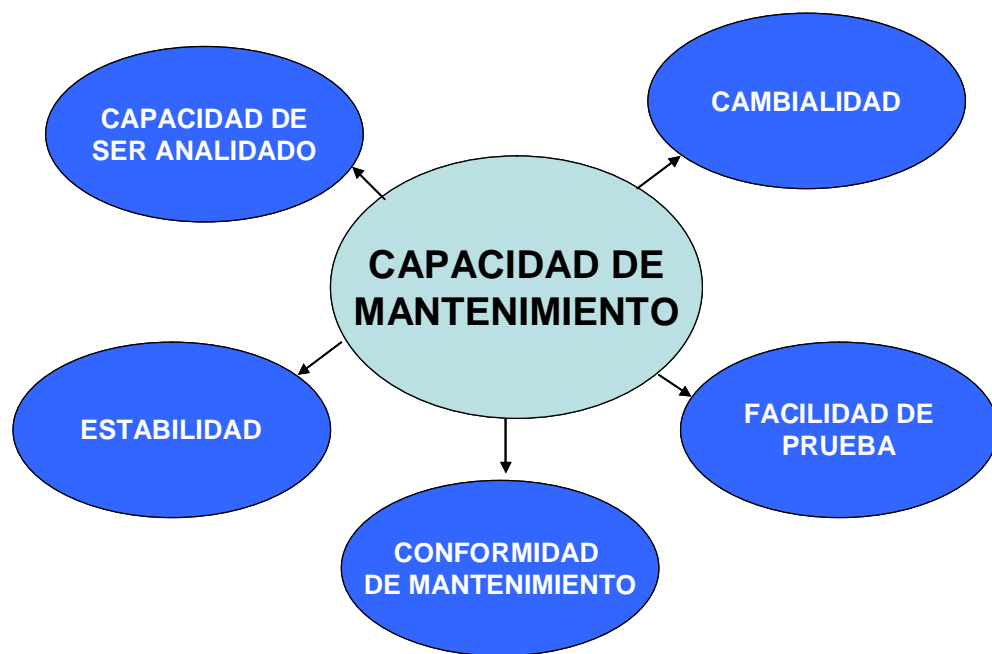
### **Utilización de recursos**

La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas. Ejemplo, los recursos humanos, el hardware, dispositivos externos.

### **Conformidad de eficiencia**

La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.

## **4.5 Capacidad de mantenimiento**



La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.

### **Capacidad de ser analizado**

La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.

### **Cambiabilidad**

La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.

### **Estabilidad**

La forma como el software evita efectos inesperados para modificaciones del mismo.

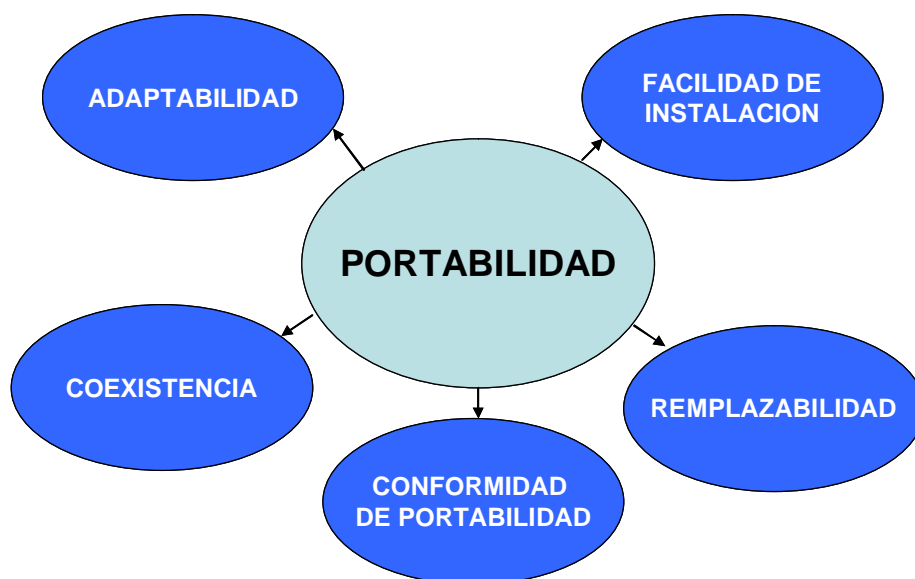
### **Facilidad de prueba**

La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.

### **Conformidad de facilidad de mantenimiento**

La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

## **4.6 Portabilidad**



La capacidad que tiene el software para ser trasladado de un entorno a otro.

### **Adaptabilidad**

Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio.

Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).

### **Facilidad de instalación**

La facilidad del software para ser instalado en un entorno específico o por el usuario final.

### **Coexistencia**

La capacidad que tiene el software para coexistir con otro o varios software, la forma de compartir recursos comunes con otro software o dispositivo.

### **Reemplazabilidad**

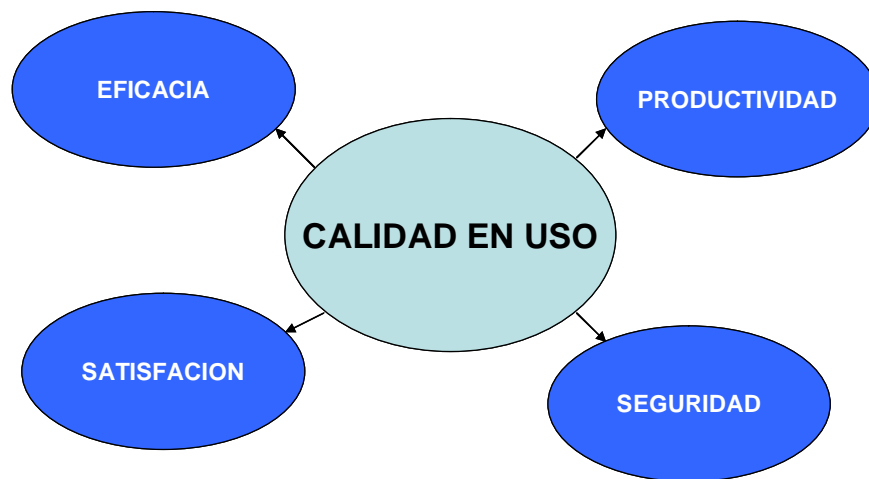
La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo.

Ejemplo, la reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.

### **Conformidad de portabilidad**

La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

## **4.7 Calidad en uso**



Calidad en uso es la calidad del software que el usuario final refleja, la forma como el usuario final logra realizar los procesos con satisfacción, eficiencia y exactitud.

La calidad en uso debe asegurar la prueba o revisión de todas las opciones que el usuario trabaja diariamente y los procesos que realiza esporádicamente relacionados con el mismo software.

### **Eficacia**

La capacidad del software para permitir a los usuarios finales realizar los procesos con exactitud e integridad.



## **Productividad**

La forma como el software permite a los usuarios emplear cantidades apropiadas de recursos, en relación a la eficacia lograda en un contexto específico de uso.

Para una empresa es muy importante que el software no afecte la productividad del empleado.

## **Seguridad**

Se refiere al que el Software no tenga niveles de riesgo para causar daño a las personas, instituciones, software, propiedad intelectual o entorno.

Los riesgos son normalmente el resultado de deficiencias en la funcionalidad (Incluyendo seguridad), fiabilidad, usabilidad o facilidad de mantenimiento.

## **Satisfacción**

La satisfacción es la respuesta del usuario a la interacción con el software, e incluye las actitudes hacia el uso del mismo.

A continuación se describe un cuadro donde podemos resumir las características y cada uno de sus atributos, este cuadro le ayudara a visualizar el proceso de evaluación.

**TABLA - RESUMEN**  
**PREGUNTAS GENERALES DE LAS CARACTERÍSTICAS Y**  
**SUBCARATERÍSTICAS.**

CARACTERÍSTICA	PREGUNTA	SUBCARATERÍSTICA	PREGUNTA
FUNCIONALIDAD	¿Las funciones y Propiedades satisfacen las necesidades Explícitas e implícitas; esto es, el qué?	ADECUACIÓN	¿Tiene el conjunto de funciones apropiadas para las tareas especificadas?
		EXACTITUD	¿Hace lo que fue acordado en forma esperada y correcta?
		INTEROPERABILIDAD	¿Interactúa con otros sistemas especificados?
		CONFORMIDAD	¿Está de acuerdo con las leyes o normas y estándares, u otras prescripciones?
CONFIABILIDAD	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?	MADUREZ	¿Con qué frecuencia presenta fallas por defectos o errores?
		TOLERANCIA A ERRORES	¿Si suceden fallas, como se comporta en cuanto a la performance especificada?
		RECUPERABILIDAD	¿Es capaz de recuperar datos en caso de fallas?
		ENTENDIMIENTO	¿Es fácil de entender y reconocer la estructura y la lógica y su aplicabilidad?
USABILIDAD	¿El software, es fácil de usar y de aprender?	APRENDIZAJE	¿Es fácil de aprender a usar?
		OPERABILIDAD	¿Es fácil de operar y controlar?
		ATRACCIÓN	¿Es atractivo el diseño del software?
		COMPORTAMINETO DE TIEMPOS	¿Cuál es el tiempo de respuesta y performance en la ejecución de la función?
EFICIENCIA	¿Es rápido y minimalista en cuanto a uso de recursos, bajo ciertas condiciones?	UTILIZACION DE RECURSOS	¿Cuántos recursos usa y durante cuánto tiempo?
		CAPACIDAD DE SER ANALAZADO	¿Es fácil diagnosticar una falla o identificar partes a modificar?
CAPACIDAD DE MANTENIMINETO	¿Es fácil de modificar y testear?	CAMBIALIDAD	¿Es fácil de modificar y adaptar?

<b>PORTABILIDAD</b> ¿Es fácil de transferir de un ambiente a otro?	ESTABILIDAD	¿Hay riesgos o efectos inesperados cuando se realizan cambios?
	FACILIDAD DE PRUEBA	¿Son fáciles de validar las modificaciones?
	ADAPTABILIDAD	¿Es fácil de adaptar a otros entornos con lo provisto?
	FACILIDAD DE INSTALACION	¿Es fácil de instalar en el ambiente especificado?
	REPLAZABILIDAD	¿Es fácil de usarlo en lugar de otro software para ese ambiente?
	COEXISTENCIA	¿Comparte sin dificultad recursos con otro software o dispositivo?
	EFICACIA	¿La eficaz el software cuando el usuario final realiza los procesos?
<b>CALIDAD EN USO</b> ¿Muestra el usuario final aceptación y seguridad del software?	PRODUCTIVIDAD	¿Muestra el usuario final rendimiento en sus tareas cotidianas del proceso específico?
	SEGURIDAD	¿El software tiene niveles de Riesgo que causan daño al usuario final?

Es la capacidad que tiene el software para cumplir con las normas o estándares relacionados con cada uno de los atributos.

## 5. MÉTRICAS

### 5.1 Conceptos básicos de Métricas

La palabra *métrica*, es muy común asociarla con las palabras medición y medida, aunque estas tres son distintas. La *medición* “es el proceso por el cual los números o símbolos son asignados a atributos o entidades en el mundo real tal como son descritos de acuerdo a reglas claramente definidas” [Fenton '91].

El IEEE “*Standard Glossary of Software Engineering Terms*” define como *métrica* como “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” [Len O. Ejiogo '91]

Varios investigadores de métricas han intentado desarrollar una sola métrica que proporcione una medida del software. Las que están documentadas tienen un punto de vista diferente, esto hace difícil obtener un solo valor de estas métricas de calidad.

En las métricas hay que tener en cuenta que la gran documentación se basa en métricas aplicadas a desarrollos realizados por codificación manual, afortunadamente hoy en día se esta trabajando métricas para desarrollo de aplicaciones orientadas a objetos.

Es importante aclarar que esta guía plantea métricas solo para la evaluación de productos finales y realizados desde la visión del cliente.

Esto lleva a plantear unas métricas sencillas pero significativas a la hora de aplicarlas en el proceso de evaluación.



## **6. PROCESO DE EVALUACIÓN DE SOFTWARE**

El proceso de evaluación de software se inicia con una visión cualitativa y deriva en una evaluación cuantitativa, siendo todo el proceso documentado y cumpliendo los siguientes pasos:

### **6.1 Estado del Software**

Conocimiento del el estado del software, estableciendo si se trata de un desarrollo sin terminar o un producto terminado para la entrega al cliente.

### **6.2 Identificar el tipo de software**

Especificar el tipo de software a evaluar, si es un sistema operativo, software de seguridad, software de ofimática, lenguaje de programación, base de datos, aplicativo a la medida, entre otros.

### **6.3 Perfiles de Evaluadores**

Teniendo como marco conceptual al estándar ISO [ISO/IEC9126], se consideran tres perfiles de usuario, a un alto nivel de abstracción para desarrollo de software, usuarios finales, desarrolladores, y gerentes.

El estándar afirma que la relativa importancia de las características de calidad (como usabilidad, funcionalidad, confiabilidad, eficiencia, portabilidad, y mantenibilidad y calidad en uso) varían dependiendo del punto de vista considerado y de la critica de los componentes del software a evaluar.

La visión del usuario final, concierne al interés de los mismos en usar el software, como así también su performance, su eficiencia, su facilidad de uso, entre otros

aspectos. Los usuarios finales no están interesados en características internas o de desarrollo del software (sin embargo, atributos internos contribuyen a la calidad de uso).

La visión de calidad del desarrollador debe considerar no sólo los requerimientos del software para la visión del usuario sino también la calidad para los desarrollos intermedios resultantes de las actividades de la fase de desarrollo.

Se debe tener en cuenta que los desarrolladores están preocupados en características de calidad del software como mantenibilidad y portabilidad.

La visión de calidad del gerente es una visión integradora, que incorporar requerimientos de negocio a las características individuales.

Ejemplo, un gerente esta interesado en el equilibrio entre la mejora del software y los costos y tiempos establecidos

## **6.4 Especificar los Objetivos**

Conocer los objetivos tanto generales como específicos del software

## **6.5 Aplicar el modelo de calidad**

Elaborar un instrumento o formato donde aplique el modelo de calidad externo e interno y calidad de uso.

Si existe un comité o conjunto de personas encargadas de la evaluacion, el instrumento debe ser aprobado por los participantes.

## 6.6 Criteríos de la evaluacion

Los criterios parten de los 7 indicadores principales los cuales fueron socializados anteriormente.

Los criterios para evaluar el software se dividen en dos grandes bloques: uno dedicado a criterios que son aplicables a cualquier tipo de software (criterios generales), y otro conjunto compuesto por criterios adaptables al grupo de software evaluados (criterios específicos). En este caso se definen los criterios de la evaluación según el tipo de software, para el cual debe conformar un equipo evaluador, este ejercicio ayuda a definir que opciones se deben evaluar con más detalle y valor.

PRINCIPALES TIPOS DE SOFTWARE DE NUESTRA REGIÓN CUADRO DE CRITERIOS A TENER EN CUENTA AL EVALUAR UN SOFTWARE			
TIPOS DE SOFTWARE	EJEMPLOS	ORDEN DEL CRITERIO DE EVALUACIÓN	EVALUADORES
FINANCIEROS	CONTABILIDAD, BANCARIOS, CARTERAS, PAGOS, COSTOS NOMINAS, ETC	1. SEGURIDAD 2. TIEMPO DE RESPUESTA 3. EXACTITUD DE LA INFORMACIÓN 4. RECUPERABILIDAD	PERSONAL DE SISTEMAS, CONTADOR O FINANCIERO, AUXILIAR, DIGITADOR
ADMINISTRATIVOS	RECURSOS HUMANOS, ADMINISTRACIÓN DE DOCUMENTOS, HOSPITALARIOS, ETC	1. TIEMPO DE RESPUESTA 2. SEGURIDAD 3. EXACTITUD DE LA INFORMACIÓN 4. RECUPERABILIDAD	PERSONAL DE SISTEMAS, ADMINISTRATIVO, AUXILIAR, DIGITADOR
EDUCATIVOS	MATERIAS ACADÉMICAS, ENCICLOPEDIAS, TUTORES, MANUALES	1. FACILIDAD DE COMPRENSIÓN 2. CALIDAD GRAFICA 3. PORTABILIDAD	PERSONAL DE SISTEMAS, DOCENTE, ALUMNO
A LA MEDIDA	PRODUCCIÓN, RADIO TERAPIA, CONTROL DE MAQUINAS, ETC	LOS CRITERIOS O INDICADORES ESTÁN SUJETOS A LA ACTIVIDAD ESPECÍFICA DEL SOFTWARE	PERSONAL DE SISTEMAS, PERSONAL QUE CONOZCA EL PROCESO MANUAL O AUTOMÁTICO, CLIENTE

## 6.7 Seleccionar métricas

La selección de métricas se obtiene a partir de los indicadores especificados en el modelo.



### **Niveles o escalas**

- A cada métrica seleccionada le asigna un puntaje máximo de referencia.
  - La suma de los puntajes máximos de todas las métricas debe ser igual o aproximado a 100 puntos.
  - El personal que participa en la evaluación debe establecer niveles de calificación cualitativa con base a los puntajes, por ejemplo:
    - De 0 a 1 Inaceptable.
    - De 2 a 3 mínimo aceptable
    - Mas de 3 Aceptable o satisfactorio
- Otro ejemplo de calificación cualitativa puede ser:
- Deficiente
  - Insuficiente
  - Aceptable
  - Sobresaliente
  - Excelente
- Se permite usar números enteros o hasta con un decimal de aproximación.
  - Definir por cada métrica, un puntaje mínimo de aprobación, y al final de de la evaluación, dependiendo del puntaje si es mayor o menor a lo propuesto, considerar si el software cumple o no cumple con los objetivos propuestos.

## **6.8 Establecer criterios**

Las persona que participa en el proceso de evaluacion debe tener criterios con

respecto al indicador que se está analizando, Es importante tener en cuenta que el criterio debe ajustarse al tipo de software que se va a evaluar.

## **6.9 Tomar medidas**

Para la medición, las métricas seleccionadas se aplican al software. Los resultados son valores expresados en las escalas de las métricas, definidos previamente.

## **6.10 Resultados**

El proceso de evaluación genera un cuadro de resultados por cada uno de los principales indicadores y el total final de resultado.

## **6.11 Documentación**

El proceso de evaluación se documenta, indicando la fecha, empresa, los cargos, nombres y apellidos, dependencia de las personas que participan en el proceso de evaluación, especificando las etapas en las que participaron.

## **6.12 Seguimiento**

Si el resultado de la evaluación tiene observaciones o indicadores de calidad bajos, y el personal que lo evalúa permite realizar la corrección, se programa otra evaluación donde se verifique que el proceso mejora, el tiempo que se estime debe influir en los criterios de la próxima evaluación.

## 7. PRÁCTICA

Teniendo en cuenta este documento se desarrollo un instrumento que recopila la información de la guía y la lleva al ejercicio de la evaluación.

Este instrumento esta desarrollado en formato Excel, permitiendo a los lectores de la guía aplicar la evaluación en cualquier ambiente que permita trabajar una hoja de cálculo, la plantilla se puede solicitar al correo electrónico [carloslargo@puntoexe.com.co](mailto:carloslargo@puntoexe.com.co) o en la pagina Web [www.puntoexe.com.co](http://www.puntoexe.com.co)

Consta de 10 hojas diseñadas para que el usuario solo pueda ingresar los datos relacionados con las preguntas así:

### HOJA 1

#### PORTADA

En la portada se documenta toda la información relacionada con el software, empresa y personal que participa en la evaluación:

**Fecha:** Fecha de e la evaluación

**Ciudad:** Ciudad donde se realiza el proceso.

**Empresa:** Nombre o razón social de la empresa donde se realiza el proceso.

**Teléfono:** Teléfono de la empresa.

**Nombre del Software:** Nombre comercial del software que se va a evaluar.

**Objetivos generales del Software:** Se describen los objetivos generales de la aplicación.

**Objetivos Específicos del software:** Se detalla los objetivos específicos.

**Participantes:** Se relaciona el cargo, nombre y la firma de las personas que

participaron en la evaluación.

Es importante definir los objetivos tanto generales como específicos antes de la evaluación y tenerlos en cuenta al momento de aplicar criterios de la evaluación. Los datos y la firma son de mucha importancia para la documentación, las personas deben cumplir un perfil relacionado con el tipo de software

## **HOJA 2**

### **PARAMETROS**

La hoja de parámetros nos muestra los 7 indicadores que se van a evaluar del software, con su respectiva descripción, número de preguntas y porcentaje que influirá en el resultado final de la evaluación. Este porcentaje puede ser modificado según el tipo de software, ejemplo si para un software bancario es más importante la seguridad que la usabilidad, entonces en la seguridad puedo aumentar el porcentaje y disminuirlo en usabilidad. El porcentaje total debe dar exactamente en 100%

## **HOJA 3 A LA HOJA 9**

En estas hojas se detalla cada uno de los 7 indicadores propuestos por la guía para ser evaluados con sus respectivas preguntas.

Las hojas constan de las siguientes características:

1. Número y nombre del indicador principal
2. Nombre, descripción, valor y observación de cada uno de los atributos que se evalúan. El campo valor y observación son los que se deben llenar en el momento de evaluación de dicho atributo
3. Total de punto del máximo posible que se pueda obtener del indicador

principal.

4. Porcentaje total del indicador evaluado.
5. Criterios de evaluación que se deben tener en cuenta.

Los valores de las métricas cuantitativas permitidos para la evaluación están en una escala de 0 a 3 indicado 0 el valor menor y 3 el valor máximo de favorabilidad del resultado.

## **HOJA 10**

### **RESULTADOS**

En esta hoja se muestra el consolidado, resultado de los 7 indicadores propuestos así:

Nombre y descripción del indicador

Valor final del máximo posible

Porcentaje del indicador del máximo posible

Porcentaje global

Total del puntaje de 99 puntos posibles

Porcentaje final del 100% posible

El porcentaje final se compara con la escala definitiva donde se indica el estado de calidad del software

## GLOSARIO

**Atributo:** Una característica física o abstracta medible de una entidad. Los atributos pueden ser internos o externos.

**Calidad:** Son todas las características de una entidad que forman parte de su habilidad para satisfacer las necesidades propias e implícitas.

**Calidad externa:** La extensión para la cual un producto satisface necesidades explícitas e implícitas cuando es usado bajo condiciones específicas.

**Calidad interna:** Es la totalidad de atributos del producto que determinan su habilidad para satisfacer las necesidades propias e implícitas bajo condiciones específicas.

**Calificación:** La acción de evaluar el valor medido al nivel de calificación adecuado. .

**Defecto:** Un paso, proceso o definición de dato incorrecto en un programa de computadora.

**Desarrollador:** Una organización que realiza actividades de desarrollo (incluyendo análisis de los requisitos, diseño y pruebas de aceptación) durante el proceso del ciclo de vida del software.

**Escala:** Un conjunto de valores con propiedades definidas.

**Falla:** La terminación de la capacidad de un producto de realizar una función requerida o su incapacidad para realizarla dentro de límites previamente especificados

**Indicador:** Una medida que se puede utilizar para estimar o para predecir otra medida.

**Medición:** Actividad que usa la definición de la métrica para producir el valor de una medida.

**Medida:** Número o categoría asignada a un atributo de una entidad mediante una medición.

**Medida directa:** Una medida de un atributo que no depende de la medida de ningún otro atributo.

**Métrica:** Es un método definido de valoración y su escala de valoración.

**Medida externa:** Una medida indirecta de un producto derivada de las medidas del comportamiento del sistema del que es parte.

**Modelo cualitativo:** Es una serie de características y la relación entre las mismas, que conforman la base de los requerimientos cualitativos específicos y la valoración cualitativa.

**Módulo de evaluación:** Un paquete de tecnología de evaluación para una característica o sub característica de calidad de un software específico.

**Necesidades implícitas:** Necesidades que pueden no haber sido especificadas

pero que son necesidades reales cuando la entidad es usada en condiciones particulares.

**Nivel de calificación:** Un punto en la escala ordinal que es utilizado para categorizar una escala de medida.

**Servicio:** Es una organización que presta servicios de mantenimiento.

**Sistema:** Una composición integrada que consiste en uno o más procesos, hardware, software, instalaciones y personas, que proveen una capacidad para satisfacer una necesidad establecida o un objetivo.

**Software:** Todo o parte de los programas, procedimientos, reglas y documentación asociada a un sistema de procesamiento de información.

**Usuario:** Un individuo que utiliza el producto de software para realizar una función específica.

**Valoración:** Emplear una métrica para asignar uno de los valores de una escala (el mismo que puede ser un número o categoría) al atributo de una entidad.

**Valoración Cualitativa:** Es una evaluación sistemática del grado o capacidad de una entidad para satisfacer necesidades o requerimientos específicos.

**Validación:** Confirmación por inspección y provisión de evidencia objetiva de que los requerimientos particulares para un uso específico son alcanzados.

**Verificación:** Confirmación por examen y provisión de evidencia objetiva que los requerimientos específicos han sido alcanzados.



## BIBLIOGRAFIA

### Referencias bibliográficas

Norma ISO/IEC TR 9126-3: 2003 - Software engineering -- Product quality --

Norma ISO/IEC 14598-5:1998 - Part 5: Process for evaluators.

Norma [ISO 8402] **ISO 8402:1994** *Quality - Vocabulary*

### Referencias electrónicas

<http://www.monografias.com/trabajos5/call/call.shtml>

<http://www.gestiopolis.com/canales2/gerencia/1/modcalidad.htm>