

Universidad de las Ciencias Informáticas

Facultad 5



Evaluación de la Portabilidad de Componentes Software

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

Autor:

Dayana García Cabrera

Tutor(es):

MSc. Liudmila Reyes Álvarez.

Ing. Yadira Morales Álamo.

Co-tutora:

Ing. Alicia Lissett Fernández

“La Habana, Marzo 2011”



“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Dayana García Cabrera

Autora

MSc. Liudmila Reyes

Tutora

Ing. Yadira Morales

Tutora

Ing. Alicia Lissett Fernández

Co-Tutora

DATOS DE CONTACTOS

Tutoras:

Msc. Liudmila Reyes Álvarez

Graduada en: Universidad de las Ciencias Informáticas

Email: lreyes@uci.cu

Ing. Yadira Morales Álamo

Graduada en: Universidad de las Ciencias Informáticas

Email: yalamo@uci.cu

Co-Tutora:

Ing. Alicia Lissett Fernández

Graduada en: Universidad de las Ciencias Informáticas

Email: alfernandez@uci.cu

AGRADECIMIENTO

“A mi familia le debo lo que soy”.

A mi mamá por ser un ejemplo de sacrificio para mí, por sus consejos y todo lo que ha hecho durante estos 23 años para darme lo mejor.

A mi papá y a mi Midalis que me han apoyado y he podido contar con ellos en todo momento.

A mis abuelos del alma que no tengo palabras para decir lo mucho que los quiero, desde que nací siempre han estado a mi lado dándome todo su amor, cariño y dedicación.

A Rosy y Juani por ser los hermanos más bellos del mundo.

A mi tía Olga, la que más me ha malcriado desde que estoy en preescolar, por ser una madre más.

A mi novio Yaniel por darme todo su amor, ayuda, apoyo incondicional, por formar parte de mi vida durante estos años.

A mis tíos Hirancito y Mary, por su preocupación, ayuda y el cariño que

siempre he recibido de ellos. A mis primas Eliany, Yeily y Yaisi por ser como hermanitas para mí, juntas para siempre.

A mi madrina Doris, por estar ahí ayudándome y dándome su cariño a pesar de la distancia, no tengo forma de agradecer todo lo que ha hecho por mí desde que nací, te quiero mucho mi madrina.

A mis amistades en la Universidad Yeni, Tania, Yanet y Lianet que siempre han estado presentes para apoyarme.

A las muchachitas del apartamento, especialmente a Camila y Yanet que todas ellas han sido mi familia durante todo este tiempo.

A mis tutoras Liudmila y Yadira, especialmente a Liudmila por su ayuda en la realización de esta investigación.

A todos aquellos amigos y familia que de alguna manera me ayudaron durante todos estos años y aunque sus nombres no estén los llevaré siempre en mi corazón.

¡Muchas Gracias!

DEDICATORIA

Este trabajo es una de las cosas más importantes que he realizado en mi vida y como tal se lo dedico a las personas que más quiero en este mundo y gracias a ellos estoy aquí.....

A mi mamá y mi papá por siempre confiar en mí.

A mis abuelos lindos Fina y Julio porque han sido como unos padres para mí, a ellos le debo la vida.

A mi familia, por ser la mejor del mundo...

¡Los quiero!

Resumen

En la actualidad el desarrollo de software implica una gama de actividades en las que la perfección es la característica más compleja de alcanzar. A pesar del desarrollo que ha logrado la Industria del Software en los últimos años, actualmente son muchos los proyectos que fracasan en el mercado, entre las causas fundamentales se considera la inadecuada evaluación de la calidad.

La presente investigación propone un procedimiento para evaluar la portabilidad de componentes software que se desarrollan en el Centro de Desarrollo Informática Industrial (CEDIN). La portabilidad es una de las características fundamentales de la calidad, se considera un atributo deseable para la mayoría de los productos software. Para la realización de la investigación se utilizan como guías las Normas ISO/IEC 9126 Calidad de los productos y la ISO/IEC 14598 Evaluación de los productos de software.

El procedimiento está compuesto por 4 etapas, estas son: Descripción del componente que se evaluará, Proceso de Medición, Ejecución de la evaluación y Procesar la evaluación de la Portabilidad. Cada una de estas tiene definidas un grupo de actividades para darle cumplimiento al procedimiento.

La validación de la solución propuesta se hace en dos componentes software que utiliza el proyecto Energía UCI, estos componentes son reutilizables y pertenecen al SCADA-ux.

Palabras Clave

Calidad, componentes software, etapas, evaluación, portabilidad

ÍNDICE

Introducción.....	1
CAPÍTULO 1: Fundamentación Teórica	5
Introducción	5
1.1 Calidad de software	5
1.1.2 Características de la calidad.....	6
1.2 Portabilidad de software.....	8
1.2.1 Definiciones de Portabilidad.....	9
1.2.2 Atributos o sub-características de la Portabilidad.....	9
1.3 Evaluación de la calidad: Su Importancia	11
1.3.1 Evaluación de la Portabilidad.....	13
1.4 Portabilidad en la práctica.....	15
1.5 Desarrollo de Software Basado en Componentes (DSBC).....	18
1.5.1 Componentes Software	18
1.5.2 Tipos de Componentes.....	18
1.6 Métricas de calidad de software.....	20
1.7 Procedimiento	21
1.8 Normas de software: ISO/IEC 9126 y 14598	21
1.8.1 Estándar	21
1.8.2 Instituciones más importantes para el desarrollo de estándares.....	21
1.8.3 Norma ISO/IEC 9126.....	21
1.8.4 Norma ISO/IEC 14598.....	22
Conclusiones Parciales	23
CAPÍTULO 2: Procedimiento para evaluar la portabilidad de Componentes Software	

Introducción	24
2.1 Nombre del procedimiento	25
2.2 Objetivos.....	25
2.3 Alcance	25
2.4 Referencias.....	25
2.5 Responsables.....	25
2.6 Términos y definiciones	25
2.7 Normas generales	26
2.8 Flujo del Procedimiento.....	27
2.9 Descripción del Procedimiento.....	28
Etapa 1: Descripción del componente que se evaluará.....	28
Etapa 2: Proceso de Medición	35
Etapa 3: Ejecución de la evaluación	43
Etapa 4: Procesar la evaluación de la Portabilidad	45
Conclusiones Parciales	48
CAPÍTULO 3: Validación del Procedimiento Propuesto.....	49
Introducción	49
3.1 Descripción del Procedimiento.....	49
3.1.1 Componente HMI	50
3.1.2 Componente Seguridad.....	54
3.2 Comparación de Resultados	57
Conclusiones Parciales	58
Conclusiones Generales	59
Recomendaciones	60

Referencias Bibliográficas	61
Bibliografía.....	63
Anexos.....	64

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Atributos de la Calidad de Software según la Norma ISO 9126.	6
Ilustración 2: Atributos específicos de la Portabilidad.....	10
Ilustración 3: Proceso de evaluación de la Norma ISO/IEC 14598.....	12
Ilustración 4: Evaluador de Portabilidad de Datos.....	16
Ilustración 5: Evaluador de Portabilidad del código fuente	17
Ilustración 6: Etapas del Procedimiento	24
Ilustración 7: Flujo central del Procedimiento.....	27
Ilustración 8: Actividades de la Etapa 1.....	28
Ilustración 9: Actividades de la Etapa 2.....	35
Ilustración 10: Actividades de la Etapa 3.....	44
Ilustración 11: Actividades de la Etapa 4.....	45
Ilustración 12: Resultados del componente HMI	57
Ilustración 13: Resultados del componente Seguridad.....	57
Ilustración 14: Por ciento de portabilidad de los componentes HMI y Seguridad.....	58
Ilustración 15: Vista Edición del HMI.....	68
Ilustración 16: Vista Ejecución del HMI	68

ÍNDICE DE TABLAS

Tabla 2.1: Sub-Características de la Portabilidad en las fases del desarrollo de software según los roles.....	33
Tabla 2.2: Especificación de métricas	35
Tabla 2.3: Métricas para evaluar la Portabilidad.....	37
Tabla 2.4: Nivel de Portabilidad.....	43
Tabla 2.5: Ejecución de la evaluación	44

Tabla 2.6: Posibles No Conformidades y Acciones Correctivas.....	47
Tabla. 3.1: Selección de métricas que se van a utilizar para realizar la evaluación	50
Tabla. 3.2: Componente HMI.....	53
Tabla. 3.3: Componente Seguridad.....	56

Introducción

A lo largo de la historia las herramientas de trabajo se crean y se desarrollan con el fin de mejorar y perfeccionar las condiciones de vida. “Al conjunto de condiciones, que contribuyen a hacer agradable y valiosa la vida, se le define calidad “[1]. El término calidad ha evolucionado con el tiempo lo que trae consigo múltiples interpretaciones, todo depende desde el punto de vista que la persona lo vea.

En la actualidad la producción de software se ha convertido en uno de los pilares más importantes para el desarrollo mundial. Uno de los grandes problemas que se afrontan actualmente en la esfera del desarrollo de soluciones informáticas es la calidad del software, tema preocupante desde la década del 70 hasta nuestros días, donde se han hecho grandes investigaciones que han dado lugar a la siguiente interrogante:

¿Cómo evaluar la calidad del software?

La calidad del software se diferencia de la calidad de otros productos de fabricación industrial, ya que el software tiene ciertas características especiales que lo diferencian de cualquier otro producto creado por el hombre.

Una de las características de la calidad de un software es la portabilidad, que se considera uno de los elementos claves que deben tenerse en cuenta para garantizar una migración exitosa de los mismos.

Hoy día una de las tecnologías más importantes que juegan un papel fundamental en la producción de software con calidad es el “Desarrollo de Software Basado en Componentes (DSBC)”. La misma busca, dentro de otros objetivos, reducir el tiempo de trabajo, el esfuerzo que requiere implementar una aplicación y los costos del proyecto, y de esta forma incrementar el nivel de productividad de los grupos desarrolladores y minimizar los riesgos globales sin incurrir en gastos exorbitantes.

El desarrollo de la economía cubana está vinculado en los momentos actuales a los avances relacionados con las nuevas tecnologías de la información y las comunicaciones. Cuba, a pesar de ser un país subdesarrollado y bloqueado, ha centrado gran parte de sus fuerzas económicas en la producción de software, una actividad cada vez más

demandada donde su objetivo fundamental es buscar las vías idóneas para el perfeccionamiento y la mejora continua.

La Universidad de las Ciencias Informáticas (UCI) es un paso adelante para el comienzo de un mundo de avance en el que la tecnología prevalece para ofrecer beneficios económicos que favorecen al país. La Universidad está conformada por facultades, estas a su vez están divididas en centros para una mejor organización de la información.

En la facultad 5 se encuentra el Centro de Desarrollo Informática Industrial (CEDIN) en el que se desarrollan Software Basado en Componentes, a estos componentes se les hace una indebida evaluación de la calidad, debido a que las métricas¹ que allí se utilizan son generales y se limitan a proporcionar información específica que pueda evaluar la característica portabilidad. Lo que trae consigo la liberación de los componentes sin una correcta evaluación de esta característica, pudiendo ocurrir en algunos casos los siguientes problemas:

- Poca capacidad de algunos componentes de ser adaptados e instalados en plataformas específicas, aplicando solamente acciones que tengan suministradas el software.
- Escasa coexistencia con otros componentes en la misma plataforma.
- Mínima posibilidad de utilizar unos componentes en lugares de otros para los mismos fines y el mismo ambiente².

Los problemas antes mencionados son aspectos negativos para los productos que se desarrollan en el CEDIN, debido a que pueden conllevar a que no se cumplan las expectativas de un cliente cualesquiera y esto implicaría una mala imagen para el centro y la Universidad.

Una vez descrita la situación problemática surge el siguiente **problema científico**:

¹ Métrica: Medida o conjunto de medidas destinadas a conocer o estimar el tamaño u otra característica de un software.

² Ambiente: Se refiere al ambiente de software, hardware u organizacional.

- Deficiencias en la forma de evaluar la Portabilidad de Componentes Software en el Centro de Desarrollo Informática Industrial.

Se define como **objeto de estudio**: Evaluación de la calidad de Componentes Software.

Para darle solución al problema propuesto, se determina como **objetivo**: Elaborar un procedimiento que permita evaluar la Portabilidad de los Componentes Software que se desarrollan en el CEDIN.

Campo de acción: Evaluación de la Portabilidad en el Desarrollo de Software basado en Componentes.

La investigación tiene como **idea a defender**: Si se elabora el procedimiento para evaluar la portabilidad de software se debe contribuir a que todos los componentes que se liberen en el CEDIN tengan la capacidad de coexistir con otros y compartir los mismos recursos en el ambiente que se encuentren; puedan ser adaptados e instalados en plataformas específicas y cuenten con la posibilidad de reemplazar a otros componentes para los mismos fines.

Las **tareas que deben realizarse** para darle cumplimiento al objetivo antes expuesto son:

- Elaboración del marco teórico a través del estudio del estado del arte.
- Descripción de los principales estándares o modelos de calidad con el objetivo de determinar el que más se adecue a la solución del procedimiento.
- Caracterización de las distintas técnicas y conceptos de medición y evaluación de la calidad de software en cuanto a la Portabilidad para un mejor entendimiento de la problemática.
- Análisis de la calidad de los componentes de software para definir como se podrá evaluar la calidad de un componente en cuanto a su Portabilidad en el Centro de Informática Industrial (CEDIN).
- Confección del procedimiento para evaluar la calidad de los componentes en cuanto a la Portabilidad de los mismos en el Centro de Desarrollo Informática Industrial.

- Validación del procedimiento en dos componentes software que se desarrollen en el CEDIN.
- Realización del informe: “Evaluación de la portabilidad de componentes software”.

Se utilizan **métodos teóricos** como el *Analítico-Sintético*, se escoge este método para el análisis de documentos, teorías, entre otros, porque permite la extracción de los elementos más importantes que se relacionan con los procedimientos de evaluación de un componente software. Otro de los métodos teóricos que se elige es el *Análisis Histórico – Lógico*, este se utiliza para verificar teóricamente si existe algún procedimiento en el centro, en el país o a nivel internacional que resuelva la problemática planteada

También se utilizan **métodos empíricos** como la *Observación*, se escoge este método para observar cómo se comporta el procedimiento propuesto para evaluar la calidad de un componente en cuanto a su portabilidad y el método *Estadístico* para el cálculo de las métricas.

La investigación está estructurada por tres capítulos, los cuales abarcan los aspectos esenciales relacionados con el contenido de la investigación. La distribución es la siguiente:

Capítulo 1: Fundamentación Teórica: En este capítulo se hace un estudio de los concepto de calidad del software, se aborda en profundidad acerca de la evaluación de la portabilidad, se hace un estado del arte de técnicas, metodologías o procedimientos que se han realizado anteriormente y un análisis crítico de las normas de calidad ISO 9126 y 14598.

Capítulo 2: Procedimiento para evaluar la portabilidad de componentes software: En este capítulo se define el procedimiento para evaluar la portabilidad de componentes software.

Capítulo 3: Validación del Procedimiento: En este capítulo se valida la solución propuesta de la investigación en dos componente del CEDIN.

CAPÍTULO 1

Fundamentación Teórica

Introducción

El objetivo fundamental de este capítulo es exponer los fundamentos generales que sustentan teóricamente la solución del problema.

Se abordan los conceptos de calidad de software, los atributos de calidad, prestando mayor atención en la característica Portabilidad, se referencian diferentes conceptos del tema y se escoge el utilizado en la investigación, para mayor comprensión se analiza la evaluación de la calidad y en específico la evaluación de la portabilidad, se hace un estudio de la portabilidad en la práctica donde se tratan herramientas que se han realizado para evaluar la misma.

Otro de los temas fundamentales que se aborda es el desarrollo de software basado en componentes, aquí se hace un panorama general para dominio del tema, también se habla de las métricas, especificando las métricas de calidad y ya para finalizar se describen las normas de calidad ISO/IEC 9126 *Calidad de los productos* y la ISO/IEC 14598 *Evaluación de los productos de software*, por ser los documentos rectores que rigen la investigación.

1.1 Calidad de software

La calidad del software es una de las mayores preocupaciones de cualquier productor, al aseguramiento de esa actividad, se empeñan muchos esfuerzos; sin embargo, eso no implica que el software sea perfecto. No existe un proyecto que no tenga entre sus objetivos, producir software de la mejor calidad posible, que cumpla y supere las

expectativas de los clientes; lograr el cumplimiento de ese objetivo, es la tarea más compleja dentro de un equipo de desarrollo.

Para la obtención de un software con calidad se hace necesario el uso de metodologías, procedimientos o estándares para el análisis, diseño, programación y prueba del software, que permitan equilibrar la filosofía de trabajo, en aras de lograr una mayor confiabilidad y facilidad de prueba, y a la vez elevar la productividad, tanto para la labor de desarrollo como para el control de la calidad del software. Hoy día este tema es de gran importancia, las compañías de todo el mundo industrializado reconocen que la calidad del producto se traduce en ahorro de costos y en una mejora general.

1.1.2 Características de la calidad

En el estándar internacional ISO/IEC 9126 se definen seis características de calidad de los productos de software. **Ver ilustración 1**



Ilustración 1: Atributos de la Calidad de Software según la Norma ISO 9126.

Funcionalidad (4)

Es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el software se usa bajo las condiciones especificadas.

Confiabilidad (4)

La capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas.

Usabilidad (4)

Capacidad del producto de software de ser comprendido, aprendido, utilizado y atractivo para el usuario, cuando se utilice bajo las condiciones especificadas.

Eficiencia (4)

Capacidad del producto de software para proporcionar una ejecución o desempeño apropiado, en relación con la cantidad de recursos utilizados, bajo condiciones establecidas.

Mantenibilidad (4)

Capacidad del producto de software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales.

Portabilidad (4)

Capacidad del producto de software de ser transferido de un ambiente a otro.

Las características de la calidad definidas en la norma ISO 9126 antes expuestas son para la calidad interna y externa, estas características son divididas en sub-características que se manifiestan externamente cuando el software se usa como una parte del sistema computarizado, y son un resultado de los atributos internos del software. Son aplicables a todo tipo de software, incluidos los programas de computación y los datos contenidos en el firmware³. Estos atributos sirven para medir un software.

³ En computación, se refiere a las instrucciones de programación almacenadas en memoria en vez de ser aplicadas a través del software.

Las características y sub-características proveen una consistente terminología sobre la calidad del producto, proveen un marco para especificar los requisitos de calidad para el software y permiten el intercambio entre las diversas capacidades del producto de software (4).

1.2 Portabilidad de software

La portabilidad es uno de los temas más importantes que afrontan los desarrolladores de aplicaciones hoy día. Este término suele confundirse por muchas personas debido a sus diversas acepciones.

¿Es lo mismo portabilidad que migración?

No es lo mismo, a continuación se dará una explicación de estos dos términos.

La Migración consiste simplemente en hacer una aplicación existente ejecutar con éxito en una nueva plataforma y a menudo puede resultar en la sustitución de un conjunto de dependencias del sistema con otro.

La portabilidad implica que el software haya sido destinado a varias plataformas desde el principio y que este factor haya sido considerado a través del diseño e implementación. Es sólo mediante la adopción de este enfoque que se puede garantizar un alto por ciento de portabilidad en el resultado final.

Son muchas las estrategias que deben tener los proyectos para lograr un producto con portabilidad. Antes de comenzar a desarrollarlo se deben tener definidas prioridades para alcanzar un mejoramiento en este indicador, no puede faltar aislar las dependencias entre clases, pensar y tener una mente clara a la hora de desarrollar, también se debe tener en cuenta el control de las interfaces, los lenguajes, las librerías y las funciones que se utilizan, ya que mientras más se desarrollen en lenguajes que son en un gran por ciento portable esto mejoraría la portabilidad del mismo, así como los distintos tipos de estándares que se utilizan tanto para el diseño como para la implementación y desarrollo del software. Teniendo en cuenta todas estas razones enunciadas, en el desarrollo de software se puede obtener un producto con la mayor portabilidad posible y un enfoque para garantizarla sería la definición de un procedimiento o metodología por la que se rijan las personas a la hora de crear un software desde sus inicios.

1.2.1 Definiciones de Portabilidad.

Se define como la característica que posee un software para ejecutarse en diferentes plataformas, el código fuente del software es capaz de reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra. A mayor portabilidad menor es la dependencia del software con respecto a la plataforma. (5)

Según la Norma ISO 9126 la Portabilidad es la capacidad de producto de software de ser transferido de un ambiente a otro. Entiéndase por ambiente, el ambiente de software, del hardware u organizacional. La portabilidad es analizada partiendo de su adaptabilidad, instalabilidad, coexistencia y reemplazabilidad. (4)

La portabilidad, es uno de los elementos de la calidad de un software, se puede definir como un atributo que puede estar dirigido a una unidad de software en un grado específico con respecto a entornos específicos. También se afirma que una unidad de software es portátil en un entorno, en la medida en que el costo del transporte y adaptación a un nuevo entorno sea menor que el costo de la reconstrucción. (2)

Según *Ashrafi*, la portabilidad puede ser una necesidad para la organización que produce software para una variedad de plataformas. Conceptualmente se define la portabilidad como la facilidad de esfuerzo para el transporte a otro entorno de software y/o de la plataforma. (3)

En la presente investigación se toma como referencia la definición de portabilidad que está dada en la Norma ISO 9126, debido a que este concepto está muy claro y conciso, en la norma se especifican claramente las sub-características de portabilidad para un mejor entendimiento de la misma, por lo que se cree que es el más idóneo si se quiere alcanzar la portabilidad total de los productos.

1.2.2 Atributos o sub-características de la Portabilidad.

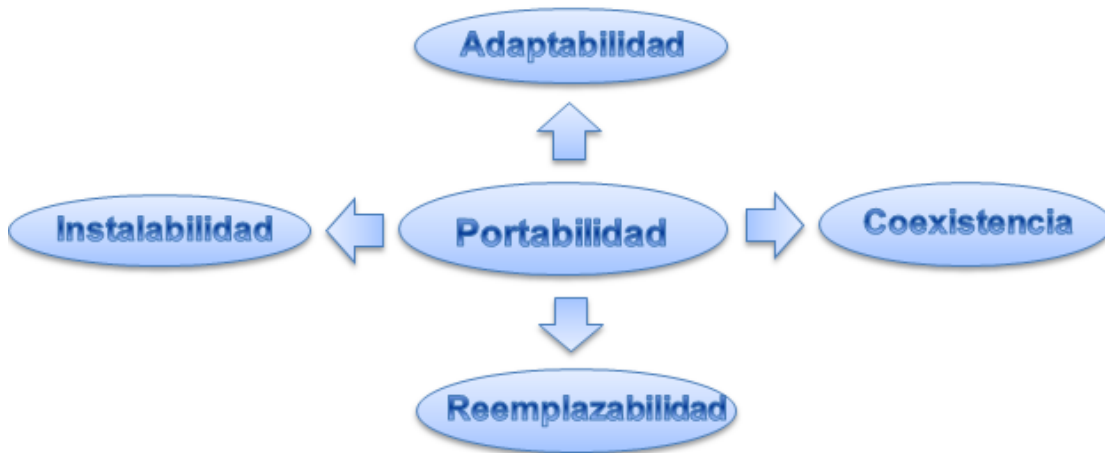


Ilustración 2: Atributos específicos de la Portabilidad

En la ISO 9126-1, se definen como sub-características de la portabilidad la adaptabilidad, instalabilidad, coexistencia y reemplazabilidad (4). En la **Ilustración 2** se muestran dichos términos, seguidamente se dará una explicación de cada uno de ellos.

Adaptabilidad: Capacidad del producto de software de ser adaptado a los ambientes especificados sin aplicar acciones o medios de otra manera que aquellos suministrados con el propósito de que el software cumpla sus fines. (4)

NOTA 1: La adaptabilidad incluye el escalado de la capacidad interna (por ejemplo los campos de la pantalla, las tablas, los volúmenes de transacción o los formatos de informes.).

Instalabilidad: Capacidad del producto de software de ser instalado en un ambiente especificado. (4)

Coexistencia: Capacidad del producto de software de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes. (4)

Reemplazabilidad: Capacidad del producto de software de ser usado en lugar de otro producto software especificado para los mismos fines y en el mismo ambiente. (4)

NOTA 2: Por ejemplo, la reemplazabilidad de una nueva versión de un producto de software es importante para el usuario cuando va a actualizar su sistema.

1.3 Evaluación de la calidad: Su Importancia

En la actualidad es importante realizar evaluaciones a todo sistema de software a través de un proceso de Control de Calidad para poder asegurar no sólo la funcionalidad a la cual está enfocado el sistema sino también para reducir al mínimo las falencias que dicho software podría tener

La inadecuada evaluación de la calidad de un producto es una de las causas fundamentales que lo hace fracasar en el mercado debido a que sale a la venta y quizás no tenga la calidad requerida para ello, provocando insatisfacción y desconfianza en el cliente, esto es un factor primordial debido a que si el cliente queda satisfecho con lo que compra habrá más venta de los productos en el mercado de lo contrario sería un fracaso. (3)

La evaluación de software para lograr la calidad de los mismos es uno de los procesos del ciclo de vida de desarrollo del software. La calidad del producto se puede evaluar al medir los atributos internos (típicamente por las medidas estáticas de productos intermedios), o los atributos externos (por lo general mediante el comportamiento del programa cuando se ejecuta) o midiendo los atributos de la calidad durante el uso. El objetivo es que el producto sea capaz de ejercer el efecto requerido en un contexto de uso particular. Por consiguiente, evaluar y mejorar un proceso es un medio para mejorar la calidad del producto, y la evaluación y mejora de la calidad del producto son una vía para mejorar la calidad durante el uso. De igual modo, la evaluación de la calidad durante el uso permite la retroalimentación para mejorar un producto, y la evaluación cuando se produce permite la retroalimentación para mejorar un proceso. (4)

El objetivo de evaluar la calidad de un producto de software es la detección de defectos y la corrección de los mismos una vez detectados en etapas tempranas durante el proceso de desarrollo del software. Con esto se logra que el producto final satisfaga las necesidades del cliente. Se prevé no alcanzar la calidad perfecta sino la calidad necesaria y suficiente para cada contexto de uso específico cuando el producto se entrega a los usuarios y éstos lo utilizan en la práctica.

Una inadecuada evaluación de la calidad del software puede causar defectos inesperados en las pruebas y fases de operación. En el peor de los casos, los productos no se entregarían al cliente o habría un aumento en el costo de mantenimiento.

Para evaluar la calidad se necesita un modelo de calidad, el método de evaluación que se va a utilizar y herramientas de soporte si se necesitan.

En la investigación se utiliza como guía para evaluar los productos, la Norma ISO/IEC 14598 en conjunto con la norma ISO/IEC 9126. Específicamente en la 14598 se define el proceso de evaluación. **Ver ilustración 3**

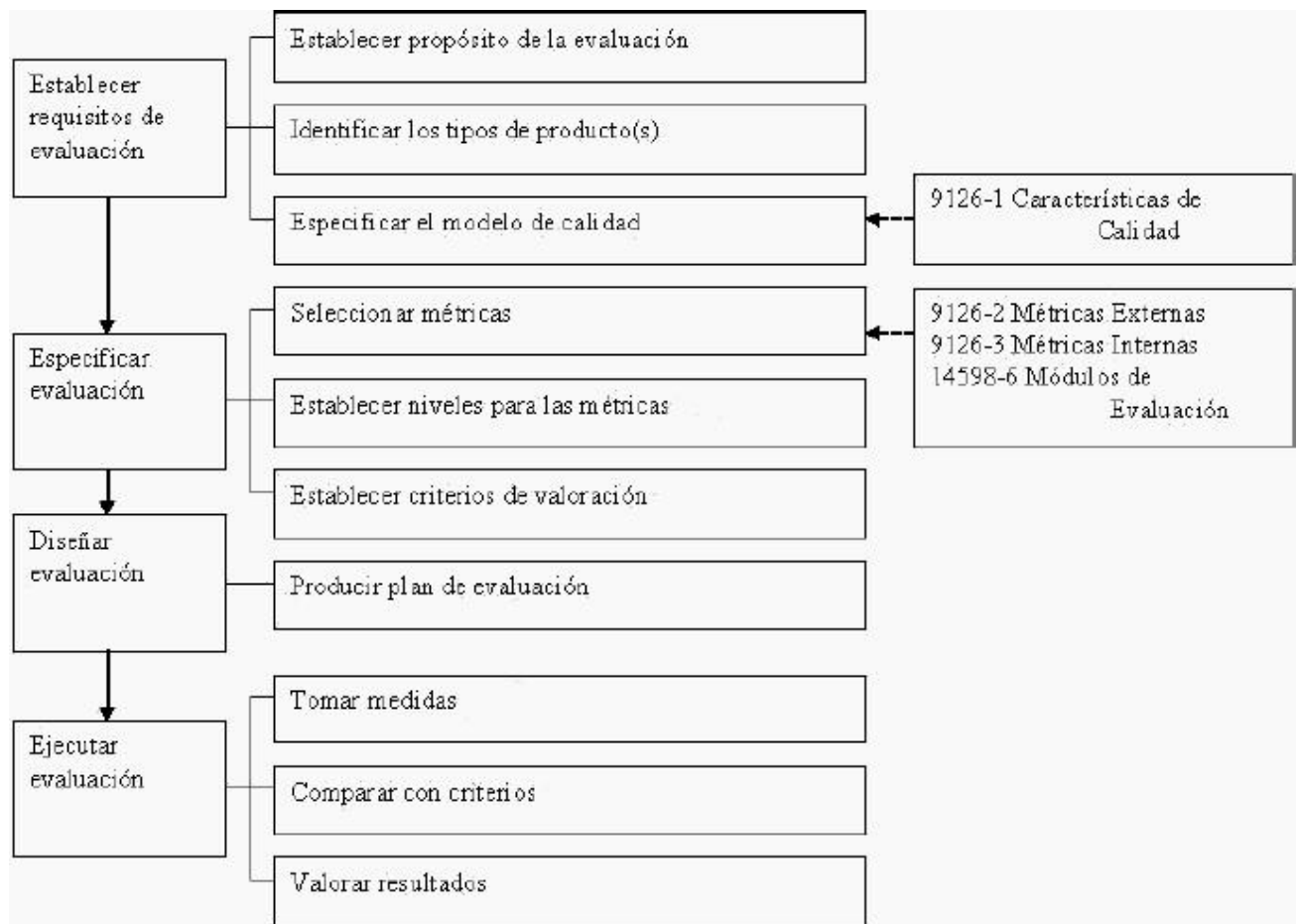


Ilustración 3: Proceso de evaluación de los productos de software en la ISO/IEC 14598

El proceso de evaluación como se muestra en la ilustración anterior está definido por las siguientes fases: Establecer requisitos de evaluación, Especificar evaluación, Diseñar evaluación y Ejecutar evaluación, cada fase se desglosa por diferentes actividades, entre las que se encuentra Especificar el modelo de calidad, aquí se elige el modelo que se define en la norma ISO/IEC 9126-1 constituido por características y sub-características las cuales se pueden medir utilizando métricas, a la hora de la selección de las métricas se escogen basándose en las definidas en la ISO/IEC 9126-2 y 9126-3.

1.3.1 Evaluación de la Portabilidad.

El tema principal de la portabilidad para el proceso de desarrollo viene dado por la siguiente interrogante: *¿Qué esfuerzo debe ser gastado para producir un diseño que pueda ser fácilmente adaptada a los ambientes en el futuro?* Una respuesta racional a esta pregunta requiere un análisis costo / beneficio para determinar el grado de portabilidad que se puede lograr antes de que los costos superen los beneficios.

El grado de portabilidad de un software sólo puede especificarse con precisión en un entorno de destino específico. Este proceso tiene mayor efectividad si es posible comparar las características de portabilidad de la interfaz externa de una unidad de software con las de la interfaz del ambiente. Con este análisis simple se puede estimar los costos de portabilidad (y por tanto el grado de portabilidad).

El grado de portabilidad depende de dos enfoques diferentes, la portabilidad y la reconstrucción del software. Durante la fase de mantenimiento, los costos de estos enfoques se pueden estimar, dado un producto y un entorno de destino específico.

1.3.1.1 Proceso de Portabilidad

Se examinan los pasos involucrados en el proceso de Portabilidad con el fin de estimar sus costos. Suponiendo dependencias en el origen de la portabilidad, el primer paso es **adquirir el código fuente en una forma procesable**. Esto es trivial debido a que la fuente original se puede suministrar desde el principio, o puede requerir el análisis y la reconstrucción del programa ejecutable. En este último caso, los costos pueden ser tan altos como para descartar este enfoque inmediatamente.

Suponiendo la disponibilidad del código fuente, los pasos principales en el proceso de transferencia son el **transporte y la adaptación**, y la segunda se puede separar en **la modificación manual y traducción automática**. Así pues, tenemos tres actividades principales:

- **Modificar** la fuente si es necesario para adaptarse a los requisitos del nuevo entorno.
- **Traducir** la fuente modificándola en formato ejecutable, usando un compilador u otro tipo de traductor que genera código para el entorno de destino.
- **Transportar** el programa físicamente, al entorno de destino en forma adecuada.

Estos pasos se pueden realizar en diversos órdenes, dependiendo de lo más idóneo y lo que se necesite.

Se utilizan herramientas de apoyo, en algunos casos, las herramientas deben ser transportadas con procedimientos similares.

Una vez instalado el software en el nuevo entorno, debe ser probado y depurado. Esta actividad también puede ser ejecutada parcialmente, en las primeras etapas mediante técnicas como la verificación formal o pruebas de simulación. Este proceso puede conducir a las iteraciones de las etapas anteriores.

Por último, la aplicación completa debe ser documentada, y el mantenimiento normal o mejora de las actividades puede ser necesario durante todo su ciclo de vida.

1.3.1.2 Proceso de Reconstrucción

En general, la Reconstrucción implica el diseño de software convencionales y el desarrollo de actividades. Un primer paso esencial es **adquirir las especificaciones de requisitos**. Si el punto de partida es una implementación existente, las especificaciones se tienen que deducir de la conducta del software actual. Esto puede ser una actividad costosa. Una vez que las especificaciones están disponibles, las actividades de desarrollo principales son:

- Diseñar una estructura de software más apropiada para el nuevo entorno.
- Implementar el software para el nuevo entorno.

También es necesario para el desarrollo, herramientas de apoyo para el nuevo entorno, y para transportar o reconstruir los archivos de datos. Las pruebas y depuración son más necesarias, y puede dar lugar a las iteraciones del diseño y pasos de implementación.

Al igual que en el caso de Portabilidad, la nueva implementación debe ser documentada, y las actividades de mantenimiento deben ser apoyadas.

1.3.1.3 Grado de portabilidad

El grado de portabilidad puede ser calculado de la siguiente manera: (2)

$$GP = 1 - (CP / CR)$$

Dónde:

GP: Grado de portabilidad.

CP: Costo de portabilidad.

CR: Costo de reconstrucción.

Si el valor de GP es mayor que cero, entonces es más conveniente realizar el proceso de portabilidad que el de reconstrucción. Por otra parte, si el GP toma valor 1 representa portabilidad perfecta. Los costos de portabilidad son inversamente proporcionales al GP.

Para obtener el costo de portabilidad y el costo de reconstrucción son métricas que define James Money en su Informe "*Bringing Portability to the Software Process*". Ver el documento. (2)

El método que se utiliza en la investigación para evaluar la portabilidad de componentes es midiendo las sub-características de Portabilidad a través de métricas ya definidas que se encuentran en las normas ISO/IEC 9126-2 (Métricas Externas) y ISO/IEC 9126-3 (Métricas Internas).

1.4 Portabilidad en la práctica

Actualmente escasean las maneras de evaluar la portabilidad de software. En investigaciones realizadas en la Universidad de las Ciencias Informáticas se encontraron varias herramientas que permiten evaluar la portabilidad, tal es el caso del Trabajo de Diploma que tiene como título: "**Herramienta para evaluar la portabilidad de los datos**

en el desarrollo de software”, de los autores Alicia del Carmen Pérez Roldán y Jorge Luis Fuoman Noguera. (15)

Esta herramienta permite la evaluación de la portabilidad de los datos entre los Sistemas Gestores de Bases de Datos (SGBD) PostgreSQL y Oracle. La misma tiene como título “Evaluador de Portabilidad” (Ilustración 4) y su objetivo es dar al cliente un por ciento final de cuán portable es el modelo de datos que posee. La evaluación se hizo mediante métricas que los autores definieron. La herramienta se desarrolló sobre Microsoft Excel. A continuación se muestra una imagen de la misma.

	A	B	C	D
1	Métricas	Valor de uso	Costo	Valor de Portabilidad
2	Uso de actualizaciones de vistas.	0,2	1	0,2
3	Uso de la cláusula LIKE		1	0
4	Utilización del límite simple		0,6	0
8	Uso del tipo de datos TIMESTAMP WITH TIME ZONE		0,4	0
9	Uso de la función CHARACTER_LENGTH		1	0
10	Uso de la función SUBSTRING		1	0
11	Concatenación de columnas que admitan valores nulos.		0,5	0
12	Obtención de descripciones de consultas		1	0
13	Obtención del tiempo de consulta		1	0
14	Generación de claves automáticas		1	0
15	Utilización de valores NULL en la expresión Order By.		1	0
16				
17	% de afectación de la portabilidad:			20
18				
19	% de portabilidad del modelo analizado:			80
20				
21	Escala del valor de uso			
22	Si la característica no se usa el valor es: 0			
23	Si la característica se usa poco el valor puede estar entre 0,1 y 0,4			
24	Si la característica se usa en gran medida el valor puede estar entre 0,5 y 1			
25				

Ilustración 4: Evaluador de Portabilidad de Datos

La herramienta propuesta por los autores se considera que es beneficiosa para los proyectos productivos que se desarrollan en la UCI, debido a que se puede detectar a tiempo los problemas de portabilidad de los datos que existan en determinado proyecto, logrando con esto que se puedan arreglar y no ocasionen pérdidas ni mal funcionamiento.

Otra de las tesis que investiga acerca de la evaluación de la portabilidad tiene como tema: **“Herramienta para evaluar la portabilidad del código fuente en las aplicaciones Web”**, de los autores Annier Jiménez Plasencia y Gioanny González. (16)

La herramienta se desarrolló con el IDE NetBeans 6.8, sobre la plataforma Java. Esta posibilita a los usuarios evaluar la portabilidad del código fuente de las aplicaciones Web que actualmente se desarrollan en la universidad. Proporciona conocimientos acerca de todos aquellos problemas que están presentes en el código de los proyectos analizados a través de un fichero reporte, brinda un valor representando el por ciento de portabilidad con que cuentan los productos examinados, así como la cantidad de problemas encontrados en su totalidad de líneas de código. Muestra también los problemas encontrados en una tabla que se encuentra en la interfaz. La herramienta cuenta con una ayuda la cual sirve como guía a sus usuarios para un mejor entendimiento y uso correcto de la misma. **Ver ilustración 5**

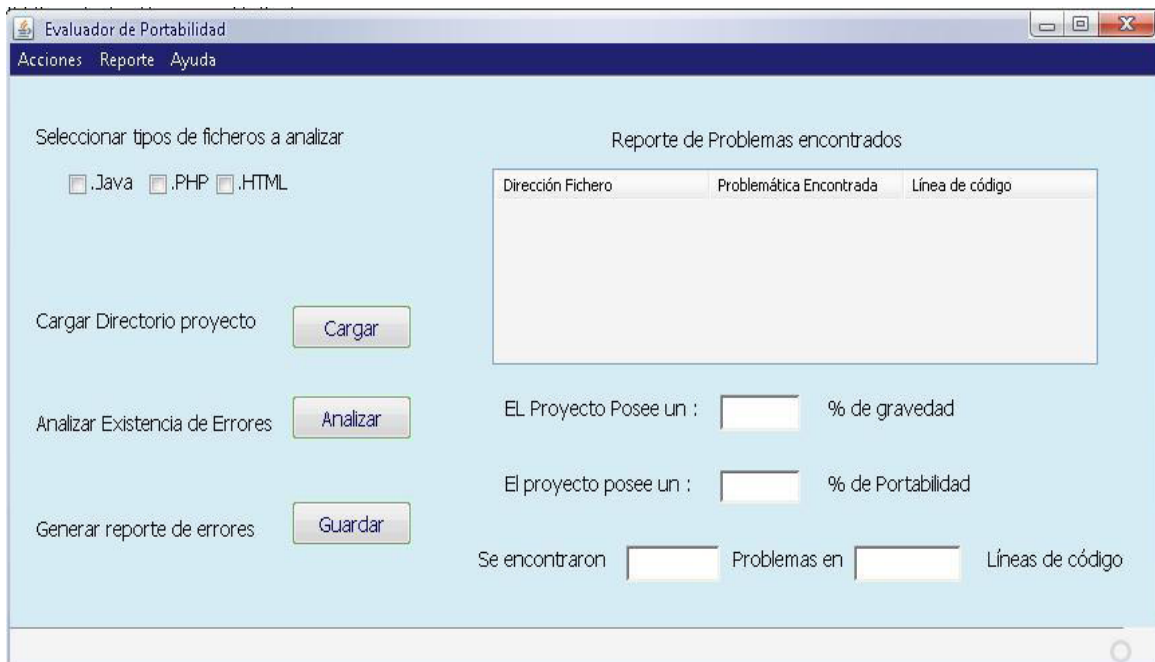


Ilustración 5: Evaluador de Portabilidad del código fuente

La implementación de la herramienta trae mejoras potenciales a la hora de hacerle pruebas a las aplicaciones Web que se desarrollan en la Universidad, debido a que puede detectar el por ciento de portabilidad que tiene una aplicación y darle una visión de la calidad que tiene el producto, logrando ahorros de tiempo y costos.

1.5 Desarrollo de Software Basado en Componentes (DSBC).

El desarrollo de software basado en componentes (DSBC)⁴, es una aproximación del desarrollo de software que describe, construye y utiliza técnicas software para la elaboración de sistemas abiertos y distribuidos mediante el ensamblaje de partes software reutilizables. La aproximación DSBC es utilizada para reducir los costos, tiempos y esfuerzos de desarrollo del software, a la vez que ayuda a mejorar la fiabilidad, flexibilidad y la reutilización de la aplicación final.

1.5.1 Componentes Software

“Es un paquete coherente de artefactos de software que puede ser desarrollado independientemente y entregado como una unidad y este puede ser compuesto, intercambiado con otro componente para construir algo mucho más grande”. (10)

Haciendo un análisis de la definición antes dicha se puede decir que los componentes son partes software que se pueden desarrollar independiente o combinar con otros componentes para generar un conjunto aún mayor (ejemplo: otro componente, subsistema o sistema).

Un componente juega el papel de una unidad software reutilizable que puede inter-operar con otros módulos mediante sus interfaces. Define una o más interfaces desde donde se puede tener acceso a los servicios que éste ofrece a los demás componentes. Puede presentarse en forma de código fuente o código objeto; puede estar escrito en un lenguaje funcional, procedural o en un lenguaje orientado a objetos.

1.5.2 Tipos de Componentes

⁴ En la literatura se puede encontrar el término como CBD (Component-Based Development)

⁴ En la literatura se puede encontrar el término como CBSE (Component-Based Software Engineering).

Según su modificabilidad (18)

- Caja negra
- Caja blanca

Según su granularidad (18)

- Componentes de negocio
- Marcos (frameworks)
- Componentes de aplicación

Según la tecnología usada (18)

- Componentes imperativos (Módulos, funciones)
- Componentes Orientados a Objeto (Clases)
- Componentes distribuidos (Componentes CORBA, Componentes.NET, Componentes J2EE, Servicios web)

Los componentes que se van a utilizar para validar la investigación en el Capítulo 3, haciendo una evaluación de la portabilidad de software en el CEDIN, son los que se clasifican según la tecnología usada en Componentes Orientados a Objeto (Clases) y Componentes imperativos (Módulos).

Componentes Orientados a Objetos (Clases)

Los componentes orientados a objetos se desarrollan con la misma filosofía que la programación orientada a objetos, aquí se utilizan objetos, no algoritmos, como bloques de construcción lógicos. Cada objeto es una instancia de una clase y las clases se relacionan unas con otras por medio de relaciones de herencia.

La programación orientada a objetos es una de las formas más populares de programar y viene teniendo gran acogida en el desarrollo de proyectos de software desde los últimos años. Esta acogida se debe a sus grandes capacidades y ventajas frente a las antiguas formas de programar. Entre las ventajas más significativas se encuentran:

- Fomenta la reutilización y extensión del código.

- Permite crear sistemas más complejos.
- Relacionar el sistema con el mundo real.
- Agiliza el desarrollo de software.
- Facilita el trabajo en equipo.
- Facilita el mantenimiento del software

Componentes Imperativos (módulos):

Los componentes imperativos módulos son partes de una aplicación, donde cada uno debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes. Estos componentes módulos se pueden compilar por separado, pero tienen conexiones con otros módulos.

Los componentes imperativos se basan en el paradigma de lenguaje de *programación imperativo* el cual soporta características entre las que se encuentran:

- Tipos de datos para números reales, caracteres, cadenas, booleanos y sus operadores, y punteros (apuntadores).
- Definición de tipos por parte del usuario.
- Estructuras de control, bucles for y while, instrucciones case (switch).
- Asignación de elementos y arrays.

Los lenguajes imperativos han sufrido una transformación desde un lenguaje de recorrido completo, hasta una mezcla de múltiples funcionalidades, según necesidades del programador.

1.6 Métricas de calidad de software

Las métricas de calidad de software se pueden definir como: “La continua aplicación de técnicas basadas en la medición al proceso de desarrollo de software y a sus productos para proveer información administrativa significativa y oportuna, junto con el uso de esas técnicas para mejorar el proceso y sus productos.” (8)

Se puede afirmar que las métricas aportan una manera de estimar la calidad de los atributos internos, permitiendo valorar la calidad del producto antes de construirlo, así se

podrá estimar el tiempo de su realización, mitigando los riesgos que puedan presentarse en su desarrollo. Esta es una de las vías más seguras que todo proyecto puede utilizar para verificar la calidad de sus productos tanto en el desarrollo como el producto final, con su uso se alcanza la seguridad de la planificación debido a la detección de problemas antes de que ocurran.

1.7 Procedimiento

El término procedimientos corresponde al plural de la palabra procedimiento, en tanto, por la misma se refiere a la acción, modo de proceder o método que se implementa para llevar a cabo ciertas cosas o tareas.

Básicamente, un procedimiento consiste de una serie de pasos bien definidos que permitirán y facilitarán la realización de un trabajo de la manera más correcta y exitosa posible.

1.8 Normas de software: ISO/IEC 9126 y 14598

Una norma establece un marco de referencia común para los procesos del ciclo de vida del software, con una terminología bien definida. Contiene procesos, actividades y tareas para aplicar durante el suministro, desarrollo, operación y mantenimiento de productos software. También incluye un proceso que se puede emplear para definir, controlar y mejorar los procesos del ciclo de vida.

1.8.1 Estándar

Un estándar es una especificación precisa de algo. Estándar o norma es un conjunto de reglas sobre cómo debe ser algo.

1.8.2 Instituciones más importantes para el desarrollo de estándares.

- ISO – International Organization for Standardization. (14)
- IEC – International Electrotechnical Commission. (13)

1.8.3 Norma ISO/IEC 9126

El ISO/IEC 9126 *Calidad del producto*, permite especificar y evaluar la calidad del producto de software desde las perspectivas de aquellos asociados con la adquisición, regulación, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría del software. Por ejemplo, puede ser utilizada por los programadores, los clientes, el personal de aseguramiento de la calidad y los evaluadores independientes, particularmente los responsables de especificar y evaluar la calidad de los productos de software. (4)

La norma está conformada por 4 partes que dirigen: un modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso.

En la primera parte el modelo de calidad se divide en dos partes: calidad externa y calidad interna, y la calidad durante el uso. La primera parte del modelo especifica seis características para la calidad interna y externa, que son además divididas en sub-características. La otra parte del modelo especifica cuatro características de calidad durante el uso del producto.

La segunda parte del estándar define las métricas externas necesarias para medir con respecto a cada una de las características y sub-características agrupadas en la primera parte.

La tercera parte define las métricas internas para estimar las características de calidad de un software que se encuentre en ejecución o desarrollo.

Por último, la cuarta parte, define las métricas para establecer la calidad en uso de un producto software.

1.8.4 Norma ISO/IEC 14598

La ISO/IEC 14598 *Evaluación de los productos software* ofrece una visión general, explica la relación entre su serie y el modelo de calidad de la ISO/IEC 9126, define los términos técnicos utilizados, contiene requisitos generales para la especificación y evaluación de la calidad del software, y clarifica los conceptos generales. Además, provee un marco de trabajo para evaluar la calidad de todos los tipos de productos de software y establece requisitos para métodos de medición y evaluación de los productos. (9)

Se compone de 6 partes, ellas son: Panorama general, Planificación y gestión, Proceso para desarrolladores, Proceso para adquirientes, Proceso para evaluadores y

Documentación de los módulos de evaluación. Esta norma es para utilizarla en conjunto con la 9126 debido a que la 14598 define el proceso de evaluación pero la 9126 tiene el modelo de calidad y las métricas necesarias para llevar a cabo el proceso.

Conclusiones Parciales

En el presente capítulo se trataron diferentes conceptos que son de vital importancia para dar solución al problema planteado. Los temas más importantes que se tienen en cuenta son el de portabilidad de software, evaluación de la calidad y evaluación de la portabilidad, debido a que constituyen el centro de la investigación.

Después de haber hecho el estudio del capítulo se llega a la conclusión que la evaluación de la portabilidad es importante y necesaria para que un producto se libere con el mayor grado posible de calidad con respecto a esta característica, trayendo consigo mejoras potenciales tanto para el producto como para la empresa que lo respalda.

CAPÍTULO 2

Procedimiento para evaluar la portabilidad de Componentes Software

Introducción

En este capítulo se explica la propuesta de solución, que puede ser aplicada a los componentes software que se desarrollan en el CEDIN. Para ello se definen 4 etapas, cada una tiene actividades asociadas que las desarrollan los roles encargados de llevar a cabo el proceso de evaluación. Las etapas definidas son:

- Descripción del componente que se evaluará
- Proceso de Medición
- Ejecución de la evaluación
- Procesar evaluación de la Portabilidad

Para el buen funcionamiento del procedimiento las etapas se ejecutan secuencialmente o, sea, una detrás de la otra, comenzando por la Etapa 1.

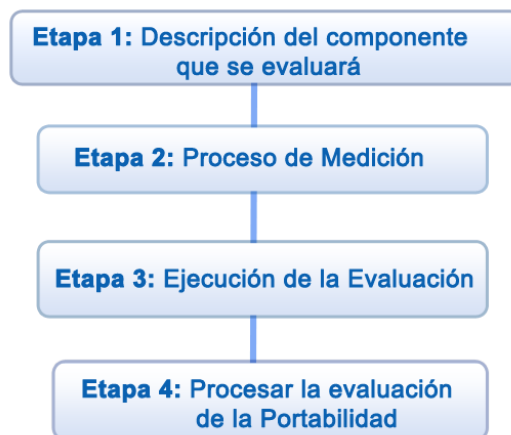


Ilustración 6: Etapas del Procedimiento

Procedimiento para evaluar la Portabilidad

2.1 Nombre del procedimiento

Evaluación de la Portabilidad de Componentes Software.

2.2 Objetivos

- Proporcionar una guía, paso a paso, que permita evaluar la portabilidad de los componentes que se desarrollan en el CEDIN tanto para el Administrador de la Calidad como para el Desarrollador.
- Establecer métricas que permitan medir el por ciento de portabilidad que posee un componente.

2.3 Alcance

Este procedimiento es aplicable a todos los componentes que se desarrollen en el CEDIN siempre y cuando estos componentes estén identificados dentro de la clasificación propuesta por la investigación en el Capítulo 1, epígrafe **1.5.2** Tipos de Componentes.

2.4 Referencias

Manual de procedimientos IPP-1000:2008.

ISO/IEC 14598 Evaluación de los productos de software.

ISO/IEC 9126-2 Métricas Externas.

ISO/IEC 9126-3 Métricas Internas.

2.5 Responsables

Ejecuta: Administrador de la Calidad o Desarrollador.

Responsable de su ejecución: Grupo de Aseguramiento de la Calidad del CEDIN.

Revisa y actualiza este procedimiento: Jefe del grupo de Gestión de la calidad (GGC) del CEDIN.

Fiscaliza su cumplimiento: Jefe del grupo de Gestión de la calidad (GGC) del CEDIN.

2.6 Términos y definiciones

Procedimiento: Serie de pasos bien definidos que permitirán y facilitarán la realización de un trabajo de la manera más correcta y exitosa posible.

Rol: Responsabilidades de cada uno de los miembros del proyecto.

Procedimiento para evaluar la Portabilidad

Etapas: Fase en el desarrollo de una acción u obra.

Fase: Representa un ciclo de desarrollo.

Acciones Correctivas: Métodos que se utilizan para eliminar las causas que provocan que existan no conformidades en el software desarrollado.

No conformidad (NC): Incumplimiento de un requisitos.

Fallos: Incumplimiento de un requisito asociado a un uso previsto o especificado.

Artefacto: Productos tangibles del proyecto, que son producidos, modificados y usados por las actividades.

Proceso de Mejora: El proceso de mejora está encaminado a que la Universidad de las Ciencias Informáticas alcance una certificación internacional del nivel 2 del modelo CMMI. Hecho que la convertiría en la primera empresa cubana certificada con este modelo.

2.7 Normas generales

Se propone que el equipo de Aseguramiento de la calidad debe ser diferente al equipo de desarrollo del software, así se detecta con mayor facilidad las posibles no conformidades que puedan afectar la calidad del software en cuanto a su portabilidad.

El administrador de la calidad debe contar con las competencias necesarias para poder aplicar el procedimiento de manera exitosa.

El Procedimiento se debe ejecutar secuencialmente comenzando por la Etapa1.

Procedimiento para evaluar la Portabilidad

2.8 Flujo del Procedimiento

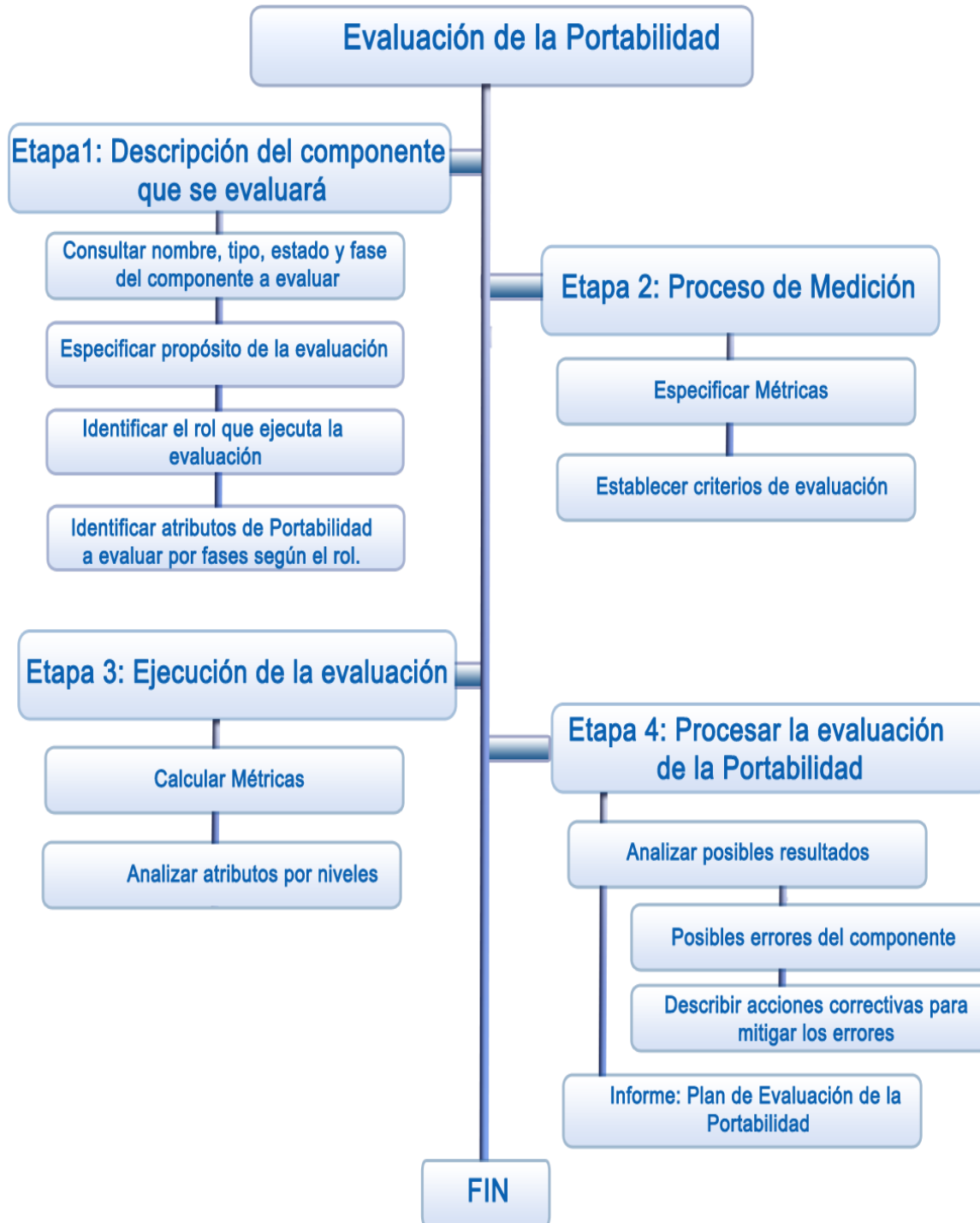


Ilustración 7: Flujo central del Procedimiento

2.9 Descripción del Procedimiento

El procedimiento propuesto está basado en las normas ISO/IEC 9126 y en la ISO/IEC 14598, estos dos estándares proponen una guía de lo que se debe hacer para evaluar la calidad de los productos, pero no dice el cómo realizarlo, el procedimiento que se describirá a continuación es una propuesta de cómo evaluar la característica portabilidad. Se realizó una adaptación de lo que proponían las normas a lo que se necesitaba.

Etapa 1: Descripción del componente que se evaluará

La etapa Descripción del componente que se evaluará está compuesta por diferentes actividades, el objetivo principal es recolectar toda la información inicial necesaria que se necesite para realizar la evaluación de la portabilidad lo más cercano posible a la realidad. A continuación se muestra para una mejor visualización un diagrama con las actividades definidas en esta primera etapa.

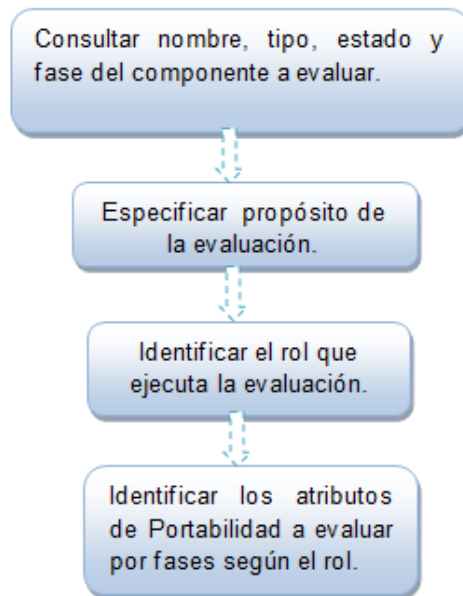


Ilustración 8: Actividades de la Etapa 1

1.1 Consultar Nombre, Tipo, Estado y Fase del componente a evaluar.

Procedimiento para evaluar la Portabilidad

Se consulta el nombre, tipo, estado y fase del componente que se va a evaluar, entrevistando al líder del proyecto donde se desarrollan los componentes que se van a evaluar. Ver Anexo 1

Tipos de Componentes: Ver Etapa 2, epígrafe 1.5.2 Tipos de Componentes.

El Estado del componente puede ser: *En Desarrollo* o *Terminado*.

Se selecciona la fase en la que se encuentra el componente. Las fases que se van a utilizar son las que están definidas en RUP (Rational Unified Process) debido a que esta es la metodología en la que se basa el Programa de mejora de la Universidad para definir sus fases, es muy probable que existan algunos componentes que se hayan desarrollado con otra metodología porque las características del proyecto donde se desarrollan lo requieran, en estos casos se especifica la fase en la que se encuentra y que metodología es la que se usa. Si el proyecto donde se desarrolla el componente está inmerso en el Programa de Mejora, se especifica en la fase que se encuentra según las especificadas que este posee. Para ver detalladamente el ciclo de vida definido en el Programa de Mejora Ver Anexo 3.

Ejemplo de cómo se pondría si no se utiliza metodología RUP:

- El componente X utiliza metodología XP para su desarrollo, se encuentra en la fase Planificación.
- El componente X se rige por el ciclo de vida que define el programa de Mejora, se encuentra en la fase Implementación.

Rational Unified Process (RUP) es una metodología que se usa para proyectos largos de gran envergadura, divide en 4 fases el desarrollo del software. **(12)**

- **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** Se especifica en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema. El resultado de esta fase es una línea base de la arquitectura.

Procedimiento para evaluar la Portabilidad

- **Construcción:** En esta fase se crea el producto, la línea base de la arquitectura crece hasta convertirse en el sistema completo.
- **Transición:** El producto se convierte en versión *beta*⁵, un número reducido de usuarios con experiencia prueban el producto e informan los defectos y deficiencias. Los desarrolladores corrigen los problemas e incorporar algunas de las mejoras sugeridas.

1.2 Especificar propósito de la evaluación

El propósito general de la evaluación de la portabilidad es detectar las no conformidades y fallos que las sub-características (adaptabilidad, instalabilidad, coexistencia y reemplazabilidad) de portabilidad que el componente podría tener y proponer acciones correctivas para el mejoramiento de las mismas.

En este acápite se especifica el propósito de la evaluación de la Portabilidad que se le vaya a realizar a un componente específico.

El propósito de la evaluación de la portabilidad del componente final podría ser:

- Decidir sobre la aceptación del componente.
- Decidir cuán adaptable e instalable es el componente.
- Decidir si el componente puede coexistir o puede ser reemplazado por otro en el mismo ambiente.
- Decidir cuándo liberar el componente.
- Comparar el componente con otros que se desarrollen en el CEDIN.
- Decidir cuándo aumentar o reemplazar el producto.
- Conocer el por ciento de portabilidad que posee el componente.

1.3 Identificar el rol que va a ejecutar el procedimiento.

Se identifica el Rol encargado de ejecutar el procedimiento, solo hay dos Roles que tienen estos permisos, Administrador de la Calidad y Desarrollador. Las responsabilidades del desarrollador y del administrador de la calidad son referenciadas del documento 0516_Roles y Responsabilidades que se encuentra publicado en <http://calisoft.uci.cu>, y

⁵ Se le llama a la primera versión del producto donde se le hacen pruebas y se informan los defectos.

Procedimiento para evaluar la Portabilidad

adaptadas durante esta investigación a las responsabilidades que deben tener estos dos roles para una correcta evaluación de la Portabilidad.

Desarrollador:

Responsabilidades

- Convierte la especificación del sistema en código fuente ejecutable.
- Desarrolla el diseño teniendo en cuenta la arquitectura para lograr que el producto final sea liberado con un alto por ciento de portabilidad.
- Integra los componentes que forman parte de la solución logrando con esto que puedan coexistir en un ambiente común y compartir los mismos recursos que tengan.
- Procesa y analiza los resultados de la Evaluación de la Portabilidad.

El desarrollador para un buen desempeño del rol que ocupa debe tener conocimientos de técnicas de programación, análisis y diseño de sistemas, debe tener habilidades sobre lógica y algoritmos, conocimientos de portabilidad de software, aptitud para identificar la mejor alternativa de solución que se pueda presentar y la más portable posible, y una de las cosas principales, una muy buena disposición para el trabajo.

Administrador de la calidad

Responsabilidades

- Elabora el Plan de Aseguramiento de la calidad donde incluye, cómo evaluar la portabilidad de software de un componente.
- Elabora el plan de Mediciones incluyendo métricas para evaluar la adaptabilidad, instalabilidad, coexistencia y reemplazabilidad.
- Participa en la elaboración del Plan de Monitoreo y en el monitoreo y análisis de las áreas de procesos según la fase en la que se encuentre el componente verificando cómo se comporta la portabilidad del mismo.
- Guía el diseño y ejecución de las pruebas internas que se hacen a los componentes no obviando la portabilidad que estos presentan.
- Participa en el análisis y recolección de los datos para aplicar las métricas de las características de portabilidad.
- Colabora en las auditorías que se les realicen al proyecto comprobando si el componente cumple con las características de portabilidad.

Procedimiento para evaluar la Portabilidad

- Crea una cultura de calidad especificando la importancia que tiene la portabilidad para los componentes que se desarrollen en el CEDIN.

El administrador de la calidad para realizar una exitosa evaluación de la portabilidad debe tener ciertas habilidades para un mejor desempeño a la hora de realizar la evaluación, debe tener amplio conocimiento de conceptos relacionados con la calidad especialmente la característica portabilidad, debe principalmente trabajar en equipo, ser buen comunicador, tener amplio poder estadístico y de síntesis, dominar el ciclo de desarrollo del software y tener amplios conocimientos en la rama Ingeniería de Software, además de tomar decisiones cuando sea preciso.

1.4 Identificar los atributos de Portabilidad a evaluar por fases según el rol.

El rol encargado de realizar la evaluación identifica los atributos de Portabilidad que necesiten para realizar la evaluación en la fase que se encuentra el componente. Los atributos son: (Se encuentran detallados en el **Capítulo 1, acápite: 1.2.2 Atributos de la Portabilidad**)

- Adaptabilidad
- Instalabilidad
- Coexistencia
- Reemplazabilidad

A la hora de seleccionar los atributos que se van a utilizar para realizar la evaluación, se escogen en dependencia de las necesidades que el rol encargado de realizar el proceso presente. Todas las sub-características son importantes pero las que no pueden faltar cuando se está evaluando la portabilidad son la adaptabilidad y la coexistencia debido a que son las más significativas y generales relacionadas con la portabilidad.

A continuación se muestran las sub-características de Portabilidad que puede utilizar cada rol en una fase determinada, para mayor descripción **Ver 1.4.1 Especificaciones de la Tabla 2.1** que está todo explicado con mayor detalle.

Procedimiento para evaluar la Portabilidad

Tabla 2.1: Sub-Características de la Portabilidad en las fases del desarrollo de software según los roles.

Roles	Sub-características	Nivel Requerido	Fases			
			Inicio	Elaboración	Construcción	Transición
Desarrollador	Adaptabilidad	Alto		x	x	x
	Instalabilidad	Alto		x	x	x
	Coexistencia	Alto		x	x	x
	Reemplazabilidad	Alto		x	x	x
Administrador de la calidad.	Adaptabilidad	Alto			x	x
	Instalabilidad	Alto			x	x
	Coexistencia	Alto			x	x
	Reemplazabilidad	Alto			x	x

1.4.1 Especificaciones de la Tabla 2.1

Para obtener un software portable una de las condiciones fundamentales que se tienen en cuenta es que se considere la portabilidad desde el levantamiento de requisitos, diseño e implementación del componente.

En la fase de inicio no se evalúan las características de portabilidad debido a que en esta fase lo que se hace es una descripción del componente final y se representa el análisis del negocio para el mismo.

En la fase de transición evalúan todos los atributos de la portabilidad tanto el rol desarrollador como el administrador de la calidad, es aquí donde se crea una versión del componente final a la que se le hacen pruebas para ver en qué estado se encuentra y si puede ser liberado.

Desarrollador

En la fase de elaboración el desarrollador no va a evaluar la portabilidad del componente, sino tener presente las sub-características que se especifican en la tabla 2.1, las cuales deben estar en el artefacto especificación de requisitos del componente bien detalladas

Procedimiento para evaluar la Portabilidad

como requisitos no funcionales, esto es importante si se quiere que al final el componente sea portable.

En la fase de construcción el desarrollador debe tener en cuenta todas las características de portabilidad como se muestra en la tabla, debe tener conocimientos y saber lo que se debe hacer para que el componente se desarrolle previendo que una vez terminado tenga un alto por ciento de portabilidad. En esta fase puede ir evaluando el componente una vez que se tenga algún resultado y así detectar fallos que podría solucionar antes que el componente este terminado.

Administrador de la calidad

En la fase de elaboración tampoco se tienen en cuenta estas características y el administrador no realiza tal evaluación debido a que en esta fase no hay ninguna versión del producto terminada puesto que aquí lo que se hace es el análisis y diseño del componente.

En la fase de construcción el Administrador de la calidad podrá aplicar el procedimiento cuando se tenga algún resultado de este, que se pueda evaluar.

1.4.2 Uso de la tabla 2.1: Sub-Características de la Portabilidad en las fases del desarrollo de software según los roles.

Para un adecuado uso de la tabla 2.1 se explican las partes por las que está conformada.

Roles: Se muestran los roles que tienen permisos para evaluar la portabilidad.

Sub-características: Nombre de la sub-característica o atributo de la portabilidad que se va a evaluar.

Nivel Requerido: Peso de las características, permite a los evaluadores (Administrador de Calidad o Desarrollador) establecer criterios de comparación con el Nivel Real (Ver tabla 2.5) cuando se establezca la evaluación.

Fases: Fases del desarrollo de software.

Nota: Esta etapa es muy importante y no se debe dejar ninguna actividad en blanco, debido a que en las posteriores etapas se van a utilizar información recogida en esta primera.

Procedimiento para evaluar la Portabilidad

Etapa 2: Proceso de Medición

La etapa Proceso de Medición tiene como objetivo aplicar las métricas necesarias para una correcta evaluación de la portabilidad de un componente software, la etapa está compuesta por dos actividades que se explicarán al detalle más adelante. Seguidamente se muestra un diagrama con las actividades específicas.

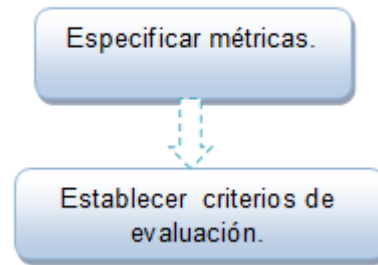


Ilustración 9: Actividades de la Etapa 2

2.1 Especificar métricas.

En esta actividad se especifican las métricas de los atributos de la Portabilidad necesarias para realizar la evaluación.

En la norma ISO/IEC 9126 se definen métricas para cada atributo de la Portabilidad, después de haber realizado un análisis de cada una de ellas se escogieron las métricas necesarias, suficientes y más importantes para una correcta evaluación subjetiva de las sub-características, se tuvieron en cuenta las necesidades específicas de la evaluación y que estas fueran lo suficientemente precisas para permitir realizar comparaciones fiables entre los atributos.

A continuación se muestra la tabla especificación de métricas donde se especifican las sub-características y el nombre de las métricas que se van a utilizar para realizar la evaluación.

Tabla 2.2: Especificación de métricas

Sub-característica	Nombre de las métricas escogidas para realizar la evaluación
--------------------	--

Procedimiento para evaluar la Portabilidad

2.1.1 Métricas de la portabilidad

Las métricas de la portabilidad se utilizan tanto para predecir el efecto que el software puede tener como ser capaces de medir un atributo como es el comportamiento del operador y el sistema durante las actividades de implementación y distribución del producto de software.

Métricas de adaptabilidad

Las métricas de adaptabilidad deben ser capaces de medir atributos tales como el comportamiento del sistema o el usuario que está tratando de adaptar el software a los diferentes entornos especificados.

Métricas de Instalabilidad

Las métricas de instalabilidad deben ser capaces de medir un atributo como es el comportamiento del sistema y del usuario cuando este está intentando instalar el software en un ambiente específico del usuario.

Métricas de Reemplazabilidad

Las métricas de reemplazabilidad deben ser capaces de medir atributos tales como el comportamiento del sistema o el usuario que está tratando de usar el software en lugar de otro software especificado en el entorno de dicho software.

Métricas de coexistencia

Las métricas de coexistencia deben ser capaces de medir atributos tales como el comportamiento del sistema o el usuario que está tratando de usar el software con otro software independiente en un entorno común compartiendo recursos comunes.

Procedimiento para evaluar la Portabilidad

Tabla 2.3: Métricas para evaluar la Portabilidad.

MÉTRICAS DE PORTABILIDAD			
Nombre	Propósito	Definición	Método de Aplicación
1. ADAPTABILIDAD			
Administrador de Calidad			
1.1 Adaptabilidad del componente software al hardware. (Adaptabilidad para dispositivos de hardware y servicios de red)	¿Puede el usuario adaptar fácilmente el componente software al hardware?	$X = 1 - A / B$ A: Número de funciones operativas donde las tareas no se hayan concluido o no cumplan con los niveles adecuados durante las pruebas de funcionamiento con el hardware. B: Número total de funciones que fueron probadas. $0 \leq X \leq 1$ Mientras más se acerque a 1 mejor.	Observar el comportamiento del usuario cuando este está tratando de adaptar el software al hardware.
	Desarrollador		
	¿Cómo se adapta el componente al hardware?	$X = A / B$ A: Número de funciones implementadas que son capaces de lograr los resultados requeridos en otros entorno de hardware. B: Número total de funciones capaces de adaptar los requisitos en el hardware. $0 \leq X \leq 1$ Mientras más se acerque a 1 mejor.	Contar el número de funciones implementadas que son capaces de lograr resultados en múltiples entornos de hardware y compararlo con el número de funciones capaces de adaptar los requisitos en el hardware.
1.2 Entornos de	Esta métrica es aplicable a los dos Roles		

Procedimiento para evaluar la Portabilidad

<p>ejecución del componente.</p>	<p>¿En cuántos entornos puede adaptarse el componente?</p> <p><i>Entorno: Se refiere a Sistema Operativo, tipos de sistema, distribución en caso que se esté trabajando con Linux.</i></p>	<p>X = Cantidad de entornos donde puede correr el componente.</p> <p>X=1 bajo, toma valor 0.2</p> <p>X = 2 medio, toma valor 0.5</p> <p>X> 2 alto, toma valor de 0.6 a 1 en dependencia de la cantidad, esto es apreciación del rol realice la evaluación.</p>	<p>Observar el comportamiento del usuario cuando este está tratando de adaptar el software al entorno de operación.</p>
Administrador de la Calidad			
<p>1.2.1 Adaptabilidad del componente software al Sistema Operativo probado</p>	<p>¿Puede el usuario adaptar fácilmente el componente software en el Sistema Operativo?</p>	<p>$X = 1 - A / B$</p> <p>A: Número de funciones operativas donde las tareas no se hayan concluido o no cumplan con los niveles adecuados durante pruebas de funcionamiento con el sistema operativo.</p> <p>B: Número total de funciones que fueron probadas.</p> <p>0 <= X <= 1</p> <p>Mientras más se acerque a 1 mejor.</p>	<p>Observar el comportamiento del usuario cuando este está tratando de adaptar el software al entorno de operación.</p>
Desarrollador			
<p>1.2.1 Adaptabilidad del componente software al Sistema Operativo probado</p>	<p>¿Cómo es adaptado el producto al sistema Operativo?</p>	<p>$X = A / B$</p> <p>A: Número de funciones implementadas que son capaces de lograr los resultados requeridos en determinados sistemas Operativos.</p> <p>B: Número total de funciones con capacidad de adaptación en el Sistema Operativo.</p> <p>0 <= X <= 1</p> <p>Mientras más se acerque a 1 mejor.</p>	<p>Contar el número de funciones implementadas que son capaces de lograr resultados en determinados entornos del sistema y compararlo con el número de funciones capaces de adaptarse en el sistema.</p>
2. INSTALABILIDAD			

Procedimiento para evaluar la Portabilidad

Administrador de la Calidad			
2.1 Facilidad de instalación.	<p>¿Puede el usuario o servidor instalar fácilmente el producto de software en su ambiente de operación?</p>	<p>$X = A/B$</p> <p>A: Número de casos en los cuales el usuario tiene éxito en las operaciones para una instalación adecuada a su conveniencia.</p> <p>B: Número total de casos en los cuales el usuario intenta adecuar la instalación a su conveniencia.</p> <p>$0 \leq X \leq 1$ Mientras más se acerque a 1 mejor.</p>	<p>Observe el comportamiento del usuario o servidor cuando tratan de instalar el producto de software en su ambiente de operación.</p>
Desarrollador			
2.1 Esfuerzo de Instalación	<p>¿Qué nivel de esfuerzo es necesario para la instalación?</p>	<p>$X = A / B$</p> <p>A: Número de pasos de instalación automatizada.</p> <p>B: Número de pasos de instalación necesarios.</p> <p>$0 \leq X \leq 1$ Mientras más se acerque a 1 mejor</p>	<p>Contar el número de pasos de instalación automatizados y compararlo con el número de pasos de instalación prescritas.</p> <p>Prescrito: <i>Ejemplo, número de ventanas, comandos, operación manual para alcanzar operativo de destino.</i></p>
3. COEXISTENCIA			
Administrador de la Calidad			
	<p>¿Con qué frecuencia el usuario se encuentra con todas las limitaciones o fallas inesperadas cuando se trabaja simultáneamente con</p>	<p>$X = A / T$</p> <p>A= Número de todas las limitaciones o fallas inesperadas que encuentre el usuario durante el funcionamiento del componente al mismo tiempo con otro software.</p>	<p>Utilizar el software de evaluación al mismo tiempo con otro software que el usuario utiliza a menudo.</p>

Procedimiento para evaluar la Portabilidad

4.1 Coexistencia disponible.	otro software?	<p>T= Tiempo de duración del funcionamiento del componente al mismo tiempo con otro software.</p> <p>0 <= X Mientras más se acerca a 0 mejor.</p> <p>Importante: Cuando la métrica tome valor 0 se le dará 1, en otro caso restar 1 menos el valor que dio la métrica y ese sería el que se coge para calcular la portabilidad.</p>	
	Desarrollador		
	¿Cuán flexible es el componente en compartir su ambiente con otro software sin efectos adversos?	<p>$X = A / B$</p> <p>A: Número de entidades con las que el componente puede coexistir.</p> <p>B: Número de entidades en el entorno de producción que requieren coexistencia</p> <p>0 <= X <= 1 Mientras más se acerque a 1 mejor</p>	<p>Contar el número de entidades con las cuales los componentes pueden coexistir y compararlo con el número de entidades en el entorno de producción que requieren coexistencia.</p>
4. REEMPLAZABILIDAD			
Administrador de la Calidad			
5.1 Uso Continuo de los datos.	¿Puede el usuario facilitar el uso de los mismos datos después de reemplazar este software por otro?	<p>$X = A / B$</p> <p>A: Número de datos que se utilizan para ser reemplazados y son capaces de utilizarse continuamente.</p> <p>B: Número de datos que se utilizan después que se sustituya el software y se prevé su continua reusabilidad.</p> <p>0 <= X <= 1 Mientras más se acerque a 1 mejor</p>	<p>Observar el comportamiento del usuario cuando este está reemplazando al software anterior, puede ser por otra versión del mismo.</p>

Procedimiento para evaluar la Portabilidad

Desarrollador		
¿Cuál es la cantidad de datos originales que permanecen sin cambios después de la sustitución de este producto?	$X = A / B$ A: Número de datos de software que continúan siendo utilizados después de la sustitución. B: Número de datos requeridos para ser utilizados desde el software antiguo, es decir desde el software antes de reemplazarse. $0 \leq X \leq 1$ Mientras más se acerque a 1 mejor.	Contar el número de elementos de datos, que se siguen utilizando después de la sustitución y compararlo con el número total de elementos de datos necesarios para ser utilizados desde el software antiguo.

2.1.2 Uso de la tabla de métricas

Para un correcto uso de las tablas de métricas se explican las partes por las que está compuesta la tabla, donde se especifican las métricas para evaluar la Portabilidad tanto para el Desarrollador como para el Administrador de la calidad.

Nombre de la métrica: Denominación de la métrica.

Propósito de la métrica: Se expresa como la interrogante a ser respondida por la aplicación de la métrica.

Definición: Provee la fórmula de medición y el significado de los datos empleados, además de los valores preferidos y su delimitación.

Se especifican las métricas para cada rol, se escoge la métrica a utilizar en dependencia del rol que a va a ejecutar el procedimiento.

Método de aplicación: Provee una secuencia de aplicación de la métrica.

2.2 Establecer criterios para la evaluación.

Se especifican los criterios para poder evaluar el resultado de las métricas y luego llegar a una conclusión, para ello se definen métricas y una tabla con los niveles establecidos.

2.2.1 Obtener valor de una sub-característica específica.

Procedimiento para evaluar la Portabilidad

El objetivo de esta métrica es conocer el valor de la sub-característica debido a que hay varios casos donde existe más de una métrica para la misma sub-característica, se pretende con esto conocer un único valor.

$$V_{Sub} = \frac{\sum C_{mS}}{CantM} \quad (1)$$

V_{Sub} : Valor de la sub-característica.

CmS: Valor obtenido después de haber realizado el cálculo de la métrica para esa sub-característica

CantM: Cantidad de métricas de la sub-característica.

Ejemplo:

Sub-característica: Adaptabilidad = 0.5

2.2.2 Valor real de adaptabilidad en el Sistema Operativo

Esta métrica se utiliza para obtener el valor de adaptabilidad que tiene el sistema operativo, depende del resultado de dos métricas que se encuentran en la tabla 2.3, estas métricas son: la 1.2 y 1.2.1

$$V_{real} = \frac{Aso + Ejec}{2}$$

V_{real} : Valor Real de Adaptabilidad al Sistema Operativo

Aso : Resultado que obtuvo la métrica Adaptabilidad al Sistema Operativo

$Ejec$: Resultado que obtuvo la métrica Entornos de ejecución del componente

2.2.3 Por ciento de portabilidad

Con esta métrica se calcula el por ciento de portabilidad de un componente con respecto a las sub-características escogidas en la Etapa1 actividad 1.4 **Selección de los atributos de calidad a evaluar**, esta fórmula depende del resultado anterior calculado de la fórmula (1)

Procedimiento para evaluar la Portabilidad

$$P_{Pt} = \frac{\sum V_{Sub}}{C_{Sub}} * 100$$

P_{Pt} : Por ciento Portabilidad.

V_{Sub} : Valor de la sub-característica.

C_{Sub} : Cantidad de sub-características.

2.2.4 Nivel de Portabilidad

Una vez aplicada las fórmulas anteriores se necesita conocer qué nivel y por ciento final de Portabilidad posee el componente, para ello se define la siguiente tabla.

Tabla 2.4: Nivel de Portabilidad

Nivel	Intervalo(métricas de atributos)	Porcentaje(final)
Alto	$0.8 \leq X \leq 1$	80%----100%
Medio	$0.5 \leq X < 0.8$	50%-----79%
Bajo	$X < 0.5$	Menor o igual que 49%

En la tabla se define el nivel de portabilidad tanto para los atributos como para el por ciento final del componente.

Etapa 3: Ejecución de la evaluación

En esta Etapa se realiza la evaluación de la Portabilidad, es una de las más importantes debido a que en ella se ejecuta el proceso, toda la teoría dicha anteriormente se lleva a la práctica aquí y se recoge en una tabla que está al final de la etapa. A continuación se muestran las actividades por las que está compuesta.

Procedimiento para evaluar la Portabilidad

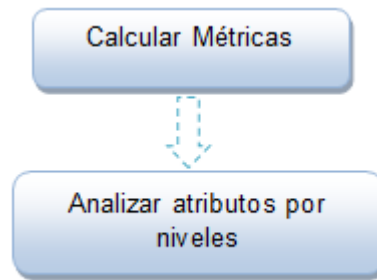


Ilustración 10: Actividades de la Etapa 3

3.1 Calcular Métricas

En esta actividad se procede a realizar el cálculo de las métricas.

3.2 Analizar atributos por niveles

En esta actividad se procede a otorgarle niveles a cada sub-característica después de haber calculado las métricas asociadas a ellas, estos niveles se seleccionan de los definidos en la Tabla 2.4 que se encuentra en el epígrafe **2.2.4 Nivel de Portabilidad**.

La tabla que se muestra a continuación es donde se resume todo el proceso de medición, en ella se recogerá la información y los resultados que denote el procedimiento una vez aplicado a un componente específico.

Tabla 2.5: Ejecución de la evaluación

Sub-característica	Nivel Requerido	Resultados	
		Valor Métrica	Nivel Real
Por ciento de Portabilidad			

3.2.1 Uso de la Tabla 2.5: Ejecución de la evaluación

Procedimiento para evaluar la Portabilidad

Sub-características: Nombre de la sub-característica de Portabilidad que se evalúa.

Nivel Requerido: Se especifica el que se encuentra en la *Etapa 1 / 1.4 Selección de los atributos de Portabilidad a evaluar según la fase / Tabla 1: Características de la Portabilidad en las fases del desarrollo de software.*

Resultados: Resultado obtenido de la evaluación.

- **Valor de la métrica:** Valor del resultado de la métrica aplicada.
- **Nivel Real:** Nivel Real que tiene la característica después de haberla evaluado.

Por ciento de Portabilidad: Por ciento total de portabilidad que posee el componente.

Etapa 4: Procesar la evaluación de la Portabilidad

Esta es la etapa final del procedimiento, se confeccionan las conclusiones de la evaluación y se conforma el informe final de la misma. Seguidamente se muestran las actividades:

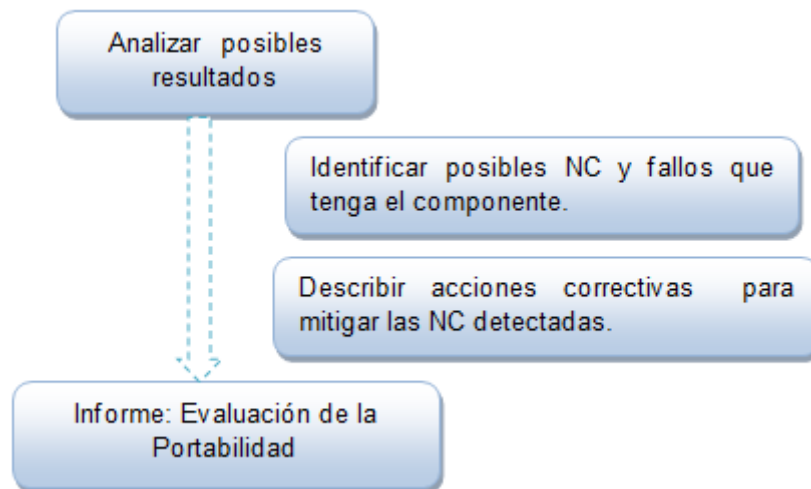


Ilustración 11: Actividades de la Etapa 4

4.1 Analizar Posibles Resultados

Una vez ejecutada la evaluación se procede a realizar el análisis de todo el proceso antes descrito teniendo en cuenta los resultados que se muestran en la tabla 2.5, donde se podrá comparar el nivel requerido con el nivel real de cada sub-característica así como el

Procedimiento para evaluar la Portabilidad

por ciento de portabilidad que tiene el componente en general. Con esta información se podrá llegar a una conclusión donde se identifiquen las posibles no conformidades y fallos que presenta el componente y las acciones correctivas que se pueden llevar a cabo para cada no conformidad. El rol encargado de ejecutar el procedimiento es el responsable de documentar los resultados y en caso que estos no sean satisfactorios debe informárselo al líder de proyecto para que este tome medidas. Todo queda registrado en el informe, Evaluación de la portabilidad que se describe en la siguiente actividad.

Los fallos en el software (o componentes software), tanto de funcionalidad como de seguridad, no son algo excepcional para los usuarios. Recientemente, un estudio publicado por el Instituto Nacional de Estándares y Tecnología (NIST) del Departamento de Comercio estadounidense ha cuantificado lo que cuestan allí: 59.500 millones de dólares al año, de los que los usuarios pagan más de la mitad del montante. Más testeo y mejores prácticas de calidad podrían reducir estas cantidades. (19)

Fallos más comunes de portabilidad

- Eliminación o sustitución de bibliotecas comunes a más de un programa o del sistema (Ejemplo: DLL Hell) ocasionando daños o pérdidas en el componente.
- Reiniciar arbitrariamente la sesión de un usuario para que la instalación tenga efecto.
- El componente no puede adaptarse en el hardware donde se está probando o desarrollando por causa de averías en el microprocesador, en la placa base o en la memoria RAM.
- Problemas a la hora de ejecutarse en el Sistema Operativo debido a pérdidas de archivos en su instalación.

Después de ver los fallos más comunes que ocurren en los software se muestran las posibles no conformidades y una propuesta de acciones correctivas que se pueden llevar a cabo para mitigar las mismas, en caso de que exista alguna no conformidad que no se especifique en la siguiente tabla, se redacta cuando se esté documentando el proceso en el informe.

Procedimiento para evaluar la Portabilidad

Tabla 2.6: Posibles No Conformidades y Acciones Correctivas

No Conformidades	Acciones Correctivas
Incapacidad de adaptabilidad del componente en el hardware.	Verificar que el componente cumpla con los requisitos no funcionales de Hardware que se definieron en el artefacto Especificación de Requisitos. Insertarle alguna funcionalidad que permita solucionar el problema.
Incapacidad de adaptabilidad del componente en el Sistema Operativo.	Verificar que el componente cumpla con los requisitos no funcionales de Software que se definieron en el artefacto Especificación de Requisitos. Comprobar que todos los software que necesite el componente para su ejecución estén instalados en el Sistema Operativo.
Dificultades en la Instalación del componente software en el ambiente de Operación.	Verificar inexistencias de incompatibilidad de idioma entre el componente que se esté instalando y el ambiente de operación.
Problemas inesperados cuando se trabaja en conjunto con otro software.	Comprobar que el software al que se une el componente sea compatible con los requisitos no funcionales del componente.
Inconvenientes a la hora de sustituir el componente por otra versión del mismo.	Verificar que la versión sustituida cumpla con los requisitos funcionales que debe tener el componente, establecidos en el artefacto Especificación de requisitos del componente.

4.2 Informe: Evaluación de la Portabilidad

Este es el artefacto salida de la evaluación, en él se documenta todo el proceso y se redactan los resultados, en el Anexo 2 se muestra la estructura.

Procedimiento para evaluar la Portabilidad

Una vez elaborado, se debe incluir en el plan de Aseguramiento de la Calidad (PPQA) del proyecto al que se le aplicó el procedimiento.

Conclusiones Parciales

La elaboración de la propuesta de procedimiento en su totalidad tiene los elementos que se consideran importantes y necesarios para lograr que el proceso de evaluación de la Portabilidad tenga éxito.

CAPÍTULO 3

Validación del Procedimiento Propuesto

Introducción

En este capítulo se realiza la validación del procedimiento propuesto en el capítulo anterior. Se efectúa en dos componentes software utilizados en el proyecto Energía UCI del CEDIN ellos son: el componente HMI y el de Seguridad. Se le aplicará a cada componente por separado y luego se hará una comparación de los dos.

3.1 Descripción del Procedimiento

El procedimiento es aplicable a dos componentes software que se utilizan en el proyecto Energía UCI, estos dos componentes son reutilizables y pertenecen al SCADA-ux.

El propósito de la evaluación es detectar las posibles no conformidades y fallos de las sub-características (adaptabilidad, instalabilidad y coexistencia) de portabilidad que los componentes podrían tener y proponer acciones correctivas para el mejoramiento de las NC. Además de conocer el por ciento de portabilidad que posee cada uno de ellos.

El **rol** encargado de ejecutar el procedimiento para los dos componentes es el administrador de la calidad y los **atributos** a evaluar son:

- Adaptabilidad
- Instalabilidad
- Coexistencia

El atributo Reemplazabilidad no se evalúa debido a que no se ha realizado una nueva versión de ninguno de los dos componentes, ni se van a reemplazar por otros.

Según las sub-características y el rol definido anteriormente se **seleccionan las métricas** necesarias de cada una de las sub-características especificadas en la actividad anterior para evaluar la portabilidad.

Tabla. 3.1: Selección de métricas que se van a utilizar para realizar la evaluación

Sub-característica	Nombre de las métricas escogidas para realizar la evaluación.
Adaptabilidad	1.1 Adaptabilidad del componente software al hardware.
	1.2 Valor real de adaptabilidad en el Sistema Operativo
	1.2.1 Entornos de ejecución del componente.
	1.2.2 Adaptabilidad del componente software al Sistema Operativo probado.
	Obtener valor de una sub-característica específica.
Instalabilidad	2.1 Facilidad de instalación.
Coexistencia	3.1 Coexistencia disponible.
-----	Por ciento de Portabilidad.

Nota: Toda la información descrita anteriormente está definido en el Capítulo 2 como actividades del procedimiento para cada uno de los componentes, se decidió redactarlo general debido a que coincide la información en los dos casos, seguidamente se trabajará cada componente por separado.

3.1.1 Componente HMI

1 Consultar Nombre, Tipo, Estado y Fase del componente a evaluar.

Después de haberle hecho una entrevista al líder del proyecto donde se utiliza el componente se obtuvieron los siguientes resultados:

Nombre del componente: HMI (Human Machine Interface)

Tipo de componente: Componentes imperativos (Módulos), utilizando para su desarrollo programación Orientada a Objetos.

Estado del componente: Terminado

Fase del ciclo de vida: En el proyecto productivo donde se desarrolló el componente se rigen por el ciclo de vida que propone el Programa de Mejora, encontrándose actualmente en **Despliegue**.

HMI (Human Machine Interface): Permite la supervisión y control de forma gráfica de los procesos y el sistema, desde cualquier PC cliente se puede acceder a los datos que se encuentran en el servidor, este comprende todos los puntos de contacto entre el usuario y el equipo, de forma amena. La calidad del funcionamiento del componente depende en gran medida del abarrotamiento y velocidad de la red. El componente HMI tiene dos vistas una para la Edición y la otra para la Ejecución. **Ver Anexo 4**

2 Calcular Métricas

Adaptabilidad

1.1 Adaptabilidad del componente software al hardware. (Adaptabilidad para dispositivos de hardware y servicios de red)

$$X = 1 - 0/10 = 1$$

Se probaron 10 funcionalidades del componente pertenecientes al menú principal de la aplicación, teniendo éxito en todos los casos. La computadora donde se probó el componente para validar el procedimiento tenía las siguientes características de hardware: Procesador Intel(R) Pentium(R) Dual- Core CPU 3.00GHz, memoria instalada (RAM) 1 GB, disco duro 250 GB y motherboard Dell.

1.2 Entornos de ejecución del componente.

$$X = 1 - 0.2 \text{ según la escala definida en la tabla 2.3}$$

El componente HMI del proyecto Energía UCI solo se ejecutó en un sistema operativo y de hecho solamente está implementado para que pueda ejecutarse sobre Linux, distribución Debian – Lenny, por tanto esta métrica arroja bajos resultados debido a su incapacidad de poder adaptarse en varios entornos.

Los desarrolladores están trabajando para que en un futuro el software pueda correr en la distribución Debian- Squeeze pero hasta el momento no hay nada terminado.

1.2.1 Adaptabilidad del componente software al Sistema Operativo probado

$$X = 1 - 0 / 10 = 1$$

Se probaron las mismas 10 funcionalidades que se utilizaron para probar la adaptabilidad al hardware en la métrica 1.1, teniendo éxito en todos los casos, por consiguiente en la evaluación que se le hizo al componente en el sistema operativo, no dio problemas debido a que la computadora tenía las características necesarias para su correcto funcionamiento.

Métricas generales de la sub- característica adaptabilidad

Valor real de adaptabilidad en el Sistema Operativo

$$V_{real} = \frac{1+0.2}{2} = 0.6$$

En este caso no dio un valor bajo a pesar de que solo puede ejecutarse en un solo sistema operativo debido a que en este funciona correctamente.

Obtener valor de una sub-característica específica.

$$V_{Sub} = \frac{1 + 0.6}{2}$$

$$V_{Sub} = 0.8$$

El valor de la sub-característica **adaptabilidad** es de **0.8** para el componente HMI.

Instalabilidad

Nombre: Facilidad de Instalación

$$X = 1/1 = 1$$

El componente fue instalado sin problemas desde el primer intento que se hizo, para su instalación depende de varias librerías, con esto no hubo inconvenientes debido a que lo habían integrado con todo lo que necesitaba, cosa que se hacía automáticamente una vez que comenzaba la instalación.

Librerías más importantes que necesitaba:

Biblioteca AsynComm de Middleware 2.0

Biblioteca DataQuery de BDH

Bibliotecas de definiciones de datos de Middleware 2.0

Coexistencia

Nombre: Coexistencia Disponible

$X = 0/2 = 0 \text{ --- } 1$ según lo especificado en la tabla 2.3

El componente coexiste rápido y con éxito junto a otros módulos y librerías implementados debido a que este ya está integrado con los software que necesita para poder correr perfectamente. Ejemplo de módulos con los que puede coexistir: **Ver Anexo 4**

Módulo Histórico.

Módulo BDTR.

Módulo Seguridad.

Para probar el tiempo de duración funcionando con otro componente se probó juntamente con el HMI, lo que dio rápidos resultados.

Por ciento de portabilidad del componente HMI

$$P_{Pt} = \frac{0.8+1+1}{3} * 100 = \mathbf{93.33\%}$$

Tabla. 3.2: Componente HMI

Sub-característica	Nivel Requerido	Resultados	
		Valor Métrica	Nivel Real
Adaptabilidad	Alto	0.8	Alto
Instalabilidad	Alto	1	Alto
Coexistencia	Alto	1	Alto
Por ciento de Portabilidad		93.33%	

2 Analizar posibles resultados

En la tabla 3.2 se muestran los resultados de la evaluación que se le hizo al componente HMI perteneciente al proyecto Energía UCI, se muestra el nivel que requiere cada sub-característica y el nivel real que poseen, no denotando resultados negativos en ningún caso, por tanto no existen no conformidades y por consiguiente acciones correctivas para mitigarlas ni fallos existentes al realizar la evaluación, por lo que se concluye que dicho componente es adaptable, instalable y puede coexistir con varios software a la vez dando como resultado que posee un alto por ciento de portabilidad.

3.1.2 Componente Seguridad

1 Consultar Nombre, Tipo, Estado y Fase del componente a evaluar.

Nombre del componente: Seguridad.

Tipo de componente: Componentes imperativos (Módulos), utilizando para su desarrollo programación Orientada a Objetos al igual que el componente HMI.

Estado del componente: Terminado

Fase del ciclo de vida: **Ver fase del ciclo de vida del componente 1**, como estos dos componentes pertenecen al mismo proyecto los dos se encuentran en la misma fase.

Seguridad: Este componente provee las funcionalidades necesarias para garantizar el trabajo autorizado por usuarios, además brinda las herramientas para la protección contra ataques maliciosos o involuntarios al sistema por parte de personas o recursos tales como fallas de corriente, problemas de red o servidores, entre otros.

2 Calcular métricas

Adaptabilidad

1.1 Nombre de la Métrica: Adaptabilidad del componente software al hardware. (Adaptabilidad para dispositivos de hardware y servicios de red)

$$X = 1 - 0 / 1 = 1$$

Se probó una funcionalidad, la cual se ejecutó con éxito en el entorno de hardware, dicho entorno fue el mismo que el del componente anterior debido que se probó en la misma máquina. Ver componente HMI

1.3 Entornos de ejecución del componente.

X = 1 -- **0.2** según la escala definida en la tabla 2.3

El componente seguridad se puede ejecutar en un solo sistema Operativo al igual que el componente HMI. Ver componente HMI

1.2.1 Adaptabilidad del componente software al Sistema Operativo probado.

X = 1- 0/1 = **1**

Se probó una funcionalidad, la cual permitía el acceso del personal autorizado al proyecto, trabaja en el mismo sistema operativo y con las mismas características que en el componente anterior, la funcionalidad se ejecutó correctamente en el sistema operativo lo que demuestra que el módulo de seguridad cumple con su objetivo y funciona perfectamente.

Métricas generales de la sub-característica adaptabilidad para el componente Seguridad

Valor real de adaptabilidad en el Sistema Operativo

$$V_{real} = \frac{1+0.2}{2} = \mathbf{0.6}$$

Obtener valor de una sub-característica específica.

$$V_{Sub} = \frac{1 + 0.6}{2}$$

$V_{Sub} = \mathbf{0.8}$

El valor de la sub-característica **adaptabilidad** es de **0.8** para el componente seguridad.

Instalabilidad

Nombre: Facilidad de Instalación

X = 1/1 = **1**

El componente Seguridad al igual que HMI se instaló perfectamente de la primera vez que se probó. Este componente posee varias dependencias aún mayores que el HMI para su funcionamiento, pero como se encuentra integrado con todas ellas en un paquete no hubo problemas y fue muy fácil la instalación.

Coexistencia

Nombre: Coexistencia Disponible

El tiempo de duración del funcionamiento al mismo tiempo con otro software fue de 1 segundo

$X = 0 / 0.1 = 0$ -----1 según lo especificado en la tabla 2.3

Puede coexistir con varios módulos, de hecho es el encargado de la seguridad del sistema en general.

Ejemplo: Para probar esta métrica se abrió la interfaz del componente HMI en su modo ejecución, aquí había que introducirle primeramente un usuario válido para abrir esta ventana, lo que se hizo fue introducir los datos de un usuario que no tenía accesos permitidos y al segundo salió una ventana donde decía que el usuario no tenía permisos para acceder a la aplicación.

Por ciento de portabilidad del componente HMI

$$P_{Pt} = \frac{0.8+1+1}{3} * 100 = \mathbf{93.33\%}$$

Tabla. 3.3: Componente Seguridad

Sub-característica	Nivel Requerido	Resultados	
		Valor Métrica	Nivel Real
Adaptabilidad	Alto	0.8	Alto
Instalabilidad	Alto	1	Alto
Coexistencia	Alto	1	Alto
Por ciento de Portabilidad		93.33%	

3 Analizar posibles resultados

En la tabla 3.3 se muestran los resultados de la evaluación que se le hizo al componente Seguridad, las sub-características arrojaron un nivel alto y por consiguiente el por ciento de portabilidad general del componente es alto.

3.2 Comparación de Resultados

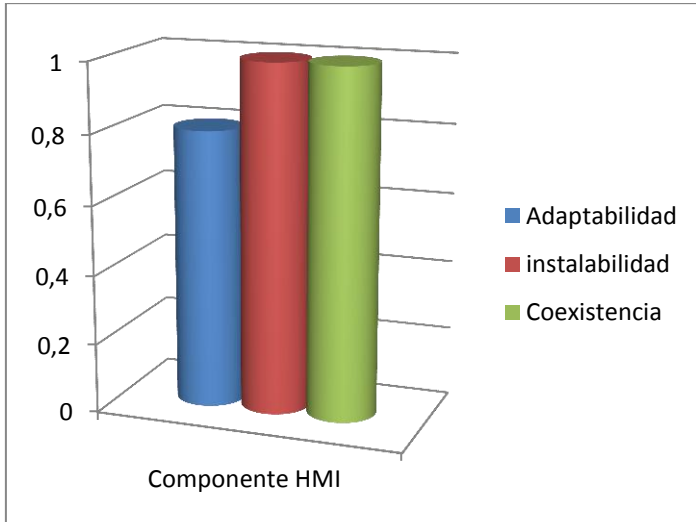


Ilustración 12: Resultados del componente HMI

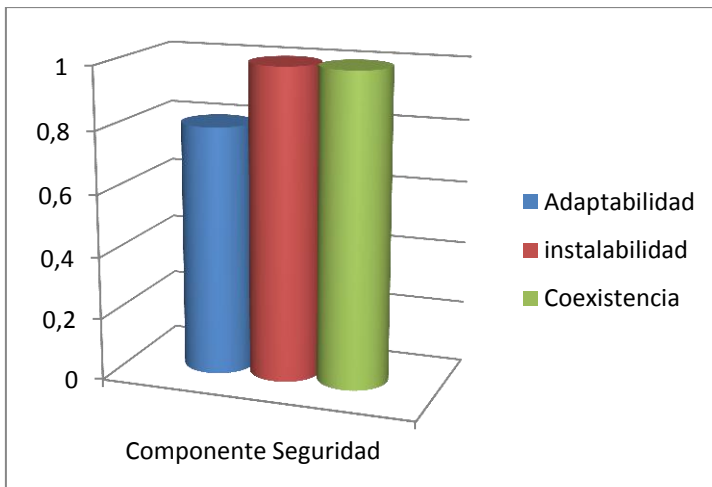


Ilustración 13: Resultados del componente Seguridad

Los componentes HMI y Seguridad poseen características similares a la hora de probar la portabilidad debido a que pertenecen al mismo proyecto, se encuentran en estado terminado y están integrados y listos para usarse con todos los software necesarios para que funcionen con éxito. Ambos son instalables, adaptables y pueden coexistir con otro software en el mismo entorno, por lo que se llega a la conclusión que poseen un alto por ciento de portabilidad. **Ver ilustración 14**



Ilustración 14: Por ciento de portabilidad de los componentes HMI y Seguridad

Conclusiones Parciales

En este capítulo se realizó la validación del procedimiento propuesto en el componente HMI y Seguridad del proyecto Energía UCI obteniendo resultados favorables en los dos casos, para ello se evaluaron los atributos de la portabilidad necesarios para realizar la evaluación y con estos conocer el por ciento de portabilidad de los componentes.

Conclusiones Generales

Luego de la investigación realizada sobre la evaluación de la portabilidad y con el fin de fortalecer la calidad de los componentes software, se le dio cumplimiento al objetivo general de la investigación, al desarrollar un procedimiento que permite evaluar la Portabilidad de los componentes software que se desarrollan en el CEDIN, logrando medir el por ciento de portabilidad que posee el componente evaluado.

Dicho procedimiento aportó a los administradores de la calidad y desarrolladores una guía paso a paso para facilitar su trabajo.

Para obtener dicho resultado se estudiaron un conjunto de modelos y guías que aportaron conocimientos relevantes para obtener la propuesta realizada.

Se aplicó el procedimiento en dos componentes que se desarrollaron en el proyecto Energía UCI, obteniéndose resultados satisfactorios para cada uno de ellos.

Recomendaciones

Para extender la investigación presentada en el presente trabajo de diploma se recomienda:

- Aplicar el procedimiento propuesto a todos los componentes que se desarrollen en el CEDIN para de esta forma contribuir a mejorar la calidad en cuanto a la portabilidad de los componentes desarrollados o que se encuentren en desarrollo.
- Elaborar una herramienta que permita gestionar el proceso de evaluación de la propuesta realizada.
- Confeccionar una comisión integrada por personal capacitado del Grupo de Aseguramiento de la Calidad del CEDIN, encargada de evaluar la Portabilidad de los Componentes Software que se desarrollen.

Referencias Bibliográficas

1. **Real Academia Española.** [En línea] [Citado el: 17 de 10 de 2010.] <http://buscon.rae.es/drael/>.
2. **Mooney, James. D.** (1997). *Bringing Portability to the Software Process*.
3. **Ashrafi.** The impact of software process improvement on quality: in theory and practice. [En línea] [http://dx.doi.org/10.1016/S0378-7206\(02\)00096-4](http://dx.doi.org/10.1016/S0378-7206(02)00096-4) .
4. **ISO, 9126.** “*Software product evaluation – Quality characteristics and guidelines for their use*”. 2005.
5. **Diccionario de Informática.** “Portabilidad”. Página 254. Editorial Cultural. 1999. Madrid, España. ISBN 84-8055-256-5
6. **Pressman.** (1998). *Aspectos y métricas de la calidad del Software*.
7. **1992.** *IEEE Standard for Software Quality Management System*. 1992.
8. **Estévez, Isabel Pérez. 2002.** *MÉTRICAS PARA EL CONTROL DE PROYECTOS*. 2002.
9. **14598, ISO. 1999.** *Evaluación de los productos de software*. 1999.
10. **Szypersky, Clemens. 2002.** “*Component Software. Beyond Object – Oriented Programming*”. 2002.
11. **Real Academia Española.** [En línea] [Citado el: 10 de 3 de 2011.] <http://buscon.rae.es/drael/>.
12. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software*.
13. **IEC – International Electrotechnical Commission.** [En línea] [Citado el: 12 de 3 de 2011] <http://www.iec.ch>
14. **ISO – International Organization for Standardization.** [En línea] [Citado el: 12 de 3 de 2011] <http://www.iso.org>
15. **Alicia del Carmen Pérez, J. L.** (2010). “*Herramienta para evaluar la portabilidad de los datos en el Desarrollo de Software*”.
16. **Annier Jiménez, G. G.** (2010). *Herramienta para evaluar la portabilidad del código fuente en las aplicaciones web*.

Referencias Bibliográficas

17. **Escobar, Lic. Miguel Angel Paco.** 2003. Calidad del software y su importancia en el mercado. 2003.
18. **S., Viridiana Figueroa.** Slideshare. [En línea] [Citado el: 5 de 2 de 2011.]
<http://www.slideshare.net/viri88/ingenieria-de-dominio>.
19. **Herrero, Toñi.** PC WORLD digital. [En línea] [Citado el: 6 de 5 de 2011.]
<http://www.idg.es/pcworld/Los-fallos-del-software:-unos-defectos-muy-caros/art141458.htm>.

Bibliografía

0516_Roles y responsabilidades. *Calisoft UCI, 2009.*

14598, ISO. 1999. *Evaluación de los productos software.* 1999.

Escobar, Lic. Miguel Angel Paco. 2003. *Calidad del software y su importancia en el mercado.* 2003.

García., Oscar Gómez. *SQuaRE: Una unificación de normas para la especificación de requisitos y la evaluación de la calidad.* Enero de 2009.

ISO, 9126. “*Software product evaluation – Quality characteristics and guidelines for their use*”. 2005.

Ivar Jacobson, Grady Booch, James Rumbaugh. *El Proceso Unificado de Desarrollo de Software.*

Mooney, J. D. (1997). *Bringing Portability to the Software Process.*

Mooney, James D. *Issues in the Specification and Measurement of Software Portability.*

Rojas, Maribel Ariza y García, Juan Carlos Molina. 2004. *INTRODUCCIÓN Y PRINCIPIOS BÁSICOS DEL DESARROLLO DE SOFTWARE BASADO EN COMPONENTES.* 2004.

Anexos

Anexo 1: Entrevista al líder de proyecto al que pertenece el componente a evaluar.

Esta pequeña entrevista tiene como objetivo recopilar toda la información inicial necesaria para evaluar la portabilidad de los componentes software que se desarrollan en el CEDIN.

Pregunta 1:

Nombre del Componente Software:

Pregunta 2:

Tipo de Componente Software

____ Orientado a Objeto.

____ Imperativos (Módulos).

Otros: _____

Pregunta 3:

Estado del Componente Software:

____ Terminado.

____ En Desarrollo

Pregunta 4:

Fase del ciclo de vida que se encuentra el Componente Software:

Nota: El procedimiento está realizado siguiendo la metodología RUP debido a que es por la que se rigió el programa de mejora de la universidad para definir su ciclo de vida.

En caso que no se utilice metodología RUP especificar:

Metodología: _____

Fase del ciclo de vida: _____

Anexo 2: Informe de Evaluación de la Portabilidad



1 Descripción del componente que se evaluará

<Breve descripción del componente que se evaluará>

- Consultar Nombre, Tipo, Estado y Fase del componente a evaluar.

<Se consulta el nombre, tipo, estado y fase del componente que se evaluará mediante una entrevista al líder del proyecto donde se desarrolla el componente>

- Especificar propósito de la evaluación

<El propósito de la evaluación es identificar los posibles errores de las sub-características [adaptabilidad, instalabilidad, coexistencia y reemplazabilidad] que el componente podría tener y proponer acciones correctivas para el mejoramiento de las mismas>

- Identificar el rol que ejecuta la evaluación

<Se especifica el rol que ejecuta el procedimiento>

- Identificar los atributos de Portabilidad a evaluar por fases según el rol

<Se identifican los atributos de portabilidad que se vayan a evaluar según las necesidades del rol encargado de ejecutar el procedimiento, los atributos son:

Adaptabilidad, instalabilidad, coexistencia y reemplazabilidad>

2 Especificar métricas

<Se procede a escoger las métricas necesarias para cada sub-característica, en la siguiente tabla se realiza esta actividad>

Sub-característica	Nombre de las métricas escogidas para realizar la evaluación

3 Ejecución de la evaluación

<Se procede a ejecutar la evaluación del procedimiento>

- **Calcular métricas**

<Se realiza el cálculo de las métricas seleccionadas.

En la siguiente tabla se recoge todo el proceso de evaluación>

Tabla: Ejecución de la evaluación

Sub-característica	Nivel Requerido	Resultados	
		Valor Métrica	Nivel Real
Por ciento de Portabilidad			

4 Analizar posibles resultados

<Se analizan los resultados dando las conclusiones de la evaluación efectuada. **VER Etapa 4 del procedimiento epígrafe 4.1 para mayor detalle de lo que se debe especificar en esta actividad**>

5 Incluir en el plan de mediciones las métricas escogidas para realizar la evaluación.

Anexo 3: Ciclo de vida de proyectos pilotos del programa de mejora

Estudio Preliminar
Modelación del Negocio
Requisitos
Análisis y Diseño
Implementación
Pruebas Internas
Pruebas de Liberación
Despliegue
Soporte

Anexo 4: Componente HMI del proyecto Energía UCI

Ilustración 15: Vista Edición del HMI

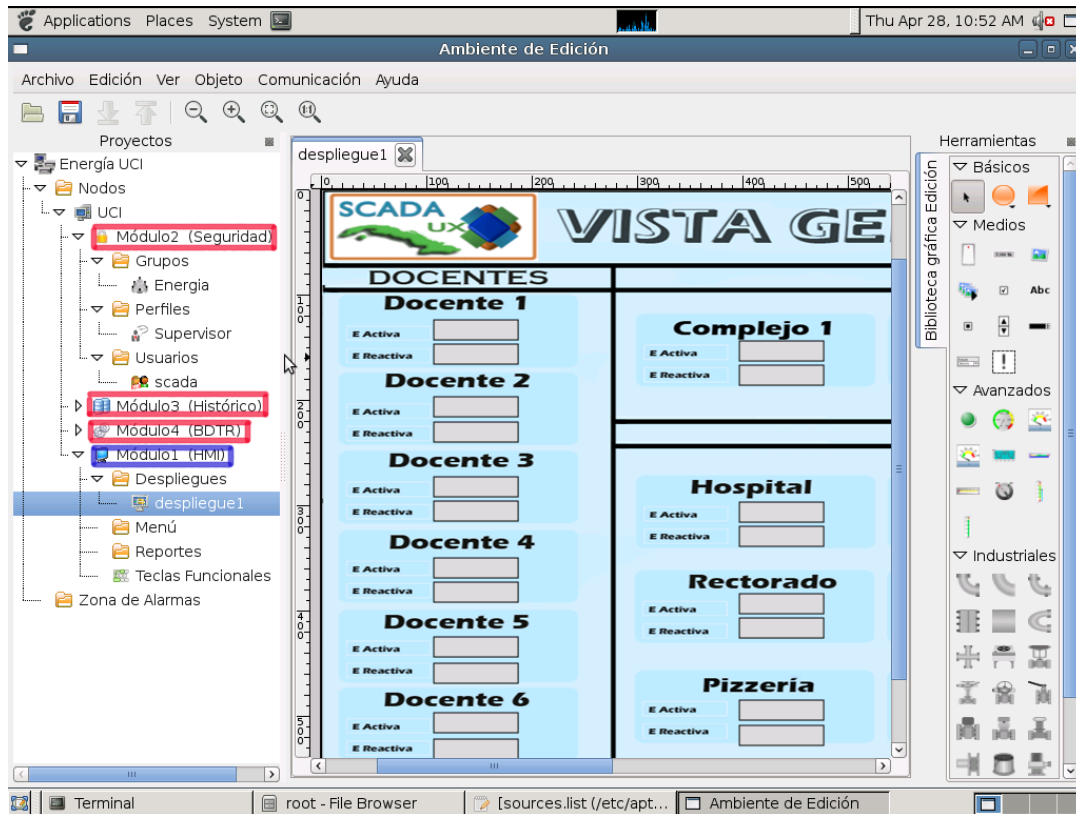


Ilustración 16: Vista Ejecución del HMI

