Universidad de las Ciencias Informáticas

Facultad 15

Título: Herramienta para evaluar la portabilidad del código fuente en las aplicaciones Web.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor: Annier Jiménez Plasencia

Autor: Gioanny González Roche

Tutor: Ing. Pedro Enrique Castiñeiras Sánchez

Cotutor: Ing. Julio Cesar Prieto Álvarez

Año del 51 Aniversario del Triunfo de la Revolución

Ciudad de la Habana

Junio de 2010

Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte"

Ernesto Che Guevara

30 de septiembre de 1960

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado: Herramienta para evaluar la portabilidad del código fuente en las aplicaciones Web. Y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Gioanny González Roche

_____

Firma del Autor

Annier Jiménez Plasencia

_____

Firma del Autor

Pedro Enrique Castiñeiras Sánchez

_____

Firma del Tutor

Julio Cesar Prieto Álvarez

_____

Firma del Cotutor

DATOS DE CONTACTO

Ing. Pedro Enrique Castiñeiras Sánchez.

Titulo: Ingeniería en Ciencias Informáticas.

Categoría Docente: Adiestrado.

Queremos agradecer:

A todas aquellas personas que aportaron su granito de arena en la realización de nuestra tesis.

A la Revolución por habernos dado la oportunidad de formar parte de un proyecto de tal magnitud y permitirnos realizarnos como profesionales.

Principalmente a nuestros padres, hermanos, tíos, abuelos en fin a todas nuestras familias que sin ellos no seríamos lo que hoy somos y no hubiéramos llegado hasta aquí. Gracias por su apoyo incondicional y por confiar siempre en nosotros.

A todas aquellas personas que de una forma u otra formaron parte de nuestro desarrollo como futuros profesionales.

A nuestros tutores Pedro y Julio por su valiosa colaboración y ayuda

A nuestros amigos por sus ayudas en los momentos más difíciles en el transcurso de nuestra carrera.

A todos muchas gracias.

Gioanny Y Annier

Gioanny:

Dedico este trabajo a mi padre, el cual siempre ha estado conmigo en los momentos difíciles de mi vida por mostrarme el camino y a iniciarme en él y luego continuarlo, a ese hermano Luis y a mi hermana Vianna que mucho tengo que agradecerles lo que hoy soy, ellos que tanto me inspiran para seguir adelante, mi Madre por entenderme y apoyarme en todo momento por darme apoyo moral brindado con infinito amor y confianza. A una persona que ya no se encuentra con nosotros y que no pude despedirme de ella físicamente mi abuela Delia. A toda mi familia es la mejor del mundo, a mis amigos y amigas que lucharon conmigo durante cinco duros años, a todas aquellas personas que me ayudaron cuando más lo necesité. Espero que nunca el destino me haga pasar por ingrato y poder agradecerles a todas esas personas las cuales me ayudaron a llegar aquí.

Annier:

Dedico este trabajo a mi querida y hermosa Madre que me ha dado todo su amor y cariño, por aconsejarme y ayudarme para que triunfe en la vida, y por decirme siempre que estudie que algún día se lo agradeceré. A mi Padre por sacrificarse tanto para ayudarme y mostrarme el camino correcto que se debe tomar en la vida. A mí querido hermano Andrey que quiero y siempre voy a querer porque para mí tú eres el mejor. A mis abuelos Teresa, Pedro y también Bernaldo que ya no está con nosotros. A toda mi familia por darme su apoyo y cariño. A mis amigos y amigas por brindarme su ayuda durante todo este tiempo. A una persona muy especial para mí, Saide, mi linda y querida novia, a quien amo con todo mi corazón y a quien siempre le voy a estar agradecido por enseñarme las cosas lindas que tiene el amor. Para todos ustedes va dedicado el fruto de todo este sacrificio.

Resumen

En este trabajo se describe el proceso de creación de una herramienta para la evaluación de la portabilidad del código fuente de las aplicaciones Web desarrolladas en la Universidad de las Ciencias Informáticas (UCI). Esta investigación está centrada principalmente a las aplicaciones desarrolladas en entornos como Windows desde su versión Windows 2000 hasta la fecha, además de Debian y Ubuntu 9.4, también se basará en los lenguajes de programación y frameworks más utilizados actualmente para el desarrollo Web en la UCI. Todas estas tecnologías que se abarcan en la investigación fueron tomadas después de un análisis y estudio realizado en los proyectos que se encuentran en desarrollo hoy por hoy en la Universidad, determinando cuáles son las que más se utilizan en los proyectos. La herramienta desarrollada abarca una serie de métricas definidas, las cuales contemplan los principales problemas que afectan la portabilidad del código fuente presentes en las tecnologías de desarrollo Web que generalmente se usan en la UCI.

La herramienta, para cumplir los objetivos propuestos, además de evaluar la portabilidad de un código fuente de acuerdo a métricas definidas anteriormente, muestra un reporte de todos aquellos problemas presentes en el código en cuestión que fueron detectados, así como brinda una estimación de qué por ciento de portabilidad y de gravedad presenta el código de la aplicación que se está evaluando. También está ajustada a los lenguajes, frameworks y plataformas a las cuales va dirigida, además de que es fácil de interpretar y de usar. Todo este proceso se tornará mucho más agradable para quien la utilice y le ahorrará tiempo en el proceso de evaluación de una aplicación Web.

Tabla de contenido

Introducción

Actualmente la necesidad del desarrollo de las tecnologías, la informática y las comunicaciones está presente en todo el mundo, impulsadas para lograr un mejoramiento en la economía de los países del tercer mundo y con esto elevar el nivel de vida de sus habitantes. Este progreso, a gran escala requiere mayor seguridad, disponibilidad y confiabilidad en los productos, menor costo de los mismos para su creación, así como que estos puedan ser utilizados en cualquier ambiente de desarrollo brindando una mayor funcionalidad y eficiencia de los mismos. La Universidad de las Ciencias Informáticas no está apartada del cumplimiento de estos requerimientos en los productos que comercializa, para así de esta manera, colaborar con el desarrollo de nuestro país.

La mayor parte de los productos que se desarrollan en la UCI son aplicaciones Web, ya que el buen uso de estas aplicaciones y la variedad de funcionalidades que estas brindan, facilitan el manejo de las informaciones entre las empresas y sus clientes, brindándole la posibilidad a estos de realizar transacciones a través de internet o de una red local. Cuba se encuentra en un constante avance en el campo de la informática y las comunicaciones, por lo que no desecha la idea de utilizar la Web como medio de comunicación, lo que trae consigo como resultado, que las relaciones entre las empresas, clientes y proveedores, se están efectuando mayormente a través de redes informáticas, mediante aplicaciones Web y esta tendencia de comunicación está aumentando cada día más.

Una aplicación Web debe contar con algunas cualidades indispensables como son: seguridad, funcionalidad, usabilidad, disponibilidad y portabilidad. De estas características la portabilidad es la que con más frecuencia cae en el olvido. Esto actualmente causa serios problemas, ya que la portabilidad es la característica que posee un software para ejecutarse en diferentes plataformas con el mínimo de cambios. Además el código fuente del software puede ser reutilizable en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra, por lo que a medida que se tiene una mayor adaptabilidad del código fuente en nuevos ambientes, traería consigo que el software fuera portable a cierto nivel en estos ambientes, lo que provocaría mejorar la relación coste-eficacia.

En la actualidad cuando se hacen pruebas de calidad a las aplicaciones Web que se desarrollan en la UCI no se cuenta con una herramienta que permita determinar el grado de portabilidad de una aplicación Web, lo cual arroja inconformidades de los clientes cuando utilizan estas aplicaciones en diferentes plataformas, e incluso, cuando en etapas de desarrollo es necesario cambiar de plataforma o de versión de las herramientas que se estén utilizando para su desarrollo. Consecuentemente es necesario crear una herramienta que permita determinar el grado de portabilidad de una aplicación Web.

Por lo tanto, considerando la necesidad de la Universidad y la problemática anteriormente mencionada se platea resolver el siguiente **Problema científico:** ¿Cómo evaluar la portabilidad del código fuente en las aplicaciones web que se desarrollan en la UCI?

Esto nos lleva a centrar nuestro **Objeto de estudio** en: El proceso de desarrollo de software. Y se tiene como **Campo de acción**: Portabilidad del código fuente en las aplicaciones Web que se desarrollan en la UCI.

El Objetivo de la investigación es: Elaborar una herramienta que permita evaluar la portabilidad del código fuente de las aplicaciones Web desarrolladas en la UCI.

Se concreta como **Idea a defender** que:

Con la creación de una herramienta que permita evaluar la portabilidad del código fuente de las aplicaciones Web, se podrá contribuir a mejorar el proceso de evaluación de las aplicaciones Web que se desarrollan en la UCI.

Se han definido las siguientes **Tareas de investigación** para dar cumplimiento al objetivo de este trabajo:

- ✓ Estudiar el estado del arte de las herramientas que permitan evaluar la portabilidad del código fuente de las aplicaciones Web.
- ✓ Analizar los aspectos más significativos en la portabilidad de las aplicaciones Web y su repercusión en la calidad de dichas aplicaciones.
- ✓ Investigar y determinar las principales tecnologías que se utilizan para el desarrollo Web en la UCI.
- ✓ Elaborar métricas de portabilidad del código fuente para aplicaciones Web
- ✓ Desarrollar una herramienta que permita evaluar la portabilidad de las aplicaciones Web que se desarrollan en la Universidad de las Ciencias Informáticas
- ✓ Evaluar la herramienta que se desarrolle.

El presente trabajo de diploma está estructurado en 3 capítulos:
Capítulo Nro.1: Fundamentación Teórica de la investigación.
Capítulo Nro.2: Métricas para evaluar la portabilidad del código fuente de aplicaciones Web.
Capítulo Nro.3: Desarrollo y validación de una herramienta para evaluar la portabilidad.

1.1. Introducción

En el presente capítulo se realiza un estudio del estado del arte sobre la evaluación de la portabilidad del código fuente en aplicaciones Web en la UCI, a nivel nacional e internacional, así como una descripción y comparación de los diferentes conceptos de calidad ya definidos por diferentes personalidades e instituciones de gran relevancia en el tema. Se mencionan los elementos que pueden dar una medida de la calidad de un software como son las normas, estándares y modelos de calidad por los que se rigen los desarrolladores de aplicaciones Web. Se analizan los parámetros para medir la calidad en un producto de software profundizando en el indicador portabilidad en las aplicaciones Web.

Con la presente investigación se puntualizan las tecnologías que se usan en la actualidad para el desarrollo de las aplicaciones Web en la UCI, lo cual brinda la oportunidad de realizar un estudio de los lenguajes, plataformas, frameworks, y otras herramientas utilizadas en los proyectos para su desarrollo, así como las potencialidades de estos.

1.2. Calidad

El concepto de calidad describe una forma de satisfacer al cliente mejorando, día a día, los procesos y resultados de un producto. En la actualidad la definición de Calidad ha evolucionado hasta convertirse en una forma de gestión que introduce el concepto de mejora continua en cualquier organización y a todos los niveles de la misma. La calidad se puede mirar desde dos puntos de vista. La calidad externa, que se dirige a la satisfacción del cliente. Y la calidad interna, que tiene como propósito implementar los medios para permitir la mejor descripción posible de la organización así como detectar y limitar los funcionamientos incorrectos de los productos.

La calidad en términos generales se puede definir como la satisfacción plena de las necesidades del cliente, de manera tal que se cumplan las expectativas de este y algunas más, creando productos y servicios de acuerdo a las normas establecidas y con la menor cantidad de defectos posibles, además de dar respuesta inmediata a las solicitudes de los clientes.

Varias son las definiciones que le han dando personalidades de gran prestigio internacional a la calidad, a los cuales les avala toda una experiencia en esta área. Pero no solo son personalidades las que definen este concepto sino también organizaciones rectoras de estándares que hacen una valoración sobre cómo se define la calidad como concepto, a continuación se proponen algunas de las definiciones que se abordan en este epígrafe:

Según **la norma ISO[1] 8402-UNE 66-001-92** la calidad es un conjunto de características de una entidad que le confiere la aptitud para satisfacer las necesidades establecidas y las implícitas. (Ing. Roberto Hugo Vázquez, 2006)

La norma ISO 9000:2000 plantea que la Calidad es el grado en el que un conjunto de características inherentes cumple con los requisitos. (Pillou, 2004)

Por su parte **Edwards Deming**[2] plantea que "la calidad no es otra cosa más que "Una serie de cuestionamientos hacia una mejora continua". (Saavedra, 2005)

Después de un estudio de las distintas definiciones de calidad, se decidió que esta investigación se acogerá al concepto que más se acomoda para su desarrollo. Varias son las definiciones hacia el punto de vista de lo que piensa el cliente y no a las funcionalidades del producto, el problema de calidad viene estrechamente relacionado tanto desde el punto de vista del cliente como del producto, pues los productos se crean para que cumplan las expectativas de los clientes.

En la definición de Edwards Deming se trata la calidad como el corregimiento de errores que pueda tener un producto en su desarrollo hasta intentar la perfección, trata los productos como contenedores de errores y no se tiene en cuenta que existe la posibilidad de que no se logre la perfección impidiendo así lograr la calidad total de un producto. La norma ISO 9000:2000 la cual es una nueva revisión de la norma ISO 8402-UNE 66-001-92 plantea que existe una concordancia entre las características de un producto y su calidad, dejando claro que las características son las que definen la calidad del producto y con este cumpliendo las expectativas del cliente. Para la presente investigación la definición más aceptada es la planteada por la norma ISO 9000:2000, cumpliendo con las expectativas de poder tener implícito y explícito los requisitos del producto.

1.2.1. Normas, estándares y modelos de calidad

Un producto con calidad implica la utilización de elementos y características para llevar una medida del nivel de calidad con que se debe trabajar, estos elementos tienen diferentes clasificaciones, una de estas son ***las normas,*** las cuales se definen como reglas previamente definidas y establecidas, aprobadas y certificadas por una organización reconocida internacionalmente, estas deben ser respetadas una vez que se rigen por ellas. (Capo, 1998)

---

[1] ISO: Organización Internacional para Normalización

[2] Edwards Deming: Estadístico estadounidense, profesor universitario, consultor y difusor del concepto de calidad total.

Otro de los mecanismos que se utilizan son ***los estándares,*** definidos como principios orientadores o guías para la evaluación, por lo general; identifican prácticas sobre las cuales existen acuerdos o conocimientos generalizados de aceptabilidad, proponen pautas que reflejan la mejor práctica vigente o tendencia y contienen precauciones, recomendaciones o alertas contra errores potenciales. (Pulido, 2004)

Por último el otro de los mecanismos que se utiliza son ***los Modelos,*** los cuales no son más que arquetipos o puntos de referencias a seguir o imitar de modo tal que se puedan comprender las situaciones más complejas, pero la definición más aceptable es un conjunto de prácticas vinculadas a los procesos de gestión y el desarrollo de proyectos. Estos suponen una planificación para alcanzar un impacto estratégico, cumpliendo con los objetivos fijados en lo referente a la calidad del producto. Los modelos de calidad te dicen qué hacer no cómo hacerlo, debido a que esto depende de las metodologías que se usen así como de los objetivos del negocio. Logrando de esta manera mejorar la calidad de los procesos o productos elaborados. (Quiñones, 2009)

En este punto se ha dejado claro que tanto las normas, los estándares y los modelos son importantes a la hora del crecimiento y madurez de la calidad de las empresas, así como de sus procesos y servicios.

1.3. Calidad de Software

En la actualidad todas las instituciones y empresas que buscan el desarrollo en sus producciones necesitan que sus productos posean una calidad con la cual podrán obtener un prestigio a nivel mundial. La calidad de software es la línea que siguen los clientes hoy en día, ya que estos para adquirir cualquier producto para sus beneficios, primero tienen en cuenta sus cualidades. Muchos han sido los profesionales que han querido profundizar en este sentido e intentando evaluar la calidad de los productos que se diseñan puesto que las interrogantes de cualquier empresa o institución desarrolladora de software es ¿cómo podemos desarrollar un software con calidad? Para dar solución a parte de esta interrogante se comenzará a estudiar qué se entiende por calidad de software, que plantean las grandes compañías de estándares y las grandes figuras que se han especializado en esta esfera. A continuación se presentan los principales conceptos estudiados y se definirá por cual se guiará esta investigación.

Roger Pressman[3] plantea que la calidad de software es la concordancia entre el software producido y los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario. (Pressman, 1998)

Según **The Institute of Electrical and Electronics Engineers (IEEE[4])** la calidad se define de dos formas, el grado en que un sistema, componente o proceso cumple con lo especificado y el grado en que un sistema, componente o proceso cumple con los clientes o las necesidades del usuario y las expectativas de este. (IEEE, 1992)

La norma ISO/IEC DEC 9126 define la calidad de software como la totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas (ISO 9000:2000)

Capability Maturity Model Integration (CMMI[5]) define como calidad de software, la satisfacción de las necesidades y expectativas del cliente otorgando a éste seguridad sobre su uso, fiabilidad de sus funciones esperadas, confianza en un producto o servicio sin fallos y duradero según tiempos establecidos y acordados. (Contreras, 2008)

El Dr. Tom DeMarco[6] plantea que la calidad de un producto es una función de cuánto cambia el mundo para mejor (DeMarco, 1999).

La IEEE se basa principalmente en la calidad orientada a los clientes, por lo que prioriza el punto de vista de cómo este ve al producto, en el caso del Dr. Tom DeMarco se refiere a que hay que satisfacer al cliente sin importar los cambios que traerá consigo. Pressman hace unas definiciones muy acertadas acerca de la calidad de software, sobre cómo debe concebirse, junto con él también lo hace la norma ISO/IEC DEC 9126, ambos coinciden con lo mencionado por CMMI, pero este último habla en su definición además sobre algunos rasgos significativos como seguridad, expectativas, fiabilidad que se ajusta a los rasgos

[3] Roger Pressman: Ingeniero de Software, autor y consultor de importantes libros, es el presidente de la RS Pressman & Associates

[4] IEEE: Asociación más grande del mundo profesional, dedicada al avance de la innovación tecnológica y excelencia en beneficio de la humanidad

[5] CMMI: Modelo para la mejora y evolución de procesos para el desarrollo y mantenimiento de sistemas de software

[6] Tom DeMarco: estadounidense, Ingeniero de software, profesor y difusor de la ingeniería de software. Fue uno de los desarrolladores de análisis estructurado en la década de 1980

antes expuestos que debería tener un software o que debe cumplir así como con la misma idea de que un producto de software satisfagan las necesidades implícitas o explicitas de los clientes, rasgo distintivo que lleva a elegir este concepto como guía para el desarrollo de este trabajo.

1.3.1. Aspectos para medir de la calidad del software

Para obtener una medida de la calidad de un producto hay que tener en cuenta que esta se obtiene atendiendo a tres grupos y a su vez estos están definidos por distintos aspectos.

Aspectos para medir la Calidad (Lovelle, 1999)

1. **Operaciones del producto:** Trata las características operativas del producto.

1.1. **Corrección** (¿Hace lo que se le pide?)

- ✓ El grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente.

1.2. **Fiabilidad** (¿Lo hace de forma fiable todo el tiempo?)

- ✓ El grado que se puede esperar de que una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida.

1.3. **Eficiencia** (¿Qué recursos hardware y software necesito?)

- ✓ La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.

1.4. **Integridad** (¿Puedo controlar su uso?)

- ✓ El grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.

1.5. **Facilidad de uso** (¿Es fácil y cómodo de manejar?)

- ✓ El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.

1.6. **Disponibilidad** (¿Puedo acceder a la aplicación?)

- ✓ El grado o el período de tiempo en que está disponible para ser usado por los usuarios autorizados.

2. **Revisión del producto:** Trata sobre la capacidad del producto de soportar cambios.

2.1. **Facilidad de mantenimiento** (¿Puedo localizar los fallos?)

- ✓ El esfuerzo requerido para localizar y reparar errores.

2.2. **Flexibilidad** (¿Puedo añadir nuevas opciones?)

- ✓ El esfuerzo requerido para modificar una aplicación en funcionamiento.

2.3. **Facilidad de prueba** (¿Puedo probar todas las opciones?)

- ✓ El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos.

3. **Transición del producto:** Trata sobre la adaptabilidad del producto en nuevos entornos

3.1. **Portabilidad** (¿Podré usarlo en otra máquina o en otro entorno?)

- ✓ El esfuerzo requerido para transferir la aplicación a otro ambiente o sistema operativo.

3.2. **Reusabilidad** (¿Podré utilizar alguna parte del software en otra aplicación?)

- ✓ Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones.

3.3. **Interoperabilidad** (¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?

- ✓ El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos

1.4. Métricas de calidad

Las métricas son medidas que se le aplica a un producto, estas medidas proporcionan una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto" [Pressman´98].

La IEEE en "Standard Glosary of Software Engering Terms" define métrica como una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado.

Atendiendo a los conceptos antes mencionados se concluye que una métrica es una forma de evaluar y una escala, definidas para realizar mediciones de uno o varios atributos.

1.4.1. Métricas de calidad de un producto 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." (Estévez, junio de 2002)

Se puede afirmar que las métricas aportan una manera de estimar la calidad de los atributos internos del producto, 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. Todo este esfuerzo sería de mucho beneficio ya que habrá disminución de errores durante el proyecto, con esta idea también se podrá controlar el proyecto y alcanzar la seguridad de la planificación de los problemas antes de que ocurran, conseguiremos minimizar el esfuerzo en la planificación del proyecto. Aún así debería ser posible desarrollar medidas para evaluar diferentes cualidades internas del programa.

1.5. Aplicaciones Web

Son aplicaciones utilizadas por los usuarios permitiéndoles a estos acceder a servicios que brinda un servidor Web a través de Internet o de una intranet local mediante un navegador. Se puede definir también como una aplicación de software que se codifica en un lenguaje soportado por los navegadores Web en la que se confía la ejecución al navegador. La Web surge debido a los cambios ocurridos dentro de las estructuras de las empresas, las cuales necesitaron una comunicación directa entre sus clientes y proveedores a través de la red, lo que significa que se necesitaba un sistema, en este caso una aplicación Web que los comunique entre sí para facilitar su comercio.

Las aplicaciones Web son populares debido a la facilidad para actualizar y mantenerlas sin tener que distribuir e instalar software a miles de usuarios. Una página Web puede contener elementos que permiten la comunicación activa entre el usuario y la información, accediendo a los datos de modo interactivo, también se destacan producto a la conectividad que proporcionan a Internet, permitiendo el acceso a la aplicación desde cualquier punto. Esto permite ahorrar en costosas inversiones para establecer comunicaciones que requieren un alto presupuesto en su instauración. Sin duda, las aplicaciones Web tienen aún un largo camino por recorrer, pero son ya una opción muy interesante, especialmente, en tareas donde tanto el intercambio de información como las negociaciones juegan un papel predominante y los usuarios se encuentran dispersos en el mundo del mercado y las comunicaciones. (Hooping.net, 2008)

Con la creación de la Web se ha logrado cambiar todas aquellas trabas que existían en la comunicación gracias a la colaboración de personas en todo el mundo, Internet ahora con el surgimiento de las aplicaciones Web es más amigable e intuitiva y es considerado como la **"*opción número uno de comunicación en el mundo*".**

1.6. Portabilidad de software

La portabilidad puede ser vista en las oficinas por ejemplo, aplicada a los datos, a la información o documentación que se quiere tener encima cuando se necesita ir a una reunión y así muchos son los ejemplos que pueden expresar la portabilidad, cuando se habla sobre este indicador en la calidad de 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 definen 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. (Mooney, 1997)

Son muchas las estrategias que deben tener en cuenta los proyectos para lograr un producto con portabilidad. Antes de comenzar a desarrollarlo deben tener definidos prioridades para alcanzar un mejoramiento en este indicador, aislar las dependencias entre clases, pensar y tener una mente clara a la hora de desarrollar teniendo en cuenta los parámetros correctos así como algo importantísimo que es el control de las interfaces.

Deben también tenerse en cuenta los lenguajes, las librerías y las funciones que se utilizan, ya que mientras más se desarrollen en lenguajes que son en un gran porciento portable esto estaría ayudando a 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.

Otro aspecto que se debe tener en cuenta también es las bibliotecas que se utilizan para el desarrollo del mismo, realizar un análisis profundo para la utilización de las mismas y valorar en qué sentido puede afectar esas bibliotecas en la portabilidad, en caso de que no la afectaran no existirían inconvenientes; pero en caso contrario se deben definir las alternativas para solucionar los problemas, y que el software no pierda las funciones ni la portabilidad así como no afecten también los demás indicadores de calidad.

Los arquitectos también juegan un papel importantísimo, ya que desde el principio deben definir los distintos estándares de implementación para garantizar una guía segura y dirigida a la portabilidad del software, así como realizar un estudio de los distintos tipos de arquitectura y seleccionar la más adecuada según las características, así como la selección del lenguaje adecuado para su implementación. Teniendo en cuenta todas estas razones enunciadas, en el desarrollo de software se puede obtener un producto con la mayor portabilidad posible.

1.7. Plataformas de software

Una plataforma es una combinación de hardware y software usados para ejecutar diferentes tipos de aplicaciones; en su forma más simple consiste únicamente de un sistema operativo, una arquitectura, o una combinación de todos ellos. La plataforma más conocida es probablemente Microsoft Windows en una arquitectura x86; otras plataformas conocidas son GNU/Linux[7], J2EE y Mac OS X[8]. El software en

[7] GNU/Linux: Termino promovido por la FSF (Free Software Fundation) para el sistema operativo que incluye las utilidades de esta y el núcleo Linux

[8] Mac OS X: Sistema operativo desarrollado y comercializado por la empresa Apple Inc.

general está escrito de modo que dependa de las características de una plataforma en particular; bien sea el hardware, sistema operativo o máquina virtual en que se ejecuta.

El software constituye el conjunto de programas, instrucciones y lenguajes que permiten al sistema la ejecución de múltiples tareas. Las plataformas de software pueden ser un sistema operativo, un entorno de programación, o más comúnmente como se ve en la actualidad una combinación de las dos. Una excepción notable es el lenguaje de programación Java, que usa una máquina virtual independiente del sistema operativo para leer el código.

Gracias a los resultados obtenidos en las encuestas realizadas en la presente investigación se pudo determinar que las plataformas que más se utilizan en la UCI para el desarrollo de aplicaciones Web son la J2EE (Java 2 Enterprise Edition) de Sun Microsystems[9] y .NET de Microsoft[10].

1.7.1. J2EE

Java 2 Enterprise Edition contiene especificaciones técnicas que describen el lenguaje, además provee las herramientas para implementar productos de software. La plataforma ofrece muy buenas perspectivas para la implementación de productos empresariales destinados a aquellos sistemas informáticos que requieran establecer su arquitectura basada en software libre.
Esta plataforma también puede ser considerada como un estándar de desarrollo de las aplicaciones Web. Como toda plataforma que se crea debe tener sus retos para la confección de software, y esta no está libre de estos objetivos destacando cuatro retos principales:
Portabilidad. Los componentes podrán ser reubicados del entorno operativo o, bien, del entorno de trabajo e incluso del sistema, sin requerir que se hagan cambios significativos que afecten al sistema.

Diversidad de ambientes. La empresa que la utilice podrá contar con una infraestructura de hardware/software que debe aprovecharse de manera conjunta en aplicaciones de uso empresarial para los desarrolladores de aplicaciones Web.

Oportunidad en su aparición. Los componentes que se desarrollan deben ser implementados y publicados para su integración a las aplicaciones, en el momento en que son requeridos.

---

[9] Sun Microsystems: Empresa líder en servidores Web y estaciones de trabajo, además es una gran productora de software

[10] Microsoft: Empresa multinacional estadounidense, creadora de sistemas operativos como MsDos, Windows así como de aplicaciones informáticas de todo tipo

Reutilización. El diseño de componentes de software, con independencia de los servicios que proveen, permite reforzar el concepto de reutilización, que no significa solamente “cortar y pegar” partes del código que se desarrolle, sino que proporciona al *integrador de aplicaciones* la facilidad de generar soluciones completas de software, aprovechando los beneficios de los componentes disponibles ya realizados.

La plataforma J2EE brinda también ventajas como dar soporte para distintos sistemas operativos, ya que es posible desarrollar arquitecturas basadas en J2EE usando cualquier sistema operativo donde se esté ejecutando una máquina virtual de Java. También se puede hablar sobre las soluciones libres pues la plataforma J2EE posibilita crear arquitecturas basadas por completo en productos de software libre.

1.7.2. .NET

Esta plataforma provee soluciones predefinidas para el desarrollo de aplicaciones administrando la ejecución de los programas escritos específicamente con la plataforma. .NET Framework se puede encontrar en Windows Server 2008, Windows Vista entre otras versiones del sistema operativo de Microsoft.

El Framework de .Net es una plataforma que necesita previa instalación para la ejecución de programas creados a través de .NET, ya que esta no viene incluida en los diferentes sistemas operativos distribuidos por Microsoft.

.NET tiene una interfaz de lenguajes comunes (CLI por sus siglas en inglés) que le permite soportar a más de veinte lenguajes existentes. .NET es un conjunto de nuevas tecnologías que podrían resumirse en las siguientes: Plataforma .NET, SDK de la plataforma .NET, Visual Studio.NET, Servicios Web y Servidores para empresas.

La plataforma .NET es una capa de software que se encuentra entre el sistema operativo y el programador. Una de las características fundamentales de esta plataforma es **la portabilidad** debido a la abstracción del programador respecto al sistema operativo. Una aplicación .NET puede ser ejecutada en cualquier sistema que disponga de una versión de la plataforma .NET framework. Esta plataforma en lugar de compilar a código máquina, utiliza un lenguaje intermedio llamado Microsoft Intermediate Language o MSIL (Lenguaje Intermedio de Microsoft), lo cual le permite realizar una ejecución de acuerdo a la plataforma en que se esté evaluando.

1.8. Lenguajes de programación

Un lenguaje es un sistema de comunicación que tiene forma, contenido y uso. La programación es, en informática, el proceso de escritura del código fuente de un software. De esta forma el programador le señala al programa informático qué tiene que hacer y cómo realizarlo. (Programacion, 2009)

Con esta noción en claro, se puede concluir que un lenguaje de programación es aquel sistema que, con una cierta estructura sintáctica y semántica, indica distintas instrucciones a un programa de computadora.

Para lograr una mejor visión, conocimiento del estado del arte y uso de los lenguajes de programación para el desarrollo de las aplicaciones Web en la UCI, se realizó una investigación en los proyectos de software que en esta se desarrollan, en la que se concluyó que en la mayoría de estos se utilizan los lenguajes de programación que a continuación se describen.

C#

C# es un lenguaje de programación que se creó tomando las mejores características de otros lenguajes ya existentes como Visual Basic, Java o C++ de manera que todas estas quedan combinadas en un solo lenguaje. Es un lenguaje orientado a objeto que soporta la declaración de tipos de datos, operaciones, colecciones, propiedades y excepciones.
Utiliza la plataforma .NET para la ejecución de aplicaciones Web o ventanas tradicionales. Contiene un kit de desarrollo denominado .NET Framework SDK, el cual brinda las herramientas necesarias para programar y compilar en C#.

PHP

PHP (por sus siglas: Hipertexto Pre-processor") es un lenguaje interpretado de alto nivel, incrustado en paginas HTML y ejecutado en el servidor. Este permite hacer cualquier cosa que se pueda hacer con un script CGI como procesar la información de los formularios, generar páginas con contenido dinámico o mandar o recibir cookies entre otras.
Una de las características más importantes de PHP es su capacidad de soporte para un gran número de bases de dato. También soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados.

Java

Java es un lenguaje de desarrollo de propósito general que permite hacer aplicaciones informáticas de todo tipo. Combina una serie de características que lo hacen único y que lo han hecho ser adoptado como herramienta básica de desarrollo por un sinfín de fabricantes de software de gran repercusión. Puede

funcionar como una aplicación independiente para el desarrollo Web o como un “applet", que es un pequeño programa hecho en Java que se puede ejecutar sobre diferentes navegadores.

El funcionamiento de java es el que hace que sea multiplataforma, ya que primeramente el compilador de java deja el programa en un código intermedio conocido como java bytecodes, los que serán interpretados por lo que se conoce como Java Virtual Machine (JVM) la maquina virtual de Java, permitiendo ejecutar el programa en cualquier plataforma.

C++

El lenguaje C++ es un lenguaje de propósito general basado en el C el cual es un lenguaje orientado básicamente a la implementación de Sistemas Operativos concretamente Unix[11], al que se han añadido nuevos tipos de datos, clases, plantillas, mecanismo de excepciones, sistema de espacios de nombres, funciones online, sobrecarga de operadores, referencias, operadores para manejo de memoria persistente, y algunas utilidades adicionales de librería en las cuales está presente la librería Estándar C la cual es un subconjunto de la librería C++.

JavaScritp

Se trata de un lenguaje de programación creado principalmente para hacer páginas Web dinámicas. Este se basa en las tecnologías del lado del cliente, ya que es el navegador el que soporta la carga de procesamiento.

Dentro de sus funcionalidades se encuentra brindar interactividad en las páginas con el cliente. Es un lenguaje con muchas posibilidades dentro de la programación Web que permite la programación de pequeños scripts, pero también de programas grandes, orientados a objetos, con funciones y estructuras de datos complejas. Además, pone a disposición del programador todos los elementos que forman la página Web, para que pueda acceder a ellos y modificarlos dinámicamente. Con JavaScritp el desarrollador es quien controla todo lo que ocurre en la página, cuando la está visualizando el cliente.

Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje script de programación del lado del cliente más utilizado.

[11] Unix: Sistema operativo portable, multitarea y multiusuario (la PC puede ser usada por varios usuarios a la vez)

XHTML

XHTML (del ingles e**X**tensible **H**yper**t**ext **M**arkup **L**anguage) o también conocido como HTML 4.01, es un leguaje para elaborar páginas Web que se ha convertido en un estándar aprobado por el Consorcio World Wide Web (W3C)[12] desde el 26 de enero del 2000. La principal razón de su uso es la creación de código limpio, separando el contenido del diseño, además dado que está basado en XML, es posible su lectura e interpretación en cualquier dispositivo móvil que soporte XML. Esto le permite a la vez, ser manejado y validado por cualquier herramienta estándar, le da una mayor facilidad de edición directa del código fuente además de que los documentos escritos conformes a XHTML pueden potencialmente presentar mejor rendimiento en las actuales herramientas Web.

1.9. CCS

CSS (por sus siglas significa Cascading Style Sheets) o lo que es lo mismo Hojas de Estilo en Cascada, es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla de acuerdo a los estilos que se definan, esta forma ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los *Estilos* definen la forma de mostrar los elementos HTML y XML. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. (W3C, 2008)

1.10. DOM

Una página web es un documento HTML que es interpretado por los navegadores en forma gráfica, pero también permiten el acceso al código para su interpretación. El Modelo de Objetos del Documento (DOM) permite ver el mismo documento de otra manera, describiendo el contenido del documento agrupando los objetos que un programa JavaScritp puede actuar sobre ellos.

El DOM abre una puerta a la estructura de una página HTML mediante el mapeo de los elementos de esta página. Cada elemento se convierte en un nodo y cada porción de texto en un nodo de texto.

[12] W3C: Consorcio internacional que establece normas para el desarrollo y uso de WWW.

1.11. Frameworks de desarrollo

Un Framework es una mini-arquitectura reutilizable que provee la estructura genérica y el comportamiento para un conjunto de abstracciones de software, unido a un contexto formado por metáforas que especifican las colaboraciones y el uso en un dominio dado.

Poseen una variedad de librerías que obligan a trabajar en un patrón de diseño determinado para crear aplicaciones, ya que por sí mismas no tienen ninguna funcionalidad. Un framework es, por tanto, el núcleo de toda aplicación Web. Un framework puede ser aplicado a un grupo de clases, lo cual lo hace compartido. Propone un modelo de colaboración al cual los desarrolladores que lo utilizan deben adaptarse, permitiendo la reutilización de diseño y código.

Spring 2.0

Es un framework de código abierto que se utiliza para el desarrollo de aplicaciones Web en la plataforma Java. Este framework ha adquirido gran popularidad en la comunidad de programadores en Java, ya que es considerado sustituto del modelo Enterprise JavaBean. Cuenta con innumerables características que a los programadores les dan grandes beneficios a la hora de utilizarlo, ejemplos de estas son las siguientes:

- ✓ POJO: (Plain Old Java Object) revalora la simplicidad de las clases Java aportando manejo de transacciones de forma no intrusiva.
- ✓ XML: Configuración basada en archivos XML.
- ✓ Seguridad:
- ✓ AOP: Brinda la posibilidad de declarar los aspectos de manera declarativa con la nueva adopción de AspectJ
- ✓ Contextos: Da la posibilidad de crear contextos que controlen la creación de los objetos a parte de los que vienen por defecto.
- ✓ Testing: Provee un paquete de prueba específico para componentes del framework e integrado con JUnit.
- ✓ Portlets MVC: incluye un nuevo parco para el desarrollo de aplicaciones con portlets
- ✓ Seguridad: Se basa en fundamentos sólidos Acegi de seguridad.

Hibernate 2.0

Como mismo existen IDEs para el desarrollo en Java, existen otras tecnologías y frameworks que manejan el acceso a bases de datos relacionales (BDR) o lo que se conoce como el manejo de la capa de acceso a datos, y proporcionan mecanismos que hacen corresponder la incompatibilidad entre el modelo de objetos y el modelo relacional. Para que se logre esta comunicación siendo lo más transparente y

eficientemente posible logrando una mayor portabilidad en su código, se implementa una capa de acceso a datos que interactúe directamente con una o más Bases de datos relacionales.

Hibernate es considerado como el producto de punta en el control de acceso a datos, el más utilizado y puesto en práctica en esta área. En la Universidad su utilización viene dado por las amplias ventajas que este trae con consigo, como su amplia documentación para los desarrolladores, prestaciones y estabilidad que le ofrece a los programadores de la persistencia de objetos en Java.

Hibernate hace uso de un conjunto de Apis de Java, las cuales incluye JDBC, la API de transacciones Java (JTA), y Java Naming and Directory Interface (JNDI). Las Apis son los principales elementos que debemos tener en cuenta en cuanto al trabajo con Hibernate para utilizarlas en la capa de persistencia de la aplicación.

Symfony 1.2

Symfony es el framework para desarrollar en PHP más utilizado en el desarrollo de aplicaciones web en la UCI, el 42% de los productos Web utilizan este framework para su desarrollo, pues múltiples son las ventajas que le brinda a los programadores, ayuda a que sean mucho más productivos, a la vez que crean código de más calidad y fácil de mantener.

Symfony es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones Web. Los archivos de configuración de este framework utilizan el formato YAML, el preferido por todos los frameworks modernos, en vez del tradicional y más aburrido formato XML. YAML (YAML Ain't Markup Language) es un lenguaje que permite describir los datos como en XML. Symfony utiliza el formato YAML como el lenguaje preferido para almacenar su configuración. No obstante, permite utilizar archivos XML o INI si es necesario, es compatible con la mayoría de gestores de bases de datos como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft.

Extjs 2.2

Extjs 2.2 es un framework que permite crea desde aplicaciones sencillas hasta las más complejas, utilizando componentes predefinidos así como un manejador de layouts similar al que provee Java Swing. Brinda una serie de funcionalidades como son: (Castillo, et al., 2008)

- ✓ Componentes UI de alto performance y personalizables.
- ✓ Modelo de componentes extensibles.
- ✓ Un Api fácil de usar
- ✓ Licencias Open Source y comerciales.

Además si se utiliza un motor de render como Extjs nos permite también:

- ✓ **Balance entre Cliente-Servidor:** la carga del servidor se distribuye permitiéndole a esta manejar más clientes al estar menos cargado.
- ✓ **Comunicación Asincrónica:** el motor render puede comunicarse con el servidor sin tener que estar sujeto a una acción del usuario.
- ✓ **Eficiencia de la red:** el tráfico de red puede disminuir al permitir que la aplicación elija que información desea trasmitir.

1.12. Tecnologías utilizadas para el desarrollo de la herramienta

1.12.1. Entorno de desarrollo Integrado (IDE)

NetBeans IDE 6.8

Es un ambiente libre de desarrollo, integrado con código abierto para el desarrollo de software. Ofrece todas las herramientas necesarias para crear escritorios profesionales, Enterprise, Web y aplicaciones móviles con el lenguaje Java, JavaFX, C / C + + y lenguajes dinámicos como PHP, Java Script, Groovy y Ruby. Es fácil su instalación y uso directamente desde la caja y se ejecuta en Windows, Linux, Mac OS X y Solaris[13] (TM).

Es el primer IDE que ofrece soporte completo para Java (TM) Plataforma Enterprise Edition 6 (Java EE 6) y Sun GlassFish (TM) Enterprise Server v3, así como otras funcionalidades innovadoras dentro de las cuales se pueden citar:

- ✓ Soporte ampliado para PHP: Expande el soporte de los lenguajes dinámicos con apoyo para PHP 5.3.
- ✓ Mejora de C / C + +: Perfila y sintoniza aplicaciones C / C + + con el nuevo indicador Microstate Accounting, supervisor de uso I/O.
- ✓ JavaFX ™: Código de finalización mejorado, sugerencias y navegación para JavaFX en el editor NetBeans.

[13] Solaris: Sistema operativo de tipo Unix utilizado principalmente para servidores y estaciones de trabajo.

1.12.2. Herramienta Case

Visual Paradigm

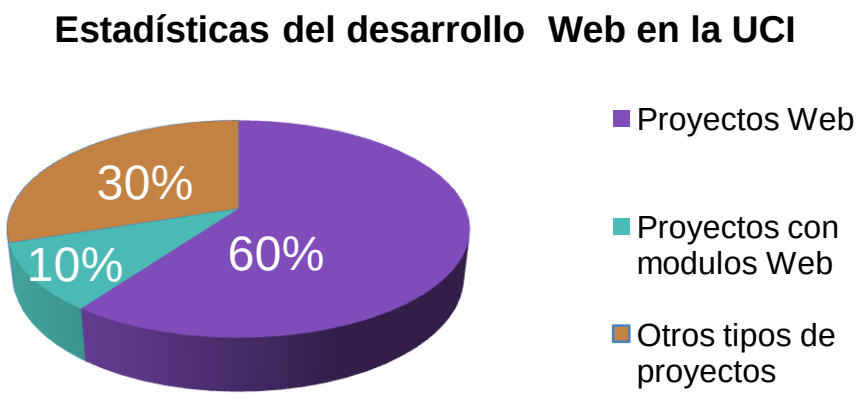
Para el diseño y modelado de la aplicación se seleccionó Visual Paradigm, pues es una herramienta CASE muy completa y fácil de usar, con soporte multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Además se utiliza UML como lenguaje de modelado para el diseño de la herramienta que se desarrollará

Es una herramienta profesional que tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML, al instante, creando de manera simple toda la documentación. (Maylen Bon Pérez, 2009)

1.13. Conclusiones parciales

Con la realización de este capítulo se logró determinar, a través una pesquisa realizada en los proyectos de la UCI, cuales son los principales lenguajes de programación, entornos de desarrollo integrado, tecnologías o frameworks que más se utilizan en la misma para el desarrollo de aplicaciones Web, se analizaron los principales conceptos de Calidad, Portabilidad, además de que se mencionaron otros aspectos que influyen en la calidad de las aplicaciones Web.

Esta investigación arrojó como resultados que más del 60% de los productos que se están desarrollando actualmente en la UCI son aplicaciones Web o poseen módulos Web. Los lenguajes de programación más utilizados son Java y PHP 5.0. Estos proyectos se encuentran trabajando mayormente sobre Windows XP o Ubuntu 9.10. También se determinó que los frameworks que se utilizan para el desarrollo son principalmente Symfony 1.2, Spring 2.0, Extjs 2.2 y Hibernate 2.0. (Ver anexo Nro. 2)



2.1. Introducción

Las funciones son el soporte principal de cualquier lenguaje de programación que se defina para el desarrollo de sistemas informáticos. Pero no solamente son las funciones los que influyen en la realización de una aplicación sino también las librerías utilizadas para su desarrollo y la arquitectura que se utilice. Los desarrolladores de aplicaciones Web para que sus productos tengan un nivel alto de portabilidad, requieren más que simplemente una implementación o un patrón de diseño; requieren el conocimiento de componentes de software que se usen para su desarrollo.

En este capítulo se exponen los principales problemas que afectan la portabilidad, presentes en las tecnologías utilizadas en los proyectos Web de la Universidad. Además, basándonos en todas las problemáticas encontradas, se definen métricas para evaluar la portabilidad del código fuente de las aplicaciones Web.

2.2. Análisis de Funciones

2.2.1. Java

Cuando se refiere a una función en Java es necesario definir que es un módulo de un programa que se encuentra separado del cuerpo principal, definido para realizar una tarea específica y que con su llamada devuelva un valor al programa u otra función o procedimiento que la invoque.

Más allá de la portabilidad básica por ser independiente de la arquitectura, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en diversos entornos. Esto permite que java contenga un alto por ciento de portabilidad de sus aplicaciones.

Java utiliza Java Runtime Environment (JRE) y Java Develoment Kit (JDK) o entorno de desarrollo de Java, proporcionándole las librerías a utilizar en el desarrollo de las aplicaciones y es el entorno de ejecución necesario para interpretar y ejecutar archivos .class. Tiene incluido también la importante máquina virtual de Java y otros componentes necesarios para poder ejecutar applets y aplicaciones en el lenguaje Java.

El uso de un JDK incluye el uso también de un JRE en una versión estándar con el JDK, así como compiladores y depuradores que son necesarios para la creación de applets y aplicaciones. Esto origina uno de los problemas en la portabilidad de aplicaciones desarrolladas en Java, pues con un JRE en una

versión moderna se pueden ejecutar programas compilados con un JDK de versiones más antiguas, pero no sucede lo mismo al revés. Un JRE en una versión antigua no sabrá ejecutar programas compilados con un JDK en una versión más avanzada.

La Plataforma Java está pensada para ser independiente del sistema operativo, por lo que las aplicaciones no deben tener vínculos con funciones dependientes de cada sistema operativo sino se estaría teniendo dependencias de las Apis nativas del mismo y podría impedir la portabilidad. Lo que hace la Plataforma Java, es brindar un conjunto de librerías estándar, que contiene mucha de las funciones reutilizables disponibles en los sistemas operativos actuales.

Al igual que otras librerías, las de Java ofrecen al programador un conjunto de funciones para realizar tareas comunes como manejar listas de elementos u operar de forma sofisticada sobre cadenas de caracteres. Además proporcionan una interfaz abstracta para tareas que son altamente dependientes del hardware de la plataforma destino y de su sistema operativo.

Finalmente, no todas las plataformas soportan todas las funciones que una aplicación Java espera. En estos casos, las librerías bien pueden emular esas funciones usando lo que esté disponible, o bien ofrecer un mecanismo para comprobar si una funcionalidad concreta está presente.

Se pudo comprobar que la plataforma Java no presenta graves problemas de portabilidad, solo algunos elementos a tener en cuenta que pueden llegar a provocar incompatibilidades en una aplicación desarrolla en este ambiente, los cuales serán utilizados para definir métricas en siguientes epígrafes para evaluar la portabilidad de aplicaciones desarrolladas en este entorno.

2.2.2. PHP

Todas las versiones de PHP que han evolucionado desde sus inicios han contenido en cada una de ellas mejoras en sus funciones para un mejor uso de las mismas por los desarrolladores, la nueva evolución de las funciones trae consigo que cambien las estructuras de las mismas y las formas de utilización, por lo que ya se entraría en un problema de portabilidad de las aplicaciones desarrolladas en una versión cuando se quiere migrar a otra versión nueva.

Con el surgimiento de las nuevas versiones y el constante soporte que se les proporciona algunas van quedando obsoletas como las anteriores a PHP 5.0 a las cuales no se les sigue brindando mantenimiento ejemplo de ellas es PHP 4 a la cual se le dejo de dar soporte tres años después de lanzada la versión PHP 5.0. El 13 de julio del 2004 fue lanzada la versión PHP 5.0, utilizando Zend Engine 2.0 que cumplía para aquel entonces una expectativa enorme por las nuevas posibilidades que brindaba comparado con

PHP 4.0. La versión más reciente de PHP es la 5.3.1 lanzada el 19 de noviembre de 2009, que incluye todas las ventajas que provee el nuevo Zend Engine 2 como: (HTML_Help, 2005)

- ✓ Mejoras en el soporte para la POO (Programación Orientada a Objetos), que en versiones anteriores era extremadamente rudimentario, con la utilización de PHP Data Objects.
- ✓ Mejoras de rendimiento.
- ✓ Mejor soporte para MySQL con extensión completamente reescrita.
- ✓ Mejor soporte a XML (XPath, DOM, etc.).
- ✓ Soporte nativo para SQLite[14].
- ✓ Soporte integrado para SOAP.
- ✓ Iteradores de datos.
- ✓ Manejo de excepciones.
- ✓ Mejoras con la implementación con Oracle.

Está definido que en noviembre del 2010 salga la rama número 6 de PHP, o como es más conocido por los programadores, PHP 6.0. Con la creación de una nueva versión serán numerosos los cambios que se piensan realizar para obtener mejorías en el lenguaje, pero no se tendrán en cuenta para la investigación ya que aun no se tiene un conocimiento concreto de cuales serán estos cambios en la nueva versión.

Funcionalidades que afectan la portabilidad en PHP 5.0

mysqli_bind_param: Alias de mysqli_stmt_bind_parames obsoleta y se eliminará en la versión PHP 5.3.0

mysqli_enable_reads_from_master: Esta función ha quedado obsoleta y eliminada desde PHP 5.3.0.

mysqli_disable_rpl_parse: Se utiliza para deshabilitar RPL, esta función ha quedado obsoleta y eliminada desde PHP 5.3.0.

mysqli_execute: Proveniente de mysqli_stmt_execute. Esta función ha quedado obsoleta y eliminada desde PHP 5.3.0.

mysqli_fetch: Esta función ha quedado obsoleta y eliminada desde PHP 5.3.0.

mysqli_get_metadata: Esta función ha quedado obsoleta y eliminada desde PHP 5.3.0.

[14] SQLite: Sistema de gestión de bases de datos relacional

mysqli_master_query: Esta función ha quedado obsoleta y eliminada desde PHP 5.3.0.

mysqli_param_count: Es obsoleta desde PHP 5.0 y se eliminará en PHP 5.3.0

mysqli_rpl_parse_enabled: Esta función ha quedado obsoleta y eliminada desde PHP 5.3.0

mysqli_rpl_parse_enabled: Examina RPL. Esta función ha quedado obsoleta y eliminada desde PHP 5.3.0.

mysqli_send_long_data: Alias de mysqli_stmt_send_long_data, obsoleta y se eliminará 5.3.0

mysqli_rpl_query_type: Es obsoleta desde PHP 5.3 y se eliminará en PHP 5.3.0

mysqli_send_query: Es obsoleta desde PHP 5.3 y se eliminará en PHP 5.3.0

mysqli_slave_query: Es obsoleta desde PHP 5.3 y se eliminará en PHP 5.3.0

define_syslog_variables: Está definida en PHP 5.0 y se elimina en PHP 5.3.0.

array_reduce: Cambiado array_reduce () para permitir mixed $ inicial. Está definida en PHP 5.0 y se cambiará en PHP 5.3.0.

php_check_syntax: Nunca funcionó correctamente. Está definida en PHP 5.0 y se cambiará en PHP 5.1.0.

Descripción de directivas php.ini

zend.ze1_compatibility_mode: Está definida en PHP 5.0 y se elimina en PHP 5.3.0.

zend_extension_ *: Está definida en PHP 5.0 y se elimina en PHP 5.3.0.

Type Operators

El operador instanceof fue introducido en PHP 5. Reemplazando a **is_a ().** A partir de PHP 5.3.0, is_a () será obsoleta.

Experimental RPL

Se ha eliminado el experimental RPL (maestro / esclavo) de las funciones de mysqli. Está definida en PHP 5.0 y se elimina en PHP 5.3.0.

Shebang sapi CGI

Eliminado de la verificación en línea Shebang sapi CGI (controlada por el escáner). Está definida en PHP 5.0 y se elimina en PHP 5.3.0.

SHELL32.DLL.

Removida la dependencia de SHELL32.DLL. Está definida en PHP 5.0 y se cambiará en PHP 5.2.1.

Stat: Es reimplementado utilizando GetFileAttributesEx (). La nueva implementación es más rápida que la aplicación en MS VC CRT, pero no es compatible con Windows 95. Está definida en PHP 5.0 y se cambiará en PHP 5.2.1.

Búsqueda de php.ini

Eliminado directorio de trabajo actual de la ruta de búsqueda de php.ini para CLI y re-añadido para otras Apis. Está definida en PHP 5.0 y se cambiará en PHP 5.2.1.

CURLOPT_FOLLOWLOCATION

Deshabilitado CURLOPT_FOLLOWLOCATION en concurrencia cuando open_basedir o safe_mode están habilitados. Está definida en PHP 5.0 y se cambiará en PHP 5.1.5.

Garbage manager (Administrador de Basura)

Garbage Manager: Eliminado Garbage Manager en Zend Engine que resulta en la liberación de datos agresivos.

Se logró determinar que en PHP 5.0 existes un gran número de problemáticas que en versiones continuas a esta son eliminadas, originando de esta manera numerosos problemas de portabilidad en aplicaciones desarrolladas en este lenguaje. Todos estos problemas serán utilizados para definir las métricas que evaluaran la portabilidad del código fuente de aplicaciones desarrolladas en este ambiente.

2.2.3. JavaScript

Las funciones son una estrategia correcta que se usan para el desarrollo de JavaScritp. Cuando se vuelve a escribir todo desde cero se pierde tiempo y recursos, la solución a este problema es solo una de las muchas ventajas que brindan las funciones en JavaScritp. El uso de un conjunto de bibliotecas que contienen funciones conocidas y aceptadas guían a minimizar los errores en la programación, lo que trae consigo un desarrollo más agilizado de las aplicaciones que se creen, y con mayor calidad.

La infinidad de ventajas que brinda este lenguaje para el desarrollo de aplicaciones Web hace que se convierta en el lenguaje de preferencia para los programadores hoy en día. Pero como todo ambiente de desarrollo a medida que pasa el tiempo surgen nuevas técnicas de programación las cuales van sustituyendo a aquellas que por un motivo u otro quedarán obsoletas. JavaScript no por ser un lenguaje de preferencia para los programadores deja de presentar algunos cambios a medida que van avanzando las versiones de dicho lenguaje. A continuación se muestran una serie de funciones u operadores que han quedado en desuso y que no se utilizarán en nuevas versiones provocando errores de portabilidad.

El uso de algunas funciones de JavaScritp para el desarrollo Web en un futuro puede causar problemas de portabilidad.

Funciones Obsoletas para JavaScript

- ✓ **getYear:** En Internet Explorer funciona correctamente lo que no es el caso en Netscape[15] y derivados, donde mostrará la diferencia de años entre el año actual y 1900.La solución a este problema viene dada por el uso de getFullYear la cual sustituye a getYear propiciando compatibilidad entre los navegadores.
- ✓ **Substr:** La cual retorna un substring derivada del string original. Esta función ha quedado obsoleta desde la versión 1.2 de JavaScript.
- ✓ **Scape:** Es una función global que codifica una cadena mediante la sustitución de caracteres con secuencias de escape. Esta ha quedado obsoleta en la versión 3 de JavaScript.
- ✓ **Unescape:** Es una función global que descodifica una cadena que se ha codificado con la función scape, ha quedado obsoleta en la versión 3.0 de JavaScript.
- ✓ **SetYear:** Devuelve la fecha ajustada en milisegundos con el argumento de CCAA. Esta función ha quedado obsoleta desde la versión 3.0 de JavaScript.
- ✓ **toGMTString:** Devuelve la fecha ajustada en milisegundos con el argumento de CCAA. Esta ha quedado obsoleta desde su versión 3.0 de JavaScript.
- ✓ **arguments [ ]:** Representa una matriz de los argumentos que se pasan a la función. Esta ha quedado obsoleta desde su versión 3.0 de JavaScript.
- ✓ **Celleris:** Es una propiedad que hace referencia que invoca al objeto. Esta ha quedado obsoleta desde su versión 3.0 de JavaScript.

[15] Netscape: Programa o browser (navegador) que permite navegar por las páginas Web.

En este lenguaje se detectaron una serie de elementos que han sido sustituidas por funcionalidades más eficientes en el uso de este ambiente de desarrollo pero que aún no han sido eliminadas por completo del lenguaje, lo cual no trae consigo problemas de portabilidad por el momento ya que todavía siguen existiendo, por lo cual no serán utilizadas para definir métricas.

2.3. Análisis de los frameworks

2.3.1. Framework Spring 2.0

La utilización de este framework se convierte cada día en una alternativa mucho más avanzada para el desarrollo web. Pero, los cambios que se le realizan a las nuevas versiones, traen consigo que muchas de las funcionalidades, métodos, clases, interfaces, etc. queden obsoletas o sean renovadas por otras más sofisticadas, donde el uso de estas puede provocar problemas de incompatibilidad en un futuro cuando se desee hacerle cambios a la aplicación que lo utilice. A continuación se muestran los elementos que pueden provocar dificultades en la portabilidad y en la compatibilidad de una aplicación en la cual se utiliza Spring Framework. Para esto el presente análisis se basa en la información que se brinda en el portal Web de la comunidad de Spring: (Community, 2007)

- **Interfaces Obsoletas**

- ✓ **Interface XmlBeanDefinitionParser:** Obsoleta a partir de Spring 2.0, sustituida por BeanDefinitionDocumentReader en la versión 2.5.
- ✓ **Interface ListenerSessionManager:** Obsoleta a partir de Spring 2.5, sustituida por DefaultMessageListenerContainer y JmsMessageEndpointManager para ser removida en Spring 3.0.
- ✓ **Interface PagedListSourceProvider:** Obsoleta a partir de Spring 2.5 y se retirará en Spring 3.0.
- ✓ **Interface ResponseTimeMonitor:** Obsoleta a partir de Spring 2.5 y se retirará en Spring 3.0.
- ✓ **Interface ServerSessionFactory:** Obsoleta a partir de Spring 2.5, sustituida por DefaultMessageListenerContainer y JmsMessageEndpointManager para ser removido en Spring 3.0.
- ✓ **Interface ThrowawayController:** Obsoleta a partir de Spring 2.5, sustituida por controladores basados en la anotación para ser removido en Spring 3.0.

- **Clases Obsoletas**

- ✓ **Class AdvisorChainFactoryUtils:** Obsoleta a partir de Spring 2.0.3, sustituida por DefaultAdvisorChainFactory. Esta se eliminará en Spring 2.1.

- ✓ **Class DynamicMethodMatcherPointcutAdvisor:** Obsoleta desde Spring 2.0, sustituida por DefaultPointcutAdvisor con un tiempo de ejecución DynamicMethodMatcherPointcut
- ✓ **Class HashMapCachingAdvisorChainFactory:** Obsoleta a partir de Spring 2.0.3, sustituida por AdvisedSupport incorporado en la memoria caché. Esta clase será eliminado en Spring 2.1.
- ✓ **Class RequestUtils:** Obsoleta a partir de Spring 2.0, se usara en su lugar ServletRequestUtils.
- ✓ **Class AbstractPathMapHandlerMapping**: Obsoleta a partir de Spring 2.5 sustituida por la anotación de mapas basados en la solicitud y será removida en Spring 3.0.
- ✓ **Class AbstractPoolingServerSessionFactory:** Obsoleta a partir de Spring 2.5, sustituida por DefaultMessageListenerContainer y JmsMessageEndpointManager y será removida en Spring 3.0.
- ✓ **Class LoaderAnalyzerInterceptor:** Obsoleta a partir de Spring 2.5 y se retirará en Spring 3.0.
- ✓ **Class LoaderUtils:** Obsoleta a partir de Spring 2.5 y se retirará en Spring 3.0.
- ✓ **Class CommonsPathMapHandlerMapping:** Obsoleta a partir de Spring 2.5 sustituida por la anotación de mapas basados en la solicitud y será removida en Spring 3.0.
- ✓ **Class CommonsPoolServerSessionFactory:** Obsoleta a partir de Spring 2.5, sustituida por DefaultMessageListenerContainer y JmsMessageEndpointManager y será removida en Spring 3.0.
- ✓ **Class PathMap:** Obsoleta a partir de Spring 2.5, sustituida por la anotación de mapas basados en la solicitud. Será removida en Spring 3.0.
- ✓ **Class PerformanceMonitorListener:** Obsoleta a partir de Spring 2.5, sustituida por ApplicationListener en su lugar. Se retirará en Spring 3.0.
- ✓ **Class ReflectiveVisitorHelper:** Obsoleta a partir de Spring 2.5, se retirará en Spring 3.0.
- ✓ **Class RefreshablePagedListHolder:** Obsoleta a partir de Spring 2.5, se retirará en Spring 3.0.
- ✓ **Class ResponseTimeMonitorImpl:** Obsoleta a partir de Spring 2.5, se retirará en Spring 3.0.
- ✓ **Class ServerSessionMessageListenerContainer:** Obsoleta a partir de Spring 2.5, sustituida por DefaultMessageListenerContainer y JmsMessageEndpointManager y será removida en Spring 3.0.
- ✓ **Class ServerSessionMessageListenerContainer102:** Obsoleta a partir de Spring 2.5, sustituida por DefaultMessageListenerContainer y JmsMessageEndpointManager y será removida en Spring 3.0.
- ✓ **Class SimpleServerSessionFactory:** Obsoleta a partir de Spring 2.5, sustituida por DefaultMessageListenerContainer y JmsMessageEndpointManager y será removida en Spring 3.0.

- ✓ **Class ThrowawayControllerHandlerAdapter:** Obsoleta a partir de Spring 2.5, sustituida por la anotación de mapas basados en la anotación. y será removida en Spring 3.0.

❖ Métodos Obsoletos

- **En la clase MockHttpServletRequest**
  - ✓ **addRole:** Obsoleto y sustituido por addUserRole
- **En la clase TimerFactoryBean**
  - ✓ **CreateTimer:** Obsoleto a partir de Spring 2.0.1.
- **En la clase NamespaceHandlerSupport**
  - ✓ **findDecoratorForNode:** Obsoleto a partir de Spring 2.0.2.
  - ✓ **findParserForElement:** Obsoleto a partir de Spring 2.0.2.
- **En la clase AbstractAutowireCapableBeanFactory**
  - ✓ **findMatchingBeans:** Obsoleto a partir de Spring 2.0.1, sustituido por findAutowireCandidates.
- **En la clase AbstractXsltView**
  - ✓ **getParameters:** Obsoleto a partir de Spring 2.0.4, sustituido por la variante getParameters (HttpServletRequest). Se le agrega un nuevo parámetro al método y este retornara un mapa de los parámetros de transformación que se aplicara a la hoja de estilos
- **En la clase AopProxyUtils**
  - ✓ **getTargetClass:** Obsoleto a partir de Spring 2.0.3, sustituido por AopUtils.getTargetClass
- **En la clase MultiActionControlle**
  - ✓ **initBinder:** Obsoleto a partir de Spring 2.0, sustituido por initBinder (HttpServletRequest, ServletRequestDataBinder), se sustituye el parámetro servletrequest por un httpservletrequest
- **En la clase JaxRpcPortClientInterceptor**

  ✓ **performJaxRpcCall**(MethodInvocation invocation): Obsoleto a partir de Spring 2.0.3, sustituido por la variante performJaxRpcCall(MethodInvocation invocation, Service service) con un argumento explícito de servicios
- **En la Interfaz ConfigurableBeanFactory**
  - ✓ **registerCustomEditor:** Obsoleto a partir de Spring 2.0.7, sustituido por addPropertyEditorRegistrar ()
- **En la clase CustomEditorConfigurer**
  - ✓ **setCustomEditors:** Obsoleto a partir de Spring 2.0.7, sustituido por setPropertyEditorRegistrars ()
- **En la clase GenericFilterBean**
  - ✓ **setFilterConfig:** Obsoleto a partir de Spring 2.0, se retira en Spring 2.1.

- **En la clase XmlBeanDefinitionReader**
  - ✓ **setParserClass:** Obsoleto a partir de Spring 2.0, sustituida por "documentReaderClass"
- **En la clase FreeMarkerConfigurationFactory**
  - ✓ **setTemplateLoaders:** Obsoleto a partir de Spring 2.0.1, sustituida por las propiedades "preTemplateLoaders" y "postTemplateLoaders"
- **En las clases XmlBeanDefinitionReader**
  - ✓ **setValidating:** Obsoleto a partir de la Spring 2.0: sustituida por "validationMode"
- **En la Clase BaseCommandController**
  - ✓ **suppressValidation**(HttpServletRequest request): Obsoleto a partir de Spring 2.0.4, sustituida por la variante suppressValidation(HttpServletRequest request, Object command).
- **En la clase ModelMap**
  - ✓ **addObject**: Obsoleto a partir de Spring 2.5, sustituida por addAttribute().
  - ✓ **addAllObjects:** Obsoleto a partir de Spring 2.5, sustituida por addAllAttributes().
- **En la clase BeanDefinitionBuilder**
  - ✓ **setSource:** Obsoleto desde Spring 2.5.
  - ✓ **setFactoryBean:** Obsoleto desde Spring 2.5.
  - ✓ **addConstructorArg:** Obsoleto desde Spring 2.5, sustituida por java.lang.ObjectaddConstructorArgValue ()
- **En la clase CollectionFactory**
  - ✓ **createIdentityMapIfPossible:** Obsoleto a partir de la Spring 2.5, para el uso de JDK 1.4 o superior
- **En la clase RmiClientInterceptorUtils**
  - ✓ **doInvoke**: Obsoleto a partir de Spring 2.5, sustituido por invokeRemoteMethod().
- **En la clase HibernateTemplate**
  - ✓ **Execute:** Obsoleto a partir de Spring 2.5, sustituido por executeWithNativeSession().
- **En la clase AopNamespaceUtils**
  - ✓ **registerAutoProxyCreatorIfNecessary(PraserContext parserContext, Object source):** Obsoleto desde Spring 2.5, sustituido por registerAutoProxyCreatorIfNecessary(PraserContext parserContext, Element sourceElement) y registerAutoProxyCreatorIfNecessary (BeanDefinitionRegistry registry, Object source) de la clase AopConfigUtils.
  - ✓ **forceAutoProxyCreatorToUseClassProxying:** Obsoleto desde Spring 2.5, sustituido por forceAutoProxyCreatorToUseClassProxying() de la clase AopConfigUtils
- **En la clase AbstractBeanDefinition**

- ✓ **overrideFrom:** Obsoleto desde Spring 2.5.
- ✓ **setFactoryBean:** Obsoleto desde Spring 2.5.

▪ **En la Interfaz BeanWrapper**

- ✓ **setWrappedInstance*:*** Obsoleto a partir de Spring 2.5, a favor de recrear un objetivo BeanWrapper.

❖ **Constructores Obsoletos**

▪ **En la clase ClassBeanDefinitionStoreException**

- ✓ **BeanDefinitionStoreException:** Obsoleto a partir de Spring 2.0, sustituido por la variante de constructor con un argumento de descripción de los recursos.

▪ **En la clase CannotGetJdbcConnectionException**

- ✓ **CannotGetJdbcConnectionException:** Obsoleto desde Spring 2.5.

▪ **En la clase DriverManagerDataSource**

- ✓ **DriverManagerDataSource:** Obsoleto desde Spring 2.5, sustituido por SimpleDriverDataSource.

▪ **En la clase RootBeanDefinition**

- ✓ **RootBeanDefinition:** Obsoleto desde Spring 2.5, sustituido por AbstractBeanDefinition.setScope()

▪ **En la clase SingleConnectionDataSource**

- ✓ **SingleConnectionDataSource:** Obsoleto desde Spring 2.5.

En este framework se detectaron varios problemas que pueden afectar la portabilidad de una aplicación, posee una serie de interfaces, clases y métodos que han sido eliminadas del lenguaje ya que las funcionalidades que estos cumplían, ahora son ejecutadas por otros elementos más eficientes y avanzados. Más adelante se definirán métricas con todas estas problemáticas para que sean utilizadas en la evaluación de aplicaciones que usen este framework.

2.3.2. Hibernate 2.0

Hibernate es un framework donde el código que se crea en la capa de acceso a datos es portable en todo sentido, pues la idea principal con que se creó fue con poder utilizar el mismo código en varias plataformas, incluso es utilizado en C# y corre perfectamente en la plataforma .Net para la cual su equivalente es denominado NHibernate, por lo que no presenta problemas que afecten la portabilidad. Se puede resumir que es portable en su totalidad, el objetivo de su análisis en este trabajo es debido a que

un 35% de los proyectos que desarrollan en la universidad que son aplicaciones Web utilizan este framework para su desarrollo en la capa de acceso a datos.

2.3.3. Symfony 1.2

En las aplicaciones realizadas con Symfony, el acceso y la modificación de los datos almacenados en la Base de Datos (BD) se realizan mediante objetos a través del PDO[16]; de esta forma nunca se accede de forma explícita a la base de datos. Sin embargo el desarrollo de este framework y el avance de sus versiones hacen que en su versión Symfony 1.2 con la utilización del ORM (por sus siglas en ingres Object-Relational mapping) se detecten una serie de plugins, funciones, clases, etc. que han quedado obsoletos y han sido eliminadas en versiones superiores a la 1.2, lo cual podría afectar la portabilidad de una aplicación Web desarrollada en este ambiente. A continuación se mencionan las problemáticas encontradas en este framework. (Javier, 2010)

❖ **Plugins que pueden afectar la portabilidad**

✓ **sfCompat10Plugin:** Obsoleto desde Symfony 1.3 y eliminado en la versión 1.4. Esto implica la eliminación de todos aquellos elementos que dependen de este plugin.

❖ **Funciones que pueden afectar la portabilidad**

- **En la clase sfToolkit**
  - ✓ **getTmpDir():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4. Este método es reemplazado por sys_get_temp_dir().
  - ✓ **removeArrayValueForPath():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4.
  - ✓ **hasArrayValueForPath():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4.
  - ✓ **getArrayValueForPathByRef():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4.
- **En la clase sfValidatorBase**
  - ✓ **setInvalidMessage():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4. Sustituida por la función setDefaultMessage
  - ✓ **setRequiredMessage():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4. Sustituida por la función setDefaultMessage.
- **En la clase sfTesterResponse**

[16] PDO: Es una extinción que provee una capa de abstracción de acceso a datos para PHP 5

- ✓ **Contains():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4. Reemplazada por la función matches() la cual es mucho más potente.

- **En la clase sfTestFunctionalBase:**
  - ✓ Las funciones **isRedirected(), isStatusCode(), responseContains(), isRequestParameter(), isResponseHeader(), isUserCulture(), isRequestFormat() y checkResponseElement()**: Fueron declaradas obsoletas desde Symfony 1.2 y se han eliminado y sustituido por las nuevas clases de tipo *tester.*
- **En la clase sfTestFunctional:**
  - ✓ Las funciones **isCached()** e **isUriCached():** Fueron declaradas obsoletas desde Symfony 1.2 y se han eliminado y sustituido por las nuevas clases de tipo *tester*.
- **En la clase sfFilesystem:**
  - ✓ **sh():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4. Reemplazada por execute().
- **En la clase sfComponent**
  - ✓ **debugMessage():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4. Sustituida por el *helper* log_message().
- **En la clase sfApplicationConfiguration:**
  - ✓ **loadPluginConfig():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4. Sustituida por la función initializePlugins().
- **En la clase sfWebRequest:**
  - ✓ **getMethodName():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4. Reemplazada por getMethod().
- **En la clase sfDomCssSelector:**
  - ✓ **getElements():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4. Sustituida por matchAll()
- **En la clase sfVarLogger:**
  - ✓ **getXDebugStack():** Obsoleta desde Symfony 1.3 y eliminada en la versión 1.4. Sustituida por getDebugBacktrace().
- **En la clase sfApplicationConfiguration:**
  - ✓ **heckSymfonyVersion():** Obsoleta y eliminada en Symfony 1.3.

❖ Clases que pueden afectar la portabilidad

- ✓ **Clase sfDoctrineLogger:** Declarada obsoleta en Symfony 1.3 y eliminada en Symfony 1.4. Sustituida por la clase sfDoctrineConnectionProfiler.

- ✓ Las clases **sfCrudGenerator, sfAdminGenerator y sfDoctrineUniqueValidator**: Fueron eliminadas en Symfony 1.4 ya que dependía del plugin **sfCompat10Plugin:**
- ✓ **Clase sfLoader:** Declarada obsoleta en Symfony 1.3 y eliminada en Symfony 1.4. Esta clase fue eliminada del framework debido a que todas sus funciones fueron eliminadas también.
- ✓ Las clases **sfDoctrineDataRetriever:** Declaradas obsoletas en Symfony 1.3 y eliminadas en Symfony 1.4. Estas clases sólo las utiliza ObjectHelper, que ha sido declarado como obsoleto
- ✓ Las clases **sfWidgetFormChoiceMany, sfWidgetFormDoctrineChoiceMany, sfValidatorChoiceMany:** Declaradas obsoletas en Symfony 1.3 y eliminadas en Symfony 1.4. Sustituidas por las clases con el mismo nombre pero sin el sufijo Many y estableciendo la opción múltiple a true

Se determinaron un gran número de problemas en Symfony que provocan problemas de portabilidad, ya que existen varios elementos que son eliminados del lenguaje debido a que fueron sustituidos por otros más avanzados. Estos problemas serán tratados para definir métricas para la evaluación de aplicaciones que utilizan este framework

3.3.4. Extjs 2.2

Extjs es un framework confeccionado con la idea de que todo su trabajo sea portable, no consta con ningún método o consultas a Apis del Sistema Operativo que permitan en algún momento afectar la portabilidad, por lo que no se detectaron problemas de portabilidad en el mismo, llegando a la conclusión de que es un framework que cualquier aplicación que lo utilice no tendrá problemas de portabilidad con el mismo.

3.4. Comportamiento del DOM en IE 6 y Firefox 3.0

Comúnmente el uso del DOM es casi generalizado por todos los programadores, pero existen problemas importantes con la utilización de este, estos problema consisten en que los navegadores interpretan de manera distinta el mismo código. Eso ocurre frecuentemente con los lenguajes interpretados en el lado del cliente, como HTML, CSS y JavaScript.

Muchos son las opciones que se están creando para dar solución a estos problemas de compatibilidad entre los navegadores a la hora de ejecutar algún elemento de la página. Para estos problemas existen una serie de técnicas que permiten detectar el navegador y según sea este, ejecutar unas u otras sentencias.

El Document Object Model (DOM) todavía no es totalmente compatible con todos los navegadores o al menos con los más utilizados como es el caso de Firefox 3.0 o una versión superior a este e internet Explorer 6. El DOM en Firefox no se comporta igual que en Internet Explorer, se han hecho algunos arreglos, lo que ofrece unas series de mejoras sobre el Modelo de Objetos del Documento (DOM), especialmente en lo que se refiere a la implementación de extensiones del DOM añadidas por otros navegadores.

Para una mejor compatibilidad se implementan en Firefox las extensiones ClientTop y CientLeft que tiene Internet Explorer, se está perfeccionando la propiedad Window.fullScreen la cual ahora es siempre exacta, sin importar donde sea leída, incluso en el contenido, antes devolvía false erróneamente.
A continuación se describen algunas de las implementaciones hechas para Firefox:

- ✓ Implementan las extensiones de DOM getClientRects y getBoundingClientRects.
- ✓ Se implementa la extensión del DOM: elementFromPoint de Internet Explorer.
- ✓ Ahora se implementan las extensiones del DOM: oncut, oncopy, onpaste, onbeforecut, onbeforecopy, onbeforepaste de Internet Explorer.
- ✓ Se han añadido capturadores para Node.nodePrincipal, Node.baseURIObject, y document.documentURIObject sólo para código con privilegios.

Firefox no soporta el acceso a elementos a través de document.elementName, algo que Internet Explorer permite (llamado también *global namespacepolluting*). Firefox no soporta el método document.all de Internet Explorer, el cual se puede usar para obtener una lista de todos los elementos del documento con un nombre determinado, como todos los elementos . (Doron Rosenberg, IBM Corporation, 2009)

Para generar y manejar contenido, Firefox no soporta el método outerHTML que añade marcado alrededor de un elemento y no tiene un equivalente estándar. Tampoco soporta innerText el cual establece el valor alfabético de su nodo. (Doron Rosenberg, IBM Corporation, 2009)

Una diferencia entre estos dos navegadores es que Firefox crea fragmentos de documento a través de document.createDocumentFragment(), el cual devuelve un fragmento del documento vacío mientras que la implementación de Internet Explorer para los fragmentos de documentos, no es compatible con el estándar del W3C y simplemente devuelve un documento normal.

Internet Explorer tiene varios métodos de manejo de contenidos que no son estándares y no son soportados en Firefox, incluyendo obtener valores, insertar texto e insertar elementos adyacentes a un nodo tales como getAdjacentElement e insertAdjacentHTML.

Firefox posee un fuerte soporte para XML y todas sus tecnologías relacionadas, tales como XSLT y servicios web. Además soporta algunas extensiones no estándar de Internet Explorer, tales como XMLHttpRequest. (Doron Rosenberg, IBM Corporation, 2009)

Amplio es el campo de las diferencias entre estos navegadores y en su comportamiento en el uso del DOM pero se ha definido las principales diferencias entre estos dos gigantes del mundo de los navegadores las cuales pueden provocar un bajo nivel de portabilidad, es necesario que no se vean estas diferencias como una comparación sino como problemas en los navegadores que repercuten en la interpretación de las aplicaciones y de esta forma provocando errores de portabilidad.

3.5. Métricas de evaluación de la portabilidad

Para un mejor entendimiento de las métricas que se definirán a continuación se da una breve descripción de la estructura por la que están conformadas.

1. La métrica representa el parámetro o la característica que evalúa la portabilidad.
2. La precondición es una condición que debe cumplirse antes de arrojar un resultado
3. Los posibles resultados son las respuestas que se pueden obtener luego de aplicada la métrica
4. El peso indica el grado en que se utiliza la métrica; para esto se define una escala la cual se divide en 3 grupos:
    - Si no se usa el peso es 0
    - Si se utiliza poco (de 1 - 4 veces) el peso es de 0.1 a 0.4 respectivamente.
    - Si se utiliza mucho (más de 4) el peso es de 0.5 a 1 respectivamente

El costo es un coeficiente que indica el grado de repercusión en la portabilidad al utilizar la métrica en cuestión y este estará dado por un valor de 0 a 1.

El grado o por ciento de portabilidad de la aplicación que se esté analizando se determinara de la siguiente manera:

La siguiente formula determina el porciento que representan los problemas encontrados en la aplicación analizada:

$$Porciento\ de\ Problemas = \left(\frac{\sum(Costo * Peso)}{Catidad\ de\ Métricas}\right) * 100$$

Entonces el porciento de portabilidad de la aplicación sería:

$$Porciento\ de\ Portabilidad = 100 - Porciento\ de\ Problemas$$

3.5.4. Métricas para PHP 5.0

Métricas de portabilidad del código fuente para aplicaciones desarrolladas en PHP 5.0

Métrica # 1	Uso de la función mysqli_bind_param
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 2	Uso de la función mysqli_enable_reads_from_master
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 3	Uso de la función mysqli_disable_rpl_parse
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 4	Uso de la función mysqli_execute

Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.7

Métrica # 5	Uso de la función mysqli_fetch
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.6

Métrica # 6	Uso de la función mysqli_get_metadata
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.5

Métrica # 7	Uso de la función mysqli_master_query
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 8	Uso de la función mysqli_rpl_parse_enabled

Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 9	Uso de la función mysqli_rpl_probe
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 10	Uso de la función mysqli_param_count
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.6

Métrica # 11	Uso de la función mysqli_send_long_data
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.5

Métrica # 12	Uso de la función mysqli_rpl_query_type
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 13	Uso de la función mysqli_send_query
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 14	Uso de la función mysqli_slave_query
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 15	Uso de la directiva zend.ze1_compatibility_mode
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.7

Métrica # 16	Uso de la directiva zend_extension_
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.7

Métrica # 17	Uso de la función define_syslog_variables
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.7

Métrica # 18	Uso de la función array_reduce
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.3
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.7

Métrica # 19	Uso de la función php_check_syntax
Precondición	Se encuentra trabajando con PHP 5.0 y se desea migrar a PHP 5.1.0
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.7

3.5.5. Métricas para Symfony 1.2.

Métricas de portabilidad para el código fuente de las aplicaciones que utilizan el Framework Symfony 1.2.

Métrica # 1	Uso del plugin sfCompat10Plugin:
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.2

Métrica # 2	Uso de la función getTmpDir
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 3	Uso de la función removeArrayValueForPath
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.5

Métrica # 4	Uso de la función hasArrayValueForPath
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***

No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.6

Métrica # 5	Uso de la función getArrayValueForPathByRef
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.6

Métrica # 6	Uso de la función setInvalidMessage
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.8

Métrica # 7	Uso de la función setRequiredMessage
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.8

Métrica # 8	Uso de la función Contains
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	

Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.7

Métrica # 9	Uso de la función isRedirected
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.8

Métrica # 10	Uso de la función isStatusCode
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 11	Uso de la función responseContains
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 12	Uso de la función isRequestParameter
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.

Posibles Resultados	
Respuesta	**Peso**
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.7

Métrica # 13	Uso de la función isResponseHeader
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	**Peso**
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 14	Uso de la función isUserCulture
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	**Peso**
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.2

Métrica # 15	Uso de la función isRequestFormat
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	**Peso**
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 16	Uso de la función checkResponseElement

Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 17	Uso de la función isCached
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 18	Uso de la función isUriCached
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 20	Uso de la función debugMessage
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 21	Uso de la función loadPluginConfig
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.5

Métrica # 22	Uso de la función getMethodName
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.5

Métrica # 23	Uso de la función getElements
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.8

Métrica # 24	Uso de la función getXDebugStack
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 25	Uso de la función heckSymfonyVersion
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.5

Métrica # 26	Uso de la clase sfDoctrineLogger
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.5

Métrica # 27	Uso de la clase sfLoader
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.8

Métrica # 28	Uso de la clase sfDoctrineDataRetriever
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1

Costo	0.6

Métrica # 29	Uso de la clase sfWidgetFormChoiceMany
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.5

Métrica # 30	Uso de la clase sfValidatorChoiceMany,
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.5

Métrica # 31	Uso de la clase sfWidgetFormDoctrineChoiceMany
Precondición	Se encuentra trabajando con Symfony 1.2 y se desea migrar a Symfony 1.3 o 1.4.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.2

3.5.6. Métricas para Spring 2.0.

Métricas de portabilidad para el código fuente de las aplicaciones que utilizan el Framework Spring 2.0.

Métrica # 1	Uso de la Interface XmlBeanDefinitionParser
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.

Posibles Resultados	
Respuesta	**Peso**
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 2	Uso de la Interface ListenerSessionManager
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	**Peso**
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 3	Uso de la Interface PagedListSourceProvider
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	**Peso**
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 4	Uso de la Interface ResponseTimeMonitor
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	**Peso**
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 5	Uso de la Interface ServerSessionFactory

Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 6	Uso de la Interface ThrowawayController
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 7	Uso de la Clase AdvisorChainFactoryUtils
Precondición	Se encuentra trabajando con Spring 2.0 hasta una versión anterior a la 2.1 y se desea migrar a Spring 2.1 o superior.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 8	Uso de la Clase DynamicMethodMatcherPointcutAdvisor
Precondición	Se encuentra trabajando con Spring 2.0 hasta una versión anterior a la 2.1 y se desea migrar a Spring 2.1 o superior.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 9	Uso de la Clase HashMapCachingAdvisorChainFactory
Precondición	Se encuentra trabajando con Spring 2.0 hasta una versión anterior a la 2.1 y se desea migrar a Spring 2.1 o superior.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 10	Uso de la Clase AbstractPathMapHandlerMapping
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 11	Uso de la Clase AbstractPoolingServerSessionFactory
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 12	Uso de la Clase LoaderAnalyzerInterceptor
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 13	Uso de la Clase LoaderUtils
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 14	Uso de la Clase CommonsPathMapHandlerMapping
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 15	Uso de la Clase CommonsPoolServerSessionFactory
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 16	Uso de la Clase PathMap
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 17	Uso de la Clase PerformanceMonitorListener
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 18	Uso de la Clase ReflectiveVisitorHelper
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

0.6

Métrica # 19	Uso de la Clase RefreshablePagedListHolder
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 20	Uso de la Clase ResponseTimeMonitorImpl
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

Métrica # 21	Uso de la Clase ServerSessionMessageListenerContainer
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 22	Uso de la Clase ServerSessionMessageListenerContainer102
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.4

Métrica # 23	Uso de la Clase SimpleServerSessionFactory
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.5

Métrica # 24	Uso de la Clase ThrowawayControllerHandlerAdapter
Precondición	Se encuentra trabajando con Spring 2.0 ó 2.5 y se desea migrar a Spring 3.0.
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.3

3.5.7. Métricas para los navegadores Firefox 3.0 e Internet Explorer 6.0.

Métricas de portabilidad en el comportamiento del DOM en los navegadores IE6 y Firefox 3.0

Métrica # 1	Uso de la función document.elementName
Precondición	Que se interpreten en IE6 y Firefox3.0 de la misma forma
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.8

Métrica # 2	Uso de la función globalnamespacepolluting
Precondición	Que se interpreten en IE6 y firefox3.0 de la misma forma
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.5

Métrica # 3	Uso de la función document.all
Precondición	Que se interpreten en IE6 y firefox3.0 de la misma forma
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.7

Métrica # 4	Uso de la función outerHTML
Precondición	Que se interpreten en IE6 y firefox3.0 de la misma forma
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.6

Métrica # 5	Uso de la función innerText
Precondición	Que se interpreten en IE6 y firefox3.0 de la misma forma
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.8

Métrica # 6	Uso de la función document.createDocumentFragment
Precondición	Que se interpreten en IE6 y firefox3.0 de la misma forma
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.8

Métrica # 7	Uso de la función getAdjacentElement
Precondición	Que se interpreten en IE6 y firefox3.0 de la misma forma
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.8

Métrica # 8	Uso de la función insertAdjacentHTML
Precondición	Que se interpreten en IE6 y firefox3.0 de la misma forma
Posibles Resultados	
Respuesta	***Peso***
No se usa	0
Se usa poco	0.1 a 0.4
Se usa mucho	0.5 a 1
Costo	0.8

3.5.8. Métricas para Java.

Métricas de portabilidad del código fuente para aplicaciones desarrolladas en Java

Métrica # 1	Uso de la JRE antigua
Precondición	Se encuentra trabajando con JRE antigua y se desea compilar con una JDK moderna
Posibles Resultados	
Respuesta	***Peso***
Se usa	Causa errores de compatibilidad
No se usa	No afecta la portabilidad

3.6. Conclusiones parciales

En este capítulo después de un estudio realizado en las diferentes tecnologías de desarrolla Web que se usan en la universidad, se logró detectar problemas de portabilidad en algunas de estas como son PHP 5.0, Symfony 1.2, Spring 2.0; los cuales poseen una serie de funciones, clases e interfaces que han sido eliminadas, originando problemas de portabilidad en los mismos. Además se detectaron incompatibilidades en la manera en que los navegadores Firefox 3.0 e IE 6.0 interpretan algunas funcionalidades del DOM. También existen otras tecnologías analizadas en los cuales no se logró detectar problemas por lo que se llegó a la concesión de que son portables, ejemplo de ello es Hibernate y Extjs. Todos estos problemas encontrados sirvieron como base para la definición de las métricas de portabilidad que se establecieron para la confección de la herramienta que se desarrollará.

CAPITULO 3. Desarrollo y validación de la herramienta

3.1. Introducción

En cuba así como en la UCI existe actualmente una tendencia hacia la utilización del software libre por diversas ventajas que este ofrece. El auge de la migración desde los sistemas con licencia comercial, que solo están en manos de los monopolios o empresas privatizadas de la informática hacia aquellos cuyo uso está libre de pagos, es cada vez mayor. La UCI necesita realizar esta migración hacia el software libre lo más rápido posible debido a que se pretende crear una industria avanzada en el campo de la informática. Este trabajo parte de esta premisa y se propone la realización de una herramienta que cumpla con las necesidades que la originaron, utilizando principalmente las herramientas y tecnologías libres.

En el presente capítulo se pretende desarrollar una herramienta para evaluar la portabilidad del código fuente de las aplicaciones Web, utilizando las métricas definidas en el capítulo anterior. Con dicha herramienta se podrá llevar a la práctica y de forma dinámica toda la investigación realizada. Una vez terminada la herramienta se pasará a su validación para asegurar que cumple con las expectativas, esto se realizará con una aplicación de la misma en proyectos de la universidad, guardando todos aquellos resultados arrojados para su aprobación y con la consulta a expertos de las diferentes aéreas presentes en este estudio.

3.2. Descripción de la Herramienta

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 encontrado 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.

3.2.1. Diagrama de clases

A continuación se muestra el diagrama de clases de la herramienta así como las principales funcionalidades con las que se dio cumplimiento a la creación de esta herramienta.

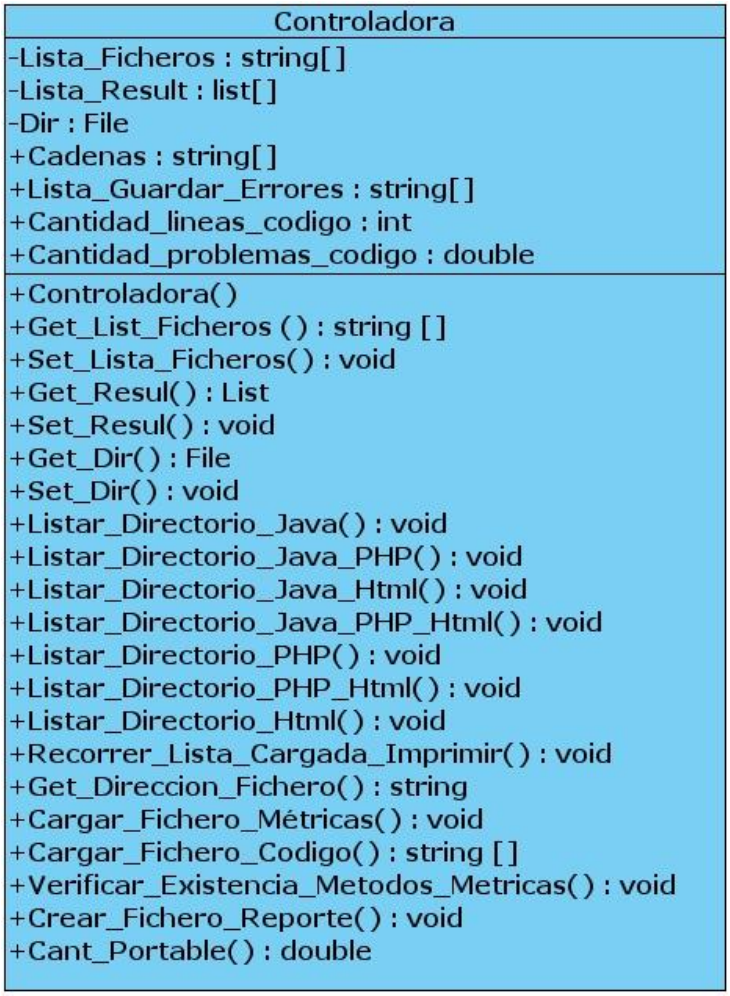


Fig. Nro.1: Diagrama de Clases

3.2.2. Patrones de diseño

Creador: Se usó para guiar el trabajo sobre las listas con la creación de objetos.

Experto: Este patrón fue utilizado para que cada objeto realizara la funcionalidad de acuerdo a la información que domina. Es decir cada objeto está relacionado con una determinada lista de información sobre la cual este será el que realice las operaciones.

3.2.3. Estándares de codificación

El estilo y la forma en que un programador traza sus líneas de código es un aspecto muy importante, ya que estos son los que le brinda al programa legibilidad y entendimiento del código escrito. Para el desarrollo de esta herramienta se propuso el uso de un código simple y entendible permitiendo de esta manera que la aplicación sea escalable y que a medida que surgen nuevas problemáticas de portabilidad, estas puedan ser agregadas y tratadas la herramienta desarrollada.

Estándar de nomenclatura

Se utilizó la notación Camel que consiste en escribir los identificadores con la primera letra de cada palabra en mayúsculas y el resto en minúscula. Para esto existen dos variantes:

1. UpperCamelCase, CamelCase o PascalCase: En esta variante la primera letra también es mayúscula.
2. lowerCamelCase, camelCase o dromedaryCase: En esta variante la primera letra es minúscula.

- **Nomenclatura de las Clases:**

Para este caso se utilizó la variante UpperCamelCase de manera tal que los nombres de las clases comenzarán con Mayúscula y a continuación el resto en minúscula.

Ejemplo: Controladora

- **Nomenclatura de las funciones:**

Para este caso también se utilizó el estilo UpperCamelCase, pero se decidió hacerle una modificación al mismo de modo tal que ahora en caso de ser un nombre compuesto se dividirá por un underscore y la siguiente palabra se escribirá de la misma forma.

Ejemplo: Listar_Directorio_Java ()

- **Normas de comentario**

Se usa el comentario para explicar algunas funciones que su entendimiento no sea claro a simple vista y se hará de la siguiente forma

Ejemplo: // aquí va el comentario

3.3. Interfaz de Usuario

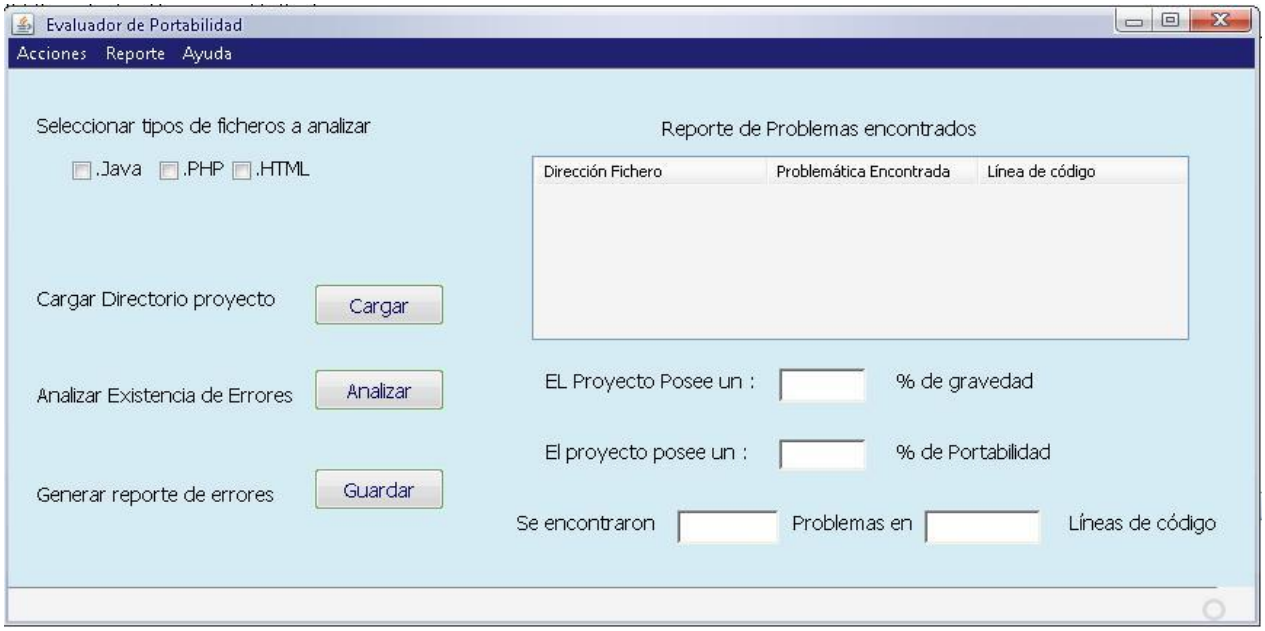


Fig. Nro.2: Interfaz de usuario de la herramienta

3.4. Validación de la herramienta

Para darle cumplimiento a los objetivos de la validación de la herramienta desarrollada, y así probar su nivel de funcionalidad en relación con los resultados que se esperan de esta, se evaluaron algunos de los proyectos que se encuentran actualmente en desarrollo en la universidad. A dichos proyectos se les aplicó el proceso de evaluación con la herramienta obteniendo resultados satisfactorios, ya que cumplió con los objetivos trazados brindando una medida de la portabilidad del código fuente de cada una de las aplicaciones evaluadas, así como señalando cada uno de los problemas encontrados en estos.

A continuación se describe el proceso de evaluación de los proyectos a los cuales se les realizaron las pruebas.

Proyecto Sistema de Gestión Fiscal (SGF): Su objetivo principal es informatizar los procesos de la fiscalía, está desarrollado en PHP y utilizan el framework Symfony.

Los resultados que se obtuvieron en la evaluación de dicho proyecto se muestran a continuación:

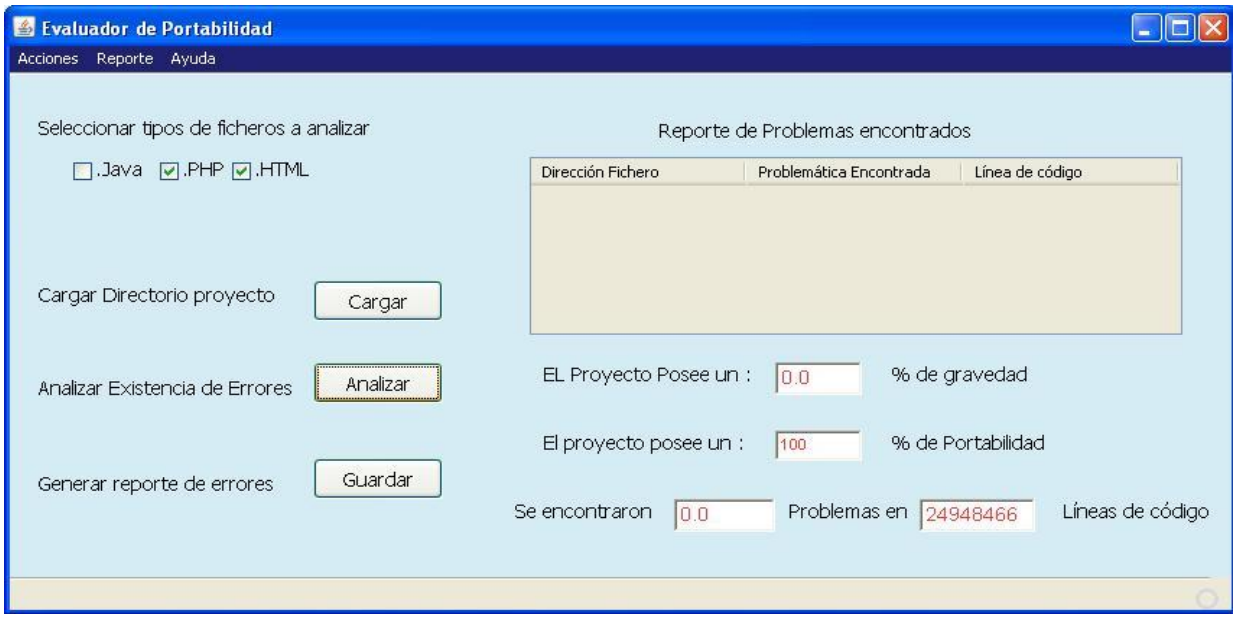


Fig. Nro.3: Resultados de la evaluación al proyecto Fiscalía (SGF)

Proyecto Convenio Cuba-Venezuela (CCV): Su objetivo principal es informatizar todos los procesos de gestión y convenios entre ambos países, está desarrollado en Java y utilizan frameworks como Hibernate, Spring.

Los resultados que se obtuvieron en la evaluación de dicho proyecto se muestran a continuación:

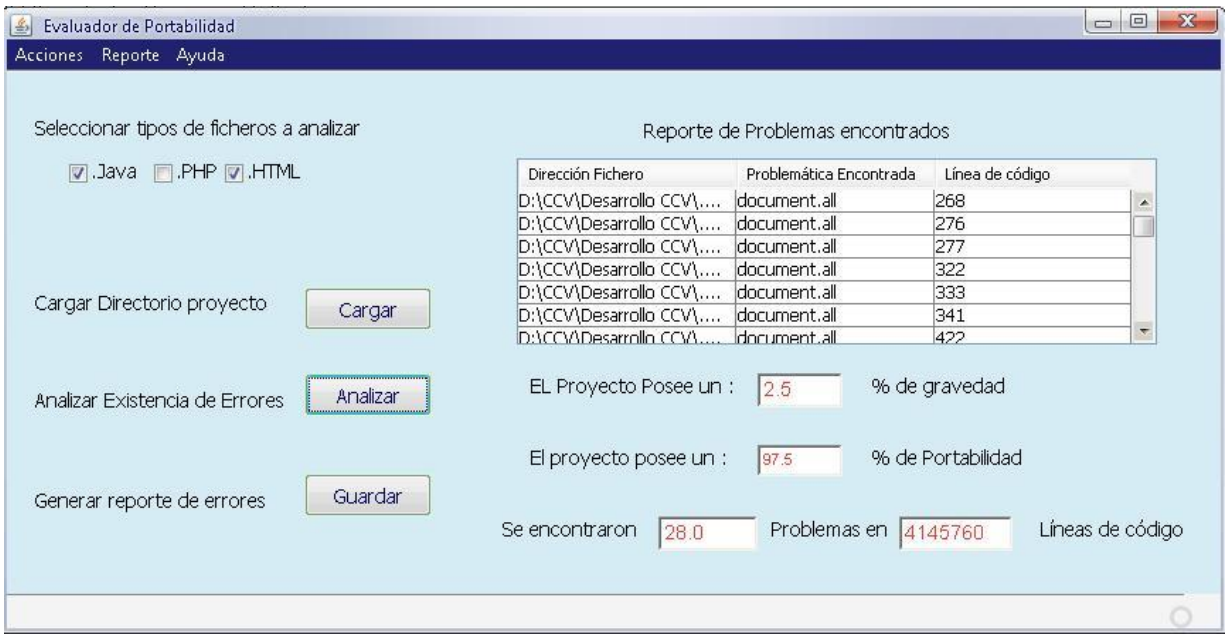


Fig. Nro.4: Resultados de la evaluación al proyecto CSV

Proyecto Planeación de Recursos Empresariales (ERP): Su objetivo es informatizar todos los procesos de gestión de recursos empresariales en Cuba, está desarrollado en PHP y utiliza el framework Symfony.

Los resultados que se obtuvieron en la evaluación de dicho proyecto se muestran a continuación:

Módulo de Seguridad:

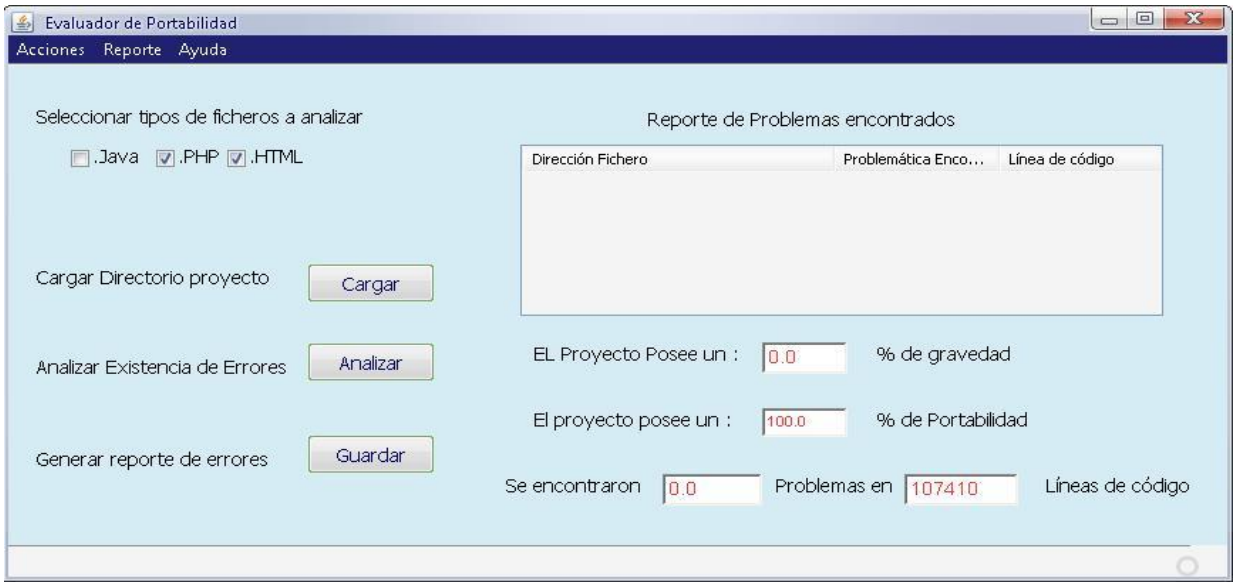


Fig. Nro. 4: Resultados de la evaluación al proyecto ERP (módulo de seguridad)

Modulo Contabilidad:

Fig. Nro. 5: Resultados de la evaluación al proyecto ERP (módulo de Contabilidad)

Modulo Logística:

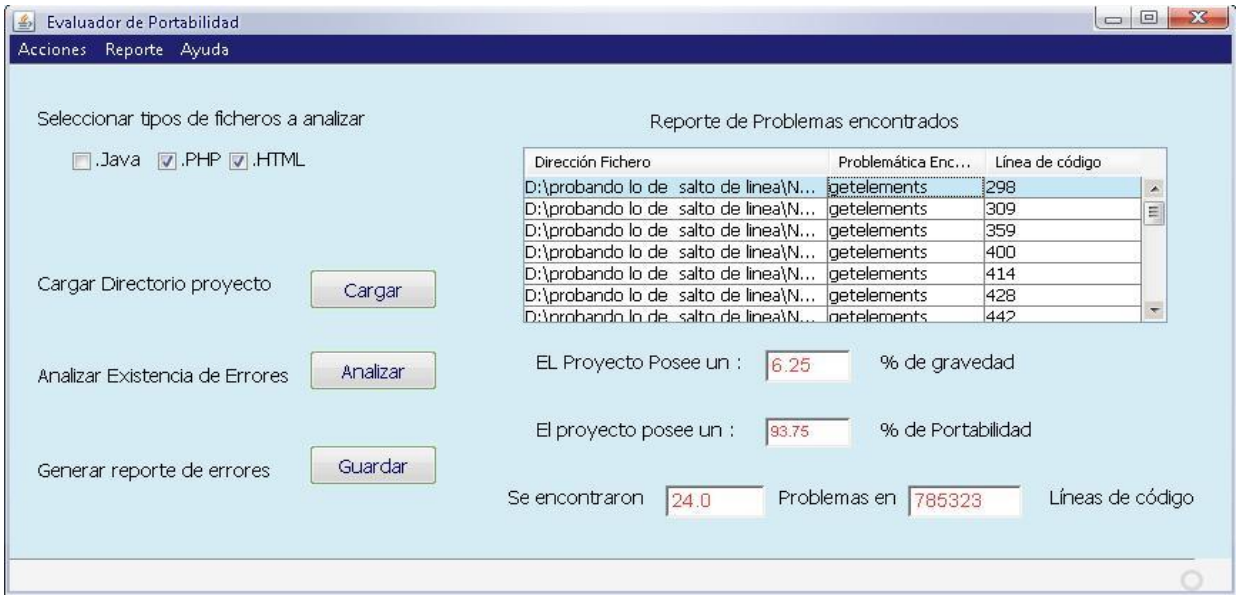


Fig. Nro. 6: Resultados de la evaluación al proyecto ERP (módulo de Logistica)

Modulo Planificación:

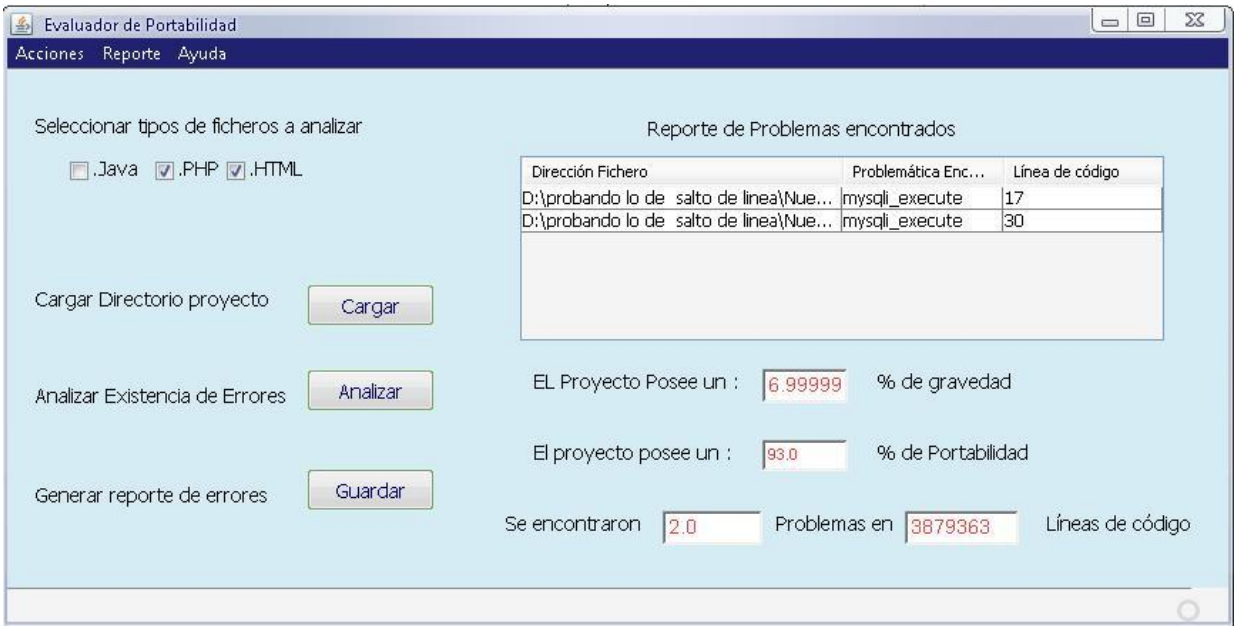


Fig. Nro. 7: Resultados de la evaluación al proyecto ERP (módulo de Planificación)

Modulo Finanzas:

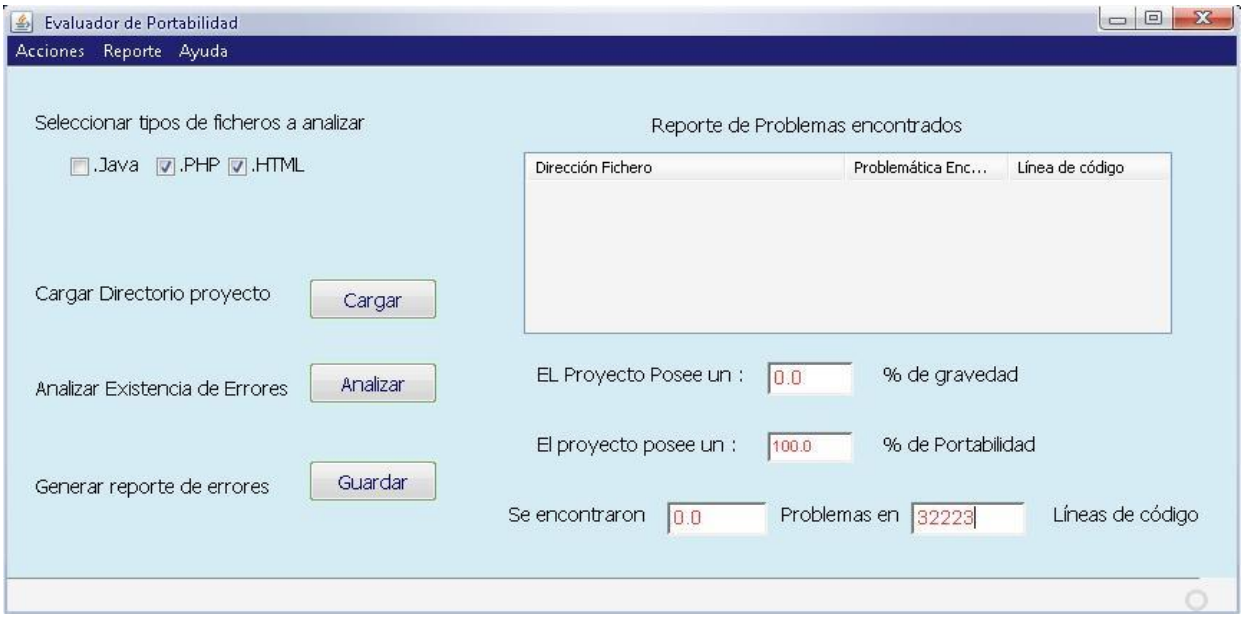


Fig. Nro. 7: Resultados de la evaluación al proyecto ERP (módulo de Finanzas)

Además se consultaron expertos de la universidad pertenecientes a los centros Calisoft, CEGEL, así como al arquitecto principal del proyecto ERP y el del proyecto Aduana, que cuentan con una experiencia y conocimiento avanzado en las diferentes tecnologías que fueron analizadas y en la definición de métricas y pruebas de calidad de software, los cuales brindaron su apoyo y aprobación de las métricas definidas para la evaluación de la portabilidad después de que estos las estudiaron y analizaron. Para ello se les hizo entrega de una planilla la cual llenaron expresando su opinión y aprobación sobre los resultados que se pueden obtener de las métricas definidas. (Ver anexo Nro. 4)

3.5. Conclusiones Parciales

Con el resultado del desarrollo y validación de la herramienta que se ha creado se ha logrado que la UCI cuente con un mecanismo para evaluar la portabilidad del código fuente de los productos Web que en la misma se desarrollan. Esta herramienta también servirá para que los miembros de los proyectos puedan tener una medida y control de los posibles errores de portabilidad que pueden tener sus aplicaciones, además de que les facilita el trabajo a la hora de detectarlos para su corrección. La validación de esta herramienta fue realizada con el apoyo de algunos proyectos de la universidad, cumpliendo con las expectativas trazadas y brindándoles a estos una medida de cuan portable estos son, así como un reporta de cada uno de los problemas de portabilidad con que cuentan actualmente estos proyectos.

Conclusiones

A partir de los resultados de la investigación realizada, del desarrollo de la herramienta para la evaluación de la portabilidad del código fuente en aplicaciones Web y de la validación de la misma para su correcto funcionamiento se arriba a las siguientes conclusiones.

Como resultado del estudio del estado del arte acerca de la existencia en la universidad de alguna herramienta que permita evaluar la portabilidad de una aplicación Web, se determinó que no existe en la UCI una herramienta que permita dicha funcionalidad y que de existir sería de gran ayuda en el proceso de evolución de las aplicaciones tanto para los miembros de los proyectos como para los grupos de calidad en la universidad.

En la investigación que se realizó en los proyectos de la universidad, se determinó que los principales lenguajes de programación que se utilizan por el momento en la misma para el desarrollo Web son PHP 5.0 y Java. Así como otras tecnologías como son Hibernate 2.0, Symfony 1.2, Spring 2.0 y Extjs 2.2. También se llegó al consenso de que más del 60 % de los proyectos de la UCI están vinculados con el desarrollo Web.

Se logró determinar en todas estas tecnologías los principales problemas que afectan la portabilidad del código fuente en aplicaciones Web, los cuales fueron utilizados para definir las métricas que darán una medida de la gravedad de la portabilidad en una aplicación Web.

En la validación de las métricas definidas que se realizó con algunos trabajadores de la universidad relacionados con el tema y que poseen una amplia experiencia, se logro obtener la aprobación de estos ya que las métricas estaban correctamente diseñadas.

En el desarrollo y validación de la herramienta, se logró implementar una aplicación la cual le brinda al usuario el conocimiento de cada uno de los problemas de portabilidad presentes en su producto Web, así como el por ciento de gravedad que representan todos estos problemas relacionados con la portabilidad de su aplicación. La validación de la herramienta demostró que será de gran ayuda para los proyectos en desarrollo permitiéndoles a esto contar con una forma o método de de conocer los problemas que pueden afectar la portabilidad.

Recomendaciones

Después de concluido el presente trabajo de diploma se ofrecen las siguientes recomendaciones:

- Que la herramienta se utilice en los proyectos de desarrollo Web de la UCI, para que ayude a mejorar el proceso de evaluación de los mismos.

- Mejorar la herramienta desarrollada para que sean tratadas las aplicaciones Web que se desarrollan en la UCI con tecnologías que no se abordaron en esta investigación para que pueda ser aplicada a todos los proyectos desarrollados por la misma.
- Darle un mejoramiento a la interfaz de la herramienta para lograr una mayor facilidad de uso.
- A medida que avancen las tecnologías de desarrollo Web y se identifiquen los nuevos problemas de portabilidad, se le agreguen a la herramienta para garantizar la actualidad de la misma.

Glosario de términos

Código fuente: El código fuente de un programa informático (software) es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está descrito por completo su funcionamiento.

Componente de Software: Los componente de Software son todo aquel recurso desarrollado para un fin concreto y que puede formar solo o junto con otro/s, un entorno funcional requerido por cualquier proceso predefinido.

Adaptabilidad: Capacidad de acomodarse o ajustarse una cosa a otra.

Portabilidad: Característica que poseen ciertos programas que les permiten funcionar o ejecutarse en distintos ordenadores sin que necesiten cambios de importancia.

Escalabilidad: Es la propiedad que posee un sistema que le permite modificar su configuración y su tamaño manteniendo su calidad y funcionalidad en sus servicios, ajustándose a los cambios.

Transacciones: Es un acuerdo comercial entre personas o empresas

Redes informáticas: Es una serie de dispositivos y ordenadores o de ambos tipos que están conectados entre sí ya sea de manera física (mediante un cable) o inalámbrica.

Servidor Web: un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP.

Navegador Web: es un programa que permite visualizar la información que contiene una página web

Hardware: corresponde a todas las partes físicas y tangibles de una computadora: sus componentes eléctricos, electrónicos, electromecánicos y mecánicos sus cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado.

Script CGI: Son programas que consisten generalmente en una serie de instrucciones escritas en un lenguaje de programación que procesan la petición de un navegador, ejecutan un programa y formatean los resultados en HTML de manera que puedan ser presentados en el navegador.

Bytecodes: Es el código independiente del hardware generado por el compilador Java y ejecutado por su intérprete.

API: Es un conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación.

Software libre: Es un tipo de software que le permite al usuario ejecutarlo con cualquier propósito, estudiar cómo funciona y adaptarlo a sus necesidades, distribuir copias, mejorarlo y brindar esas mejoras al público.

Proyecto productivo: Conjunto de actividades coordinadas e interrelacionadas para cumplir un objetivo específico en un periodo de tiempo previamente definido y acorde a un presupuesto.

Framework: Es un esquema para el desarrollo y/o implementación de una aplicación

Plataforma: Es el principio en el cual se constituye un hardware, sobre el cual un software puede ejecutarse o desarrollarse.

Patrón: Es un modelo que representa un problema que puede ocurrir en un determinado ambiente, así como también muestra la solución a este problema

Sistema Operativo: Es el software básico de una computadora que provee una interfaz entre el resto de los programas del ordenador, los dispositivos hardware y el usuario.

Underscore: Símbolo que se utiliza para dejar un espacio de manera tal que no quede vacio sino con un guión bajo.

ORM: Es una técnica de programación utilizada para convertir datos entre un lenguaje de programación orientado a objetos y una base de datos relacional

Visual Studio.NET: Es un entorno de desarrollo integrado de Microsoft, utilizado para desarrollar aplicaciones de consola, graficas de interfaz de usuario, Windows form, sitios Web, así como aplicaciones y servicios Web.

Framework Acegi: Es uno de los mejores frameworks de seguridad existentes en java.

Portlets: Componentes moduladores de las interfaces de usuarios gestionadas y visualizadas en un portal Web.

Herramienta Case: Herramientas que brindan un conjunto de funcionalidades y ayudas a los desarrolladores de software durante todo el ciclo de vida de desarrollo del software

Zend Engine: Framework que permite a los desarrolladores Web que utilizan PHP soluciones integradas que permiten desarrollar, proteger y ampliar sus aplicaciones

MySQL: Sistema de gestión de bases de dato**.**

UML: Lenguaje de modelado de sistemas de software

Bibliografía

Capo, Rosa María. 1998. El profecional de la Información . [En línea] 1998. [Citado el: 13 de 2 de 2010.] http://www.elprofesionaldelainformacion.com/contenidos/1998/septiembre/iso_9000_nueva_publicacion_de_sedic.htm.

Castillo, Javier, Theo, Otto y Corcuera, Jose Luis. 2008. Desarrollo en Web. [En línea] 22 de 10 de 2008. [Citado el: 25 de 1 de 2010.] http://blogs.antartec.com/desarrolloweb/tag/extjs/.

Community, Spring. 2007. www.springsource.org. [En línea] 2007. [Citado el: 20 de 2 de 2010.] http://static.springsource.org/spring/docs/2.0.x/api/deprecated-list.html.

Contreras, Matías Fuentes. 2008. Monografias.com. [En línea] 31 de 7 de 2008. [Citado el: 20 de 1 de 2010.] http://www.monografias.com/trabajos56/modelo-cmmi/modelo-cmmi.shtml#xcapab.

DeMarco, Tom. 1999. *Management Can Make Quality.* 1999.

Doron Rosenberg, IBM Corporation. 2009. Mozilla<developer center/>. [En línea] 26 de 7 de 2009. [Citado el: 15 de 3 de 2010.] https://developer.mozilla.org/es/Migrar_aplicaciones_desde_Internet_Explorer_a_Mozilla.

Estévez, Isabel Pérez. junio de 2002. *MÉTRICAS PARA EL CONTROL DE PROYECTOS.* junio de 2002.

Hooping.net. 2008. Hooping.net. [En línea] 2008. [Citado el: 1 de 2 de 2010.] http://www.hooping.net/glossary/aplicaciones-web-146.aspx.

HTML_Help, Maicosoft. 2005. *Manual de PHP.* 2005.

IEEE. 1992. *IEEE Standard for Software Quality Management System,.* 1992.

Ing. Roberto Hugo Vázquez. 2006. *Introduccion a la Calidad del Software.* 2006.

ISO 9000:2000.

Javier, Eguiluz. 2010. Symfony.es. [En línea] 2010. [Citado el: 17 de 3 de 2010.] http://www.symfony.es/documentacion/elementos-declarados-obsoletos-en-symfony-1-3/.

Lovelle, Juan Manuel Cueva. 1999. *Calidad del Software.* España : s.n., 1999.

Maylen Bon Pérez, Danelys Brito González. 2009. *Propuesta de Modelo para el Análisis Cuantitativo de Riesgos basado en técnicas de Softcomputing.* 2009.

Mooney, James D. 1997. *Bringing Portability to the Software Process.* Univercidad de Virginia Occidental : Departamento de Estadisticas y Ciencias Informáticas, 1997.

Pillou, Jean Francois. 2004. Kioskea.net. [En línea] 2004. [Citado el: 15 de 1 de 2010.] http://es.kioskea.net/contents/qualite/qualite-introduction.php3.

Pressman. 1998. *Aspectos y metricas de la calidad del Software.* 1998.

Programacion, Lenguajes de. 2009. Lenguajes de Programacion. [En línea] 2009. [Citado el: 10 de 2 de 2010.] http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml.

Pulido, Hernán Javier. 2004. ESTÁNDARES DE CALIDAD. [En línea] 2004. http://www.google.com.cu/url?sa=t&source=web&ct=res&cd=8&ved=0CB0QFjAH&url=http%3A%2F%2Fwww.udenar.edu.co%2Fviceacademica%2Facre_files%2FREGISTRO%2520CALIFICADO%2FEST%25C1NDARES%2520DE%2520CALIDAD.doc&rct=j&q=estandar&ei=rNrIS_KUHZWy9gSRwtjfCg&usg=AFQjCNF.

Quiñones, Ernesto. 2009. APESOL. *Modelos de Calidad de Software y Software Libre.* [En línea] 2009. [Citado el: 16 de 1 de 2010.] http://www.apesol.org.

Saavedra, Teófilo Vargas. 2005. Gestión de la Calidad y BPA. [En línea] 2005. [Citado el: 7 de 2 de 2010.] http://bpa.peru-v.com/Deming.htm.

W3C. 2008. W3C Word Wide Web Consortium. [En línea] 9 de 1 de 2008. http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo.