

Urkund Analysis Result

Analysed Document: Veleztres.docx (D44323774)
Submitted: 11/21/2018 12:18:00 PM
Submitted By: rlopezb@unemi.edu.ec
Significance: 0 %

Sources included in the report:

Instances where selected sources appear:

0

INTRODUCCIÓN

La tecnología, uno de los mayores avances de la humanidad, nació por el afán de la sociedad de querer mejorar el estilo de vida del ser humano. En la actualidad la manera de vivir en sociedad ha evolucionado, la razón principal de esto es que se han desarrollado nuevas tecnologías que han generado grandes aportes en la comunicación y los medios en los que nos comunicamos.

La vida de las personas dio un cambio inesperado con el surgimiento de lo digital y la computadora, después apareció el internet, y así sucesivamente han ido apareciendo con el paso de los años nuevas formas de tecnologías creando infinitudes de aplicaciones y software para ayudar al hombre no solo a comunicarse de manera más rápida y fácil si no también le ha ayudado a poder sobrellevar sus actividades diarias tanto como laborables y educativas de una manera más fácil y con eficiencia, sin embargo, llega el momento en el que debemos despedirnos de un programa o un sistema operativo que llevamos utilizando por años porque este deja de ser útil ya que se vuelven lentos o presentan diversos problemas y en ese momento el mismo se vuelve obsoleto.

En el siguiente proyecto investigativo se analizará como el avance de la tecnología influye en la obsolescencia del software, además, se determinarán posibles soluciones como la migración de una aplicación obsoleta a una más actual.

PROBLEMA DE INVESTIGACIÓN

En un mundo cada vez más eficiente y sustentable, en el que los avances de la ciencia y la tecnología logran que innovaciones inimaginables se hagan realidad, vemos que ciertas aplicaciones se queden atrás al no ser compatibles con las nuevas exigencias tecnológicas.

La obsolescencia del software es un problema que no solo afecta a las empresas ya que esto conlleva a una serie de problemas en los equipos que son utilizados en labores diarias, también afecta a los pequeños usuarios que al verse familiarizados con un programa después de un periodo de adaptación que en algunos casos dependiendo de las circunstancias suele darse por un tiempo alargado como en las personas adultas mayores, tienen que decir adiós a un programa o un sistema que llevan años manejando ya que el mismo deja de ser útil porque los mismos hacen que los dispositivos se vuelvan lentos, pesado y presentan problemas de seguridad, pese a esto hay personas que se resisten al cambio y buscan alternativas para seguir trabajando de una manera parecida.

Con los avances tecnológicos el problema más común en cuento a un software es que en un momento determinado dejan de dar soportes a las versiones antigua de las bases de datos o se instalan nuevas versiones dejando obsoleta la que tenemos instaladas. En los computadores personales es normal que los se dejen de recibir soporte a los sistemas operativos después de años de servicio, uno de los casos más conocidos es el de Windows XP que finalizó en el 2014 y la versión actualizada fue la de Windows7.

El software que queda obsoleto no deja de funcionar en primeras instancias, lo primero que sucede es que deja de recibir actualizaciones y a partir de ahí inicia su inutilidad, esto provoca

que se vuelvan más inseguros ya que a mayoría de los problemas de seguridad que afectan al sistema operativos más modernos suponen un riesgo para ellos al dejar agujeros de seguridad que los atacantes pueden aprovechar para dañar los equipos, por otro lado también las aplicaciones dejan de funcionar y actualizarse, por ejemplo, Chrome uno de los buscadores más utilizados en la actualidad, dejó de recibir actualizaciones en Windows XP y Vista.

Los dispositivos móviles también se ven afectados como es el caso de Android ya que Google solo da soporte a sus dispositivos Nexus que también ya dejaron de recibir soporte, con iOS también ocurre algo parecido, aunque el ciclo de soporte es algo mayor. En el caso de los televisores la obsolescencia puede llegar incluso antes, por ejemplo, Smart tv ha llevado diferentes sistemas operativos al televisor que poco a poco van perdiendo aplicaciones que no se pueden actualizar y por lo tanto dejan de funcionar.

En el caso de las empresas que suelen quedarse estancadas en versiones inferiores de software obsoleto influye el factor económico ya que prefieren evitar el costo de adquirir versiones más recientes y actualizadas, esto trae como consecuencias que sean vulnerables ante un ciberataque.

Por lo general el software siempre va a tener un tiempo limite de vida por el avance tecnologico que hara que el software caduque y comience a tener fallas hasta llegar al punto de quedar obsoleto, lo que nos levara a actualizar o a adquirir un software mas actual acorde con la epoca.

Según Teresa, generalmente cuando creamos o compramos software lo hacemos pensando en que ellos tienen un tiempo limite por el avance tecnologico y las nuevas tendencias que día a día se siguen superando. Por eso es recomendable que al momento de adquirir algun software tener en cuenta que para adquirirlo primero necesitaremos saber los antecedentes del mismo, como son los soportes a usuarios y actualizaciones del software para poder prevenir algun contratiempo. Algunas empresas se quedan atrapadas con algun software obsoleto por evitar gastos de soporte y actualizaciones y que conllevan a tener problemas a largo plazo hasta el que el software este en un punto que ya no sea operable, es por eso que se recomienda tener siempre actualizado y con sus respectivas actualizaciones para que este en optimo funcionamiento al software. CITATION Ter15 \l 1033 (Teresa, 2015).

Mantener un software obsoleto generan algunos inconvenientes con su funcionamiento al intentar seguir utilizandolo, algunas causas son las siguientes:

- Lentitud de software
- Hardware incompatible
- Problemas de seguridad
- Costos de mantenimiento

Los problemas que conyeban a mantener funcionando un software o programas obsoleto se pueden notar muy facilmente con el transcurso del tiempo como, el mal funcionamiento de los programas, al momento de estar en el area de trabajo notar lentitud en el software, contagio de virus informaticos al ordenador y fallas de conectividad a otros dispositivos por

problemas de actualizaciones o versiones, todo esto repercute en el usuario que pierden tiempo al momento de trabajar CITATION Osc17 \l 1033 (Oscar, 2017).

Seguir operando con software o programas obsoleto deja a las pequeñas, medianas y grandes empresas vulnerables a cualquier tipo de ataque informático. Según un estudio de la empresa VMware el 38% los encargados del área tecnológica de las empresas en España, dicen que tener funcionando un sistema o software obsoleto son muy vulnerables a los ciberataques, ya que estos aprovechan las vulnerabilidades que tienen los diferentes software por no contar con actualizaciones que corrigen los fallos que se llegan a generar por la brechas de seguridad que no se corrigieron en su debido tiempo con los parches de actualizaciones CITATION Vic18 \l 1033 (Vicente, 2018).

OBJETIVOS OBJETIVO GENERAL Analizar la posible solución para evitar que el software se haga obsoleto por el avance tecnológico.

OBJETIVOS ESPECÍFICOS

- Determinar la influencia de los avances tecnológicos en la obsolescencia de un software.
- Identificar las causas principales por las que un software se hace obsoleto con el avance de la tecnología.
- Determinar los riesgos de utilizar un software obsoleto.
- Identificar posibles soluciones para la obsolescencia de un software.

JUSTIFICACIÓN

Los avances tecnológicos suponen logros significativos para la humanidad sin embargo en el caso del software puede significar cambios negativos ya que estos en su mayoría quedan Obsoletos. Esto supone problemas ya que alguno de ellos almacena información importante que algunos usuarios y empresas necesitan asegurar para que pueda ser fácilmente puesta a disposición para su futuro acceso, estudio y utilización.

Es importante analizar y determinar las causas por las que ocurre este suceso para así encontrar posibles soluciones y no perder información indispensable, además, para que en el futuro se pueda evaluar correctamente la utilización de un software.

MARCO TEÓRICO CONCEPTUAL La teoría del software fue propuesta por primera vez en un ensayo de Alan Turing en 1935 sobre los números que pueden ser computables con un intérprete para la toma de decisiones y leída por Jhon Tukey en 1958, desde esa fecha los estudios del software se dividieron en ingeniería en software e informática CITATION tip \l 1033 (tiposdesoftware, n.d.).

El software es la parte intangible de un ordenador compuesta por un conjunto de programas que permiten ejecutar tareas de un computador.

Según Revilla se puede evitar que el sistema se quede obsoleto, hay diversas formas de modelos de negocios en el ámbito de los sistemas operativos pero los sistemas con código abierto tienen más ventajas en la industria CITATION Dav15 \l 1033 (David, 2015).

El software es cualquier aplicación o programa informático como (Microsoft Office, sistemas operativos, reproductores de audio y video, etc), los cuales son creados por los diferentes tipos de lenguajes de programación que existen en el mercado del desarrollo de software, los cuales nos ayudan a interactuar con el ordenador mediante los códigos escritos e interpretados por algún IDE para poder tener el control total de todas las acciones del ordenador e interactuar con los usuarios o programas.

De acuerdo a, Castellano Casas Ricardo (2001). el software es el conjunto de instrucciones y datos en formato binario de ceros y unos (0,1) los cuales son gestionados por la memoria principal, que le indica al ordenador los pasos a seguir como y cuando se necesitan, es decir, el software manipula al hardware.

En las empresas para proteger a los ordenadores vulnerables por la obsolescencia es una prioridad, por el riesgo que estas traen consigo con los famosos ataques informáticos que cuando el software es obsoleto no habrá una actualización que pueda corregir esa brecha y la organización pueda correr el riesgo de pérdida de información CITATION Ism17 \l 1033 (Ismael, 2017).

Clasificación básica de software Software de sistema, denominados programas del sistema es el tipo de software que ayuda a funcionar al hardware y a la computadora sin ellos un ordenador no puede funcionar. Estos se encargan de tareas como el mantenimiento de archivos del disco y la administración de pantallas.

Software de aplicación, De acuerdo, Alcalde García Eduardo (1994). El software de aplicación: Es el compuesto por el conjunto de programas que han sido diseñados para que la computadora pueda desarrollar un trabajo. Permite al usuario hacer determinadas tareas como editar texto, guardar datos, sacar informes sacar cálculos y comunicarnos con otros usuarios y algunos trabajos típicos en el uso de las computadoras.

Software malicioso, en este grupo encontramos los tipos de programas que han sido creados para causar algún daño específico al ordenador como los virus, troyanos, gusanos, entre otros.

Software de programación, Estos permiten la creación de software al proveer herramientas de asistencia al programador.

Clasificación del software por su naturaleza de ejecución • Aplicaciones de escritorio, estos programas se ejecutan localmente en el computador o dispositivo móvil, por ejemplo, navegadores web, editores de texto, editores gráficos entre otros.

- Scripts, estas piezas de software se incluyen en sitios web para que puedan correrse dentro de un navegador cuando una página es abierta.

- Software de servidor, son aplicaciones web que se ejecutan en un servidor remoto y el resultado lo podemos observar en el navegador web.

- Software embebido, este tipo de software reside en el firmware y suelen ser dispositivos de un único propósito.
- Microcódigo, es un software especial que le da la orden al procesador como ejecutar código máquina.
- Clasificación del software por su arquitectura.
- Plataforma, es el que permite al usuario interactuar con el computador y sus periféricos.
- Aplicación, son los programas típicos que el usuario utiliza como navegadores, juegos, editores entre otros.
- Software escrito para el usuario, está adaptado para satisfacer las necesidades específicas del usuario.

Proceso de creación de un software Un proceso es un conjunto de pasos ordenados ejecutados para llegar a la solución de un problema o la obtención de un producto, en el caso de la creación de un software estos procesos pueden llegar a ser muy complejos dependiendo de sus características. Así para la creación de un sistema o programa se debe tener en cuenta la gestión que el programa que se esta desarrollando se dirija en los pasos adecuados, recursos para poder ejecutar el proyecto, equipo de trabajo con cualidades y aspectos adecuados para desarrollar el trabajo.

Para que el proyecto de creación de un software no tenga inconvenientes y fracase es necesario aplicar las reglas preestablecida en los procesos de desarrollo y se debe aplicar un modelo ese ciclo de vida, caso contrario el mismo no logrará cumplir con los objetivos previstos, esta es la principal causa de las fallas y fracasos al no aplicar una buena metodología.

Carácter evolutivo del software El software es un producto evolutivo en todo su ciclo de vida ya que cuando evoluciona genera versiones cada vez mas complejas, mejoradas, optimizadas, adecuadas a nuevas plataformas, en otras palabras, cada vez mas completas CITATION Ros12 \l 1033 (Rosero, 2012). El software cumplirá su ciclo de vida cuando este deje de evolucionar y se producirá lo que le llamamos obsolescencia, este software obsoleto será reemplazado por un producto actual tarde o temprano. Una de las principales causas por las que evoluciona el software es porque debe adaptarse a los cambios del entorno.

Meir M. Lehman y Belady, tuvieron la mayor contribución en el área del software, analizaron la evolución y el crecimiento de varios sistemas software llegando a la conclusión basadas en su análisis en las siguientes ocho leyes.

1) Cambio Continuo. Un programa debe ser continuamente adecuado, si no, gradualmente siempre no será muy satisfactorio.

2) Complejidad incremental. Cuando un programa progresa, su complejidad se aumenta, salvo que se trabaje en conservarla o en memorarla.

- 3) Autorregulación. El proceso de evolución de un programa se regula automáticamente según una colocación normal de atributos como el producto y el proceso.
- 4) Conservación de la estabilidad organizacional. En lo extenso del ciclo de vida del software, la velocidad intermedia global que tiene como actividad efectiva del sistema tiene un progreso constante. El esfuerzo en la evolución determina los actos de dirección de su dominio queda excluida por elementos externos, como los recursos utilizables, personal competente, etc. Por ello, la acción dedicada a la estabilización del software es aproximadamente constante.
- 5) Conservación de la familiaridad. En el ciclo de vida de un programa en evolución, el tamaño de las continuas versiones es constante estadísticamente. Uno de los elementos que establece la evolución software es la familiaridad según las versiones, cuantos más cambios se hacen a una versión más arduo es que todos los implicados los conozcan, un avance "grandioso" entorpecería esa familiaridad.
- 6) Crecimiento continuo. La funcionalidad de un programa debe aumentar continuamente para conservar al usuario satisfecho. Los sistemas aparecen con prohibiciones de funcionalidad, por motivos de recursos o tiempo, la causa por tiempo es por los requisitos que se apartaron al principio para después volver a aparecer.
- 7) Calidad decreciente. Se percibirá que un programa de tipo E pierde calidad a menos que se mantenga de manera rigurosa y se adapte al entorno operacional.
- 8) Sistema de realimentación. El procedimiento para la programación de un programa categoría E forma un sistema de realimentación multinivel y para tener éxito en su transformación debe ser tratado como tal.

La tecnología obsoleta es la necesidad de reemplazar equipamiento tecnologico tanto en hardware como software por el hecho de que cada año sale una version actual del mismo haciendo que cada año que pase llegue a su vida util y tenga que ser desechada CITATION INF18 \l 1033 (INFAIMON, 2018).

La obsolescencia del software La obsolescencia del software ocurre cuando con el tiempo el rendimiento del mismo tiende a deteriorarse a medida que corre y acumula errores. Existen softwares obsoletos que tienen consecuencias irreversibles ya que el entorno operativo del mismo y sus componentes se vuelven incompatibles, sin embargo, otros pueden remediar su estado al reinstalar completamente todos los componentes relevantes del software. Entre los factores que inciden tenemos los siguientes: • Cambios producidos por el entorno del programa que no anticipo el diseñador del mismo, esto provoca que el programa no pueda operar como lo hizo inicialmente. • Los cambios que no son relacionado con el diseñador si no con los usuarios, estos últimos pueden manejar la configuración del programa por un tiempo, pero después que el sistema presenta problemas quieren acceder a la configuración y no son capaces de repetir ese paso inicial ya que el contexto suele ser diferente y existe información que no está disponible. • Los códigos que normalmente no son usados como por ejemplo filtros de documento e interfaces usados por otros programas los cuales contienen errores que pasan desapercibidos y que después se expanden con los cambios en los requerimientos

de los usuarios. • Las actualizaciones que se producen al realizar un mantenimiento de software y sistemas pueden generar conflictos o fallos de seguridad que fueron encontrados en la versión anterior.

Existen dos formas de software obsoleto, la activa en donde el software esta continuamente siendo modificado puede perder con el tiempo su integridad si no se aplican los procesos de mitigación adecuados, sin embargo, alguno de estos sistemas requiere continuos cambios para corregir errores y satisfacer exigencias de los usuarios. Por otro lado, tenemos la forma inactiva de software obsoleto la cual consiste en que el software que no esta siendo usado se torne inservible poco a poco a medida que se vayan dando cambios en la aplicación.

La vida util de un software se puede notar al saber que en nuestras impresoras vienen con chips programables que cuando lleguen a un numero determinado de impresiones esta hagan que se bloquee y deje de funcionar, es un claro ejemplo de obsolescencia programada CITATION Ele13 \l 1033 (Elena, 2013).

Los riesgos de utilizar una plataforma de software obsoleto Estos riesgos dependen de la forma y las circunstancias en las que el sistema este siendo utilizado, así por ejemplo al comprar un equipo o al utilizar un software se debe analizar que eventualmente este quedara obsoleto con el tiempo por esta razón se debe tener en cuenta algunos factores al momento de adquirir estos productos. Una de los problemas de adquirir una nueva versión de software, ya que el que tenemos paso a ser obsoleto, es que los costos suelen ser elevados y no están a disposición económica de la mayoría de usuarios, es por eso que se deben de tomar en consideración los siguientes riesgos de utilizar un software obsoleto. • Alta vulnerabilidad y falta de actualizaciones, al no contar con un sistema actualizados los equipos tienen mas posibilidades que ser vulnerables antes ataques cibernéticos, cuando el software es obsoleto limita el acceso a actualizaciones que contienen parches para ayudar a evitar riesgos de este tipo. • Al no contar con actualizaciones los usuarios pierden soporte de sus equipos ya que a través de estas los fabricantes realización actualizaciones de seguridad y avisos para poder proteger su equipo de alguna amenaza. • Cuando tenemos un software obsoleto no se cuenta con un buen procesamiento de los procesos de información y el desempeño del equipo pierde eficiencia por la lentitud del equipo. • Los sistemas operativos obsoletos presentan problemas de incompatibilidad con las nuevas versiones de los equipos, esto significa que el equipo esta expuesto a posibles fallas, suspensión de ayuda y garantía.

Elementos que permiten evaluar la calidad en el software La calidad debe ser expresada por medio de otros términos que tengan más sentido para el usuario para poder ser entendida de una mejor manera y posteriormente ser medida con eficacia. En el software estos factores son el medio por el cual se traduce el término "calidad" al lenguaje de las personas que manejan la tecnología. (Juran, 1992) McCall, Richards & Walters (1977), clasificaron los factores que directamente afectan la calidad de software la misma que se concentra en tres aspectos importantes: • Capacidad para adaptarse a nuevos entornos. • Capacidad para experimentar cambios. • Características operativas.

McCall, Richards & Walters. Proponen de forma descriptiva los siguientes elementos: • Corrección, disposición del sistema cuando cumple y satisface todos los requerimientos del

cliente. • Confiabilidad, disposición del sistema cuando cumple su desempeño con la exactitud requerida por el cliente. • Eficiencia, códigos y recursos que son necesarios para que el sistema se ejecute correctamente. • Integridad, forma en que las personas no autorizadas pueden acceder al software y los datos. • Facilidad de uso, operatividad del programa que facilita el uso del mismo, • Facilidad de mantenimiento, grado de esfuerzo para poder localizar y corregir algún error. • Portabilidad, grado de esfuerzo para transferir algún sistema de un entorno de hardware o software a otro. • Facilidad de reutilización, al poder reutilizar en otras aplicaciones un programa o partes de él. • Interoperabilidad, manera en la que un sistema se puede acoplar a otro.

Según Irene: Algunas de las mas grandes empresas u organizaciones del pais todavia utilizan software obsoleto que dia a dia dificulta a la organización para que mejore y avance tecnologicamente haciendo que no sea competitivo en el mercado bursatil CITATION Ire16 \l 1033 (Irene, 2016).

Con lo expuesto podemos analizar que algunos de estos factores se deben tener en cuenta para evitar la obsolescencia del software ya que se relacionan directamente con este problema.

¿Podemos evitar que un sistema quede obsoleto? No existe alguna opción definitiva que pueda asegurar que un software no quede obsoleto sin embargo existen alternativas mismo como el uso de un sistema operativo de código abierto como las herramientas CASE que tienen ventajas claras para reducir la problemática CITATION Cen10 \l 1033 (Valdéz, 2010).

METODOLOGÍA

La finalidad del presente trabajo investigativo se basa en analizar la obsolescencia del software por el avance de la tecnología, es necesario recalcar que la investigación engloba la búsqueda y el descubrimiento de posibles soluciones al software obsoleto. A continuación, se detallan los tipos de investigación que abarca el proyecto.

Investigación explicativa La metodología explicativa busca explicar las causas que originaron el problema que está siendo analizado estableciendo conclusiones y explicaciones para así poder esclarecer las teorías propuestas y enriquecer las mismas en el transcurso de la investigación.

En este proyecto investigativo se utilizó esta metodología ya que se tuvo que explicar de manera detallada las causas de la obsolescencia del software para así comprobar la teoría planteada en el inicio.

Investigación documental Esta metodología consiste en la obtención de datos para ser analizados y así buscar resolver el problema planteado, esta recolección de datos se la puede realizar de manera empírica o diseño de campo mediante la recolección de datos primarios o el diseño bibliográfico que obtiene datos secundarios o de investigaciones ya existentes.

En esta investigación se aplicó el diseño bibliográfico al recolectar información a través de artículos científicos de fuente confiables los cuales permitieron ampliar mas el tema para poderlo analizar mejor.

DESARROLLO DEL TEMA

TEMA: "Análisis de la obsolescencia del software por el avance de la tecnología"

El software se esta volviendo obsoleto por el avance de la tecnología es una realidad que se palpa diariamente, sin embargo, no podemos detener estos avances tecnológicos que en otras circunstancias sirve de manera positiva a la humanidad. Pese a que no se puede detener los avances tecnológicos y con ello que el software que utilizamos quede obsoleto existe una manera de evitar que el software termine su tiempo de vida o en todo caso que este quede obsoleto muy rápido. La solución se encuentra en las denominadas Herramientas CASE (Computer Aided Software Engineering), actualmente existen diferenciaciones de lo que se entiende por CASE.

Ilustración 12Ciclo de vida del software

En ingeniería de software, una herramienta CASE es la que ayuda a desarrollar y conservar un software minimizando en términos de dinero y tiempo el costo de las mismas.

La creación de la primera CASE o herramienta CASE salió a la luz en el año 1984 fue Excelerator la cual trabajaba bajo una plataforma PC, actualmente, la adquisición de herramientas CASE se han extendido por parte de muchas empresas que busca automatizar los procesos de desarrollo de un sistema a pesar de los elevados costos de estas herramientas y los costos de los cursos para el personal.

El proceso de evolución de las herramientas CASE están siempre relacionadas con la Ingeniería de Software, esta ingeniería es una rama de las aplicaciones tecnologicas de diferentes metodologías del desarrollo y de técnicas, las cuales tiene como unico objetivo es incrementar la velocidad del proceso de desarrollo, por lo tanto en el caso de las herramientas CASE automatiza y apoya las fases del período de existencia del desarrollo de software a partir de su creación hasta la última etapa del uso del mismo.

El desarrollo del software debe estar guiado por una metodología que vaya acompañando cada una de las fases para su correcta creación, para esto se pueden escoger algunas herramientas CASE la cuales permiten llevar a cabo todas estas tareas de una manera mas eficiente, estas suelen incluir:

- Diccionario de datos donde se almacenan la información de los datos de la aplicación de la base de datos y cuta gestión es realizado por el apoyo de un sistema de gestión de base de datos.
- Herramientas de diseño que sirve de apoyo al analizar datos
- Herramientas que pueden permitir desarrollar el modelo de los datos corporativos.
- Herramientas que desarrollan prototipos de las aplicaciones.

Posibilidades del uso de una herramienta CASE

Las herramientas CASE tienen como propósito dar soporte automatizado en la aplicación de técnicas usadas por metodologías, contribuyendo a elevar la productividad y calidad en el desarrollo de los sistemas de información, además, estas herramientas están orientadas a resolver problemas que tienen que ver con la crisis del software para lo cual existen soluciones generales y particulares dependiendo de los niveles de los problemas como se muestran a continuación:

- Comparar el proceso de desarrollo de software como un proyecto de ingeniería de software.
- Implementar aplicaciones de tal forma que de una u otra técnica estas se pueden utilizar para cubrir todos los periodos del ciclo de vida del software.
- Apoyar los métodos utilizados mediante la utilización de una herramienta CASE.

El hardware con el paso del tiempo ha ido disminuyendo sus costos de manera considerable, mientras que con el precio del software ocurre lo contrario, sin embargo, estos precios son recompensados con los beneficios de traer estas herramientas al ofrecer una de las mejores opciones hoy en día para enfrentar proyectos de software de muy alta complejidad.

Son importantes los beneficios de la utilización de las herramientas case, entre ellos tenemos mejoramiento de la calidad del producto que se obtiene y el acrecentamiento de la productividad que tiene que ver con la disminución del costo del proyecto, tiempo y esfuerzo en fases de desarrollo y mantenimiento es la recompensa que tiene el fuerte impacto en la ejecución de la planificación, análisis y diseño que promueve una herramienta CASE. Adicionalmente obtenemos otros beneficios por la utilización de un CASE actual como lo son las siguientes: • Simplificar la comprobación de errores y mantenimiento engorrosos que son consistentes a la hora de cambiar la información del proyecto a realizarse.

- Simplificar la creación de estándares en procesos de desarrollo y documentación.
- Proporcionar mantenimiento a los sistemas y proporcionar sus respectivas documentaciones.
- Proporcionar la aplicación de técnicas en una sola metodología.
- Utilización de algunas funciones automatizadas como las siguientes: elaboración de prototipos, producción de código, reproducción de informes y plantillas, reproducción de esquemas físicos de las bases de datos, examinadores automáticos de estabilidad.
- Proporcionar la organización y gestión de proyectos informáticos.
- Proporcionar aplicación y metodologías de reutilización y reingeniería.

Después de analizar los beneficios de la utilización de las herramientas CASE se puede determinar que su uso en los procesos de desarrollo del software ayudará para que el mismo no quede obsoleto en un futuro o en todo caso según las circunstancias buscare soluciones para que su tiempo de vida se prolongue.

Organización de una herramienta CASE

La organización de las herramientas CASE se basan en los siguientes niveles:

- CASE alto nivel, herramienta que se utiliza para automatizar o contribuir con las fases finales del ciclo de vida del desarrollo de un software.
- CASE bajo nivel, herramienta que se utiliza para apoyar y automatizar las etapas finales o menores del ciclo de vida y el diseño minucioso del sistema.
- CASE cruzado de ciclo de vida, herramienta que aplica evaluaciones y actividades de cómo se gestionan los proyectos, que tienen lugar a lo extenso del ciclo de vida del software.

Ilustración 21 Bloques constitutivos de las herramientas case

Clasificación de las herramientas case

La clasificación de las herramientas CASE se puede determinar que no existe una clasificación única ya que es difícil contenerla en una clase determinada, sin embargo, se podrían clasificar tomando en cuenta características como plataformas que soportan las fases de ciclo de vida del software.

¿Cuándo se deben utilizar las herramientas case? Según sisinggroup. Cuando se comienza algún proyecto de software sea cual sea, se lo debe de comenzar bajo normas de calidad en todas las etapas del desarrollo del software para poder utilizar herramientas case las cuales ayudan al desarrollador de software y al usuario desde la primera hasta la última fase de desarrollo y se las utiliza en la etapa de análisis cuando el diseñador y desarrollador realizan la recolección de requisitos para diseñar los diagramas de uso y es allí cuando se los debe de utilizar CITATION SIS17 \l 1033 (SISINGGROUP, 2017).

Tabla 11 COMPARACIONES DE HERRAMIENTAS CASE UTILIZADAS Herramienta

Definicion Ventajas Desventajas GENEXUS Es una herramienta multiplataforma, agil, hecha para crear desarrollar y mantener aplicación, que se adapta a los cambios del estilo del negocio y a los cambios tecnologicos. Escalable Muy rapida en la creacion de aplicaciones ABM (Alta Baja y Modificaciones) permitiendo en tener resultados en el menor tiempo posible. NETBEANS Herramienta multiplataforma open source de desarrollo intuitivo, orientado a objetos y muy practico.

El código es escrito en java es leído por un interprete, lo que hace posible que su programa corra en cualquier plataforma. Administracion automático de la memoria. Es Gratis Lentitud al momento de ejecutar las aplicaciones (aunque con el tiempo a corregido este inconveniente). Necesita de un intérprete y el que utiliza es java. Algunas librerias y heramientas tienen costos adicionales StarUML Herramienta diseñada para el modelamiento del software, apoyado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture).

Es software libre. Muy parametrizable.

Tiene inconvenientes con la ingeniería inversa si se encuentran caracteres locales y tildes. No cuenta con una version versión para Linux reconocida solo con Wine.

Según la tabla se puede observar que la herramienta case Genexus se adapta mejor como una medida ante la obsolescencia del software por ser adaptable y escalable a los cambios del estilo de juego del negocio por los cambios constantes que surgen por los clientes o los cambios tecnológicos que cada día avanzan así.

GeneXus es una herramienta que funciona como un respaldo tecnológico, lo que hace es especificar los conocimientos. Un ejemplo de esto sería que si el día de mañana se nos presenta algún inconveniente con la nueva tecnología que siempre es cambiante, se podrá orientar el viejo conocimiento hacia el nuevo conocimiento tecnológico CITATION mob13 \l 1033 (mobileworldcapital, 2013).

Hoy en día la gente apuesta en el software como un servicio de retorno de inversión, que mientras más rápido el software se cree para el cliente más rápido se logrará recuperar la inversión invertida pero eso solo pasará cuando el cliente esté satisfecho con el resultado final CITATION Han \l 1033 (Hanna, n.d.).

CONCLUSIONES

Mediante el análisis de la información debidamente investigada y justificada se puede determinar lo siguiente.

- Los avances tecnológicos influyen de manera significativa en la obsolescencia del software, pese a que estos avances significan grandes logros en la humanidad desfavorecen la utilización de programas y sistemas ya que existen softwares obsoletos que tienen consecuencias irreversibles ya que el entorno operativo del mismo y sus componentes se vuelven incompatibles.
- Los factores que más inciden en la obsolescencia de un software tienen que ver con los cambios que se realizan en un tiempo determinado y que no estaban previstos en el desarrollo del mismo y con las actualizaciones que se producen al realizar un mantenimiento las mismas que generan conflictos o fallos de seguridad.
- Mantener un software obsoleto genera inconvenientes al intentar seguir utilizándolo como lentitud del software, hardware incompatible, problemas de seguridad y costos de mantenimiento, esto a su vez provocará molestia al usuario que está utilizando este software
- No se puede evitar que un software quede obsoleto por el avance de la tecnología, sin embargo, existen opciones como la migración del software a otra plataforma siempre y cuando se den las condiciones necesarias o que pueden ayudar a extender la vida del software como son la utilización de herramientas CASE.
- Las herramientas CASE es la solución más factible frente a la obsolescencia del software al facilitar la reutilización del software.
- Existe variadas herramientas CASE que son utilizadas con fines específicos para las que fueron creados, sin embargo, en el análisis de esta investigación se pudo determinar que la mejor opción de herramienta CASE para la obsolescencia del software es "Genexus", porque

es multiplataforma, ágil y hecha para crear, desarrollar y mantener aplicaciones que se adaptan a los cambios del estilo del negocio y a los cambios tecnológicos.

9

Hit and source - focused comparison, Side by Side:

Left side: As student entered the text in the submitted document.

Right side: As the text appears in the source.
