**Universidad Abierta Interamericana**

Colegio universitario

****

**Facultad de Tecnología Informática**

**Ingeniería en Sistemas Informáticos**

**Materia:** Trabajo Final de Diploma

**Localización:** Centro

**Curso:** 3ro. “B”

**Alumno****:** Settino Pablo

**Año:** 2016

Contenido

[1.1 La Complejidad del Software 2](#_Toc460775096)

[1.2 Costo-Beneficio de migrar a nuevas tecnologías. 5](#_Toc460775097)

[1.3 Ciclo del Proyecto 6](#_Toc460775098)

[1.4 Análisis de actualizar, compra, adoptar vs Desarrollar. 7](#_Toc460775099)

[1.5 Bibliografía 10](#_Toc460775100)

## [La Complejidad del Software](http://www.mitecnologico.com/Main/LaPooYLaComplejidadDelSoftware)

De manera general, podemos definir a la complejidad como la dificultad que tenemos los humanos para ordenar el conocimiento.

Así, un tema nos parecerá más complejo en cuanto sea más difícil entenderlo y explicarlo en un contexto determinado.

La complejidad en el software.

En un sistema, la complejidad es directamente proporcional al número de elementos involucrados y a la complejidad de cada elemento.

La mayoría de las aplicaciones que son especificadas, desarrolladas, mantenidas y utilizadas por una sola persona no son complejas.

Por otro lado, las aplicaciones que forman parte del software de dimensión industrial exhiben un conjunto muy rico de comportamientos, y son propensas a tener un ciclo de vida muy largo.

Estas aplicaciones son complejas y no pueden ser desarrolladas por una sola persona. Su mantenimiento y su utilización también involucran a varios individuos.

La complejidad es parte esencial de los sistemas de software de gran tamaño. Se puede manejar, pero no se puede eliminar.

La complejidad de los sistemas de software se deriva de cuatro elementos:

la complejidad del dominio del problema,

la dificultad de gestionar el proceso de desarrollo,

la flexibilidad que se puede alcanzar a través del software y

los problemas de caracterizar el comportamiento de sistemas discretos.

1.- La complejidad del dominio del problema.

Los problemas que se intentan resolver son inherentemente complejos, con una gran cantidad de requisitos que compiten entre sí.

2.- La dificultad de gestionar el proceso de desarrollo.

Los desarrolladores de software enfrentan el reto de dar a los usuarios la impresión de simplicidad, esto es reducir al mínimo la complejidad externa. Este reto les obliga a incrementar el tamaño de los sistemas, a inventar mecanismos ingeniosos, o a reutilizar diseños y código ya existentes.

3.- La flexibilidad que se puede alcanzar a través del software.

La elaboración de software es una actividad muy laboriosa porque empuja al desarrollador a construir por sí mismo prácticamente todos los bloques fundamentales sobre los que se apoyan las abstracciones de más alto nivel. Esto es propiciado, en gran medida, por la existencia de pocos estándares para el control de calidad.

4.- Los problemas de caracterizar el comportamiento de sistemas discretos.

Los comportamientos de la mayoría de los objetos se representan por sistemas analógicos en los que, a través de funciones continuas, pequeños cambios en las entradas siempre producen pequeños cambios en las salidas.

Por el contrario, puesto que el software se ejecuta en computadoras digitales, se tienen sistemas con un número finito de estados discretos. En sistemas grandes, este número puede crecer a cantidades enormes. Como no existen herramientas matemáticas para modelar el comportamiento completo de los grandes sistemas discretos, se debe aceptar la pérdida de cierto grado de confianza en cuanto a que las salidas sean correctas.

El papel de la descomposición.

Si la complejidad de los grandes sistemas de software no se puede eliminar, ¿cómo se puede manejar?

Desde tiempos remotos, el ser humano ha enfrentado la complejidad por medio de la descomposición de un gran problema original en problemas más y más pequeños.

Este precepto, que ha sido aplicado en diversos ámbitos, también se aplica en el diseño de sistemas de software complejos, descomponiéndolos hasta llegar a un nivel tal que puedan manejarse como problemas independientes.

Para manejar la descomposición es importante definir el criterio básico de descomposición eligiendo un modelo o paradigma.

Descomposición algorítmica.

En este proceso de descomposición cada módulo del sistema representa alguna parte de un proceso global, centrándose en la funcionalidad de los objetos y dejando aparte los datos que representan sus atributos.

La mayoría de los de los desarrolladores han sido adiestrados en el diseño estructurado descendente; y por eso suelen afrontar la descomposición como un proceso de descomposición algorítmica.

Descomposición orientada a objetos.

En este tipo de descomposición el problema no se resuelve centrándose en el código, sino que se modela como un conjunto de agentes autónomos, llamados objetos, que colaboran para generar la solución.

Cada objeto sabe realizar ciertas tareas que otros objetos, a través de mensajes, les piden realizar.

El criterio de descomposición orientado a objetos es mejor para ayudar a organizar la complejidad innata de los sistemas de software, porque modela mejor a los objetos del mundo real.

Uno de los **negocios más rentables** y competidos en todo el mundo sigue siendo hoy por hoy el desarrollo de software en sus diversas presentaciones.   
Sin embargo, como en todo negocio para ser exitoso y próspero es necesario conocer y dominar todo lo relacionado con el mismo.  
Si eres informático, experto en programación o tienes conocimientos de **desarrollo de software**, hoy te comparto como puedes convertir esta habilidad en un lucrativo negocio.  
  
**La industria del software**

El desarrollo de software es una industria multimillonaria que muchos visionarios han sabido explotar demasiado bien. Basta con escuchar las extraordinarias historias de Microsoft, Macromedia, Linux y otros cientos de grandes desarrolladores para imaginar los alcances que una buena estrategia de marketing puede representar para un fabricante de programas de computación.  
  
Hasta hace algunos años, los inversionistas de riesgo daban cualquier cantidad de dinero por una [compañía que ofreciera grandes ideas](http://www.1000ideasdenegocios.com/2008/02/como-las-grandes-empresas-eligen-sus.html) en materia de programación y aunque hoy en día esta oferta ha disminuido, aún siguen existiendo nichos potenciales para desarrolladores creativos que sueñan con hacerse ricos.

**El producto**

El éxito del **negocio del software** consiste en la versatilidad de su producto que permite ofrecer programas de computación para la solución de infinidad de necesidades. Desde programas de productividad, ingeniería, entretenimiento hasta sofisticadas aplicaciones científicas, los desarrolladores de software tienen en sus manos un producto que les puede representar utilidades extraordinarias, incluso de forma residual.

**La estrategia de Marketing**

El negocio del software fundamentalmente consiste en la fabricación de un programa para solucionar una necesidad específica que funcione bien y vender las licencias (copias del mismo) a la mayor cantidad posible de consumidores obteniendo así un retorno a las horas de inversión en programación.  
  
Y para que la estrategia sea exitosa se debe visualizar desde distintos puntos de vista y encontrar la alternativa que sea más viable tanto en términos de promoción como de precio.

**Desarrollar un programa que compita con los ya existentes**. Esta opción es muy poco recomendable no por falta ingenio sino porque los competidores en el ámbito del software comercial suelen ser muy poderosos y con infinidad de opciones bastante terminadas a precios muchas veces menores a $100 dólares.

**Desarrollar un programa a la medida para un nicho particular**. Para los pequeños desarrolladores, esta suele ser una opción mas viable por el hecho que les permite competir en mejores condiciones ya que sus soluciones están mucho más enfocadas.

Desde luego, el verdadero retorno de utilidades ocurre cuando logramos vender la mayor cantidad posible de licencias de nuestro software y para ello Internet es un aliado extraordinario.

## Costo-Beneficio de migrar a nuevas tecnologías.

Hoy en día no es raro encontrar sistemas heredados de veinte a treinta años, que se han actualizado y reparchado. Los bancos, las agencias gubernamentales y otras empresas dependen de estos sistemas para realizar transacciones diarias de clientes y negocios. Algunos sistemas no son capaces de mantenerse al día con las exigencias cada vez mayores, particularmente cuando se trata de grandes cantidades de transacciones y datos. Como resultado, muchas agendas de arquitectos de software y de CIO están[sopesando los riesgos de la modernización de aplicaciones heredadas](http://searchdatacenter.techtarget.com/es/cronica/Lecciones-de-actualizacion-tecnologica-y-migracion-a-la-nube).

Para mantenerse al día con los cambios tecnológicos,[algunos sistemas viejos de lento rendimiento y que consumen muchos recursos se han actualizado](http://searchdatacenter.techtarget.com/es/cronica/Cinco-claves-para-reducir-costos-de-TI), y otros están siendo considerados para la modernización de aplicaciones heredadas en nuevas plataformas. Las últimas plataformas ofrecen optimización de recursos, una mayor escalabilidad, mayor velocidad de datos y un balance más rápido de cargas de trabajo a través de internet.

Un enfoque para la migración de aplicaciones heredadas es cambiar las viejas aplicaciones en aplicaciones óptimas, que utilicen los recursos de manera eficiente, sean internas o para la nube. Si una aplicación heredada se ha modernizado con éxito en la empresa, no significa que no se pueda utilizar también como una aplicación tipo software como servicio o SaaS.

No todas las aplicaciones que se ejecutan con éxito en la empresa se ​​pueden migrar a la nube. A fin de que la migración tenga éxito, se debe cambiar el comportamiento de la aplicación para su uso a través de internet.

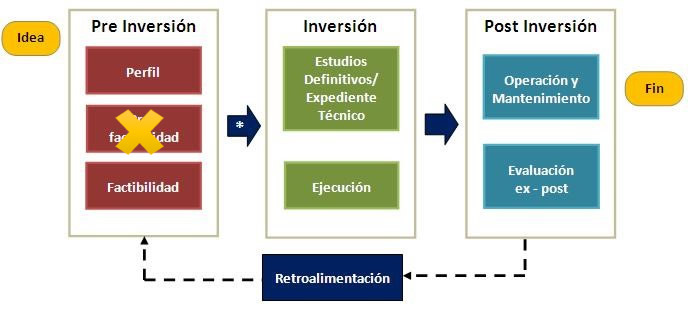
Sin embargo, cambiar el comportamiento de las aplicaciones viene con riesgos. Los peligros incluyen un rendimiento más lento; la generación de la salidas basura; pobre escalabilidad; alto consumo de recursos; fallas de aplicación frecuente; y en el peor de los escenarios posibles, fallos del sistema.

## Ciclo del Proyecto

El Ciclo de Proyecto contempla las Fases dePre inversión,InversiónyPos inversión.

 Durante la Fase de Pre inversión de un proyecto se identifica un problema determinado y luego se analizan y evalúan - en forma iterativa - alternativas de solución que permitan para encontrar la de mayor rentabilidad social.

En la Fase de Inversión se pone en marcha la ejecución proyecto conforme a los parámetros aprobados en la declaratoria de viabilidad para la alternativa seleccionada de mientras que, en la Fase de Post Inversión, el proyecto entra a operación y mantenimiento y se efectúa la evaluación ex post.



## Análisis de actualizar, compra, adoptar vs Desarrollar.

Las empresas de hoy deben usar las tecnologías de información, cambiar sus procesos para adoptar nuevas formas de hacer negocios y hacer que los procesos y sistemas de información asociados se conviertan en un recurso estratégico para generar una ventaja competitiva.

**Las TI y el software como producto**

Las tecnologías de información (TI), nombre con el que suele denominarse al software, hardware, redes de comunicaciones y otros medios que permitan transmitir y recibir información, están dando un soporte a los procesos de negocios para que sean eficientes y puedan lograr sus objetivos empresariales.

El objetivo fundamental de las TI es permitir a la empresa lograr mayor captación del mercado, mejores márgenes de rentabilidad, menores costos operativos al interior de la organización, con nuevas estructuras organizacionales que sean flexibles, más planas y que se adapten al cambio, el cual será posible si incorporamos las TI adecuadas a la empresa de hoy en un mercado en evolución.

El software en la empresa tiene que ser visto como un producto; es decir, debe ser rentable para la organización. Si invertimos en él debe ser considerado como un proyecto de desarrollo o de adquisición de software, lo cual es muy importante dado que va a apoyar los procesos del negocio y permitir crear valor para la empresa.

El desarrollo de software requiere personal, compra de equipos, las bases de datos, la participación de los usuarios, una definición de las nuevas formas de ejecución de las actividades y, como cualquier otro producto que hoy día se desarrolla en una organización, debe tener calidad. Debemos reducir al mínimo los costos posteriores por un mantenimiento indeseable como consecuencia de que no se hubieran definido adecuadamente las funcionalidades que necesitamos, esto se logra haciendo uso de metodologías formales con aseguramiento de calidad del proyecto.

Muchas empresas lo que necesitan es contar con una solución y proceder prontamente a una implantación del mismo dado que las condiciones cambiantes del mercado requieren una reacción rápida y un desarrollo puede tomar algunos meses o años. Sin embargo, no es suficiente desear un software integrado tipo paquete y acudir al mercado y comprarlo directamente como cualquier producto para decir que ya se tiene una solución. Se exige que éste sea de calidad y que satisfaga nuestros diferentes requerimientos y expectativas.

**Factores a considerar para la evaluación de software**

La evaluación de un software o paquete pre programado requiere tomar en cuenta algunos factores que son fundamentales en un proceso formal de selección, entre ellos: la funcionalidad del software; es decir, la forma como soporta los procesos y actividades con el uso de las TI; también es importante considerar los aspectos técnicos del mismo, debe tratarse de una empresa sólida y con un buen conocimiento del producto que nos está ofreciendo.

El siguiente aspecto es el económico, que debe incluir el costo del software del entrenamiento y el costo de implantación del mismo que está relacionado al tiempo de implantación, entre otros costos. Asimismo, es necesaria la participación de los usuarios de las diferentes áreas que apoyen el software y el involucramiento de la Alta Dirección de la empresa. A continuación, resumiremos todas las características que deben ser tomadas en cuenta en un proceso formal.

**Factor funcionalidad del software**

La funcionalidad del software se refiere a las nuevas características de los procesos o procedimientos que la empresa desea implantar; es decir, que en algunas ocasiones la empresa no desea seguir manteniendo las políticas ni los procedimientos obsoletos anteriores, con actividades rutinarias o con aquellas que no dan valor agregado. Por ese motivo, para realizar una adecuada evaluación de software es conveniente que la empresa realice un trabajo denominado análisis funcional de las diferentes áreas y procesos que van a ser sujetos de soporte por el software a adquirir.

Este trabajo se realiza con la participación de los usuarios involucrados en el proceso y consiste en una radiografía de los "que se hace". Es un análisis de descomposición de procesos, subprocesos y actividades. Por ejemplo, si queremos definir la funcionalidad de logística descompondríamos el área de logística en sus subprocesos gestión de compras, gestión de almacenes, gestión del transporte. El trabajo consiste en identificar algunas funcionalidades adicionales o nuevas formas de hacerlas, eliminar aquello que no da valor agregado e identificar el nivel de apoyo tecnológico.

Estos requerimientos funcionales serán necesarios para determinar el grado de adecuación del software que se adquiera. Podemos aceptar ciertos grados de variación respecto a lo que requerimos, siempre y cuando el software permita parametrizar o adaptar los programas para que puedan soportar finalmente la funcionalidad definida en esta etapa.

**Factor empresa**

La empresa proveedora del software que se evalúe debe ser sólida, con el suficiente respaldo como para poder generar confianza en el producto que se nos está ofreciendo, además de toda la experiencia necesaria y conocimiento de nuestro giro de negocio. Es conveniente contactar empresas locales e internacionales. El mercado local tiene representaciones directas de los proveedores internacionales o en todo caso son empresas locales que tienen la representación y que son de suficiente confianza como para acceder a ellas y solicitarles la presentación de una propuesta.

**Factor técnico**

Está relacionado con los aspectos tecnológicos de software, hardware o arquitectura de redes requerido por el paquete a considerar y debe estar acorde con los requerimientos de la arquitectura tecnológica de la empresa actual y proyectada para los próximos años de acuerdo a la estrategia informática de nuestro negocio. Pueden considerarse aspectos como Business Intelligence, ambiente web, redes, manejadores de bases de datos, así como lenguajes de programación, la posibilidad de modificaciones, el desarrollo de nuevas funcionalidades y plataformas, entre otros. Estos aspectos técnicos son importantes porque están relacionados con el crecimiento de la organización y evolución del uso de la tecnología considerada en la estrategia informática de la organización que debe estar alineada a la estrategia empresarial.

**Factor económico**

La evaluación de este factor considera los costos de las licencias del software, de entrenamiento, de mantenimiento que es un pago anual, así como los costos de implantación y otros costos relacionados que deben ser tomados en cuenta para tener un costo total que constituye el denominado costo de apropiación del software que será un factor de decisión para la empresa. Es conveniente realizar una evaluación financiera del proyecto.

**Consideraciones finales**

Los factores mencionados son trabajados para cada una de las empresas que está siendo evaluada a fin de tomar la decisión correcta. Hoy es posible encontrar software integrado que satisfaga los requerimientos de las empresas que soportan nuevas formas de ejecución de procesos y en sus últimas versiones trabajan con TI de vanguardia. Por ello, es conveniente una evaluación y selección donde participen las áreas directamente involucradas, como usuarios y sistemas.

Existe software desarrollado que puede apoyar los procesos de la pequeña, mediana y gran empresa. El software adquirido puede promover el rediseño de los procesos empresariales y por ello es útil tener una idea de la nueva funcionalidad requerida. Es importante realizar una cuidadosa definición del contrato teniendo en cuenta aspectos como: las funcionalidades y tecnologías ofrecidas, entregables, pagos contra entregables, penalidades, personal asignado por el proveedor, entre otros.

Seleccionar un software es una decisión que requiere de un trabajo integrado entre los proveedores y la empresa que asegure la calidad en los resultados; un aspecto es la confianza que de la empresa proveedora por los logros y éxitos del producto y por su actuación en el mercado, y por parte de la empresa contratante, esta debe asignar los recursos adecuados para la implantación.

## Bibliografía

<http://fundamentosdeprogramacion.wikispaces.com/file/view/1.3+La+POO+y+la+Complejidad+del+Software.doc>

<http://www.1000ideasdenegocios.com/2009/06/el-rentable-negocio-del-software.html>

<http://searchdatacenter.techtarget.com/es/consejo/Cinco-pasos-para-una-exitosa-modernizacion-de-apps-y-migracion-de-tecnologia-heredada>

<http://www.esan.edu.pe/conexion/bloggers/el-blog-de-eddie-morris/2011/03/evaluacion-de-software-para-la-empresa-la-decision-correcta/>