**INSTITUTO TECNOLOGICO DE SALTILLO**

**INGENIERA EN SISTEMAS COMPUTACIONALES**

**FUNDAMENTOS DE INGENIERIA DE SOFTWARE**

**TRABAJO DE INVESTIGACION: HERRAMIENTAS CASE**

**ALUMNO: CARLOS ALBERTO VANEGAS HERNANDEZ**

**PROFESORA: M.C. ARACELI CAMPOS**

**-INTRODUCCION:**

Como en cada una de las tareas que realizamos en nuestro día a día, siempre hay una herramienta que puede facilitar y hacer más eficiente esta tarea, para aumentar la calidad del producto final de este trabajo.

En cuanto a la Ingeniería del Software, las Herramientas CASE son ese apoyo que nos brindara un producto de alta calidad, bajo costo y larga vida útil en el mercado, que es lo que nos interesa en cierto momento.

En el trabajo que a continuación les presento, veremos algunas de las funciones que ayudan a este tipo de herramientas a cumplir con su tarea, como es que lo hacen, los orígenes de estas herramientas y finalmente sus ventajas y desventajas, que nos ayudaran a escoger si utilizar o no estas herramientas, y en caso de utilizarlas, cuál de estas nos conviene más, según la tarea que queramos realizar. Comencemos.

**-CONTENIDO:**

**DEFINICION DE UNA HERRAMIENTA CASE:**

Primero que nada hay que definir lo que significa CASE. La palabra CASE son siglas en el idioma ingles que significan **C**omputer **A**ssisted **S**oftware Engineering lo cual, hablando en idioma español significa, Ingeniería de Software Asistida por Computadora. Muy bien, habiendo definido lo que las siglas CASE significan, vamos a definir una herramienta CASE, condensando definiciones de los autores Roger S. Pressman y Henry David Crockett.

Como Pressman nos dice en su libro “Ingeniería de Software”, una herramienta CASE es toda herramienta que ayuda a la automatización de cualquier etapa del desarrollo de software, entre estas herramientas se integran herramientas de diagramación, para depósito de información, generadores de interfaces y de código y herramientas de administración.

Para concluir con esta definición, y agregando información que nos brinda el Instituto Nacional de Estadística e Informática, una herramienta CASE es todo programa o herramienta generalmente informática que ayuda en la automatización y mejora de cualquiera de los procesos que integran el desarrollo de software, y que ayudan a que sea más fácil, rápida y eficaz.

**CARACTERISTICAS DE UNA HERRAMIENTA CASE:**

Entre las principales características o tareas que una herramienta CASE cumple están:

* Proporcionar topologías de aplicación flexibles.
* Proporcionar aplicaciones portátiles
* Control de versión.
* Crear código compilado en el servidor.
* Trabajar con una variedad de administradores de recursos.
* Trabajar con una variedad de software intermedio.

**EVOLUCION DE LAS HERRAMIENTAS CASE:**

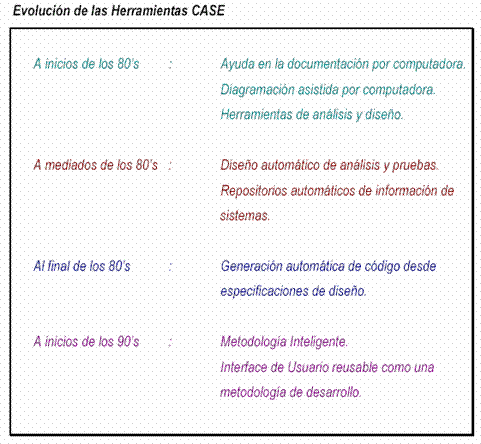
Se podría decir que la historia de este tipo de herramientas comienza en la década de los ochentas, ante la inminente crisis del software en la que el software construido, no solo no cumplía con los requerimientos ni las necesidades del cliente, sino que excede los costos y los tiempos de producción.

Fue en esta década cuando surge una nueva industria llamada CASE (Ingeniería de Software Asistido por Computadora), con la primera generación de herramientas de desarrollo de software, mismas que estaban más que nada dirigidas a automatizar la documentación y la comunicación, clave en la producción del software. La primera herramienta comercial de este tipo fue llamada Excelerator que se lanzó en el año de 1984 y trabajaba bajo una plataforma PC.

No fue hasta 1985 en que las herramientas CASE se volvieron realmente importantes en el proceso de desarrollo de software. Los proveedores prometieron que muchas áreas de la empresa serian beneficiadas con la ayuda de dichas herramientas.

Ya a finales de los ochenta y principios de los noventa incrementa la venta de estas herramientas, y empieza una etapa de asimilación de la tecnología, debito a las limitantes que la “Primera Generación” de herramientas tenía, además de las falsas promesas sobre sus capacidades. Fue por ello que a mediados de los 90 aparece una “Segunda Generación”, que incluyeron algunas herramientas que ya no llevaban el nombre de CASE, para no recordar el fracaso de la primera generación. Esta Segunda Generación supera en gran parte las limitaciones que su antecesora tenia.

Por ultimo para concluyo esta reseña histórica con un cuadro que engloba las características y la evolución de las herramientas a lo largo de los años:



**CLASIFCIACION DE LAS HERRAMIENTAS CASE:**

No existe una clasificación única de estas herramientas, puesto que se pueden clasificar de acuerdo a diferentes y diversos criterios; este tipo de herramientas se pueden clasificar atendiendo criterios como:

Las plataformas que soportan.

Las fases del ciclo de vida del desarrollo de sistemas que cubren.

La arquitectura de las aplicaciones que producen.

Su funcionalidad.

A continuación les presentare algunas de estas clasificiones, y las herramientas que entran en cada una de ellas:

Las herramientas CASE, en **función de las fases del ciclo de vida abarcadas**, se pueden agrupar de la forma siguiente:

1. Herramientas integradas, I-CASE (Integrated CASE, CASE integrado): abarcan todas las fases del ciclo de vida del desarrollo de sistemas. Son llamadas también CASE workbench.

2. Herramientas de alto nivel, U-CASE (Upper CASE - CASE superior) o front-end, orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.

3. Herramientas de bajo nivel, L-CASE (Lower CASE - CASE inferior) o back-end, dirigidas a las últimas fases del desarrollo: construcción e implantación.

4. Juegos de herramientas o Tools-Case, son el tipo más simple de herramientas CASE. Automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

Otra posible clasificación, utilizando la funcionalidad como criterio principal, es la siguiente:

1.- Herramientas de planificación de sistemas de gestión. Sirven para modernizar los requisitos de información estratégica de una organización. Proporcionan un "meta modelo" del cual se pueden obtener sistemas de información específicos. Su objetivo principal es ayudar a comprender mejor cómo se mueve la información entre las distintas unidades organizativas. Estas herramientas proporcionan una ayuda importante cuando se diseñan nuevas estrategias para los sistemas de información y cuando los métodos y sistemas actuales no satisfacen las necesidades de la organización.

2.- Herramientas de análisis y diseño. Permiten al desarrollador crear un modelo del sistema que se va a construir y también la evaluación de la validez y consistencia de este modelo. Proporcionan un grado de confianza en la representación del análisis y ayudan a eliminar errores con anticipación. Se tienen:

a) Herramientas de análisis y diseño (Modelamiento).

b) Herramientas de creación de prototipos y de simulación.

c) Herramientas para el diseño y desarrollo de interfaces. Máquinas de análisis y diseño. (Modelamiento).

3.- Herramientas de programación. Se engloban aquí los compiladores, los editores y los depuradores de los lenguajes de programación convencionales. Ejemplos de estas herramientas son:

1. Herramientas de codificación convencionales.
2. Herramientas de codificación de cuarta generación.
3. Herramientas de programación orientadas a los objetos.

4.- Herramientas de integración y prueba: Sirven de ayuda a la adquisición, medición, simulación y prueba de los equipos lógicos desarrollados. Entre las más utilizadas están:

1. Herramientas de análisis estático.
2. Herramientas de codificación de cuarta generación.
3. Herramientas de programación orientadas a los objetos.

5.- Herramientas de gestión de prototipos. Los prototipos son utilizados ampliamente en el desarrollo de aplicaciones, para la evaluación de especificaciones de un sistema de información, o para un mejor entendimiento de cómo los requisitos de un sistema de información se ajustan a los objetivos perseguidos.

6.- Herramientas de mantenimiento: La categoría de herramientas de mantenimiento se puede subdividir en:

1. Herramientas de ingeniería inversa.
2. Herramientas de reestructuración y análisis de código.
3. Herramientas de reingeniería.

7.- Herramientas de gestión de proyectos. La mayoría de las herramientas CASE de gestión de proyectos, se centran en un elemento específico de la gestión del proyecto, en lugar de proporcionar un soporte global para la actividad de gestión. Utilizando un conjunto seleccionado de las mismas se puede: realizar estimaciones de esfuerzo, coste y duración, hacer un seguimiento continuo del proyecto, estimar la productividad y la calidad, etc. Existen también herramientas que permiten al comprador del desarrollo de un sistema, hacer un seguimiento que va desde los requisitos del pliego de prescripciones técnicas inicial, hasta el trabajo de desarrollo que convierte estos requisitos en un producto final. Se incluyen dentro de las herramientas de control de proyectos las siguientes:

1. Herramientas de planificación de proyectos.
2. Herramientas de seguimiento de requisitos.
3. Herramientas de gestión y medida.

8.- Herramientas de soporte. Se engloban en esta categoría las herramientas que recogen las actividades aplicables en todo el proceso de desarrollo, como las que se relacionan a continuación:

1. Herramientas de documentación.
2. Herramientas para software de sistemas.
3. Herramientas de control de calidad.
4. Herramientas de bases de datos.

Otra clasificación, diferencia las funciones CASE en cinco grupos:

1.- Repositorio. Funcionan en torno a un repositorio central, siendo éste el núcleo fundamental que contiene todas las definiciones de objeto y sus relaciones. Los objetos pueden ser especificaciones del sistema en forma de diagramas de flujo de datos, diagramas entidad-relación, esquemas de bases de datos, diseños de pantallas, etc. El repositorio es un concepto más amplio que el de diccionario de datos y soporta a los demás grupos de funciones. No es fácil encontrar en el mercado productos Case con funcionalidades estrictamente a las de repositorio, ya que, a pesar de su innegable importancia, tienen un carácter auxiliar de los demás grupos de funciones. Cualquier sistema Case poseerá un repositorio propio o bien, trabajará sobre un repositorio suministrado por otro fabricante o vendedor.

2.- Reingeniería. Los sistemas Case permiten establecer una relación estrecha y fuertemente formalizable entre los productos generados a lo largo de distintas fases del ciclo de vida, permitiendo actuar en el sentido especificaciones-código (ingeniería "directa") y también en el contrario (ingeniería "inversa"). Ello facilita la realización de modificaciones en la fase más adecuada en cada caso y su traslado a las demás. Al conjunto de facilidades proporcionadas por la ingeniería «directa» e "inversa" se le denomina "re-ingeniería".

3.- Soporte del ciclo de vida. El ciclo de vida de una aplicación o de un sistema de información se compone de varias etapas, que van desde la planificación de su desarrollo hasta su implantación, mantenimiento y actualización. Aunque el número de fases puede ser variable en función del nivel de detalle que se adopte, pueden de modo simplificado, identificarse las siguientes:

- Planeamiento.

- Análisis y Diseño.

- Implantación (programación y pruebas).

- Mantenimiento y actualización.

Los sistemas Case pueden cubrir la totalidad de estas fases o bien especializarse en alguna(s) de ellas.

En este último caso se pueden distinguir sistemas de "alto nivel" ("Upper Case"), orientados a la autonomía y soporte de las actividades correspondientes a las dos primeras fases y, sistemas de "bajo nivel" ("Lower Case"), dirigidos hacia las dos últimas. Los sistemas de "alto nivel" pueden soportar un número más o menos amplio de metodologías de desarrollo.

Soporte de proyecto. Este tipo de funciones hace referencia al soporte de actividades que se producen durante el desarrollo, derivadas fundamentalmente del trabajo en grupos, tales como facilidades de comunicación, soporte a la creación, modificación e intercambio de documentación, herramientas personales, controles de seguridad, etc. Los sistemas Case pueden conceder a estas cuestiones una importancia variable por lo cual el soporte de proyecto constituye un factor de diferenciación.

Mejora continua de calidad. Aunque frecuentemente se asocia a los sistemas Case con la mejora de la productividad en el desarrollo de aplicaciones, debe tenerse en cuenta que una de las principales ventajas estriba también, en la mejora de la calidad de los desarrollos realizados. Determinados sistemas Case enfatizan más sobre este punto que sobre el anterior, introduciendo herramientas que permiten ejercer un control intenso de garantía de calidad del software desarrollado desde las primeras fases de su ciclo de vida.

**COSTOS ESTIMADOS DE LAS HERRAMIENTAS CASE:**

En esta sección nos basaremos en algunos ejemplo de herramientas case, como lo es la famosa Microsoft Project, para comparar las cualidades y funcionalidades que cada una de estas tiene, y comparar a la vez el costo que estas tienen ante el mercado, y de esta manera poder hacernos de una idea de cuánto cuesta tener una herramienta con buena funcionalidad.

**Microsoft Project:**

Empecemos con esta herramienta. Las cualidades que esta herramienta nos ofrece son las siguientes:

Crear Planificaciones Dinámicas

La programación controlada por el usuario le pone al mando. Reunir la flexibilidad y la facilidad de una herramienta como Excel 2010, con el poder del motor de programación de Project 2010, la

programación controlada por el usuario ofrece opciones flexibles para el desarrollo y gestión de proyectos. Las tareas programadas manuales le permiten establecer los valores de inicio, final, y los campos de Duración, déjelos en blanco, o inserte notas para futuras referencias.

El nuevo Inspector de Tareas ofrece un análisis intuitivo en la orientación. Obtenga entendimiento sobre los conflictos de recursos y cuestiones de programación de tareas mediante la herramienta Inspector, para que pueda identificar y tomar medidas para resolver los problemas.

Simplifique la planificación con tareas inactivas. Utilizando Project Professional 2010, las tareas inactivas le permitirán experimentar fácilmente con su plan de proyecto y ejecutar análisis "Que-Sí" "What If". Un simple buscador inserta o elimina una o más tareas y su efecto en el calendario de proyecto. Mejores Proyectos, Mejores

Resultados

Nueva Vista de Línea en el Tiempo y señales visuales. Con una nueva vista y completa de Línea del Tiempo, usted tendrá una visión más clara de las tareas, hitos y fases asociadas a su proyecto. Recién ampliadas paletas de colores y efectos de texto ayudarán a ver mejor todo el cronograma y el plan, por lo que rápidamente se pueden ver y compartir fechas importantes como los resultados finales. Los indicadores visuales alertan de posibles problemas.

Crear planes a nivel de resumen en el modo de tarea programada manualmente. Al principio en un proyecto usted puede utilizar tareas de alto nivel para introducir detalles limitados en el resumen de tareas programadas manualmente. Usted puede agregar tareas, recursos, y más tarde los

detalles fecha, cuando estén disponibles, y luego comparar visualmente el resultado de

un programa construido con detalles acumulados contra los planes de resumen de nivel superior. Nuevas Fuentes y formatos de controles gráficos. Utilice el color y los controles de formato con la vista Línea en el tiempo para resaltar y comunicar los aspectos clave de su proyecto.

Compartir fácilmente su línea de tiempo. Mejorada la funcionalidad de copiar y pegar que significa organizar y reorganizar su línea de tiempo, manteniendo la fidelidad de los colores, fuentes y gráficos.

Guarde sus proyectos en formatos PDF y XPS. Usted puede guardar un archivo en formato PDF o XPS, para que pueda compartir con los demás y para guardar instantáneas de la evolución de su proyecto como referencia. Tome control de los Recursos

Entre estas y otras características, Microsoft Project 2010, ofrece una buena opción como herramienta para la administración de proyectos, esta herramienta tiene un precio en el mercado actual de $14,930.

**ERwin**

PLATINUM ERwin es una herramienta de diseño de base de datos. Brinda productividad en diseño, generación, y mantenimiento de aplicaciones. Desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las características específicas de la base de datos diseñada. ERwin permite visualizar la estructura, los elementos importantes, y optimizar el diseño de la base de datos. Genera automáticamente las tablas y miles de líneas de stored procedure y triggers para los principales tipos de base de datos. Erwin hace fácil el diseño de una base de datos. Los diseñadores de bases de datos sólo apuntan y pulsan un botón para crear un gráfico del modelo E-R (Entidad-relación) de todos sus requerimientos de datos y capturar las reglas de negocio en un modelo lógico, mostrando todas las entidades, atributos, relaciones, y llaves importantes. Más que una herramienta de dibujo, ERwin automatiza el proceso de diseño de una manera inteligente. Por ejemplo, ERwin habilita la creación de un diccionario de atributos reusable, asegurando la consistencia de nombres y definiciones para su base de datos. Se mantienen las vistas de la base de datos como componentes integrados al modelo, permitiendo que los cambios en las tablas sean reflejados automáticamente en las vistas definidas. La migración automática garantiza la integridad referencial de la base de datos. Erwin establece una conexión entre una base de datos diseñada y una base de datos, permitiendo transferencia entre ambas y la aplicación de ingeniería reversa. Usando esta conexión. ERwin genera automáticamente tablas, vistas, índices, reglas de integridad referencial (llaves primarias, llaves foráneas), valores por defecto y restricciones de campos y dominios. ERwin soporta principalmente bases de datos relacionales SQL y bases de datos que incluyen Oracle, Microsoft SQL Server, Sybase, DB2, e Informix. El mismo modelo puede ser usado para generar múltiples bases de datos, o convertir una aplicación de una plataforma de base de datos a otra.

**Especificaciones Técnicas:**

Systems Operativos Compatibles: Windows NT, Windows 95, Windows 98

Requerimientos Técnicos: Mínimo 10 MB de espacio de disco duro, 16 MB RAM (32 MB RAM recomendado para modelos largos.)

**PRECIO: Gratis en varias páginas de Internet.**

**EasyCASE**

Es un producto para la generación de esquemas de base de datos e ingeniería reversa. Esta herramienta permite automatizar las fases de análisis y diseño dentro del desarrollo de una aplicación, para poder crear las aplicaciones eficazmente ² desde procesamiento de transacciones a la aplicación de bases de datos de cliente/servidor, así como sistemas de tiempo real. EasyCASE permite capturar los detalles de diseño de un sistema y comunicar las ideas gráficamente, para que sean fáciles de ver y entender. Para un diseño legítimo y modelamiento de datos, procesos y eventos. Permite crear y mantener diagramas de flujo de datos, diagramas de entidad-relación, mapas de estructura y más. Posee herramientas de corrección avanzadas que permiten revisiones generales en minutos, en lugar de horas o días. Permite re-usar diagramas o partes de diagramas para economizar el diseño de un proyecto. EasyCASE determina los tipos de esquemas según la metodología del proyecto seleccionada y notifica de errores a medida que el modelo está construyéndose. El verdadero poder de EasyCASE se encuentra en el soporte comprensivo al modelamiento de datos, procesos y eventos. Posee desde el editor de diagramas flexible y un diccionario de los datos integrado en formato dBASE, así como una extensa cantidad de reportes y análisis. Porque EasyCASE Profesional, una herramienta multi-usuario, es ideal para aquellos que necesitan compartir datos y trabajar en un proyecto con otros departamentos. El equipo completo puede acceder proyectos localizados en el servidor de la red concurrentemente. Para asegurar la seguridad de los datos, existe el diagrama y diccionario de los datos que bloquean por niveles al registro, al archivo y al proyecto, y niveles de control de acceso.

TIPOS DE DIAGRAMAS:

Data Flow Diagrams (DFDs)Transformation Schema (real-time DFDs)Structure Charts (STCs)State Transition Diagrams (STDs)Entity Relationship Diagrams (ERDs)Data Model Diagrams (DMDs)Data Structure Diagrams (DSDs)Entity Life History Diagrams (ELHs)Logical Data Structure Diagrams (LDSs)

EasyCASE Professional 4.2 o superior requiere:

EasyCASE Database Engineer; PC·s 386/486/Pentium y compatibles; Microsoft Windows 3.1 o superior, 8 MB RAM, 8 MB de espacio en disco duro; VGA o mejor color; Mouse.Requerimientos para Red: NetBIOS-compatible DOS, MS-Windows o Novell Newark.

**PRECIO: Disponible gratis en algunas páginas web.**

**Oracle Designer**

Oracle Designer es un juego de herramientas para guardar las definiciones que necesita el usuario y automatizar la construcción rápida de aplicaciones cliente/servidor flexibles y gráficas.

Integrado con Oracle Developer, Oracle Designer provee una solución para desarrollar sistemas empresariales cliente/servidor de segunda generación. Sofisticadas aplicaciones cliente/servidor pueden ser 100% generadas usando la lógica de la aplicación y el módulo de componentes reusables. Oracle Designer también habilita la captura del diseño de sistemas existentes, salvaguardando la versión actual. Todos los datos ingresados por cualquier herramienta de Oracle Designer, en cualquier fase de desarrollo, se guardan en un repositorio central, habilitando el trabajo fácil del equipo y la dirección del proyecto. Muchas metodologías diferentes para base de datos y desarrollo de aplicaciones existen actualmente. Juego de herramientas que le permiten que use la metodología de desarrollo que elija.

Desarrollo Rápido de Aplicaciones (RAD)

Ingeniería de la Información (IE)

Modelamiento de Procesos Asistido

Captura de Diseño Asistido

**PRECIO: No existe alguna referencia donde se pueda encontrar gratis.**

Como podemos ver, existe una diferencia abismal entre los precios que estas herramientas tienen, pero a la vez, hay una gran diferencia también en las características que cada una de ellas nos ofrece. De esta manera, sin embargo, podemos darnos una idea de lo que podemos hacer con un presupuesto limitado para esta área de nuestro proyecto.

**VENTAJAS Y DESVENTAJAS DEL USO DE LAS HERRAMIENTAS CASE:**

Entre los beneficios ofrecidos por la tecnología CASE se encuentran los siguientes:

**Facilidad para la revisión de aplicaciones**

La experiencia muestra que una vez que las aplicaciones se implementan, se emplean por mucho tiempo. Las herramientas CASE proporcionan un beneficio substancial para las organizaciones al facilitar la revisión de las aplicaciones. Contar con un depósito central agiliza el proceso de revisión ya que éste proporciona bases para las definiciones y estándares para los datos. Las capacidades de generación interna, si se encuentran presentes, contribuyen a modificar el sistema por medio de las especificaciones más que por los ajustes al código fuente.

**Soporte para el desarrollo de prototipos de sistemas**

En general, el desarrollo de prototipos de aplicaciones toma varias formas. En ocasiones se desarrollan diseños para pantallas y reportes con la finalidad de mostrar la organización y composición de los datos, encabezados y mensajes. Los ajustes necesarios al diseño se hacen con rapidez para alterar la presentación y las características de la interface. Sin embargo, no se prepara el código fuente, de naturaleza orientada hacia procedimientos, como una parte del prototipo.

Como disyuntiva, el desarrollo de prototipos puede producir un sistema que funcione. Las características de entrada y salida son desarrolladas junto con el código orientado hacia los procedimientos y archivos de datos.

Muchas herramientas CASE soportan las primeras etapas del desarrollo del prototipo. Muy pocas brindan apoyo durante todo el proceso de desarrollo del prototipo. Las que proporcionan la capacidad para generar código soportan de hecho todo proceso, ya que el código puede ser generado al inducir la actividad de generación después de cambiar las especificaciones o requerimientos.

**Generación de código**

Como ya se mencionó, algunas herramientas CASE tienen la capacidad de producir el código fuente. La ventaja más visible de esta característica es la disminución del tiempo necesario para preparar un programa. Sin embargo, la generación del código también asegura una estructura estándar y consistente para el programa (lo que tiene gran influencia en el mantenimiento) y disminuye la ocurrencia de varios tipos de errores, mejorando de esta manera la calidad. Las características de la generación del código permiten volver a utilizar el software y las estructuras estándares para generar dicho código, así como el cambio de una especificación modular, lo que significa volver a generar el código y los enlaces con otros módulos. Ninguna de las herramientas que existen en el presente es capaz de generar un código completo en los dominios.

**Mejora en la habilidad para satisfacer los requerimientos del usuario**

Es bien conocida la importancia de satisfacer los requerimientos del usuario, ya que esto guarda relación con el éxito del sistema. De manera similar, tener los requerimientos correctos mejora la calidad de las prácticas de desarrollo. Parece ser que las herramientas CASE disminuyen el tiempo de desarrollo, una característica que es importante para los usuarios. Las herramientas afectan la naturaleza y cantidad de interacción entre los encargados del desarrollo y el usuario. Las descripciones gráficas y los diagramas, así como los prototipos de reportes y la composición de las pantallas, contribuyen a un intercambio de ideas más efectivo.

**Soporte interactivo para el proceso de desarrollo**

La experiencia ha demostrado que el desarrollo de sistemas es un proceso interactivo. Las herramientas CASE soportan pasos interactivos al eliminar el tedio manual de dibujar diagramas, elaborar catálogos y clasificar. Como resultado de esto, se anticipa que los analistas repasarán y revisarán los detalles del sistema con mayor frecuencia y en forma más consistente.

**DEBILIDADES DE LAS HERRAMIENTAS CASE**

Las herramientas CASE tienen puntos débiles significativos, que van desde la confiabilidad en los métodos estructurados hasta su alcance limitado, los cuales amenazan con minar los beneficios potenciales descritos con anterioridad.

**Confiabilidad en los métodos estructurados**

Muchas herramientas CASE están construidas teniendo como base las metodologías del análisis estructurado y del ciclo de vida de desarrollo de sistemas. Por si sola, esta característica puede convertirse en la principal limitante ya que no todas las organizaciones emplean métodos de análisis estructurado.

Los métodos estructurados, introducidos en la década de los setenta, fueron muy elogiados por su habilidad para mejorar la exactitud de los requerimientos específicos de las aplicaciones. El nivel de conocimiento de los métodos estructurados es lato entre los profesionales de sistemas de información - de acuerdo con algunas estimaciones (Yourdon), casi el 90% de todos los analistas está familiarizado con estos métodos -. Aproximadamente la mitad de todas las organizaciones en Estados Unidos han utilizado alguna vez estos métodos. A pesar de lo anterior, si la organización o el analista no utilizan los métodos propios del análisis estructurado y tampoco desean considerar su uso, entonces el valor del CASE disminuye. En algunos casos, los analistas evitan del todo emplear herramientas CASE.

**Falta de niveles estándar para el soporte de la metodología**

Aún no aparece un conjunto "estándar" de herramientas CASE. Por tanto, debe tener precaución al seleccionar una herramienta de este tipo.

Existen dos significados para las palabras "soporte de la metodología". Una herramienta puede: 1) dar soporte a los diagramas que emplea una metodología o 2) soportarlos e imponer la metodología, sus reglas y procesos.

Las herramientas CASE que existen en el presente, tienen una de las siguientes características:

Son independientes de la metodología.

Permiten que los usuarios definan sus propias metodologías.

Soportan una metodología.

Soportan las metodologías más diseminadas.

En todas ellas existen ciertos compromisos. Las herramientas que son independientes de la metodología, no pueden fomentar el uso de las reglas y estándares de la misma. Estas herramientas quizá proporcionen los componentes de una metodología (por ejemplo: diagramas de flujos de datos, un diccionario de datos y facilidades para la descripción de procesos), pero no el marco de referencia, reglas y procedimientos que en realidad constituyen el núcleo de la metodología. Aunque se puede llevar a cabo acciones básicas para la validación de diseños y diagramas para detectar componentes faltantes, éstas son sólo funciones mecánicas. Por otra parte, esta clase de herramientas no puede proporcionar ayuda metodológica o pedir al usuario que realice tareas necesarias para la metodología que aún está sin terminar.

Estas herramientas mejoran la productividad al efectuar tareas tediosas y de documentación, aunque ellas no puedan asegurar buenos resultados. Desde el punto de vista funcional, las capacidades que brindan para garantizar la calidad son mínimas.

**Conflictos en el uso de los diagramas**

Las herramientas difieren en el uso que hacen los diagramas. Algunas son herramientas exclusivamente para gráficas, que se abocan al dibujo de diagramas para el análisis de entrada y salida de datos. Este tipo de herramientas puede restringir ya sea el proceso de desarrollo normal seguido por una organización o el estilo particular de trabajo de los analistas.

Otros vendedores de herramientas consideran los diagramas como documentación y aceptan entradas por medio de formas o lenguajes de especificación y, en ocasiones, en forma gráfica. Por tanto, se debe tener cuidado cuando se selecciona una herramienta para apoyar los métodos existentes en una organización.

**Diagramas no utilizados**

En general, los productos CASE emplean gráficas para modelar y generar informes sobre el análisis y desarrollo de sistemas. Una de las afirmaciones de los vendedores de herramientas es que las presentaciones gráficas y la documentación mejoran la comunicación entre los miembros del equipo de desarrollo, propician una calidad mayor de la entrada proporcionada por el cliente y mejoran la productividad de desarrollo de software. Sin embargo, los investigadores han encontrado que, en algunos casos, las herramientas gráficas, automatizadas o manuales, no se emplean del todo. O tal vez no se utilicen en la forma que deberían emplearse. Por otra parte, algunos analistas prefieren para algunas tareas un lenguaje estructurado o descriptivo.

Muchos profesionales de los sistemas de información no hacen uso de herramientas gráficas en el desarrollo de software; más bien las emplean para automatizar la producción de informes y documentación del sistema, como los diagramas de flujo utilizados por los programadores para documentar un programa una vez terminado.

**Función limitada**

Aunque una herramienta puede apoyar varias fases del ciclo de vida de desarrollo de sistemas o adaptarse a diferentes metodologías de desarrollo, por lo general su enfoque primario está dirigido hacia una fase o método especifico. Por ejemplo, los encargados de desarrollar un nuevo producto pueden afirmar que éste apoya todo el proceso de análisis y diseño. Sin embargo, las capacidades de comprobación y verificación de errores del producto quizá sean más rigurosas ya sea en el área de análisis o en la de diseño, pero no en ambas. Algunos productos están dirigidos hacia el diseño de bases de datos para la organización y al desarrollo de aplicaciones que giren en torno a la base de datos, omitiendo el soporte para pantallas de presentación visual, los informes sobre requerimientos o las necesidades de seguridad. Algunos productos capaces de generar el código hacen mayor hincapié en el desarrollo de prototipos como el principal método de desarrollo de sistemas de información. Muchas herramientas para la fase de desarrollo recalcan el mantenimiento y la reestructuración del código, pero ofrecen un soporte débil durante la fase de análisis para la determinación y especificación de requerimientos.

**Alcance limitado**

Aunque muchas herramientas basadas en computadoras incluyen la capacidad de verificar las especificaciones para determinar su complementes o consistencia, virtualmente no llevan a cabo ningún análisis de los requerimientos de la aplicación. Por tanto, el alcance de las actividades de desarrollo asociado con las herramientas existentes es bastante limitado.

La mayor parte de productos CASE describe (documenta) pero no analiza. De poca ayuda es proporcionar una regla de inclusión en los mejores enfoques y una regla de exclusión para los que son poco satisfactorios. No ofrecen o evalúan, soluciones potenciales para los problemas relacionados con sistemas. Y tampoco existe una garantía clara para que dos analistas que utilicen los mismos métodos aplicados a información idéntica, formulen recomendaciones igualmente aceptables.

Este fue todo el contenido de mi trabajo, pasemos a conclusión del mismo.

**CONCLUSIONES PERSONALES DEL TRABAJO:**

Después de todo lo que he leído e investigado durante este trabajo, concluyo que este tipo de herramientas ayudan bastante en el desarrollo del software, dándole mucho apoyo al ingeniero en desarrollo de software en tareas que pueden llegar a ser muy tediosas, y a veces poco fructíferas en la fase de desarrollo, pero de igual manera importantes y esenciales para el producto final.

Y aunque como en muchas de las tareas en que se emplean herramientas de software, estas mismas pudieran ser desarrolladas a mano, las herramientas case brindan mucha ayuda al darle más eficacia y velocidad en si al proceso de desarrollo que en la actualidad ha tenido la necesidad de proveer cada vez a más y más usuarios finales.

Otro punto importante en esta conclusión es la relación que tiene el precio de las herramientas case con el número de funciones y beneficios que estas nos dan. Con la información que este trabajo nos ofreció, podemos apreciar que en la mayoría de los casos el precio no es muy elevado, en muchos de ellos es gratis, y siguen teniendo un gran número de funciones y beneficios, lo cual las hace una opción muy viable para cualquier desarrollador de proyecto, ya sea que este iniciando su carrera como tal, es decir, que tenga un bajo presupuesto, o que sea un profesional en el ámbito con un presupuesto muy amplio.

Para finalizar la conclusión de este trabajo, debo decir que las herramientas case son una de las mejores opciones que un administrador de proyectos pueda utilizar, en comparación con otro tipo de herramienta, tanto en cuanto a las funcionalidades que estas nos ofrecen, como el bajo o nulo costo que estas pueden llegar a tener en el mercado. También agregar que con el uso de este tipo de herramientas se puede optimizar el trabajo que cada una de las partes que forman el trabajo aporta.

Este fue todo mi trabajo, espero que les haya gustado. Gracias.

**BIBLIOGRAFIA:**

 1.- Pressman R., "Ingeniería del Software, un Enfoque Práctico" - Tercera Edición - Editorial Mc Graw-Hill - 1993.

**CARACTERISTICAS DE LAS H. CASE:**

2.- [**http://www.slideshare.net/elsuse/herramientas-case-1402924**](http://www.slideshare.net/elsuse/herramientas-case-1402924)

**HISTORIA:**

3.-[**http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P\_terminados/HerramAuto/UNIDAD%20III/UNIDAD%203.3.htm**](http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P_terminados/HerramAuto/UNIDAD%20III/UNIDAD%203.3.htm)

**CLASIFICACION:**

4.- [**http://tpsis324.blogspot.mx/2008/09/3-clasificacion.html**](http://tpsis324.blogspot.mx/2008/09/3-clasificacion.html)

**PRECIOS DE LAS HERRAMIENTAS CASE:**

5.- [**http://es.scribd.com/doc/30987249/Herramientas-CASE-Aplicadas**](http://es.scribd.com/doc/30987249/Herramientas-CASE-Aplicadas)

**CARACTERISTICAS DE LA HERRAMIENTA PROJECT:**

6.- [**http://www.shopmania.com.mx/q-microsoft-project**](http://www.shopmania.com.mx/q-microsoft-project)

**VENTAJAS Y DESVENTAJAS DE LAS H. CASE:**

7.- [**http://docente.ucol.mx/al961223/public\_html/centro6.htm**](http://docente.ucol.mx/al961223/public_html/centro6.htm)