

INTRODUCCION

Hoy en día vivimos en un mundo de exigencias en donde la competitividad y la tecnología ha cambiado por completo los paradigmas en cuanto a los procesos manuales, la automatización a pasado a ser una de las principales herramientas utilizadas por las organizaciones, motivado al crecimiento que se ha tenido en los últimos años de la industria, donde la tecnología a jugado un papel muy importante.

En este sentido, los software de gestión han jugado un papel primordial en el crecimiento de dichas industrias, debido a los diversos sistemas personalizados creados con el objetivo de mejorar y facilitar los procedimientos en las empresas.

Es por esta razón, que las PyMES buscan hoy en día la inversión en tecnología que ayude a la toma de decisiones de una forma más rápida y efectiva siguiendo los parámetros que establecen las leyes de cada País, obteniendo software de gestión más efectivos y a más bajos costos.

La programación de este tipo de software conlleva a una serie de pasos los cuales se deben seguir para una optimización de los procesos productivos, realizados por los especialistas en la materia como creadores de las innovaciones utilizadas por las empresas para mejorar sus operaciones disminuyendo los tiempos de entrega y los pasos manuales, es por esta razón que dichos autores necesitan la protección legal para el uso de los sistemas creados por ellos, los cuales no sean el producto de replicas que ocasionen pérdidas para dichos creadores.

Por ello, es que se necesita internalizar cuáles son los pasos a seguir para la utilización adecuada de este tipo de material el cual ha beneficiado a gran parte de la población, tanto en la mejora de los procesos productivos como a los autores de dichos programas .

SOFTWARE DE GESTION

El software es una herramienta desarrollada especialmente para adecuarse a los diferentes requerimientos de las empresas. Es una solución especialmente diseñada para empresas medianas dinámicas, que buscan la eficiencia en sus procesos internos y en la gestión con terceros. Desde la administración, finanzas, comercialización, y hasta los procesos productivos, todo está contemplado.

Como ambiente de negocios, está un paso más allá de los software de los sistemas de Planificación de Recursos Empresariales (ERP), comúnmente conocidos, ya que permite la configuración dinámica de los procesos de negocios y es un programa de gestión empresarial completo (facturación, almacén, ingresos, gastos, beneficios, informes...) que, además, permite llevar un control de los costes de mano de obra y productos usados en un trabajo.

Este va dirigido a empresas medianas y grandes con necesidades de alta competitividad, venta de servicios, distribución comercial y/o producción industrial, debido a que es en definitiva, un programa con el que, de una forma muy fácil e intuitiva, se puede llevar la gestión administrativa de una empresa; es probablemente la aplicación para la gestión integral de las empresas más cómoda, intuitiva y potente del mercado.

CLASIFICACIÓN DE LOS SOFTWARE DE GESTIÓN

- ✓ Gestión Dinámica (Información directiva para toma de decisiones).
Puede presentarse en forma analítica o resumida el origen y

aplicación de fondos. Es la mirada financiera general de ingresos y egresos de fondos a la empresa, también puede visualizarse en forma analítica o resumida, general o mensualizada. El Cuadro de resultados económicos permite entender los resultados por centros de costos o grupos de centros de costo, y como tal especializa la información de balance, detallando por cuenta y grupo de cuentas la información.

- ✓ Gestión de Ventas (Clientes, Pedidos, Facturación, Cobranzas, Logística). El objetivo de este módulo es ordenar la administración de ventas, mejorando la relación con los clientes y brindando mayor y mejor información comercial. A su vez contar con información de los clientes para tomar decisiones y aprovechar las oportunidades comerciales que se presenten.
- ✓ Gestión de Compras (Proveedores, Cuentas a Pagar, Requisiciones, Cumplimientos). El objetivo de este módulo es ordenar la administración de las compras locales y de importación, mejorando el control de proveedores en lo que respecta a entregas y pagos, brindando mayor y mejor información para una buena relación comercial con los proveedores y a su vez contar con información clave de los cumplimientos de las Ordenes de Compra. Contar con información integrada para tomar decisiones y aprovechar las oportunidades comerciales que se presenten.
- ✓ Gestión de Stocks y Logística (Terminados, S Elaborados, M Primas, Bienes de Capital). El objetivo de este módulo es organizar el movimiento de stocks y la administración de inventarios en lo que respecta a unificar criterios e información de absolutamente todos los materiales involucrados en el negocio, artículos, materias primas, productos en proceso, productos terminados, etc. Brindar información de costos e información clave para tomar decisiones de logística y abastecimientos.
- ✓ Gestión de Finanzas (Planificación, Control Presupuestario, Cash

Flow). El objetivo de este módulo es el de integrar toda la información de los otros módulos, potenciando las posibilidades de visualizar reportes e informes para tomar decisiones financieras, respetando la filosofía de todo el sistema, siempre en forma automática se produce el registro contable permitiendo visualizar también, informes contables y financieros.

- ✓ Gestión de Contabilidad (Plan de cuentas, Impuestos, Bienes de uso, entre otros.). El objetivo de este módulo es integrar el plan de cuentas en donde se apoya toda la registración contable. Una vez definidos los circuitos y comprobantes, el sistema administra y graba en la BASE DE DATOS toda la información con el formato que se haya definido. Aquí es importante la participación de los responsables de la información contable o asesores externos que tengan relación directa con las salidas o reportes para cubrir los aspectos legales. La metodología de presupuestación debe estar definida antes de comenzar con la implementación.
- ✓ Gestión de Producción Industrial (Procesos, Órdenes, Plan de Producción, Costos.). El objetivo de este módulo es poder optimizar la administración de la producción, integrando los procesos productivos con los módulos de compras e inventarios. De esta manera se logra evitar la duplicidad de carga de información y la coherencia e integración de todos los recursos que intervienen en los procesos productivos.
- ✓ Gestión de Recursos Humanos (Liquidación de sueldos y gestión del personal). Es el módulo que permite principalmente, procesar las liquidaciones de sueldos y jornales generando la contabilización correspondiente. Pero además ofrece otras funcionalidades adicionales para la gestión de RRHH como llevar el registro de cada uno de los empleados pertenecientes a la empresa, sus datos personales y todo tipo de información relacionada, tal como: familiares y sus datos relativos, registro de licencias, fotos del empleado y familiares, información fiscal. Para

el procesamiento de las liquidaciones se cuenta con la posibilidad de definir conceptos de cálculo y sus fórmulas de cálculo correspondientes, las que pueden ser particularizadas por el usuario de acuerdo a la necesidad de la empresa, convenios, etc. Además se puede llevar todos los reportes relativos a los módulos, como por ejemplo los correspondientes a: credenciales, recibo de sueldos, libro de ley, informe de novedades, pueden ser particularizados por el usuario para ajustarlos a la necesidad de la empresa.

DIFERENCIA ENTRE SOFTWARE DE GESTIÓN Y PLANIFICACIÓN DE RECURSOS EMPRESARIALES ERP

La principal diferencia estriba en la definición. Un ERP es una aplicación que integra en un único sistema todos los procesos de negocio de una empresa. Adicionalmente se pretende que todos los datos estén disponibles todo el tiempo para todo el mundo en la empresa (obviando por el momento permisos sobre disponibilidad, etc) de una manera centralizada.

El software de gestión aplica programas basados en múltiples aplicaciones (denominados comúnmente suites) independientes o modulares que duplican la información (aún cuando la enlacen automáticamente) o no la centralizan en una única base de datos. También utiliza aquellos programas que se basan en sistemas de base de datos de ficheros independientes (sin motor de base de datos). Los mismos se adaptan totalmente a la organización de la empresa asegurando toda su administración comercial, incluso en aquellas empresas con procesos de venta complejos y gestión de reaprovisionamiento multi-deposito. Los procesos de gestión aceleran el tratamiento de los pedidos y facturación de compras y ventas, controlando la correcta aplicación de la política comercial y el riesgo de clientes. Su enlace con Contabilidad optimiza el circuito administrativo,

facilitando el puente contable de ventas, compras, cobros, pagos y parámetros contables.

INTEGRACIÓN DE SOFTWARE

Una de las fases del ciclo de vida del software es la de integración. Es imprescindible poder integrar los desarrollos de software en forma de productos y soluciones para que puedan ponerse en uso. Esto exige manejarse en varias disciplinas, no perder de vista el punto de vista del usuario, definición y aplicación de procedimientos con rigor, llevar a cabo gestiones de configuración. Adaptando algunas tareas que son realizadas con la creación de paquetes, que proporcionan una configuración e instalación personalizada de software (estándar, personalizada o de propiedad) en nuestro entorno a las necesidades.

Los servicios de integración de software

- Cargas de imágenes de software del cliente (Windows 2000 / Windows XP / Windows 9X / NT4 y Linux)
- Cargas de sistema operativo y sistema operativo de red, incluidos parámetros de sistema operativo personalizados
- Cargas de aplicaciones de propiedad y de otros proveedores, incluidos software de copia de seguridad y comunicación de administración de servidores
- Partición de disco duro
- Agrupación en clúster personalizada
- Personalización del sistema, incluida la personalización de parámetros individuales para que cada uno tenga una dirección IP, un nombre de ordenador y un identificador de sistema exclusivos

APLICACIÓN DE SOFTWARE EN LAS PyME

Una empresa es un organismo. Vive, respira, muta, crece. Durante la vida de una empresa se notan épocas de crecimiento y reorganización que en forma cíclica marcan su pulso. Generalmente, cuando una empresa comienza su crecimiento se presenta todo un análisis por parte de la gerencia general en el cual se redunda en explicaciones de visión, misión y valores a fin de poder poner un rumbo a ese organismo que o bien está pasando un buen momento y no lo aprovecha del todo o está mal y hay que levantarlo con algún antibiótico. En estos momentos, una decisión inteligente es la de pensar cuales son los recursos con los que se cuenta tanto humanos como de producción. Pero hay algo que muchas organizaciones se olvidan. Su capital tecnológico, pensar en cualquier PyME sin recursos de tecnología (llamase comunicación, equipamiento informático, capacidad humana para mantenerla, etc) es sentenciarla a muerte, ya que las actividades requieren un uso exhaustivo de estos recursos.

El avance tecnológico del cual somos testigos ha hecho posible que el mercado de las computadoras tenga un desarrollo importante y por lo tanto nos ha facilitado el acceso a hardware barato y con buenas características de desempeño para las Pequeñas y Medianas Empresas (PyME). Pero al invertir en equipo de cómputo para una empresa olvidamos un paso muy importante ¿Qué software vamos a usar? de toda la amplia gama de opciones ¿cuál es el apropiado para mi empresa?. El gran dilema de comprar un sistema de información comienza.

El mercado del software para la PyME's ha crecido increíblemente, además el desarrollo natural de las empresas empieza a aumentar el número de sus operaciones a un grado que el control se hace mas difícil, la competencia es cada vez mas agresiva y las dificultades por sobrevivir a las condiciones económicas hacen que la información necesaria para la toma de decisiones sea cada vez más importante. Es aquí donde complementar el hardware con un programa orientado a la administración adquiere relevancia en los procesos empresariales.

SOFTWARE DE NOMINAS

Hoy en día los software para nominas son una poderosa herramienta que ha sido creada y pensada para que nuestros en las empresas ya que pueden elaborar de forma rápida, fácil y segura las relaciones de pago al personal, y así mismo administrar todas las obligaciones contractuales o de ley que siempre está vigente con el marco legal y exigencias de la misma, en estos tiempos es tan importante para cualquier empresa.

Las características de este tipo software son múltiples; son capaces de manejar múltiples empresas, sin límite en el número de trabajadores en cada una de ellas, genera la Declaración Trimestral y la Declaración de Utilidades acorde con lo establecido en la Ley Orgánica del Trabajo (L.O.T.), generan planillas e Informes vigentes para el cumplimiento de los deberes impuestos por la LOPCYMAT, entre otras ventajas. Como el ahorro de tiempo y esfuerzo gracias al manejo sencillo y práctico de los Software, el mismo siempre estará vigente con el marco legal y exigencias de la ley permitiendo la creación de Asignaciones y Deducciones básicas de ley precargadas que le permiten al usuario un rápido inicio en el sistema. La creación de la Nómina de Tickets de alimentación, al igual que permitir generar la información electrónica para el envío de: Nómina, Fideicomiso y aportes del Ahorro Habitacional al Banco de su preferencia y siempre al día con los cambios que los bancos requieran para efectuar estos pagos electrónicamente.

SOFTWARE DE CONTABILIDAD

Los programas Contables han sido diseñados tomando en cuenta la importancia que tiene la información contable para las empresas, es por ello que estos brindan soluciones programadas de acuerdo con el

criterio de profesionales contables, siendo estos creados de una forma fácil y práctica para su manejo, tanto en la carga, en la utilidad, en los informes como así también en el tiempo de implementación. Estos Sistemas buscan que la información sea ordenada en pocos pasos y la preparada como desee para el análisis contable como resultado de sus operaciones, valiéndose de recursos como formularios, reportes, libros etc. y que presentados a la gerencia le permitirán a la misma tomar decisiones financieras.

Es por ello que este tipo de software busca consolidar el registro de todas las operaciones financieras de la empresa siguiendo un esquema innovador y práctico implementando nueva alternativa que tienen como finalidad satisfacer las diferentes gamas de usuarios del mercado venezolano.

SOFTWARE DE BANCO

Estos software cuentan con un acceso remoto seguro a las aplicaciones, tanto a los niveles gerenciales, sucursales y oficinas remotas, así como a los múltiples canales de atención al cliente, los cuales constituyen los principales desafíos para las organizaciones en el actual entorno de negocios. Para acudir en auxilio de estas necesidades surge la tecnología "Acceso Universal" es especialmente recomendable para bancos y compañías del sector financiero, ya que en su mayoría, dichas organizaciones deben convivir con múltiples redes, plataformas, sistemas operativos y aplicaciones, en un entorno donde la información crece y se requiere en cualquier lugar y momento. De esta forma, un ejecutivo, un gerente puede realizar desde un cualquier lugar conferencias en línea con sus pares compartiendo estadísticas o balances del negocio, accediendo a las aplicaciones de software del banco o la compañía aseguradora mientras realiza una visita en terreno, o incluso un agente de call center ubicado en una oficina remota podrá disponer de los datos actualizados en tiempo real sobre

el status de un cliente.

SOFTWARE LIBRE Y LAS PyME

El software libre es un programa operativo de computadoras que, una vez adquirido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. La empresa que internacionalmente ha promovido el uso de esta tecnología es la empresa Linux. al usar software libre la persona puede adaptarlo a sus propios requerimientos sin necesidad de pagar costosas licencias o permisos de la corporación fabricante del programa para operarlo

MODALIDADES DE LAS LICENCIAS

Licencia: contrato entre el desarrollador de un software sometido a propiedad intelectual y a derechos de autor y el usuario, en el cual se definen con precisión los derechos y deberes de ambas partes. Es el desarrollador, o aquél a quien éste haya cedido los derechos de explotación, quien elige la licencia según la cual distribuye el software.

Patente: conjunto de derechos exclusivos garantizados por un gobierno o autoridad al inventor de un nuevo producto (material o inmaterial) susceptible de ser explotado industrialmente para el bien del solicitante por un periodo de tiempo limitado .

Derecho de autor o copyright: forma de protección proporcionada por las leyes vigentes en la mayoría de los países para los autores de obras originales incluyendo obras literarias, dramáticas, musicales, artísticas e intelectuales, tanto publicadas como pendientes de publicar.

Software libre : proporciona la libertad de

- Ejecutar el programa, para cualquier propósito;
- Estudiar el funcionamiento del programa, y adaptarlo a sus necesidades;
- Redistribuir copias;
- Mejorar el programa, y poner sus mejoras a disposición del público, para beneficio de toda la comunidad.

Software de fuente abierta . sus términos de distribución cumplen los criterios de

- Distribución libre;
- Inclusión del código fuente;
- Permitir modificaciones y trabajos derivados en las mismas condiciones que el software original;
- Integridad del código fuente del autor, pudiendo requerir que los trabajos derivados tengan distinto nombre o versión;
- No discriminación a personas o grupos;
- Sin uso restringido a campo de actividad;
- Los derechos otorgados a un programa serán válidos para todo el software redistribuido sin imponer condiciones complementarias;
- La licencia no debe ser específica para un producto determinado;
- La licencia no debe poner restricciones a otro producto que se distribuya junto con el software licenciado;
- La licencia debe ser tecnológicamente neutral.

Estándar abierto: según Bruce Perens, el basado en los principios de

- Disponibilidad;
- Maximizar las opciones del usuario final;
- Sin tasas sobre la implementación;
- Sin discriminación de implementador;

- Permiso de extensión o restricción;
- Evitar prácticas predatorias por fabricantes dominantes.

Software de dominio público: aquél que no está protegido con *copyright* .

Software con *copyleft*: software libre cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional cuando lo redistribuyen o modifican, o sea, la versión modificada debe ser también libre.

Software semilibre: aquél que no es libre, pero viene con autorización de usar, copiar, distribuir y modificar para particulares sin fines de lucro.

Freeware: se usa comúnmente para programas que permiten la redistribución pero no la modificación (y su código fuente no está disponible),

Shareware: software con autorización de redistribuir copias, pero debe pagarse cargo por licencia de uso continuado.

Software privativo: aquél cuyo uso, redistribución o modificación están prohibidos o necesitan una autorización.

Software comercial: el desarrollado por una empresa que pretende ganar dinero por su uso.

PROTECCION LEGAL DEL SOFTWARE

La aparición de la informática en la sociedad ha generado múltiples relaciones entre ella y el derecho. Los vertiginosos avances tecnológicos en esta materia, con el uso masivo de los computadores y la comunicación

interactiva, presentan un nuevo y original desafío al derecho. La informática ha provocado la aparición de reglas de derecho que le son particulares, dispersas algunas, inadecuadas otras, algunas veces contradictorias, que han desafiado a las tradicionales instituciones del derecho. Programas de computación, bases de datos, semi-conductores, entre otros, se han convertido en objetos de derecho.

La informática como producto de la inteligencia del hombre reclama la protección que el derecho concede a los bienes intelectuales. En esta materia, todavía los productores de programas de computación se debaten entre las reglas del derecho de autor y las de la propiedad industrial, en general las consideran inadecuadas para proteger sus derechos en la época digital. Las reglas que convienen a los productores no convienen a las empresas de servicios on-line o de información por computadora vía telefónica.

La informática como bien económico exige al derecho el privilegio de un acceso controlado al mercado para protegerse de la piratería y del *counterfeiting*. En los Estados Unidos según la BSA (Asociación de Productores de Software), el 35 por ciento de los programas en uso son ilegales. Según INVESOF T de Venezuela, por cada programa que se vende legalmente en el país, existen ocho copias ilegales.

En Venezuela la propiedad intelectual se protege por la vía del derecho de autor, que incluye el derecho sobre las obras de ingenio de carácter creador, ya sea de índole literaria, científica o artística, cualquiera que sea su género, forma de expresión, mérito o destino.

La legislación venezolana sobre derecho de autor se encuentra sustentada en el reconocimiento constitucional de ese derecho como uno de los Derechos Culturales y Educativos consagrados en la Constitución de la República Bolivariana de Venezuela, expresado en su artículo 9. Para lograr el efectivo desarrollo y seguimiento de esa norma constitucional, existe vigente la Ley sobre Derechos de Autor, en la que se reconocen los derechos propios del autor, así como los derechos conexos.

Igualmente forman parte de la legislación venezolana sobre el

Derecho de Autor, los convenios internacionales que han sido ratificados por Venezuela, Convenio de Berna para la protección de obras literarias y artísticas; Convenio universal sobre Derecho de Autor; Acuerdo sobre la protección de los derechos de propiedad intelectual relativo a los bienes en el comercio.

En tal sentido, los programas de computación (Software) se protegen en los mismos términos que las obras literarias. Se permite hacer una reproducción cuando sea destinada exclusivamente como resguardo.

La ley en Venezuela prohíbe la reproducción, distribución y comercialización de copias ilícitas de software, así como el uso no autorizado de programas de computación.

ALGORITMO

(Del latín, *dixit algorithmus* y éste del matemático persa al-Jwarizmi) Un algoritmo es un sistema por el cual se llega a una o varias soluciones, teniendo en cuenta que debe ser definido, finito y eficiente. Por eficiente se entiende que cada paso a seguir tiene un orden; finito implica que tiene un determinado número de pasos, o sea, que tiene un fin; y definido, que si se sigue el mismo proceso más de una vez se llega siempre al mismo resultado

El término "algoritmo" no está exclusivamente relacionado con la matemática, las ciencias de la computación o la informática. En la vida cotidiana se emplean algoritmos en multitud de ocasiones para resolver diversos problemas. Algunos ejemplos son el uso de una lavadora (se siguen las instrucciones), pero no la preparación de una comida (porque no están perfectamente definidos los pasos) o el mismo lenguaje humano que "transforma" pensamientos en sonidos y hace que otro humano pueda entender. También existen ejemplos de índole matemática, como el algoritmo de la división para calcular el cociente de dos números, el algoritmo de Euclides para calcular el máximo común divisor de dos enteros positivos, o el método de Gauss para

resolver Sistema lineal de ecuaciones.

Lenguaje de programación

Es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

Un lenguaje de programación es una notación para escribir programas, a través de los cuales podemos comunicarnos con el hardware y dar así las ordenes adecuadas para la realización de un determinado proceso. Un lenguaje esta definido por una gramática o conjunto de reglas que se aplican a un alfabeto constituido por el conjunto de símbolos utilizados. Los distintos niveles de programación existentes nos permiten acceder al hardware, de tal forma que según utilicemos un nivel u otro, así tendremos que utilizar un determinado lenguaje ligado a sus correspondientes traductores.

CLASIFICACION DE LOS LENGUAJES DE PROGRAMACIÓN

1.- LENGUAJE MÁQUINA

El lenguaje máquina es el único que entiende directamente la computadora, ya que esta escrito en lenguajes directamente inteligibles por la máquina (computadora), utiliza el alfabeto binario, que consta de los dos únicos símbolos 0 y 1, denominados bits (abreviatura inglesa de dígitos binarios). Sus instrucciones son cadenas binarias (cadenas o series de caracteres de dígitos 0 y 1) que especifican una operación y, las posiciones (dirección) de memoria implicadas en la operación se denominan instrucciones de máquina o código maquina. Fue el primer lenguaje utilizado en la programación de computadoras, pero dejo de utilizarse por su dificultad y complicación, siendo sustituido por otros

lenguajes más fáciles de aprender y utilizar, que además reducen la posibilidad de cometer errores. El lenguaje máquina es el conocido código binario. Generalmente, en la codificación de los programas se empleaba el sistema hexadecimal para simplificar el trabajo de escritura. Todas las instrucciones preparadas en cualquier lenguaje máquina tienen por lo menos dos partes. La primera es el comando u operación, que dice a las computadoras cual es la función que va a realizar. Todas las computadoras tienen un código de operación para cada una de las funciones. La segunda parte de la instrucción es el operando, que indica a la computadora donde hallar o almacenar los datos y otras instrucciones que se van a manipular, el número de operándos de una instrucción varía en distintas computadoras.

VENTAJAS DEL LENGUAJE MÁQUINA

Posibilidad de cargar (transferir un programa a la memoria) sin necesidad de traducción posterior, lo que supone una velocidad de ejecución superior a cualquier otro lenguaje de programación.

DESVENTAJAS DEL LENGUAJE MÁQUINA

- ✓ Dificultad y lentitud en la codificación.
- ✓ Poca fiabilidad.
- ✓ Gran dificultad para verificar y poner a punto los programas
- ✓ Los programas solo son ejecutables en el mismo procesador (CPU). En la actualidad, las desventajas superan a las ventajas, lo que hace prácticamente no recomendables a los lenguajes máquina.

2.- LENGUAJES DE BAJO NIVEL

Son más fáciles de utilizar que los lenguajes máquina, pero al igual

que ellos, dependen de la máquina en particular. El lenguaje de bajo nivel por excelencia es el ensamblador. El lenguaje ensamblador es el primer intento de sustituir el lenguaje máquina por otro más similar a los utilizados por las personas. Este intenta desflexibilizar la representación de los diferentes campos. Esa flexibilidad se consigue no escribiendo los campos en binario y aproximando la escritura al lenguaje.

A principios de la década de los 50 y con el fin de facilitar la labor de los programadores, se desarrollaron códigos mnemotécnicos para las operaciones y direcciones simbólicas. Los códigos mnemotécnicos son los símbolos alfabéticos del lenguaje máquina. La computadora sigue utilizando el lenguaje máquina para procesar los datos, pero los programas ensambladores traducen antes los símbolos de código de operación especificados a sus equivalentes en el lenguaje máquina.

En la actualidad los programadores no asignan números de dirección reales a los datos simbólicos, simplemente especifican donde quieren que se coloque la primera localidad del programa y el programa ensamblador se encarga de lo demás, asigna localidades tanto para las instrucciones como los datos. Estos programas de ensamble o ensambladores también permiten a la computadora convertir las instrucciones en lenguaje ensamblador del programador en su propio código máquina. Un programa de instrucciones escrito en lenguaje ensamblador por un programador se llama programa fuente. Después de que el ensamblador convierte el programa fuente en código máquina a este se le denomina programa objeto. Para los programadores es más fácil escribir instrucciones en un lenguaje ensamblador que en código de lenguaje máquina pero es posible que se requieran dos corridas de computadora antes de que se puedan utilizar las instrucciones del programa fuente para producir las salidas deseadas.

El lenguaje de bajo nivel es el lenguaje de programación que el ordenador puede entender a la hora de ejecutar programas, lo que

aumenta su velocidad de ejecución, pues no necesita un intérprete que traduzca cada línea de instrucciones.

Visto a muy bajo nivel, los microprocesadores procesan exclusivamente señales electrónicas binarias. Dar una instrucción a un microprocesador supone en realidad enviar series de unos y ceros espaciadas en el tiempo de una forma determinada. Esta secuencia de señales se denomina código máquina. El código representa normalmente datos y números e instrucciones para manipularlos. Un modo más fácil de comprender el código máquina es dando a cada instrucción un mnemónico, como por ejemplo STORE, ADD o JUMP. Esta abstracción da como resultado el ensamblador, un lenguaje de muy bajo nivel que es específico de cada microprocesador.

VENTAJAS DEL LENGUAJE ENSAMBLADOR FRENTE AL LENGUAJE MÁQUINA

- ✓ Mayor facilidad de codificación y, en general, su velocidad de calculo, ahorran tiempo y requieren menos atención a detalles.
- ✓ Se incurren en menos errores y los que se cometen son más fáciles de localizar.
- ✓ Tanto el lenguaje maquina como el ensamblador gozan de la ventaja de mínima ocupación de memoria y mínimo tiempo de ejecución en comparación con el resultado de la compilación del programa equivalente escrito en otros lenguajes.
- ✓ Los programas en lenguaje ensamblador son más fáciles de modificar que los programas en lenguaje máquina.

DESVENTAJAS DEL LENGUAJE ENSAMBLADOR

- ✓ Dependencia total de la maquina lo que impide la transportabilidad de los programas (posibilidad de ejecutar un programa en diferentes máquinas).

- ✓ El lenguaje ensamblador del PC es distinto del lenguaje ensamblador del Apple Machintosh.
- ✓ La formación de los programadores es más compleja que la correspondiente a los programadores de alto nivel, ya que exige no solo las técnicas de programación, sino también el conocimiento del interior de la máquina
- ✓ El programador ha de conocer perfectamente el hardware del equipo, ya que maneja directamente las posiciones de memoria, registros del procesador y demás elementos físicos.
- ✓ Todas las instrucciones son elementales, es decir, en el programa se deben describir con el máximo detalle todas las operaciones que se han de efectuar en la máquina para la realización de cualquier proceso.

Los lenguajes ensamblador tienen sus aplicaciones muy reducidas, se centran básicamente en aplicaciones de tiempo real, control de procesos y de dispositivos electrónicos.

Vistos a muy bajo nivel, los microprocesadores procesan exclusivamente señales electrónicas binarias. Dar una instrucción a un microprocesador supone en realidad enviar series de unos y ceros espaciadas en el tiempo de una forma determinada. Esta secuencia de señales se denomina código máquina. El código representa normalmente datos y números e instrucciones para manipularlos. Un modo más fácil de comprender el código máquina es dando a cada instrucción un mnemónico, como por ejemplo STORE, ADD o JUMP. Esta abstracción da como resultado el ensamblador, un lenguaje de muy bajo nivel que es específico de cada microprocesador.

Los lenguajes de bajo nivel permiten crear programas muy rápidos, pero que son a menudo difíciles de aprender. Más importante es el hecho de que los programas escritos en un bajo nivel sean altamente específicos de cada procesador. Si se lleva el programa a otra máquina se debe reescribir el programa desde el principio.

Este intenta deflexibilizar la representación de los diferentes

campos, esa flexibilidad se consigue no escribiendo los campos en binario y aproximando la escritura al lenguaje.

3.- LENGUAJES DE ALTO NIVEL

Estos lenguajes son los más utilizados por los programadores. Están diseñados para que las personas escriban y entiendan los programas de un modo mucho más fácil que los lenguajes máquina y ensamblador. Un programa escrito en lenguaje de alto nivel es independiente de la máquina (las instrucciones no dependen del diseño del hardware o de una computadora en particular), por lo que estos programas son portables o transportables.

Los programas escritos en lenguaje de alto nivel pueden ser ejecutados con poca o ninguna modificación en diferentes tipos de computadoras. Son lenguajes de programación en los que las instrucciones enviadas para que el ordenador ejecute ciertas órdenes son similares al lenguaje humano. Dado que el ordenador no es capaz de reconocer estas órdenes, es necesario el uso de un intérprete que traduzca el lenguaje de alto nivel a un lenguaje de bajo nivel que el sistema pueda entender.

Por lo general se piensa que los ordenadores son máquinas que realizan tareas de cálculos o procesamiento de texto. La descripción anterior es sólo una forma muy esquemática de ver una computadora. Hay un alto nivel de abstracción entre lo que se pide a la computadora y lo que realmente comprende. Existe también una relación compleja entre los lenguajes de alto nivel y el código máquina.

Los lenguajes de alto nivel son normalmente fáciles de aprender porque están formados por elementos de lenguajes naturales, como el inglés. En BASIC, el lenguaje de alto nivel más conocido, los comandos como "IF CONTADOR=10 THEN STOP" pueden utilizarse para pedir a la computadora que pare si CONTADOR es igual a diez. Por desgracia para muchas personas esta forma de trabajar es un poco frustrante,

dado que a pesar de que las computadoras parecen comprender un lenguaje natural, lo hacen en realidad de una forma rígida y sistemática.

Los lenguajes de alto nivel, también denominados lenguajes evolucionados, surgen con posterioridad a los anteriores (lenguaje máquina, lenguajes de bajo nivel o ensamblador) con los siguientes objetivos, entre otros:

Lograr independencia de la máquina, pudiendo utilizar un mismo programa en diferentes equipos con la única condición de disponer de un programa traductor o compilador, que es suministrado por el fabricante, para obtener el programa ejecutable en lenguaje binario de la máquina que se trate. Además, no se necesita conocer el hardware específico de dicha máquina. Aproximarse al lenguaje natural, para que el programa se pueda escribir y leer de una forma más sencilla, eliminando muchas de las posibilidades de cometer errores que se daban en el lenguaje máquina, ya que se utilizan palabras (en inglés) en lugar de cadenas de símbolos sin ningún significado aparente.

Incluir rutinas de uso frecuente, como las de entrada / salida, funciones matemáticas, manejo de tablas, etc., que figuran en una especie de librería del lenguaje, de manera que se puedan utilizar siempre que se quiera sin necesidad de programarlas cada vez.

VENTAJAS DE LOS LENGUAJES DE ALTO NIVEL:

- ✓ El tiempo de formación de los programadores es relativamente corto comparado con otros lenguajes.
- ✓ La escritura de programas se basa en reglas sintácticas similares a los lenguajes humanos, nombres de las instrucciones tales como READ, WRITE, PRINT, OPEN, etc.
- ✓ Las modificaciones y puestas a punto de los programas son más fáciles.
- ✓ Reducción del costo de los programas.
- ✓ Transportabilidad.

- ✓ Permiten tener una mejor documentación.
- ✓ Son más fáciles de mantener.

DESVENTAJAS DE LOS LENGUAJES DE ALTO NIVEL

- ✓ Incremento del tiempo de puesta a punto al necesitarse diferentes traducciones del programa fuente para conseguir el programa definitivo.
- ✓ No se aprovechan los recursos internos de la máquina que se explotan mucho mejor en lenguajes máquina y ensambladores.
- ✓ Aumento de la ocupación de memoria.
- ✓ El tiempo de ejecución de los programas es mucho mayor.

Se puede decir que el principal problema que presentan los lenguajes de alto nivel es la gran cantidad de ellos que existen actualmente en uso, además de las diferentes versiones o dialectos que se han desarrollado de algunos de ellos. Es difícil establecer una clasificación general de los mismos, ya que en cualquiera que se realice habrá lenguajes que pertenezcan a más de uno de los grupos establecidos. Una clasificación muy extendida, atendiendo a la forma de trabajar de los programas y a la filosofía con que fueron concebidos, es la siguiente:

- ✓ **Lenguajes imperativos.** Utilizan instrucciones como unidad de trabajo de los programas (Cobol, Pascal, C, Ada).
- ✓ **Lenguajes declarativos.** Los programas se construyen mediante descripciones de funciones o expresiones lógicas (Lisp, Prolog).
- ✓ **Lenguajes orientados a objetos.** El diseño de los programas se basa más en los datos y su estructura. La unidad de proceso es el objeto y en él se incluyen los datos (variables) y las operaciones que actúan sobre ellos (Smalltalk, C++).
- ✓ **Lenguajes orientados al problema.** Diseñados para problemas específicos, principalmente de gestión, suelen ser generadores de

aplicaciones.

- ✓ **Lenguajes naturales.** Están desarrollándose nuevos lenguajes con el principal objetivo de aproximar el diseño y construcción de programas al lenguaje de las personas.

Otra clasificación que se puede hacer es la de atendiendo al desarrollo de los lenguajes desde la aparición de las computadoras, que sigue un cierto paralelismo con las generaciones establecidas en la evolución de las mismas:

- ✓ **Primera generación.** Lenguajes maquina y ensambladores.
- ✓ **Segunda generación.** Primeros lenguajes de alto nivel imperativo (FORTRAN, COBOL).
- ✓ **Tercera generación.** Lenguajes de alto nivel imperativo. Son los mas utilizados y siguen vigentes en la actualidad (ALGOL 8, PL/I, PASCAL,MODULA).
- ✓ **Cuarta generación.** Orientados básicamente a las aplicaciones de gestión y al manejo de bases de datos (NATURAL, SQL).
- ✓ **Quinta generación.** Orientados a la inteligencia artificial y al procesamiento de los lenguajes naturales (LISP, PROLOG).

Para la mejor comprensión se harán unas definiciones

Programa: es un conjunto de instrucciones escritas en un lenguaje de programación que indican a la computadora la secuencia de pasos, para resolver un problema.

Código fuente: esta creado en algún lenguaje de alto nivel, por lo que es entendido 100% por el ser humano. Este debe estar complementado por su documentación o manuales donde se indica el desarrollo lógico del mismo.

Código objeto: es creado por los compiladores y nos sirve como enlace entre el programa fuente y el ejecutable.

INTÉRPRETES Y COMPILADORES

La traducción de una serie de instrucciones en lenguaje ensamblador (el código fuente) a un código máquina (o código objeto) no es un proceso muy complicado y se realiza normalmente por un programa especial llamado compilador. La traducción de un código fuente de alto nivel a un código máquina también se realiza con un compilador, en este caso más complejo, o mediante un intérprete. Un compilador crea una lista de instrucciones de código máquina, el código objeto, basándose en un código fuente. El código objeto resultante es un programa rápido y listo para funcionar, pero que puede hacer que falle el ordenador si no está bien diseñado. Los intérpretes, por otro lado, son más lentos que los compiladores ya que no producen un código objeto, sino que recorren el código fuente una línea cada vez. Cada línea se traduce a código máquina y se ejecuta. Cuando la línea se lee por segunda vez, como en el caso de los programas en que se reutilizan partes del código, debe compilarse de nuevo. Aunque este proceso es más lento, es menos susceptible de provocar fallos en la computadora.

PASOS PARA LA ELABORACIÓN DE UN PROGRAMA

1. Especificación del programa

Se conoce también como definición del problema o análisis del programa. En este paso se determinan la información inicial para la elaboración del programa. Es donde se determina qué es lo que debe resolverse con el computador, de qué presupuestos se debe partir en definitiva, el planteamiento del problema. Se requieren cinco tareas:

- a) Determinación de objetivos del programa. Debe definirse claramente los problemas particulares que deberán ser resueltos o las tareas que hay que realizar, esto nos permitirá saber qué es

lo que se pretende solucionar y nos proporcionará información útil para el planeamiento de la solución.

b) Determinación de la salida deseada. Los datos seleccionados deben ser arreglados en una forma ordenada para producir información. Esta salida podría ser una salida de impresión o de presentación en el monitor.

c) Determinación de los datos de entrada. Una vez identificada la salida que se desea, se pueden determinar los datos de entrada y la fuente de estos datos. Los datos deben ser recolectados y analizados.

d) Determinación de los requerimientos de procesamiento. Aquí se definen las tareas de procesamiento que deben desempeñarse para que los datos de entrada se conviertan en una salida.

e) Documentación de las especificaciones del programa. Es importante disponer de documentación permanente. Deben registrarse todos los datos necesarios para el procesamiento requerido. Esto conduce al siguiente paso del diseño del programa.

2. Diseño del programa

Es diseñar cualquier sistema nuevo o las aplicaciones que se requieren para satisfacer las necesidades. Esta actividad se debe dividir en:

- Operaciones de entrada/salida
- Cálculos
- Lógica/ comparación
- Almacenamiento/ consulta

En este paso se genera una solución con técnicas de programación como diseño descendente de programas, pseudocódigos, flujogramas y estructuras lógicas.

3. Codificación del programa

Es la generación real del programa con un lenguaje de programación. En esta etapa se hace uso de la lógica que desarrolló en el paso del diseño del programa para efectivamente generar un programa. Se debe seleccionar el lenguaje apropiado para resolver el problema.

4. Prueba

Depurar es correr el programa en una computadora y corregir las partes que no funcionan. En esta fase se comprueba el funcionamiento de cada programa y esto se hace con datos reales o ficticios. Cuando los programas están depurados, se prueban. Cuando los programas se depuran, se pueden encontrar los siguientes errores:

a) Errores; Es una violación de las reglas del lenguaje de programación. Son más fáciles de corregir, ya que son detectados por el compilador (posible error de escritura), el cual dará información sobre el lugar donde está y la naturaleza de cada uno de ellos mediante un mensaje de error.

b) Errores de Ejecución; Se deben generalmente a operaciones no permitidas como dividir por cero, leer un dato no numérico en una variable numérica, exceder un rango de valores permitidos, etc. Se detectan porque se produce una parada anormal del programa durante su ejecución.

c) Errores de Lógica; Corresponden a la obtención de resultados que no son correctos y la única manera de detectarlos es realizando suficientes pruebas del programa. Son los más difíciles de corregir, no sólo por la dificultad de detectarlos, sino porque se deben a la propia

concepción y diseño del programa.

d) Errores de Especificación; Es el peor tipo de error y el más difícil de corregir. Se deben a mal diseño del programa posiblemente por mala comunicación usuario programador y se detectan cuando ya se ha concluido el diseño e instalación del programa, lo cual puede implicar repetir gran parte del trabajo realizado.

Prueba; Consiste en verificar la funcionalidad del programa a través de varios métodos para detectar errores posibles.

Métodos de Prueba

Chequeo de escritorio; El programador se sienta frente a un escritorio y corrige una impresión del programa. Revisa el listado línea por línea en busca de errores de sintaxis y lógica.

Prueba manual de datos de muestra; Se corre el programa en forma manual aplicando datos tanto correctos como incorrectos para comprobar que funciona correctamente.

Intento de traducción; El programa corre en una computadora usando un programa traductor para convertirlo a lenguaje de máquina. Para ello debe estar ya libre de errores de sintaxis, de lo contrario serán identificados por el programa de traducción.

Prueba de datos de muestra en la computadora; Después del intento de traducción y corregidos los errores de sintaxis, se procede a buscar errores de lógica utilizando diferentes datos de muestra.

Prueba por un grupo selecto de usuarios potenciales; Esto se conoce como prueba beta. Se trata por lo general del paso final en la prueba de un programa. Usuarios potenciales ponen a prueba el programa y ofrecen retroalimentación.

5. Documentación

Consiste en describir por escrito a nivel técnico los procedimientos relacionados con el programa y su modo de uso. También se debe documentar el programa para que sea más entendible.

¿Para quiénes son la documentación?

- Usuarios (Digitadores): Manual de Usuario A los usuarios se les elabora un manual de referencia para que aprendan a utilizar el programa. Esto se hace a través de capacitaciones y revisión de la documentación del manual de usuario. El manual del usuario no está escrito a nivel técnico sino al de los distintos usuarios previstos y explica en detalle cómo usar el programa: descripción de las tareas que realiza el programa, instrucciones necesarias para su instalación puesta en marcha y funcionamiento, recomendaciones de uso, menús de opciones, método de entrada y salida de datos, mensajes de error, recuperación de errores, etc.
- Operadores: por si se presentan mensajes de error, sepan cómo responder a ellos. Además que se encargan de darle soporte técnico al programa
- Programadores, a través del manual del analista para que recuerden aspectos de la elaboración del programa o en caso que otras personas puedan actualizarlo o modificarlo (darle mantenimiento) y no son necesariamente las personas que lo diseñaron. Es por ello, A que la documentación debe contener algoritmos y flujogramas de los diferentes módulos que lo constituyen y las relaciones que se establecen entre ellos; listados del programa, corridas, descripción de variables que se emplean en cada módulo, cuáles son comunes a diferentes módulos y cuáles locales; descripción de los ficheros de cada módulo y todo lo que sea de importancia para un programador.
- Analistas de sistemas que son las personas que deberán proporcionar toda la información al programador. Estos se encargan de hacer una investigación previa de cómo realizar el programa y documentar con las herramientas necesarias para que el

programador pueda desarrollar el sistema en algún lenguaje de programación adecuado

6. Mantenimiento

Es el paso final del desarrollo del software. Alrededor del 75% del costo total del ciclo de vida de un programa se destina al mantenimiento. El propósito del mantenimiento es garantizar que los programas en uso estén libres de errores de operación y sean eficientes y efectivos

HERRAMIENTAS DE PROGRAMACION

Las herramientas de programación, son aquellas que permiten realizar aplicativos, programas, rutinas, utilitarios y sistemas para que la parte física del computador u ordenador, funcione y pueda producir resultados.

Hoy día existen múltiples herramientas de programación en el mercado, tanto para analistas expertos como para analistas inexpertos.

Las herramientas de programación más comunes del mercado, cuentan hoy día con programas de depuración o debugger, que son utilitarios que nos permiten detectar los posibles errores en tiempo de ejecución o corrida de rutinas y programas.

Entre otras herramientas de programación encontramos librerías y componentes, dados por algunos lenguajes de programación como son el C++ y delphi.

Otras herramientas de programación son los lenguajes de programación, que nos permitenn crear rutinas, programas y utilitarios.

Entre algunas de estas herramientas de programación tenemos:

- ✓ Basic y Pascal que son herramientas de programación, idoneas para la inicialización de los programadores.

- ✓ C y C++ que sirven para la programación de sistemas.
- ✓ Cobol, que es una herramienta de programación orientada hacia sistemas de gestión empresarial como nóminas y contabilidad.
- ✓ Fortran, que son lenguajes específicos para cálculos matemáticos y/o numéricos.
- ✓ Herramientas de programación para ambientes gráficos como son Visual Basic, Delphi y Visual C.
- ✓ Html y Java, que permiten la creación de páginas WEB para internet.

COPIADORES

Un compilador es un programa traductor que a diferencia de los intérpretes, lleva a cabo la fase de traducción de dos formas, primero traduce completamente el programa fuente a código máquina y seguidamente lo ejecuta. Un programa compilado no necesita traducciones del programa fuente en ejecuciones sucesivas, lo que hace mucho más rápido el tiempo de ejecución. Se puede decir por tanto que los compiladores presentan todo lo bueno de los ensambladores y los intérpretes.

Las etapas del proceso de compilación son:

1. Edición: Consiste en la escritura del programa utilizando un lenguaje de programación precisamente seleccionado y su posterior grabación en un soporte de almacenamiento permanente. La edición del programa debe realizarse mediante un editor de textos que puede formar parte o no del compilador. En esta fase se obtiene el denominado programa fuente.
2. Compilación: En esta fase se traduce el programa fuente a su equivalente en código máquina, obteniendo en caso de no

producirse ningún error el denominado programa o modulo objeto. En caso de producirse errores, el compilador los mostrará utilizando los mensajes correspondientes, lo cual nos permitirá corregir el programa fuente y proceder de nuevo a su compilación.

3. Linkado: Esta fase también recibe el nombre de montaje y consiste en unir o enlazar el programa objeto con determinadas rutinas internas del lenguaje. Si el método de compilación es modular se enlazarán los distintos módulos para obtener así el programa ejecutable.
4. Ejecución: Esta fase consiste en la llamada del programa ejecutable a través del sistema operativo. Inicialmente se debe comprobar el buen funcionamiento del programa mediante el uso de unos juegos de pruebas que especifican los resultados que se desean obtener en función de unos determinados datos de entrada.

Los principales errores en la ejecución de un programa son:

- ✓ Datos de entrada incorrectos que puedan producir una parada del sistema, por ejemplo, un dividendo con valor 0 en una operación dará como resultado infinito y producirá el error.
- ✓ Bucles mal definidos que producen un funcionamiento continuo
- ✓ Datos de salida o resultados incorrectos producidos por un mal desarrollo del programa o por la existencia de una cierta ambigüedad en las especificaciones del usuario.

CONCLUSIONES

En la realización de este estudio se llegó entre otras a la siguiente conclusión.

- Como se ha observado en los últimos años, el gran crecimiento que han tenido el área tecnológica a nivel mundial, la cual facilita el gran número de procesos en las compañías haciendo de estas cada vez más innovadoras y competitivas, a través de un mejor rendimiento tanto del capital humano como de los procesos utilizados los cuales han llevado a la toma de decisiones más efectivas y de manera más directa.
- Cabe interpretar, que la complejidad en la elaboración de un programa trae a la reflexión a los usuarios debido a que los autores de los mismos no deben ser producto de la piratería en cuanto al plagio de sus trabajos, por ello como usuarios no se debe ser participe de esta actividad que ha perjudicado a más de un autor a nivel mundial.
- Por otra parte, es importante la aceptación de nuevos y más económicos software como es el caso del “software Libre” el cual ha innovado la adquisición de los sistemas operativos ya que es uno

de los mas economicos y completos que existen en la actualidad, el cual permite a los usuarios personalizar sus sistema sin necesidad de inversiones de realizar inversiones costosas.

- Es por esta razon, que se debe mantener este crecimiento tomando en cuenta las bases legales para la utilizaciòn de los software de gestiòn ya que a medida que estos sean mejorados se obtendrian mejores rendimientos en su posteriores actualizaciones, ya que esto motivaria cada vez mas a los creadores de dichos sistemas hacerlos cada vez mas rápidos y efectivos para la utilizaciòn de los usuarios
- Asimismo, se debe seguir incentivando tanto a las empresas como a los usuarios comunes en la utilizacion de este tipos de software, el cual constituye una herramienta indispensable para agilizar los procesos.