

# SISTEMA EXPERTO DE REQUERIMIENTOS Y COMPATIBILIDAD SOBRE ENSAMBLAJE DE COMPUTADORAS

Rita Azucena Díaz Vásquez \* , Jorge Lenin Acosta Espinoza \* , Glenda Cecibel Intriago Alcívar \*  
Universidad Regional Autónoma de los Andes, Riobamba, UNIANDES, Ecuador \*\* Universidad Técnica Estatual de Quevedo, UTEQ, Ecuador

## ABSTRACT

The purpose of this system is to satisfy the expectations and requirements of the customers of computer assembly, which is a procedure that consists on correctly placing all the parts of a computer in order such that everything works efficiently. Then, it will be able to install the operating system and other programs according to the user's needs, thus a computer selection is based on information on compatibility and assembly of computers. The information has been compiled from books, scientific papers, and based on research tools such as: surveys, interviews, for the collection of information. The development of this system serves as a tool that in the future may bring some comfort to users with little knowledge of hardware to make purchases according to the needs and customer's economy, which is an important aspect in people's daily lives. The Expert System of requirements and compatibility in assembly of computers realizes what an expert engineer in assembly of computers can do, but the system does it in a fast way and it is always available.

**KEYWORDS:** Assembly, compatibility, expert system

**MSC:** 68T35

## RESUMEN

El propósito de este sistema es cumplir con las expectativas y requerimientos de los clientes de ensamblaje de computadora que es un procedimiento que consiste en colocar correctamente todas las partes de una computadora con la finalidad que todo funcione eficientemente para luego poder instalar el sistema operativo y demás programas de acuerdo a las necesidades del usuario, para que al momento de elegir una computadora se fundamente en información sobre compatibilidad y ensamblaje de computadores. La información ha sido recopilada a través de libros, artículos científicos, con instrumentos de investigación como: encuestas, entrevistas, para la recopilación de la información. El desarrollo de este sistema sirve como herramienta que en el futuro pueda portar cierta comodidad a los usuarios con poco conocimiento de hardware para realizar compras acordes a las necesidades y de acuerdo a la economía del cliente, aspecto importante en la vida diaria de las personas. El Sistema Experto de requerimientos y compatibilidad en ensamblaje de computadoras realiza lo que un ingeniero experto en ensamblaje de computadoras puede hacer, pero el sistema lo hace de una manera rápida y siempre está disponible.

**PALABRAS CLAVES:** Ensamblaje, compatibilidad, sistema experto

## 1. INTRODUCCIÓN

La empresa MEGASYSTEM se dedica a la venta de computadoras además de todo tipo de dispositivos tecnológicos. Esta empresa proporciona a sus clientes proformas manuales de los equipos que tiene a la venta, por lo cual se pudo observar la falta de un servicio eficaz hacia los clientes, al no brindarles proformas con los equipos de características precisas en cuanto a los requerimientos, funciones, presupuesto y necesidades de cada cliente, razones por las cuales se vio en la necesidad de crear un sistema que ensamble un computador con los componentes que reúnen las características que el cliente realmente necesita y si está en el presupuesto con que el cliente cuenta.

Conociendo que los Sistemas Expertos (SE) son programas de computador que buscan simular los procesos intelectuales de los expertos humanos como: la interpretación, el diagnóstico, la monitorización, el control, la predicción, la planificación, el diseño y la enseñanza, entre otros. Una interesante orientación que caracteriza los sistemas en Inteligencia Artificial (IA) es llamado Principio de Generalidad: Al tener una estructura lógica básica necesaria para solucionar un problema, entonces, en principio, puede usarse la potencia del computador para hallar la solución. Este principio acepta que hay un método general para expresar y solucionar problemas con conocimiento común. Hoy día, los SE o Sistemas Basados en el Conocimiento (SBC) son base para desarrollar sistemas inteligentes directamente asociados con la representación de conocimiento y procesos de razonamiento, [1]. La idea de la construcción de los Sistemas Expertos se simplifica en dos aspectos:

- a) La adquisición, la representación y el uso de conocimiento especializado.
- b) El reconocimiento que los sistemas, además de resolver problemas, puedan poseer atributos propios de expertos humanos como:

- Capacidad para adquirir nuevo conocimiento.
- Capacidad para justificar sus conclusiones.
- Capacidad de hacer preguntas y explicar por qué las hacen.
- Capacidad conversacional.

La utilización de software y materiales virtuales computarizados como un recurso para apoyar los procesos de venta de las diversas áreas del conocimiento, se ha convertido en una necesidad y constituye una respuesta ante la problemática que gira en torno de la comprensión de conceptos y nociones en un tema de escoger un computador acorde a la necesidad del cliente.

En particular el surgimiento de la Inteligencia Artificial como una disciplina científica y el desarrollo tecnológico que ha impulsado en el campo de los Sistemas Expertos ha abierto una gama de posibilidades al público en general.

A pesar de que tanto el desarrollo como la adquisición de un Sistema Experto es relativamente caro, con la adquisición de un sistema para el buen funcionamiento de determinada empresa, se puede tener un ahorro de tiempo futuro, ya que el tratamiento y el coste marginal de su uso constante resulta ser más económico. De igual forma, con el uso de Sistemas Expertos se obtienen ganancias muy altas en términos monetarios, tiempo, y precisión, así como también la amortización resulta ser muy rápida. Aun así, es aconsejable realizar un análisis de factibilidad y de costo-beneficio antes de adquirir o desarrollar un Sistema Experto, [5]. Existen varias razones de peso que explican por qué usar Sistemas Expertos, esto tiene sus ventajas entre ellas podemos mencionar:

1. Un Sistema Experto permite que un problema complejo, pueda ser resuelto por personas con poca experiencia en el tema. Es esta una ventaja para la empresa MEGASYSTEM que cuenta con poco personal experto en determinadas áreas. Por lo tanto, se puede deducir que con el uso de los Sistemas Expertos aumenta el número de personas con acceso al conocimiento.
2. Es posible reunir y combinar el conocimiento de varios expertos humanos lo cual conlleva a tener Sistemas Expertos confiables, puesto que usan la inteligencia y la sabiduría de varios humanos con conocimientos del tema en vez de usar un solo criterio.
3. Una gran ventaja es que su tiempo de respuesta es menor al que tardaría una persona, aunque sea muy experta en el tema. Es esta cualidad la que los hace más valiosos en circunstancias donde el tiempo de respuesta es un factor crítico.
4. Dependiendo del grado de complejidad de determinado problema es posible o no resolverlo por un experto humano. Es aquí donde se puede notar la importancia de los Sistemas Expertos, ya que, aprovechando la evolución tecnológica, estos usan herramientas de cómputo para procesar gran cantidad de operaciones complejas de forma rápida y acertada, suministrando respuestas confiables en situaciones en las que los expertos humanos podrían tener dificultades.
5. Es posible usar Sistemas Expertos para la solución de problemas monótonos en medios en los cuales los expertos humanos se sienten aburridos e incómodos. Un Sistema Experto es la única solución a problemas donde la tarea a realizar desborda al ser humano.
6. Por medio de un Sistema Experto, es posible obtener un ahorro tanto económico como en el tiempo requerido para la solución de determinado problema. El uso de los Sistemas Expertos es recomendado en diversas situaciones, entre ellas están las siguientes:
  - Cuando el experto humano no cuenta con el conocimiento, o cuando dicho conocimiento es basado en reglas las cuales requiere de la experiencia para poder ser aprendidas.
  - Cuando el problema depende de reglas que se encuentran en constante cambio.
  - Cuando la mejora continua del conocimiento es lo esencial para una organización.
  - Cuando no se cuenta con cantidad suficiente de expertos humanos.
  - Cuando contratar expertos humanos es demasiado costoso.
  - Cuando no se tiene el conocimiento suficiente sobre el problema que se pretende solucionar.

Este artículo tiene como objetivo mostrar el diseño del Sistema Experto desarrollado por la empresa MEGASYSTEM para el requerimiento y compatibilidad sobre ensamblaje de computadoras. Se destacarán las ventajas y utilidad del sistema para ser usado.

## 2. MATERIALES Y MÉTODOS

Para el desarrollo de la aplicación se optó por la metodología de desarrollo de software. Se trabajó con encuestas a clientes para conocer las necesidades como compradores de equipos de computación y así

implementar las funciones requeridas en el desarrollo del sistema. También se realizaron entrevistas para verificar la factibilidad del sistema y los procesos que este debe cumplir. Para la investigación se utilizó libros, internet, requerimientos en software y hardware, véase Tablas 1 y 2.

<b>Software</b>	
Adobe Dreamweaver CS6	Para la programación PHP, Ajax, html5
Servidor Web IIS	Para ejecutar el sistema de manera local.
Gestor de Base de Datos: Mysql 5.0	Para guardar la información que contiene el sistema.
Browser: Mozilla, Firefox, Google, Chrome, Internet explorer	Para visualizar el sistema

**Tabla 1** Software necesario para el desarrollo del sistema.

<b>Hardware</b>	
Computador HP coreI7	Para la elaboración del sistema experto
Impresora Epson TX730 Series	Impresión de proformas.

**Tabla 2** Hardware necesario para el desarrollo del sistema

Fue necesario usar la Programación Extrema (*eXtreme Programming* en inglés) o Metodología XP para que el sistema pueda estar orientado hacia las necesidades del cliente, haciendo del cliente parte fundamental del equipo desarrollador del sistema quien estará presente desde el inicio hasta el final de su desarrollo y de esta manera lograr satisfacer sus necesidades, [7, 11].

La metodología XP cuenta con las siguientes fases:

1. **Planificación del proyecto.** - Detalla las historias de usuarios en donde el cliente establece las entradas, las salidas, y los procesos que debe tener el sistema para lograr los objetivos planteados. Estas historias de usuario también se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia, [11]. Véase Tabla 3.

<b>HISTORIA DE USUARIO</b>			
Nombre de la historia:	HU 101 Creación de la interfaz de ingreso a la aplicación		
Días asignados:	3	Prioridad:	Alta
Disposición:	Planificada	Estado:	Definida
Iteración asignada:	Iteración 1		
Descripción:	Diseño de la interfaz para Autenticación de usuarios además del formulario de registro de usuarios.		
Observación:	La interfaz contendrá enlaces a otros formularios		

**Tabla 3** Historia de usuario

2. **Diseño.** - En esta fase se sugiere conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar. De esta manera se da mayor importancia a la codificación para que el sistema sea eficiente, confiable y pueda cumplir con las funciones requeridas por los usuarios, [10]. La etapa de diseño de un Sistema Experto incluye el diseño de estructuras para archivar el conocimiento, véase [6], al igual que el motor de inferencia, el subsistema de explicación, la interfaz de usuario, entre otros.
3. **Codificación.** - Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que esta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada. La codificación debe hacerse ateniendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad. XP opta por la programación en pareja ya que permite un código

más eficiente y con una gran calidad. Esta metodología sugiere un modelo de trabajo usando repositorios de código donde las parejas de programadores publican cada pocas horas sus códigos implementados y corregidos junto a los test que deben pasar, [2].

4. **Pruebas.** - Uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos que se van implementando. Las pruebas ayudan a saber qué es exactamente lo que tiene que hacer el sistema a implementar y se sabrá que una vez implementado pasará dichas pruebas sin problemas. El construir y probar es necesario en esta etapa, [8], si las pruebas no arrojan los resultados esperados, se hace necesario realizar modificaciones a las etapas anteriores, y ejecutar nuevas pruebas tantas veces como sea necesario hasta obtener un sistema satisfactorio. El refinamiento y generalización es en esta etapa donde aparte de corregir los errores encontrados, se incorporan nuevas opciones las cuales no habían sido incluidas en el diseño inicial, [9].

El mantenimiento y puesta al día presenta el primer contacto del usuario final con el Sistema Experto, lo cual permitirá al usuario plantear problemas o defectos de la primera versión obtenida del Sistema Experto, y estas observaciones hechas por el usuario permitirá realizar correcciones y actualizar el producto con nuevos avances. Este conjunto de etapas, enmarcan la calidad del Sistema Experto obtenido, [4], el cual debe ser evaluado constantemente en función de los aportes entregados por los usuarios finales, véase [8].

La modelación matemática del SE se basa en relaciones n-arias, ver [3], en este caso con  $n = 8$ .

**Definición 1.** Una *relación n-aria* sobre  $X = X_1 \times X_2 \times \dots \times X_n$  se define como  $R = \{(x_1, x_2, \dots, x_n) \in X: x_1 \in X_1 \wedge x_2 \in X_2 \wedge \dots \wedge x_n \in X_n \wedge R(x_1, x_2, \dots, x_n) \text{ es verdadera}\}$ .

También se utilizará un árbol, que formalmente se define según la Definición 2.

**Definición 2** Un *árbol* es un grafo conexo acíclico, véase [11].

Entre la nomenclatura del árbol se encuentran los *nodos* que son los vértices en la Teoría de Grafos. Se dice que *a* es padre de *b*, o que *b* es hijo de *a*, donde *a* y *b* son nodos del árbol, si existe una arista que va de *a* hasta *b*. Dos nodos se dicen *hermanos* si tienen el mismo padre. El nodo que no tiene padre se le llama *raíz*, mientras que los nodos que no tienen hijos se llaman *hojas*.

El *nivel* de un nodo se calcula por  $1 +$  el número de conexiones entre este y la raíz.

## 2.1. Proceso de selección de un computador armado

Los pasos del sistema cuando el usuario selecciona el computador ya ensamblado se dan detallan en el Algoritmo 1.

<b>Algoritmo 1: De Selección del Computador Ensamblado</b>
<ol style="list-style-type: none"> <li>1. Cada usuario crea o accede a una cuenta. Esto se hace accediendo a una serie de ventanas visuales que interactúan con el Sistema Experto.</li> <li>2. Aparece un asistente para determinar la selección del usuario sobre el hardware que desea obtener.</li> <li>3. El usuario selecciona el tipo de computador que desea. <ul style="list-style-type: none"> <li>• De escritorio</li> <li>• Portátil</li> </ul> </li> <li>4. El usuario selecciona el uso que le dará al computador. Véase Figura 1. <ul style="list-style-type: none"> <li>• Para el hogar o la pequeña oficina</li> <li>• Para juegos</li> <li>• Para diseño gráfico</li> <li>• Profesional</li> </ul> </li> <li>5. El usuario selecciona la arquitectura con la que desea que cuente su computador <ul style="list-style-type: none"> <li>• AMD</li> <li>• Intel</li> </ul> </li> <li>6. El usuario ingresa el presupuesto en dólares con que cuenta.</li> <li>7. Si el usuario está seguro de sus opciones pulsa el botón <b>construir</b>.</li> <li>8. El sistema devuelve las opciones de hardware con sus precios, que satisfacen las características pedidas por el usuario.</li> <li>9. Si el usuario cambia de opinión en alguna de sus selecciones puede editar de nuevo su pedido.</li> </ol>



**Figura 1** Pantalla que permite escoger opciones de acuerdo con el uso que se le dará al computador

## 2.2. Proceso de selección de un computador personalizado

En el presente artículo se definirán los conjuntos siguientes:

1.  $X_1$ : es el conjunto de posibles usos que el usuario pudiera darle a la PC que desea armar. Estos usos se representan en forma de cadena de caracteres y son los siguientes:
  - Para el hogar o la pequeña oficina
  - Para juegos
  - Para diseño gráfico
  - Profesional
2.  $X_2$ : es el conjunto de precios disponibles.
3.  $X_3$ : es el conjunto de todos los *procesadores* disponibles. Cada elemento del conjunto será una cadena de caracteres con el nombre del procesador.
4.  $X_4$ : es el conjunto de cadenas de caracteres con el nombre de los tipos de *mainboard* disponibles por la empresa.
5.  $X_5$ : es el conjunto de cadenas de caracteres con el nombre de los tipos de *memoria RAM* disponibles por la empresa.
6.  $X_6$ : es el conjunto de cadenas de caracteres con el nombre de los tipos de *discos duros* disponibles por la empresa.
7.  $X_7$ : es el conjunto de cadenas de caracteres con el nombre de los tipos de *Unidades de DVD* disponibles por la empresa.
8.  $X_8$ : es el conjunto de cadenas de caracteres con el nombre de los tipos de *Sistemas Operativos* propuestos por la empresa.

La relación  $R$  de la Definición 1, se considerará de *compatibilidad* entre las componentes, es decir,  $R(x_1, x_2, \dots, x_8)$  es verdadera si y solo si existe compatibilidad a la vez, entre el uso que el usuario puede dar a la PC, con el precio, con el tipo de procesador, con el mainboard, con la memoria RAM, con el disco duro, las unidades de DVD y el sistema operativo.

Los subíndices de los conjuntos  $X_i$  significan el orden de importancia de la componente. Por ejemplo, se considera que el uso que el usuario desea darle a su PC es lo más importante para el armado de esta, por ello se representa por el conjunto de subíndice 1, luego lo más importante es que el usuario cuente con el presupuesto necesario para pagar el precio de una PC según su uso, por eso se representa por el subíndice 2, luego es más importante identificar qué tipo de procesador es necesario para cumplir con tal uso y tal precio, por lo tanto se asocia con el subíndice 3, y así sucesivamente.

Cada conjunto  $X_i$  definido anteriormente coincide con el nivel  $i$ -ésimo del árbol. Si existe alguna PC tal que  $(x_i, x_{i+1})$  son compatibles, entonces en el árbol se tendrá que  $x_i$  es el padre de  $x_{i+1}$ . Véase un árbol genérico en la Figura 2 con los tres niveles superiores. El árbol está diseñado de tal manera que cada regla de la Base de

Conocimiento se obtiene si se desciende desde la raíz hasta el último nivel.

Se definirá la Ecuación 1 sobre compatibilidad, dado  $\mathbf{x} = (x_1, x_2, \dots, x_8)$ .

$$C_i(x_{i-1}, x_i) = \begin{cases} 1, & \text{si existe } R(\mathbf{x}) \text{ verdadera con } x_{i-1} \text{ y } x_i \\ 0, & \text{en caso contrario} \end{cases} \quad (1)$$

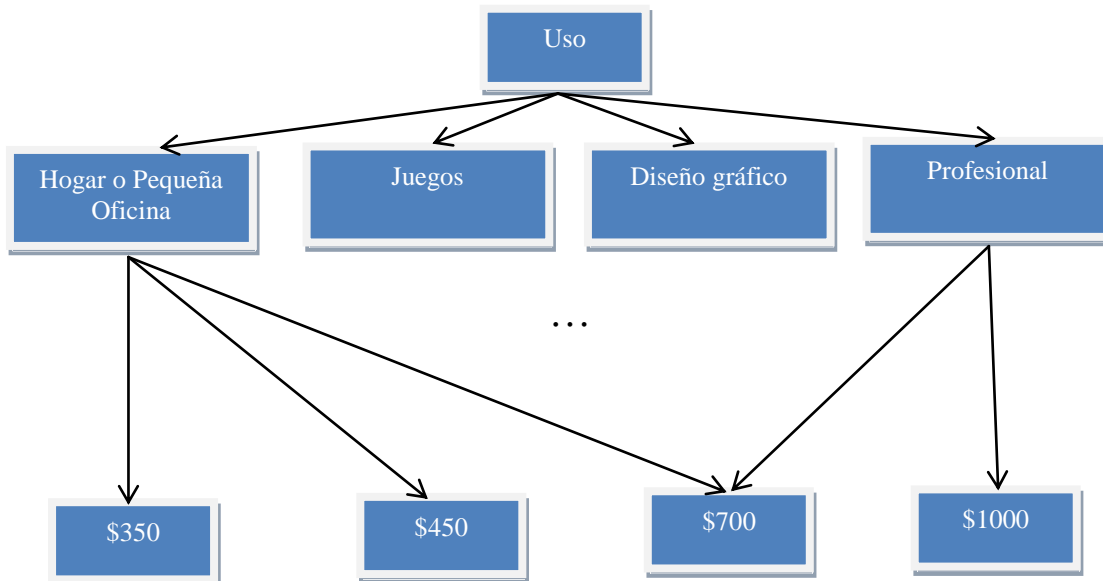
$i = 2, \dots, 8$ .

Las Ecuaciones 2 y 3 representan la similaridad entre opciones, dadas  $\mathbf{x} = (x_1, x_2, \dots, x_8)$  e  $\mathbf{y} = (y_1, y_2, \dots, y_8)$ .

$$s_i(x_i, y_i) = \begin{cases} 1, & \text{si } x_i = y_i \\ 0, & \text{en otro caso} \end{cases} \quad (2)$$

$$S^i(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^i s_j(x_j, y_j) \quad (3)$$

Para garantizar la eficiencia de los algoritmos, es importante que los elementos de cada  $X_i$  tengan un orden dentro del árbol. Por ejemplo, la Figura 2 contiene en el primer nivel un orden del *uso* según la complejidad, por otro lado, en el segundo nivel el orden de los precios también está ordenado de menor a mayor. De igual manera se pueden ordenar los procesadores, de los más simples a los más complejos etc. Esto permite que haya una concentración entre los diferentes tipos de PC, por ejemplo, se espera que una PC de uso para el hogar (ubicado más a la izquierda del árbol), tenga un precio más barato que los demás tipos, y así sucesivamente.



**Figura 2** Gráfico hipotético de representación del árbol para los tres niveles superiores.

Basado en lo anterior se propone el Algoritmo 2.

**Algoritmo 2: De Selección del Computador Personalizado**

1. Cada usuario crea o accede a una cuenta. Esto se hace accediendo a una serie de ventanas visuales que interactúan con el Sistema Experto.
2. Aparece un asistente para determinar la selección del usuario sobre el hardware que desea obtener.
3. El usuario debe de seleccionar la opción que desea: una PC personalizada.
4. Se inicializa  $i = 1$ .
5. El usuario propone  $\tilde{x}_i \in X_i$ 
  - 5.1. Si  $i = 1$ ; o  $1 < i < 8$  y  $C_i(x_{i-1}, x_i) = 1$ , entonces hacer  $i = i + 1$ . Repetir Paso 5.
  - 5.2. Si  $1 < i$  y  $C_i(x_{i-1}, x_i) = 0$ , entonces el SE busca en la Base de Conocimiento algún  $y_i$  tal que  $C_i(x_{i-1}, y_i) = 1$ , con  $S^i((x_1, x_2, \dots, x_i))$  sea el más cercano a  $S^i((x_1, x_2, \dots, y_i))$ . Véase Ecuaciones 1, 2 y 3.  
El SE le recomienda al usuario el o los  $y_i$ . Se espera a que el usuario acepte. Una vez que acepta, esto lleva a dos pasos diferentes:

- Si  $i < 8$ , se hace  $i = i + 1$ . Repetir Paso 5.
  - Si  $i = 8$ , Terminar.
- 5.3. Si  $i = 7$  y  $C_g(x_7, x_8) = 1$ , Terminar.

### 3. RESULTADOS

Los autores de este artículo programaron adicionalmente un simulador del comportamiento de un usuario que desearía obtener una PC personalizada, para comprobar el funcionamiento del Algoritmo 2.

Se utilizó la fórmula de desempeño  $D = \frac{NA^+}{N} \times 100$ , donde N es la cantidad de veces que se realiza la simulación y  $NA^+$  es el número de veces que el sistema determina que hay un verdadero positivo.

Se utilizan expertos para determinar si el resultado es un verdadero positivo. Si P es la variable aleatoria que determina si el experimento es un acierto o una fracaso, cada vez que el sistema llega a una solución que es un verdadero positivo se asigna  $P = 1$ , de lo contrario  $P = 0$ .

Primero se evaluó P para 50 ocasiones con el simulador y se obtuvo una desviación estándar muestral  $s_p = 0,2399$  aproximadamente y una media muestral de  $\bar{P} = 0,94$ . Se utilizó la fórmula  $N = \left[ \frac{s_p Z_{\alpha/2}}{e} \right]^2$  para calcular el número de simulaciones necesarias, donde  $Z_{\alpha/2}$  es el valor crítico para un nivel de confianza de  $1 - \alpha$ , de la distribución normal de media 0 y varianza 1. Para el nivel de confianza  $\alpha = 0,05$  se tiene  $Z_{\alpha/2} = 1,96$ . Mientras que  $e = 0,05$  es el error máximo admitido. Por tanto se toma  $N = 89$ .

Luego de comprobar el resultado del Algoritmo 2 con 89 casos, se obtuvo  $D = 95,506\%$ , que es un número alto de aciertos.

### 4. CONCLUSIONES

En este artículo se expuso el diseño de un Sistema Experto que permite a los usuarios no necesariamente expertos, de una manera sencilla y rápida la determinación de la compatibilidad para el ensamblaje de computadores. Se comprobó mediante simulación que se puede esperar alrededor de un 95% de acierto del Sistema Experto en su Algoritmo 2.

El Sistema Experto ayuda al usuario bajo dos circunstancias, cuando el usuario desea obtener un computador ensamblado, el sistema lo ayuda a seleccionar el más adecuado a sus necesidades. Por otra parte, si el usuario desea obtener una PC personalizada, el sistema le permite que diseñe la PC de acuerdo a sus necesidades y corrige en caso de incompatibilidad.

Debido a que se usó un árbol, es posible incluir casos nuevos donde se diseñen nuevos computadores compatibles. Por tanto, es posible actualizar el sistema de manera fácil. También fue posible desarrollar algoritmos con un grado alto de exactitud debido a lo relativamente pequeño que es el número de combinaciones para armar una PC. Si se tuviera un número mayor habría que considerar otras representaciones del conocimiento, por ejemplo árboles de decisión que usan algoritmos como ID3 o C4.5.

**RECEIVED: DECEMBER, 2018.**

**REVISED: JUNE, 2019**

### REFERENCIAS

- [1] FLORES FONSECA, V. M. (2010) **Generación automática de resúmenes de comportamiento de sistemas dinámicos mediante modelos de conocimiento del dominio**, Tesis Doctoral Informática.
- [2] HEURTEL, O. (2009) **PHP y MySQL: domine el desarrollo de un sitio web dinámico e interactivo**. Ediciones Eni.
- [3] JECH, T. (2003) **Set Theory**. Springer-Verlag Berlin Heidelberg.
- [4] LARSON, R. E., TINNEY, W. F., HAJDU, L. P. y PIERCY, D. S. (1970) State estimation in power systems part II: Implementation and applications, **IEEE Transactions on Power Apparatus and Systems**, 1970, 353-363.
- [5] LÓPEZ SEPÚLVEDA, G. P. (2011) **Creación de un sistema experto aplicado a estimación de estado para centros de control de energía**, Tesis Doctoral. Universidad Tecnológica de Pereira, Pereira, Colombia.
- [6] MENDOZA CÁRDENAS, S. R. y CÁCERES ZÁRATE, J. N. (2016) **Sistema experto para la selección de postulantes en puestos de una agencia bancaria usando la metodología CommonKads**. Tesina para optar el Título Profesional de Ingeniero de Sistemas, Universidad

- Nacional Mayor De San Marcos, Lima, Perú
- [7] MEX.TL. (2014). Ingeniería de Software . Obtenido de Mex.tl: Disponible en: [http://ingenieriadesoftware.mex.tl/52753\\_XP---Extreme-Programing.html](http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html) . Consultado 20-2, 2018.
  - [8] PREECE, A. D. (1990) Towards a Methodology for Evaluating Expert Systems. **Expert Systems**, 7, 215–293.
  - [9] RUIZ FLOREZ, H. A. (2009) **Estimación de estado de sistemas eléctricos Usando algoritmos de optimización combinatorial**. Tesis Doctoral. Universidad Tecnológica de Pereira, Pereira, Colombia.
  - [10] SCHILDT, H. (2002) **Fundamentos de programación en Java 2**, Osborne Mc Graw Hill, Colombia.
  - [11] WALLIS, W.D. (2007) **A Beginner’s Guide to Graph Theory**, Birkhäuser, Boston.