

# DESARROLLO DE UN SISTEMA RECOMENDADOR WEB PARA LA TOMA DE DECISIONES DURANTE EL PROCESO DE ADQUISICIÓN DE EQUIPOS DE CÓMPUTO UTILIZANDO ÁRBOLES DE DECISIÓN

Eduard Wbeimar Díaz García

Andrés Lozano Garzón

Angela Patricia Arenas Amado

José Fernando Sotelo

Universidad de Cundinamarca, Facultad de Ingeniería, Programa de Ingeniería de Sistemas  
Fusagasugá, Cundinamarca, Colombia

e-mail: eduarddz85@gmaili.com , omegaandres@gmail.com , apatriciaaa1@gmail.com ,  
jfsotelo2000@gmail.com

**Resumen-** El sistema recomendador web TechCity tiene como objetivo proporcionar asesoría a los posibles clientes para que puedan adquirir un equipo de cómputo, basándose en 1 de los 3 perfiles de usuario con los que cuenta el sistema junto con el presupuesto proporcionado. Cada uno de estos perfiles engloba, a grandes rasgos, los posibles clientes que pueden acceder al sistema: el primero, “Básico”, comprende los clientes que desean adquirir un equipo para el hogar, capaz de llevar a cabo actividades básicas como acceder a internet, redactar documentos de texto u otras tareas que no requieran mayor capacidad de procesamiento; el segundo perfil, “Estación de trabajo”, comprende los usuarios que desean adquirir un equipo de procesamiento medio, con el cual tendrán la capacidad de realizar tareas que necesiten un poder de procesamiento gráfico medio, tales como: diseño, renderizado, desarrollo de software, entre otras; y por último el tercer perfil, usuario “Entusiasta”, con miras a obtener el mayor rendimiento posible en su equipo, capacidad de utilizar una configuración con varios monitores y los componentes del más

alto rendimiento, con el objetivo de ejecutar tareas demandantes como lo son, el renderizado de imágenes en 4K y 8K, la ejecución de juegos de última generación en alta definición, o cualquier otra que requiera de altas capacidades de procesamiento.

La asignación de los componentes a cada una de las categorías se realiza utilizando árboles de decisión proporcionados por las librerías de Python “SciKit-Learn”, los cuales permiten la clasificación de acuerdo a sus características, y así mediante un algoritmo combinatorio generar las posibles opciones (combinaciones), teniendo en cuenta las condiciones necesarias para que las partes del nuevo equipo sean compatibles entre sí, asegurando su utilización sin ningún inconveniente o percance.

*Palabras Clave:* Sistema Recomendador, Datos, Información, Predilección, Árboles de Decisión, Machine Learning, Aprendizaje Supervisado.

## I. INTRODUCCIÓN

**Abstract-** The TechCity web recommendation system aims to provide advice to potential customers to acquire computer equipment, based on 1 of the 3 user profiles that the system counts along with the budget provided. Each of these profiles broadly encompasses potential customers who can access the system: the first, "Basic", includes customers who want to purchase a home computer, capable of carrying out basic activities such as accessing the internet, Writing text documents or other tasks that do not require more processing capacity; The second profile, "Workstation", comprises users who want to acquire medium processing equipment, with which they will be able to perform tasks that require medium graphics processing power, such as: Design, rendering, software development, among others; And finally the third profile, user "Enthusiast", with a view to obtaining the highest possible performance in your computer, ability to use a configuration with multiple monitors and components of the highest performance, aiming to perform demanding tasks as they are, Rendering 4K and 8K images, running high-end games in high definition, or any other that requires high processing capacities.

The assignment of the components to each of the categories is done using decision trees provided by the "SciKit-Learn" Python libraries, which allow classification according to their characteristics, and thus through a combinatorial algorithm generate the possible options (Combinations), taking into account the conditions necessary for the parts of the new equipment to be compatible with each other, ensuring their use without any inconvenience or mishap.

Keywords: System Recommendation, Data, Information, Predilection, Decision Trees, Machine Learning, Supervised Learning.

Actualmente adquirir un computador es común para la gran mayoría de las personas debido a sus diversas aplicaciones en el hogar, estudio, trabajo y ocio, pero su razón principal es el abaratamiento de los componentes electrónicos, lo cual hace que tener un computador sea más asequible; por ejemplo, cuando una persona desea adquirir un equipo de cómputo, si se ve desde un punto de vista de conocimiento donde el cliente está poco familiarizado con términos relacionados con la informática en el momento de escoger el mejor computador. Lo anterior trae consigo inconvenientes a la hora de adquirir un equipo de cómputo debido a las necesidades del usuario, las cuales no pueden ser solventadas a cabalidad, esto conlleva al paradigma de aquellas personas para quienes un computador barato es mejor sin saber si realmente cumple con los requisitos que posee el cliente o si entiende el funcionamiento de determinado componente.

Existe una mínima asesoría hacia el cliente por parte de los vendedores especializados, lo que hace que a éste le resulte confuso el proceso de selección y compra, adicional al cúmulo de detalles que el vendedor le pueda mencionar sobre la máquina. Por estos motivos, sería ideal presentar una solución a estos problemas, ofreciendo un sistema de información que brinde asesoramiento a los clientes en el momento de adquisición de un equipo de cómputo, mediante la implementación de un sistema recomendador capaz de catalogar dichos componentes de cómputo de acuerdo al perfil de cada cliente, y con la idea de una herramienta tecnológica que sea fácilmente accesible y entendible para cualquier usuario.

Para la creación de un sistema recomendador dentro de un sistema de información, se pensó en implementar un sistema de recomendación basado en contenido, siendo así posible ajustar la información de

cada componente a una categoría que sea de fácil entendimiento para el usuario.

## II. CONTEXTUALIZACIÓN

### A. Árboles de decisión

Soler Morejón, C., & Lombardo Vaillant, A. (2012). En apoyo al método clínico. *Revista Cubana de Medicina*, 51(1), 99-104. Resalta que los algoritmos de árboles de decisión se basan en las características de la selección de atributos útiles, cuya finalidad es la de optimizar tiempos de procesamiento.

De acuerdo a la referencia anterior donde hacen mención de los árboles de decisión, a la hora de tomar decisiones con base en información relevante, se tomó dicha funcionalidad, pero en este caso adaptándola a un contexto comercial.

### B. Manejo y usabilidad

Hassan, Y., et al. (2007) realizaron un artículo titulado “Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información”, donde determinan las características necesarias para que una aplicación web sea desarrollada de manera correcta, orientada a la interacción entre usuario y dicha aplicación.

Este artículo nos sirvió de guía en el desarrollo de la accesibilidad, usabilidad, diseño (interfaz y contenido), y la evaluación del desarrollo, obteniendo una aplicación a la medida y destinada a cualquier tipo de usuario.

Lorés, J., & i Saltiveri, T. G. (2004). *La Ingeniería de la Usabilidad y de la Accesibilidad aplicada al diseño y desarrollo de sitios web*. Universitat de Lleida. Menciona aspectos de la usabilidad del software, el aprendizaje, la efectividad y la satisfacción del usuario (p.3).

Del artículo anterior recogimos pautas para el desarrollo del aplicativo, pensando en

características como la facilidad de uso y el diseño de la interfaz.

### C. Problema de la mochila

El problema de la mochila es un problema simple de entender: hay una persona que tiene una mochila con una cierta capacidad y tiene que elegir que elementos pondrá en ella. Cada uno de los elementos tiene un peso y aporta un beneficio. El objetivo de la persona es elegir los elementos que le permitan maximizar el beneficio sin excederse de la capacidad permitida. (Tomás Bruno, 2013, p.2)<sup>1</sup>

El problema de la mochila es una parte complementaria al árbol de decisión anteriormente mencionado, cuya función es la de realizar la combinación de los componentes de cómputo previamente catalogados, implementando un algoritmo que realiza la combinación óptima y que funciona de acuerdo a un tope que desde el punto de vista del problema sería la capacidad de la mochila.

## III. OBJETIVO

Diseñar un sistema recomendador web que integre una tienda virtual de equipos de cómputo con un espacio en el cual los usuarios puedan obtener asesoría del sistema, y la presencia de distribuidores nacionales donde puedan realizar sus compras.

## IV. METODOLOGÍA

### A. Metodología Kanban

Kanban, que se define como “un sistema de producción altamente efectivo y eficiente”, ha

---

<sup>1</sup> Cita tomada de: Tomás Bruno. (2013). Problema de la mochila. Artículo de la Facultad de Ingeniería de la Universidad de Buenos Aires, [materias.fi.uba.ar/7114/Docs/ProblemaMochila.pdf](http://materias.fi.uba.ar/7114/Docs/ProblemaMochila.pdf)

contribuido a generar un panorama manufacturero óptimo y competitivo.

Laia Gilibets. Qué es la metodología Kanban y cómo utilizarla, Comunidad IEBS

Sitio

<http://comunidad.iebschool.com/iebs/general/metodologia-kanban/>

El proyecto se decidió enmarcarlo en la metodología kanban pensando en la asignación de tareas a realizar, dando una prioridad y una estimación de tiempo de acuerdo a la complejidad de cada una.

La comunicación resulta importante para el desarrollo del proyecto ya que a la hora de trabajar, los integrantes del equipo conocen de primera mano las tareas más importantes de cada miembro que lo conforman.

Kanban permite hacer mejoras en la calidad y la flexibilidad en lo que respecta al desarrollo, dado que aporta para hacer una revisión de las actividades realizadas, en curso y pendientes por hacer, logrando una mejor gestión del proyecto en sus diferentes aspectos funcionales y manteniendo control sobre los tiempos de trabajo propuestos.

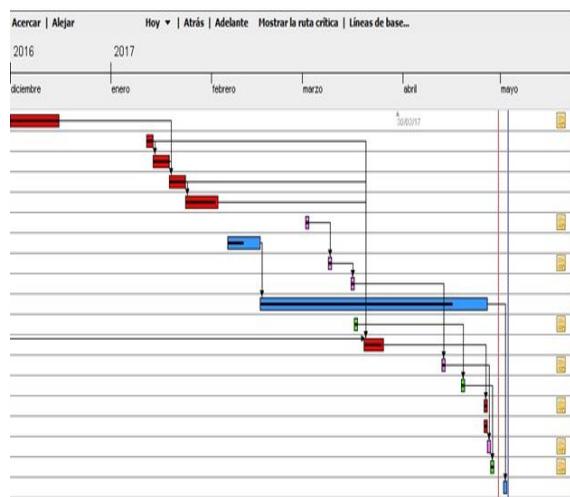


Figura 1. Diagrama de gantt implementado para el desarrollo del aplicativo Techcity

## B. Proceso de diseño y creación del árbol de decisión

Para el desarrollo del algoritmo de árboles de decisión se realizaron los siguientes pasos:

- **Recolección de datos:** Consistió en la toma de datos de las tiendas participantes del proyecto, en este caso el listado correspondiente al inventario de cada tienda.
- **Preprocesamiento de los datos:** Los datos previamente obtenidos pasan a ser convertidos en tablas para posterior catalogación.
- **Explorar los datos:** Se catalogan los atributos que se consideran relevantes organizándose en formato para su posterior análisis.

Para el sistema de recomendación se tomó el modelo de recomendación basado en contenido, el cual se enfoca en la obtención de los atributos de un objeto haciendo una comparativa con el perfil preestablecido.

Para el desarrollo del aplicativo web se utilizaron técnicas de Machine Learning haciendo énfasis en la aplicación de aprendizaje supervisado implementado en un sistema de recomendación, el cual permite la toma de decisiones con base en datos que sirven de respaldo para la máquina al momento de realizar la tarea de toma de decisiones y de aprendizaje.

### 1. Detalle del diseño del aplicativo

El sistema recomendador TechCity está diseñado para trabajar en entornos WEB, lo que permite acceder a él desde cualquier ordenador con acceso a la red, además de contar con un

inventario de productos disponibles que ofrecen los distintos proveedores

## 1. Detalle del diseño del aplicativo TechCity

<b>Identificación del requerimiento:</b>	RF01
<b>Nombre del Requerimiento:</b>	Registrar Usuarios.
<b>Características:</b>	Los usuarios deberán registrarse en el sistema para realizar alguna compra.
<b>Descripción del requerimiento:</b>	El sistema permitirá al usuario (Proveedor, cliente) registrarse. El usuario debe suministrar datos como: correo electrónico y contraseña.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>• RNF01</li> <li>• RNF02</li> <li>• RNF05</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta

Tabla 1. Requerimientos funcionales del aplicativo Techcity

<b>Identificación del requerimiento:</b>	RF02
<b>Nombre del Requerimiento:</b>	identificar Usuarios.
<b>Características:</b>	Los usuarios deberán identificarse en el sistema para realizar alguna compra.
<b>Descripción del requerimiento:</b>	El sistema requerirá que el usuario se identifique para poder realizar alguna compra.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>• RNF01</li> <li>• RNF02</li> <li>• RNF05</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta

Tabla 2. Requerimientos funcionales del aplicativo Techcity

<b>Identificación del requerimiento:</b>	RF03
<b>Nombre del Requerimiento:</b>	Carrito de compras.
<b>Características:</b>	Agregar y quitar objetos del carrito de compras.
<b>Descripción del requerimiento:</b>	Durante el proceso de compra el cliente debe contar con la capacidad de agregar y quitar productos de su lista o carrito de compras.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>• RNF01</li> <li>• RNF02</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta

Tabla 3. Requerimientos funcionales del aplicativo Techcity

<b>Identificación del requerimiento:</b>	RF04
<b>Nombre del Requerimiento:</b>	Cotización.
<b>Características:</b>	El sistema debe generar una cotización.
<b>Descripción del requerimiento:</b>	Durante la compra el sistema debe proporcionarle al cliente una posible configuración del sistema que este podría comprar utilizando las piezas disponibles basándose en el uso que se le dará y el presupuesto disponible.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>• RNF01</li> <li>• RNF02</li> </ul>
<b>Prioridad del requerimiento:</b>	Alta

Tabla 4. Requerimientos funcionales del aplicativo Techcity

<b>Identificación del requerimiento:</b>	RF05
<b>Nombre del Requerimiento:</b>	Tienda virtual.
<b>Características:</b>	Lista de todos los artículos disponibles
<b>Descripción del requerimiento:</b>	El sistema contará con una tienda virtual en la que están disponibles para la venta todos los artículos de los proveedores.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>• RNF01</li> <li>• RNF02</li> </ul>

<b>Prioridad del requerimiento:</b> Alta
---

Tabla 5. Requerimientos funcionales del aplicativo Techcity

<b>Identificación del requerimiento:</b>	RF06
<b>Nombre del Requerimiento:</b>	Agregar
<b>Características:</b>	Se podrán incluir productos nuevos
<b>Descripción del requerimiento:</b>	Permite al proveedor incluir productos nuevos en su inventario disponible para la venta
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>● RNF01</li> <li>● RNF02</li> <li>● RNF05</li> </ul>
<b>Prioridad del requerimiento:</b> Alta	

Tabla 6. Requerimientos funcionales del aplicativo Techcity

<b>Identificación del requerimiento:</b>	RF07
<b>Nombre del Requerimiento:</b>	Modificar.
<b>Características:</b>	Permite gestionar información perteneciente a los artículos en venta.
<b>Descripción del requerimiento:</b>	Permite a los distintos proveedores modificar los precios y disponibilidad de los artículos presentes en la tienda.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>● RNF01</li> <li>● RNF02</li> <li>● RNF05</li> <li>● RNF06</li> <li>● RNF07</li> </ul>
<b>Prioridad del requerimiento:</b> Alta	

Tabla 7. Requerimientos funcionales del aplicativo Techcity

<b>Identificación del requerimiento:</b>	RF08
<b>Nombre del Requerimiento:</b>	Eliminar
<b>Características:</b>	Permite eliminar artículos de la tienda.
<b>Descripción del requerimiento:</b>	El proveedor puede eliminar de la lista cualquier artículo ya sea por motivos de

	disponibilidad o pocas ventas.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>● RNF01</li> <li>● RNF02</li> <li>● RNF05</li> <li>● RNF06</li> <li>● RNF07</li> </ul>
<b>Prioridad del requerimiento:</b> Alta	

Tabla 8. Requerimientos funcionales del aplicativo Techcity

## V. RESULTADOS

Se pensó para el desarrollo en una interfaz simple que fuese entendida por cualquier usuario, donde al oprimir la opción “Arma tu PC” el sistema de información solicita requisitos como el presupuesto y cuál va a ser el perfil de uso de la máquina. Existe una opción adicional y es la de compra de los componentes por separado en caso que el usuario así lo requiera.

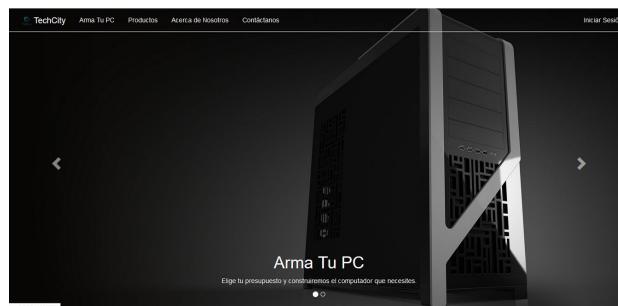


Figura 2. Interfaz inicial del aplicativo Techcity

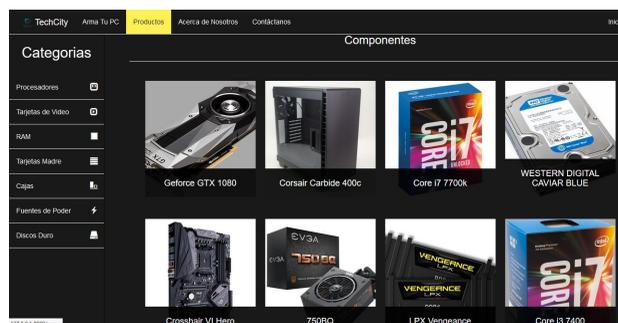


Figura 3. Vista de catálogo de componentes

El usuario al seleccionar la opción “Arma tu PC” le aparecen en la ventana dos opciones previas para muestra del equipo de cómputo, una de ellas es el presupuesto que el usuario está

dispuesto a pagar y la otra es la categoría o el tipo de uso que el cliente le piensa dar a la máquina.



Figura 4. Ingreso de condiciones al generador de combinaciones

Una vez ingresados estos dos requisitos entra en funcionamiento el algoritmo combinatorio, donde lo que hace es llamar a los componentes que se encuentran agrupados mediante una etiqueta específica, la cual corresponde al nombre de alguna de las tres categorías (Básico, Equipo de Trabajo y Entusiasta), que representan los posibles usos del computador a adquirir.

La función del algoritmo combinatorio es tomar el presupuesto como un límite, y obtener una combinación de los precios de los componentes siempre y cuando estos no superen la totalidad de 7 componentes mínimos requeridos para armar un computador. El algoritmo funciona con base en el planteamiento del “Problema de la Mochila”, donde éste algoritmo inicia por la toma de los valores menores de los componentes, y posteriormente hace una comparación entre la sumatoria de estos precios junto con el precio del presupuesto dado por el usuario.

El resultado esperado es que el algoritmo combinatorio arroja una combinación óptima que se aproxima al presupuesto asignado por el usuario.

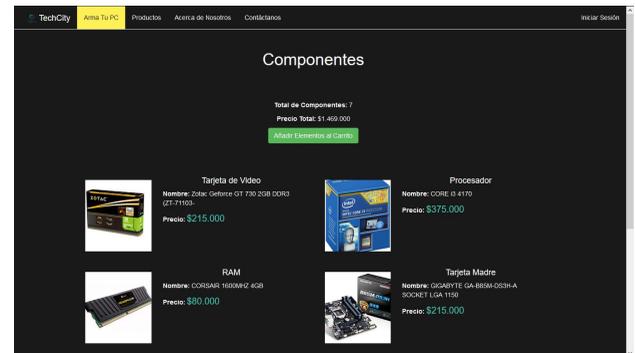


Figura 5. Resultado obtenido del generador de combinaciones

El principal algoritmo que le antecede al algoritmo combinatorio es el algoritmo de árboles de decisión. La función de este algoritmo es la de analizar la información de los componentes alojados en la base datos, para el caso, atributos de los componentes considerados de relevancia para el entrenamiento del algoritmo.

En fase de entrenamiento al árbol de decisión se le asignan etiquetas para que sea capaz de clasificar los componentes de cómputo según su grupo, posteriormente el algoritmo es capaz de hacer una comparación de las características de un nuevo componente ingresado, junto con los datos que fueron utilizados durante su fase de aprendizaje.

## VI. EVALUACIÓN

## VII. CONCLUSIONES

Se desarrolló un aplicativo web llamado TechCity, el cual funciona como un sistema de recomendación que contiene los datos de inventarios o existencias de componentes de cómputo facilitados por las tiendas comerciales Smartgamer y Tauret Computadores, más el presupuesto dado por el usuario y la categoría seleccionada, que en operación conjunta o combinada generan una recomendación de componentes para adquirir de manera óptima un equipo de cómputo acorde a las necesidades del usuario (cliente).

El sistema de información TechCity permitió evidenciar la aplicabilidad y el uso de los árboles de decisión que implementados por sí solos no serían lo suficientemente efectivos o eficientes para satisfacer las necesidades del sistema recomendador. Por ello, fue necesario emplear un algoritmo de optimización y combinación basado en el concepto del “Problema de la Mochila”, en el tratamiento a la base de datos, la cual trabaja con la agrupación de los datos de acuerdo a determinada información otorgada por los establecimientos comerciales y manejada mediante etiquetas.

Los resultados obtenidos a partir de las pruebas realizadas y la experiencia de usuario por parte de la Gerencia de la Tienda Smartgamer donde dichas pruebas se hicieron, es que la implementación de un sistema recomendador de este tipo ayuda a optimizar, en términos de tiempo, la carga de trabajo que demanda un proceso de construcción y compra de un equipo de cómputo, por lo que la solución desarrollada permite que muchos más clientes puedan hacer su proceso de consulta, cotización y compra a la hora de adquirir un computador, obteniéndose una mejora significativa en el servicio prestado.

## BIBLIOGRAFIA

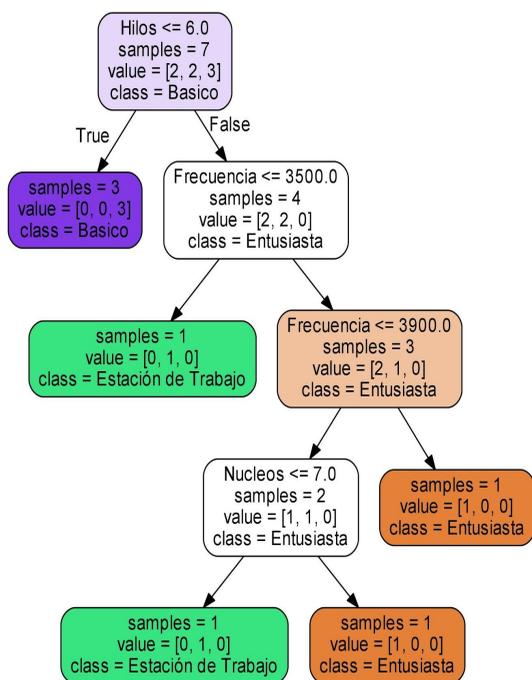


Figura 6. Ejemplo de árbol de decisión aplicado a procesadores

Dado a que en el sistema recomendador se hace uso de 7 árboles de clasificación, uno por cada componente necesario para armar un equipo de cómputo funcional, se debe tratar de manera individual cada árbol considerando que cada una de las piezas de Hardware cuenta con características muy diferentes y por ende el proceso de clasificación varía considerablemente de un árbol a otro.

Gradualmente la funcionalidad de los árboles se verá necesariamente alterada según los estándares para la construcción de un equipo de cómputo que van cambiando con el paso del tiempo, por ejemplo si actualmente un procesador de 8 núcleos puede ser calificado como un componente de gama alta; puede que en unos 5 y 6 años un procesador con la misma cantidad de núcleos pertenezca a una categoría inferior a la mencionada, es por ello que llegado a un punto del tiempo es necesario realizar un proceso de sobreentrenamiento con el fin de mitigar el efecto de entropía en el desempeño del algoritmo.

[1] Puello, J. (2011). Sistema Recomendador Adaptable al Contexto (Tesis de maestría). Universidad de Vigo, Pontevedra, España.

[2] Cabal C., Claudia C., Martínez L., Francisco J. y Molina M., Valentín (2010). Sistemas de recomendación en el Comercio Electrónico y la E-educación. Criterio Libre, 8 (12), 161-182.

[3] Richert, W., Coelho L. (2013). Building Machine Learning Systems with Python. Packt Publishing Ltd. 2(1), 7 - 48. Recuperado de <http://totoharyanto.staff.ipb.ac.id/files/2012/10/Building-Machine-Learning-Systems-with-Python-Richert-Coelho.pdf>

[4] Almaraz A., Goddard, J. (2015, enero - abril). Sistemas de recomendación, Komputer Sapiens, 7(1), Recuperado de <http://smia.mx/komputersapiens/>