

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



**DÍAZ GARCÍA EDUARD WBEIMAR**

**LOZANO GARZÓN ANDRÉS**

**UNIVERSIDAD DE CUNDINAMARCA**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA DE SISTEMAS**

**FUSAGASUGÁ**

**2017**



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

**DÍAZ GARCÍA EDUARD WBEIMAR  
LOZANO GARZÓN ANDRÉS**

**DOCUMENTO TRABAJO DE GRADO PARA OBTENER EL TÍTULO DE  
INGENIEROS DE SISTEMAS**

**DIRECTORES  
ANGELA PATRICIA ARENAS AMADO  
JOSE FERNANDO SOTELO**

**UNIVERSIDAD DE CUNDINAMARCA  
FACULTAD DE INGENIERÍA  
INGENIERÍA DE SISTEMAS  
FUSAGASUGÁ**

**2017**





## Agradecimientos

Agradecemos a la Ingeniera Ángela Patricia Arenas A. y a los Ingenieros Edgar Roa y Fernando Sotelo.

*Por su orientación, colaboración y asesoría en el desarrollo de este proyecto.*

**Eduard Wbeimar Díaz García, Andrés Lozano Garzón**

Agradezco a mis padres.

*Por mis estudios, el haberme ayudado, apoyado y por todos sus consejos en ser una mejor persona.*

A Gloria Patricia Muñoz y Edgar Eduardo Muñoz.

*Por la constante motivación para poder alcanzar mis metas en el campo académico y además ser ejemplo a seguir.*

**Eduard Wbeimar Díaz García**

Agradezco a mi padre Carlos Arturo Lozano Páez.

*Por su gran apoyo desde el inicio de mis actividades académicas, su compromiso, dedicación continua durante mis estudios y por todos los consejos que me ha dado a lo largo de mi vida.*

Agradezco a mi madre Martha Luz Garzón Pérez.

*Por su apoyo incondicional durante toda mi vida, por todos sus consejos, por la comprensión y colaboración que me ha dado siempre que he tenido alguna dificultad y por haber estado siempre junto a mí.*

Agradezco a mi familia, tías, abuelas, primos, primas y amigos.

*Que de alguna u otra manera han aportado para que pueda alcanzar esta meta.*

**Andrés Lozano Garzón**



## LISTA DE ANEXOS

|   | Pág |
|---|-----|
| <i>Anexo A-1.</i> Lista de precios de los componentes de la tienda Smartgamer.....          |     |
| 58  |     |
| <i>Anexo B-1.</i> Lista de precios de los componentes de la tienda Tauret Computadores..... | 58  |



## Tabla de contenido

|  |           |
|--|-----------|
| <b>Abreviaturas</b>                                  | <b>11</b> |
| <b>Resumen</b>                                       | <b>12</b> |
| <b>Abstract</b>                                      | <b>14</b> |
| <b>INTRODUCCIÓN</b>                                  | <b>16</b> |
| <b>1. PLANTEAMIENTO DEL PROBLEMA</b>                 | <b>18</b> |
| 1.1. Descripción del problema                        | 18        |
| <b>2. OBJETIVOS</b>                                  | <b>20</b> |
| 2.1. Objetivo General                                | 20        |
| 2.2. Objetivos Específicos                           | 20        |
| <b>3. JUSTIFICACIÓN</b>                              | <b>21</b> |
| <b>4. MARCO TEÓRICO</b>                              | <b>23</b> |
| <b>4.1. ANTECEDENTES</b>                             | <b>23</b> |
| <b>4.2. BASE TEÓRICA</b>                             | <b>25</b> |
| 4.2.1. Sistema Recomendador (SR)                     | 25        |
| 4.2.2. Sistemas de Recomendación (SR)                | 25        |
| 4.2.3. Técnicas de Sistemas de Recomendación         | 27        |
| 4.2.4. Aprendizaje basado en contenido               | 28        |
| 4.2.5. Aplicación                                    | 29        |
| 4.2.6. Aprendizaje supervisado                       | 29        |
| 4.2.7. Árboles de decisión                           | 29        |
| 4.2.8. El Problema de la Mochila                     | 30        |
| 4.2.9. Algoritmo combinatorio                        | 31        |
| <b>5. MARCO INGENIERIL</b>                           | <b>31</b> |
| <b>5.1.1. Lenguajes de programación</b>              | <b>31</b> |
| 5.1.1.1. Python                                      | 31        |
| 5.1.1.2. SQL   | 32        |
| 5.1.1.3. HTML 5                                      | 32        |
| 5.1.1.4. PHP   | 33        |
| <b>5.1.2. Sistema Gestor de Base de Datos (SGBD)</b> | <b>33</b> |
| 5.1.2.1. phpMyadmin                                  | 33        |
| <b>5.1.3. Software de desarrollo</b>                 | <b>34</b> |
| 5.1.3.1. Anaconda Continuum                          | 34        |
| 5.1.3.2. XAMPP                                       | 34        |
| 5.1.3.3. Django                                      | 35        |
| <b>5.1.4. Software de análisis</b>                   | <b>35</b> |
| 5.1.4.1. Scikit learn                                | 35        |



|   |           |
|---|-----------|
| <b>6. METODOLOGÍA</b>                               | <b>37</b> |
| <b>6.1. Localización</b>                            | <b>37</b> |
| <b>6.2. Perfiles</b>                                | <b>37</b> |
| 6.2.1. Cliente                                      | 37        |
| 6.2.2. Vendedor                                     | 37        |
| <b>6.3. Diagnóstico Inicial</b>                     | <b>38</b> |
| 6.3.1. Solución Ofrecida                            | 39        |
| <b>6.4. Metodología de Desarrollo</b>               | <b>39</b> |
| <b>6.5. Desarrollo de la herramienta</b>            | <b>40</b> |
| 6.5.1. Artefactos                                   | 40        |
| 6.5.2. Cronograma                                   | 41        |
| 6.5.3. Diseño de Base de Datos                      | 43        |
| 6.5.4. Product Backlog                              | 45        |
| 6.5.5. Desarrollo de algoritmo de árbol de decisión | 48        |
| 6.5.6. Funcionamiento del aplicativo                | 49        |
| <b>ANÁLISIS</b>                                     | <b>58</b> |
| <b>ANEXOS</b>                                       | <b>59</b> |
| <b>Conclusiones</b>                                 | <b>60</b> |
| <b>Lista de referencias</b>                         | <b>62</b> |



## LISTA DE TABLAS

|   | Pág |
|---|-----|
| <i>Tabla 1.</i> Tipos de prioridades del Product Backlog, con sus colores correspondientes..... | 48  |



### Lista de figuras

|   | Pág |
|---|-----|
| Figura 1. Esquema sobre el funcionamiento de un SR.....                 | 25  |
| Figura 2. Logo de Python.....   | 31  |
| Figura 3. Logo de SQL.....  | 32  |
| Figura 4. Logo de HTML5.....  | 32  |
| Figura 5. Logo de PHP.....  | 33  |
| Figura 6. Logo de phpMyAdmin.....                                       | 33  |
| Figura 7. Logo de Anaconda Continuum.....                               | 34  |
| Figura 8. Logo de XAMPP.....  | 34  |
| Figura 9. Logo de django framework.....                                 | 35  |
| Figura 10. Logo de la librería scikit learn.....                        | 35  |
| Figura 11. Cronograma.....  | 41  |
| Figura 12. Diagrama de Gantt.....                                       | 42  |
| Figura 13. Modelo entidad relación de la base de datos de TechCity..... | 44  |
| Figura 14. Tablero Kanban.....  | 45  |
| Figura 15. <i>Caso de uso del aplicativo</i><br><i>TechCity</i> .....   | 49  |



Figura 16. Interfaz inicial del aplicativo TechCity.....  
50

Figura 17. Vista de catálogo de componentes.....51

Figura 18. Vista de generador de  
combinaciones .....52

Figura 19. Ingreso de condiciones al generador de combinaciones..... 52

Figura 20. Resultado obtenido del generador de combinaciones..... 54

Figura 21. Ejemplo de árbol de decisión aplicado a procesadores..... 55

Figura 22. Ejemplo de árbol de decisión aplicado a tarjetas gráficas..... 56



## Abreviaturas

**API** Application Programming Interface (Interfaz de programación de aplicaciones)

**CPU** Central Processing Unit (Unidad Central de Procesamiento)

**CSV** Comma-Separated Values (Valores Separados por Coma)

**GPU** Graphics Processor Unit (Unidad Procesadora de Gráficos)

**HTML** HyperText Markup Language (Lenguaje Marcado de Hipertexto)

**IDE** Integrated Development Environment (Entorno de Desarrollo Integrado)

**MER** Modelo Entidad Relación

**PC** Personal Computer (Computadora Personal)

**PHP** Hypertext Processor (Procesador de Hipertexto)

**PSU** Power Supply Unit (Unidad de Suministro de Energía)

**RAM** Random Access Memory (Memoria de Acceso Aleatorio)

**SQL** Structured Query Language (Lenguaje de Consulta Estructurada)

**SR** Sistema de Recomendación o Sistemas Recomendadores



## 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.



### **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.



## INTRODUCCIÓN

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.

## **2. PLANTEAMIENTO DEL PROBLEMA**

### **2.1. Descripción del problema**

En Suramérica y en el territorio colombiano es muy frecuente que los usuarios que desean adquirir un computador de escritorio se encuentren con la gran variedad de precios y componentes que ofrece el mercado. De igual manera, existe una amplia gama de computadores pre ensamblados que han puesto a la venta grandes empresas como Dell, Acer, Asus, Lenovo, entre otros, lo que genera una gran confusión a la hora de escoger qué equipo comprar. Dado que los equipos ensamblados utilizando partes individuales generalmente presentan una mejor relación precio-beneficio además de brindarle al usuario la libertad para determinar la configuración ideal que permita satisfacer sus necesidades, es recomendable tener en cuenta esta opción.

Sin embargo, para ello es necesario tener conocimientos previos que le permitan al usuario construir y adquirir un equipo que se ajuste a sus necesidades. Aunque se puede encontrar esta información fácilmente en la red, actualmente en el país no se cuenta con un entorno web que le proporcione al usuario una asesoría adecuada durante el proceso de construcción y adquisición de su equipo de cómputo. Por lo tanto, el presente trabajo pretende desarrollar esta herramienta tecnológica como un apoyo ajustado a las necesidades económicas y de usabilidad



propias de cada usuario, proporcionando una tienda virtual en la que se encuentren todos los componentes necesarios para su construcción y posterior compra.

### 3. OBJETIVOS

#### 3.1. Objetivo General

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.

#### 3.2. Objetivos Específicos

- Definir todos los componentes necesarios para el funcionamiento del equipo de cómputo, incluyendo sus características, compatibilidad, precio, proveedores, etc.
- Obtener la información necesaria que permita construir un equipo de acuerdo a las necesidades del cliente.
- Identificar las posibles mejores configuraciones e integraciones entre los componentes.
- Construir el sistema recomendador web con base en la información recolectada.
- Incluir a los distribuidores locales que estén interesados en participar en el proyecto, como participantes de la tienda virtual del aplicativo.

#### 4. JUSTIFICACIÓN

En el momento en que un usuario se da a la tarea de obtener la información necesaria para construir un equipo de cómputo personalizado, es bastante frecuente encontrar foros y páginas web. Aunque éstas cuentan con una gran cantidad de información detallada y presentan una idea aproximada sobre los aspectos básicos de cada componente, por lo general, omiten aspectos claves como por ejemplo la incompatibilidad entre tarjetas madre y algunos procesadores, la necesidad de tener una fuente de poder de acuerdo a las exigencias de la tarjeta de video (GPU) en conjunto con los demás componentes, entre otros. Esta situación puede llevar a que los usuarios cometan errores cuando no cuentan con el conocimiento necesario o un asesoramiento adecuado. Adicionalmente, en los foros y páginas web no se encuentran precios concretos de estos componentes. Por tanto, se genera la necesidad de crear un aplicativo web en el que se ofrezcan las partes de los equipos de cómputo, información básica para que el usuario pueda darse una idea acertada de lo que necesita, un espacio en el que pueda obtener asesoramiento y una tienda virtual en la que el usuario pueda contactar a los proveedores de manera directa, y realizar el pedido de su producto (equipo de cómputo) basado en sus necesidades.

En el documento “Sistemas de recomendación en el Comercio Electrónico y la Educación” (Cabal *et al.*, 2010), hacen énfasis en que los sistemas de recomendación aportan de una forma significativa a los establecimientos comerciales virtuales, dado que convierten a los



usuarios que navegan por su sitios web en potenciales compradores, ofreciendo productos con base en las preferencias del individuo en cuestión y a unos cuantos clic.

De esta manera, la toma de decisiones y el acceso a información pertinente por parte de los individuos, se hace más eficiente con el apoyo de fuentes de información externa como las recomendaciones de terceros (Price & Feick, 1984; Rosen & Olshavsky, 1987).

## 5. MARCO TEÓRICO

### 5.1. ANTECEDENTES

Actualmente se encuentra una plataforma web llamada PcPartpicker, la cual brinda soporte a la hora de construir y adquirir el equipo de cómputo deseado, centrándose en Estados Unidos, Canadá y España, principalmente. Dado que la aplicación mencionada anteriormente tiene como objetivo resolver los mismos problemas que se desean solventar en el presente proyecto, resultó ideal tomarla como modelo inicial ya que la página cuenta con gran variedad de proveedores, partes disponibles en el mercado, y de acuerdo a los componentes seleccionados realiza sugerencias para maximizar el rendimiento del equipo y minimizar su costo. Además de ello, proporciona un canal de comunicación entre el usuario y el posible proveedor.

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.

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.

## 5.2. BASE TEÓRICA

### 5.2.1. Sistema Recomendador (SR)

Un sistema recomendador utiliza un filtrado cuyo funcionamiento principal es la obtención o toma de datos de un sistema de información como una base de datos, a la cual se le aplica un análisis de información, de acuerdo a los parámetros establecidos por el usuario (cliente), y diseñador (desarrollador o programador).

El proyecto está basado en conceptos orientados a las técnicas de Machine Learning para el desarrollo de sistemas de información, y más específicamente, en una de estas técnicas comúnmente conocida como árboles de decisión. Esta técnica es un método conveniente para presentar y analizar una serie de decisiones que se deben tomar en diferentes puntos del tiempo. (Kilmer, J., & Salinas, P. Árboles de decisión).

### 5.2.2. Sistemas de Recomendación (SR)



Figura 1. Esquema sobre el funcionamiento de un SR

## Importancia de los SR

“Los SR surgieron como área independiente de investigación a mediados de la década de los 90’s, aumentando drásticamente el interés sobre estos en los últimos años. Algunas de las razones por las cuales se desea explotar esta tecnología son”(Almaraz y Goddard, 2015)<sup>2</sup>:

- Incrementar su número de ventas. Probablemente es la razón más importante de los SR comerciales. Cuando el usuario está realizando una compra y le sugieren algo que le puede ser de utilidad es muy probable que el usuario también adquiera este producto incrementando así las ventas.
- Incrementar la satisfacción del usuario. Es muy importante que el SR dé recomendaciones afines al usuario para que éste se vaya contento con la recomendación y la decisión que ha tomado mediante la ayuda recibida, ya sea al comprar un producto, escuchar una canción o ver una película.
- Incrementar la fidelidad del usuario. Si el SR da una recomendación del interés del usuario y éste queda satisfecho, seguramente regresará a utilizar este sistema para futuras consultas.
- Comprender mejor lo que el usuario quiere. Un SR va aprendiendo a través de la actividad del usuario, por lo cual puede ofrecer mejores recomendaciones, ya que conoce un poco más los gustos del usuario.

---

<sup>2</sup> Almaraz A., Goddard, J. (2015, enero - abril). Sistemas de recomendación, *Komputer Sapiens*, 7(1), 12-13.

## Aplicaciones de los SR

- Entretenimiento: Proveedores de contenido como música, películas y videos.
- Contenido: Motores de búsqueda, sitios web de noticias, filtros de correo electrónico.
- Comercio electrónico: Recomendaciones al consumidor en sitios de compras virtuales.
- Servicio: Servicios de viajes.

### 5.2.3. Técnicas de Sistemas de Recomendación

El aprendizaje automático o Machine Learning es una técnica de la computación que ofrece una amplia gama de aplicaciones, en donde la finalidad principal está orientada a que la máquina adquiera cierto nivel de autonomía al momento de realizar tareas específicas.

- **Aprendizaje colaborativo:** Siendo la técnica más utilizada, esta técnica de aprendizaje se caracteriza por su funcionamiento con base en las calificaciones que proporcionan los usuarios.
- **Basado en contenido:** Esta técnica se caracteriza en tomar en cuenta las características de un producto y hacer una comparación con el perfil de un usuario, mostrando a modo de recomendación productos alternos que tengan características similares.

- **Basado en demografía:** Esta técnica se basa en el suministro de la información al usuario de acuerdo a características como: Edad, género, localidad, lenguaje, entre otros.
- **Basado en conocimiento:** Se basa en la recolección de reseñas que han aportado los usuarios acerca de un producto, servicio, contenido, etc.
- **Mixto (híbrido):** Esta técnica simplemente puede hacer uso de cualquiera de las anteriores técnicas ya mencionadas.

#### 5.2.3.1. Aprendizaje basado en contenido

Recientemente las técnicas de recomendación se han comenzado a utilizar en el ámbito académico, y en particular en el contexto del aprendizaje. Coincidimos con los trabajos descritos a continuación, en que el ámbito académico impone unos requisitos específicos en el proceso de recomendación. (Ruiz-Iniesta, Jiménez-Díaz y Gómez-Albarrán, 2010). Personalización en Recomendadores Basados en Contenido y su Aplicación a Repositorios de Objetos de Aprendizaje. *IEEE-RITA*, 5(1), 31-38.)

## **5.2.4. Aplicación**

### **5.2.4.1. Aprendizaje supervisado**

“En los problemas de aprendizaje supervisado se enseña o entrena al algoritmo a partir de datos que ya vienen etiquetados con la respuesta correcta. Cuanto mayor es el conjunto de datos, más el algoritmo puede aprender sobre el tema. Una vez concluido el entrenamiento, se le brindan nuevos datos, ya sin las etiquetas de las respuestas correctas, y el algoritmo de aprendizaje utiliza la experiencia pasada que adquirió durante la etapa de entrenamiento para predecir un resultado. Esto es similar al método de aprendizaje que se utiliza en las escuelas, donde se nos enseña problemas y las formas de resolverlos, para que luego podamos aplicar los mismos métodos en situaciones similares”. Raul E. Lopez Briega. Obtenida 10 Octubre 2015, de <http://relopezbriega.github.io/blog/2015/10/10/machine-learning-con-python/>

Para nuestro proyecto se hizo implementación de aprendizaje supervisado dado que resultaba conveniente manejar la información mediante el uso de etiquetas, a su vez para el uso de algoritmo de árboles de decisión, siendo este fácil de interpretar a la hora de obtener resultados.

### **5.2.5. Árboles de decisión**

Los árboles de decisión son una de las opciones más utilizadas para el razonamiento y el aprendizaje basado en ejemplos, para tratar con el lenguaje natural y la incertidumbre. Han

sufrido una serie de alteraciones a lo largo del tiempo a fin de mejorar su capacidad de representación e inferencia (Janikow, C.Z. ,1998)<sup>3</sup>.

Para el desarrollo del aplicativo se implementó un método de recomendación llamado árbol de decisión, conocido como una técnica de machine learning, el cual tiene dos enfoques: El primero es de agrupación y el segundo de predicción.

Para este proyecto se utilizaron árboles de decisión enfocados hacia la predicción y asignación de categorías a los componentes de cómputo, estos al ser almacenados en una base de datos el algoritmo del árbol de decisión analiza determinados atributos de estos, y por consiguiente su clasificación.

#### **5.2.6. El 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>4</sup>

---

<sup>3</sup> Janikow, C.Z. (1998) Fuzzy decision trees: Issues and methods. IEEE Trans. Syst., Man, Cybern., vol. 28, pp. 1–14.

<sup>4</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)

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.

#### **5.2.6.1. Algoritmo combinatorio**

Se utilizó un algoritmo que permite obtener la combinación óptima en función de obtener prioridad de los precios de componentes, siendo este algoritmo una parte complementaria para el algoritmo del árbol de decisión, y manteniendo una funcionalidad correspondiente al problema de la mochila.

### **5.3. MARCO INGENIERIL**

#### **5.3.1. Lenguajes de programación**

##### **5.3.1.1. Python**



Lenguaje interpretativo cuyo enfoque se basa en la creación de una sintaxis en la que da como resultado un código legible, siendo un lenguaje de programación de múltiples paradigmas incluyendo la programación orientada a objetos y la programación imperativa, además de ser un lenguaje flexible y multiparadigma.

*Figura 2.* Logo de Python. Fuente:

<https://www.paradigmadigital.com/wp-content/uploads/2016/07/python-logo.png>

Este lenguaje fue utilizado para la creación de los árboles de decisión, que son la parte central del desarrollo del aplicativo al igual que el framework llamado django.

#### 5.3.1.2. SQL



Por sus siglas Structured Query Language (Lenguaje de Consulta Estructurada) es el lenguaje estándar para manipulación y acceso a bases de datos, mediante consultas desde un gestor de bases de datos o desde una aplicación que esté enlazada a dicha base de datos.

*Figura 3.* Logo de SQL. Fuente:

<https://azure.microsoft.com/svghandler/sql-server-stretch-database/?width=600&height=315>

#### 5.3.1.3. HTML 5



Lenguaje marcado de hipertexto, empleado para el diseño de interfaces en aplicaciones.

*Figura 4.* Logo de HTML5. Fuente:

[https://www.w3.org/html/logo/downloads/HTML5\\_Logo\\_512.png](https://www.w3.org/html/logo/downloads/HTML5_Logo_512.png)

Este lenguaje nos permitió realizar un sitio web fácilmente adaptable a diversos frameworks.

#### 5.3.1.4. PHP



Lenguaje de programación para el procesamiento de hipertexto incrustado al lenguaje de programación de HTML, el cual se caracteriza por ser manejado del lado del servidor, como por ejemplo, el manejo de bases de datos o de conexiones de red.

*Figura 5.* Logo de PHP.

Fuente:<https://upload.wikimedia.org/wikipedia/commons/thumb/2/27/PHP-logo.svg/1200px-PHP-logo.svg.png>

#### 5.3.2. Sistema Gestor de Base de Datos (SGBD)

##### 5.3.2.1. phpMyAdmin



Herramienta gráfica para la gestión de bases de datos basada en MySQL, manejada mediante el uso de instrucciones del lenguaje de programación PHP.

*Figura 6.* Logo de phpMyAdmin.

Fuente:<http://www.elchecibernetico.com/wp-content/uploads/2015/12/phpmyadmin.png>

Mediante el uso de esta herramienta se hizo la gestión de la base de datos de la aplicación.

### 5.3.3. Software de desarrollo

#### 5.3.3.1. Anaconda Continuum



Conjunto de aplicaciones destinado al análisis de datos, compuesto por grupos de programas y librerías ya listas para el uso de análisis en Machine Learning. Anaconda Continuum es un conjunto de diversas aplicaciones y librerías relacionadas con temas de Machine Learning que nos permitió realizar la instalación del framework django con suma facilidad.

*Figura 7.* Logo de Anaconda Continuum. Fuente:

[https://www.continuum.io/sites/default/files/Anaconda\\_Logo\\_0702\\_0.png](https://www.continuum.io/sites/default/files/Anaconda_Logo_0702_0.png)

#### 5.3.3.2. XAMPP



# XAMPP

Paquete de instalación basado en software libre que contiene herramientas de gestión de bases de datos como MySQL, gestor de páginas web Apache e intérprete de instrucciones PHP.

*Figura 8.* Logo de XAMPP.

Fuente: <http://managewp.com/wp-content/uploads/2012/08/xampp-logo.png>

### 5.3.3.3. Django



Framework escrito en python para el desarrollo de aplicaciones web, cuyo objetivo es facilitar la creación de sitios web complejos.

*Figura 9.* Logo de django framework.

Fuente:[https://upload.wikimedia.org/wikipedia/commons/thumb/7/75/Django\\_logo.svg/1200px-Django\\_logo.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/7/75/Django_logo.svg/1200px-Django_logo.svg.png)

Django es un framework escrito en python que permite la implementación del modelo Vista-Controlador, facilitando la gestión de formularios de sitios web mediante una interfaz gráfica que este incorpora.

### 5.3.4. Software de análisis

#### 5.3.4.1. Scikit learn



Scikit learn es una biblioteca con librerías destinadas para el uso de aprendizaje de máquina, basada en el lenguaje de programación python.

*Figura 10.* Logo de la librería scikit learn.

Fuente:<https://avatars2.githubusercontent.com/u/365630?v=3&s=400>



Utilizamos scikit-learn por su enfoque al análisis de datos y técnicas de Machine Learning, base fundamental para la creación del árbol de decisión.

## **6. METODOLOGÍA**

### **6.1. Localización**

El proyecto se inició en el mes de noviembre de 2016 con la búsqueda de establecimientos interesados en participar. De todos los visitados encontramos que las compañías Smartgamer y Tauret Computadores, ubicadas en el Centro de Alta Tecnología y en el Centro Comercial Unilago de la ciudad de Bogotá, manifestaron su interés en participar y nos facilitaron información referente a listas de precios de los componentes comercializados por las tiendas.

### **6.2. Perfiles**

#### **6.2.1. Cliente**

El cliente habitual que va a un establecimiento comercial dedicado a la comercialización de equipos de cómputo, a menudo no tiene una idea clara de cuáles son las necesidades para comprar un computador, por lo que espera que el vendedor le aconseje durante el proceso de compra de un equipo de cómputo.

#### **6.2.2. Vendedor**

El vendedor común espera al cliente que por lo general es indeciso, le muestra varios componentes de cómputo con ciertas especificaciones o características técnicas, pero en algunos

casos el cliente no cuenta con el presupuesto, y el vendedor debe empezar de nuevo el proceso de asesoramiento teniendo en cuenta la restricción económica manifestada por el cliente.

### **6.3. Diagnóstico Inicial**

Los establecimientos comerciales dedicados a la venta de equipos y componentes de cómputo ubicados en Unilago y en el Centro de Alta Tecnología de la ciudad de Bogotá, pretenden ofrecer sus productos al mejor precio posible y dentro de los horarios de atención establecidos.

Otra situación que también se presenta es que el cliente comúnmente no se familiariza con detalles técnicos de las características de los componentes que conforman un computador, y por consiguiente, en algunos casos surge cierta insatisfacción por parte del cliente dado que la compra de su computador no fue la esperada, o el gasto excesivo de dinero por la adquisición cuando sus necesidades reales eran menores.

Situaciones como las presentadas anteriormente nos permitieron argumentar ante los establecimientos interesados la necesidad de contar con su apoyo, tomar sus nombres e información base requerida para el desarrollo del aplicativo TechCity, como idea de tienda online y explicando las intenciones del proyecto.

Al final del proceso de diagnóstico dos tiendas nos proporcionaron el aval de usar sus nombres como interesados en apoyar el desarrollo del proyecto, y como constancia de ello entregamos las respectivas cartas explicando nuestros objetivos con el presente proyecto.

### **6.3.1. Solución Ofrecida**

La solución que se encontró para el cliente fue el desarrollo de un aplicativo web, el cual contiene árboles de decisión que trabajan con etiquetas junto con un algoritmo combinatorio, el cual analiza la información que se encuentra en la base de datos, dando como resultado la selección más óptima de acuerdo a las necesidades del usuario previamente ingresadas.

## **6.4. Metodología de Desarrollo**

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

## **6.5. Desarrollo de la herramienta**

### **6.5.1. Artefactos**

Para hacer un registro más eficiente y un seguimiento de cada una de las actividades a realizar se utilizaron elementos y métodos por parte de Kanban, los cuales se mencionan a continuación:

- **Tablero Kanban:** El tablero Kanban permite hacer un seguimiento de las actividades relacionadas con el proyecto (pendientes, en curso y ya realizadas).

- **Diagrama de Gantt:** Permite generar un esquema de actividades relacionadas a un tiempo específico de desarrollo de un proyecto.

### 6.5.2. Cronograma

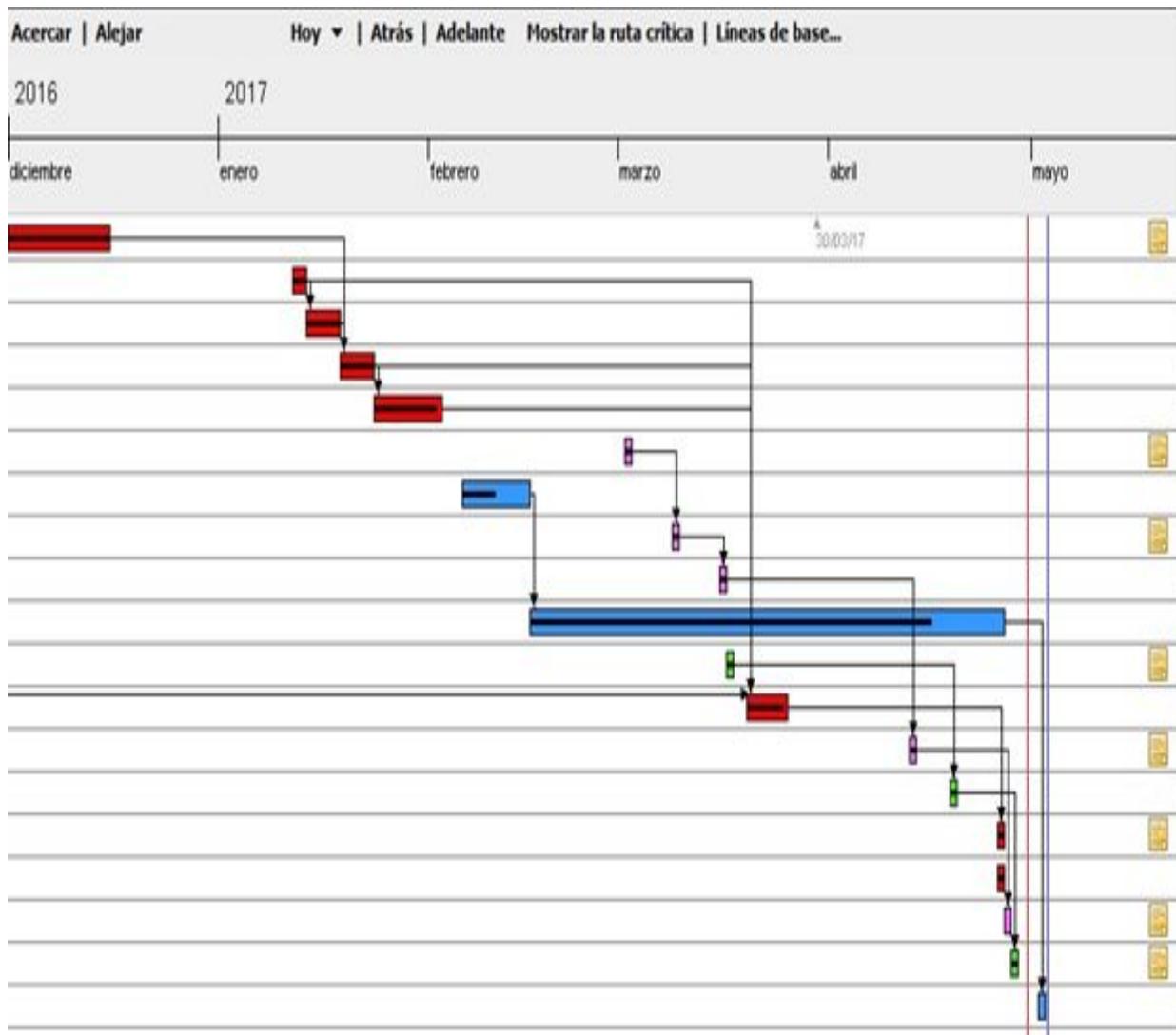


Figura 11. Cronograma



Figura 12. Diagrama de Gantt



### **6.5.3. Diseño de Base de Datos**

Posterior a la toma de los datos y a su debida organización, en este caso, la lista de precios de los productos del inventario que ofrecen las tiendas Smartgamer y Tauret Computadores, se procede al diseño y creación de la base de datos para el sistema de información TechCity con el gestor MySQL, siendo este el ideal para la integración entre los demás componentes que son HTML5, PHP y el framework django.

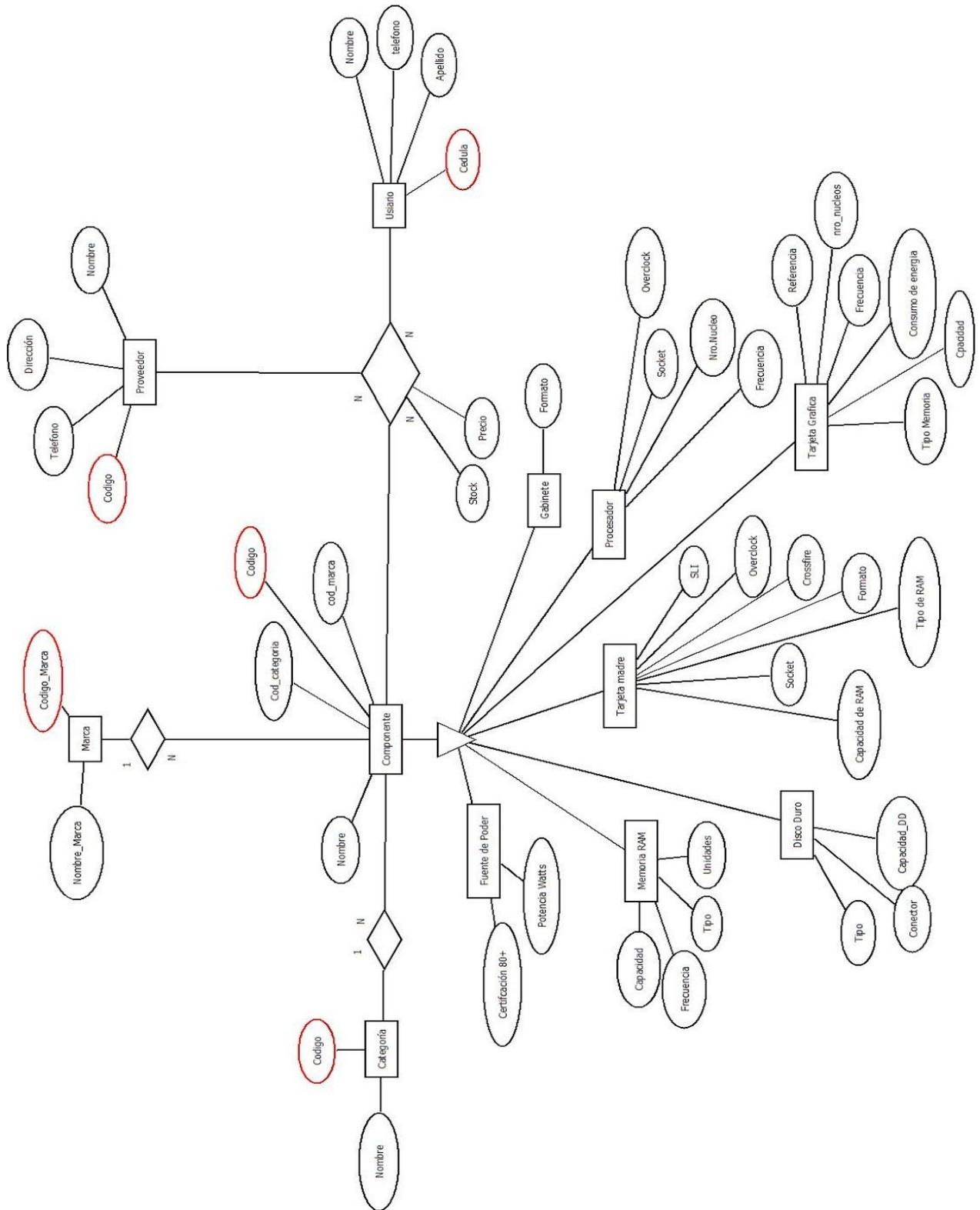


Figura 13. Modelo entidad relación de la base de datos de TechCity.

### 6.5.4. Product Backlog

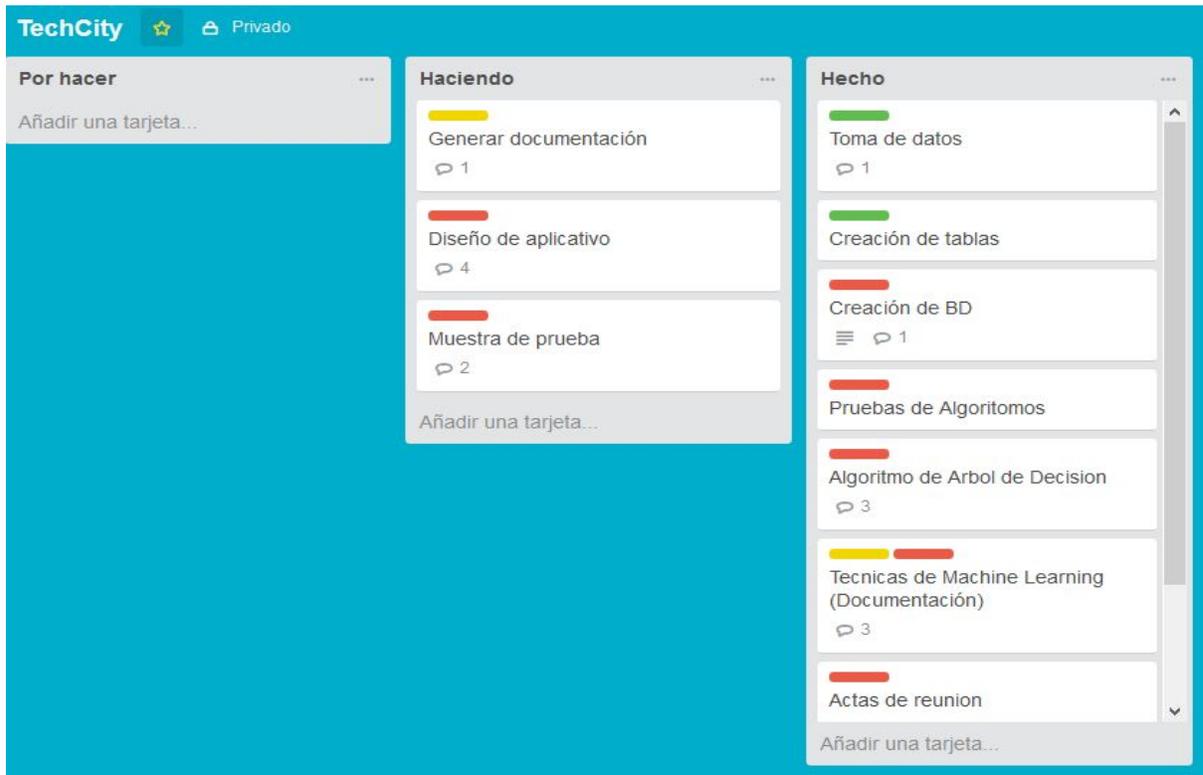


Figura 14. Tablero Kanban

Un tablero Kanban es una de las herramientas más populares para aumentar la productividad. Ninguno de los métodos de gestión de proyectos existentes es a la vez tan fácil de usar y tan eficaz como el método Kanban.

Los tableros Kanban se han hecho populares entre muchas empresas de diferentes industrias del mundo entero para mejorar la transparencia, la productividad y la eficiencia de su organización.

La metodología Kanban está ganando gran popularidad en corporaciones y empresas de todo el mundo como una manera de gestionar el trabajo de forma fluida. Proveniente de Japón, Kanban es un símbolo visual que se utiliza para desencadenar una acción. A menudo se representa en un tablero Kanban para reflejar los procesos de su flujo de trabajo.

Kanban, representada por una tarjeta Kanban, se mueve a través de las diversas etapas del trabajo hasta su finalización. A menudo, se habla de éste como un método de extracción, de forma que “usted” tira de sus tareas a través de su flujo de trabajo, ya que permite a los usuarios mover de sitio libremente las tareas en un entorno de trabajo.

### **¿Cómo funciona Kanban?**

Existe una serie de principios básicos con el fin de obtener el máximo rendimiento de su flujo de trabajo.

**Visualice lo que hace (su flujo de trabajo):** Una visualización de todas las tareas y elementos en una tabla, contribuye a que todos los miembros del equipo se mantengan al corriente con su trabajo.

**Limite la cantidad de Trabajo en Proceso (límites del TEP):** Se establecen metas asequibles. Se mantiene el equilibrio del flujo de trabajo mediante la limitación de los trabajos en proceso, para prevenir el exceso de compromiso en la cantidad de tareas que son difíciles de terminar.

**Realice un seguimiento de su tiempo:** El seguimiento del tiempo confluye con la metodología Kanban. Se realiza un seguimiento del tiempo de forma continua y se evalúa el trabajo con precisión.

**Lectura fácil de indicadores visuales:** Se conoce lo que está ocurriendo de un solo vistazo, haciendo uso de tarjetas de colores para distinguir los Tipos de trabajo, Prioridades, Etiquetas, Fechas límite, entre otros.

**Identifique los cuellos de botella y elimine lo que resulta descartable:** Se aprovecha al máximo los plazos y ciclos de ejecución, del Flujo Acumulativo y de los informes de tiempo. Estos criterios permiten evaluar el rendimiento, detectar los problemas y ajustar el flujo de trabajo en consecuencia.

Incluso hasta el método Kanban más básico produce un aumento en el rendimiento. Una distribución simple de las tareas, sumado a una monitorización del flujo de trabajo y el realizar los ajustes apropiados a lo largo del proceso incrementan la eficiencia. A manera de caso de éxito, un equipo de desarrollo de software contratado por la BBC Worldwide London ha experimentado unas mejoras formidables en el transcurso de 12 meses, luego de la implementación de Kanban. Su tiempo de entrega se ha visto reducido en un 37% y la consistencia en la entrega ha repuntado un 47%. Este no es un caso aislado, numerosos equipos que han implementado el método Kanban han informado de una mejora extraordinaria en sus plazos de ejecución y tiempos de entrega.

Para el desarrollo del proyecto fue necesario establecer objetivos claves en el proceso de creación del aplicativo TechCity; los cuales, fueron analizados posteriormente con el fin de establecer su prioridad en cuanto al funcionamiento dentro de este.

Para cada objetivo se pensó en agregar una etiqueta de color para determinar la prioridad:

*Tabla 1.* Tipos de prioridades del Product Backlog, con sus colores correspondientes

| <i>Tipo de Prioridades</i> | <i>Color</i>    |
|----------------------------|-----------------|
| <i>Baja</i>                | <i>Verde</i>    |
| <i>Media</i>               | <i>Amarillo</i> |
| <i>Alta</i>                | <i>Rojo</i>     |

De la anterior tabla se planteó la utilización de etiquetas de colores, dichas etiquetas como se observa en la *Figura 14*, fueron utilizados para delimitar la prioridad de la tarea en el tablero Kanban.

#### 6.5.5. Desarrollo de algoritmo de 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.

#### 6.5.6. Funcionamiento del aplicativo

El sistema de información funciona con base en la información obtenida de la lista de catálogos y precios que dispone cada establecimiento comercial, para ello la información almacenada en la base de datos es analizada por un algoritmo encargado de su clasificación, y es aquí en donde entra en funcionamiento el sistema de recomendación.

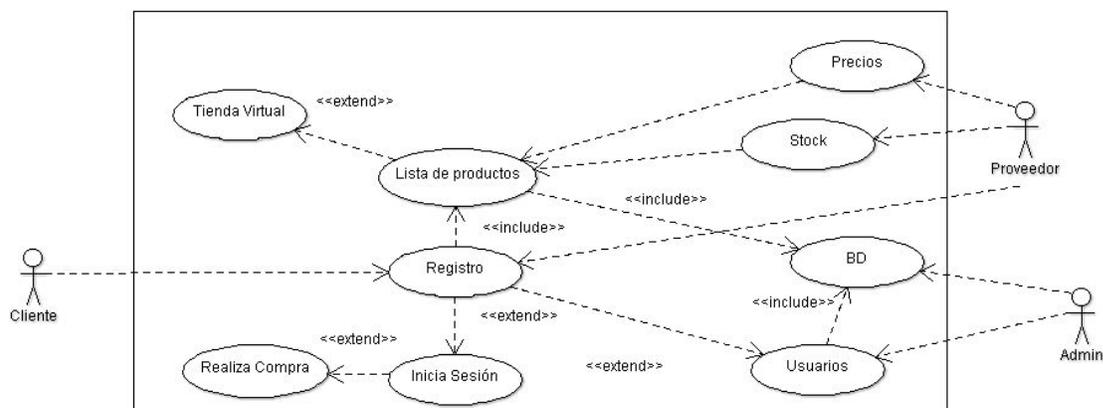


Figura 15. Caso de uso del aplicativo TechCity

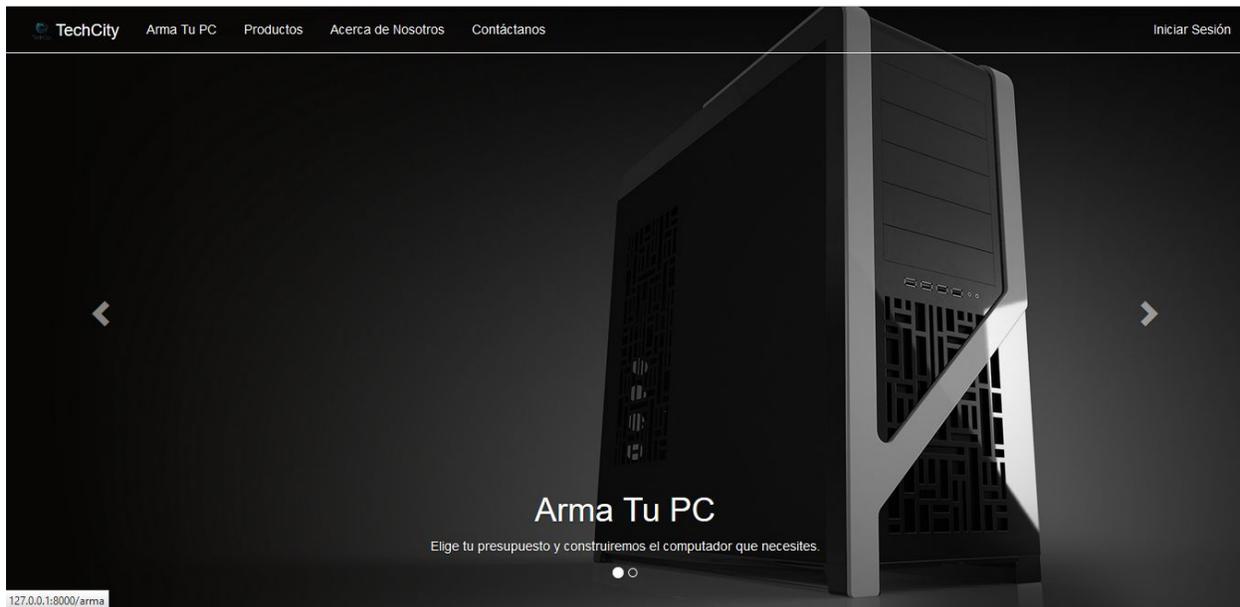


Figura 16. Interfaz inicial del aplicativo TechCity

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.

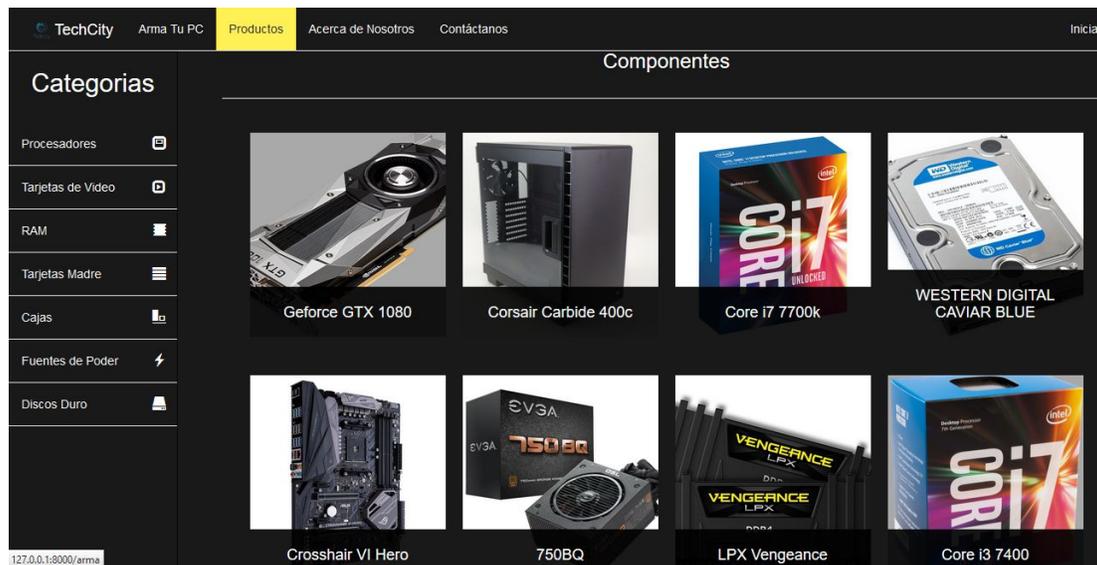


Figura 17. Vista de catálogo de componentes

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.

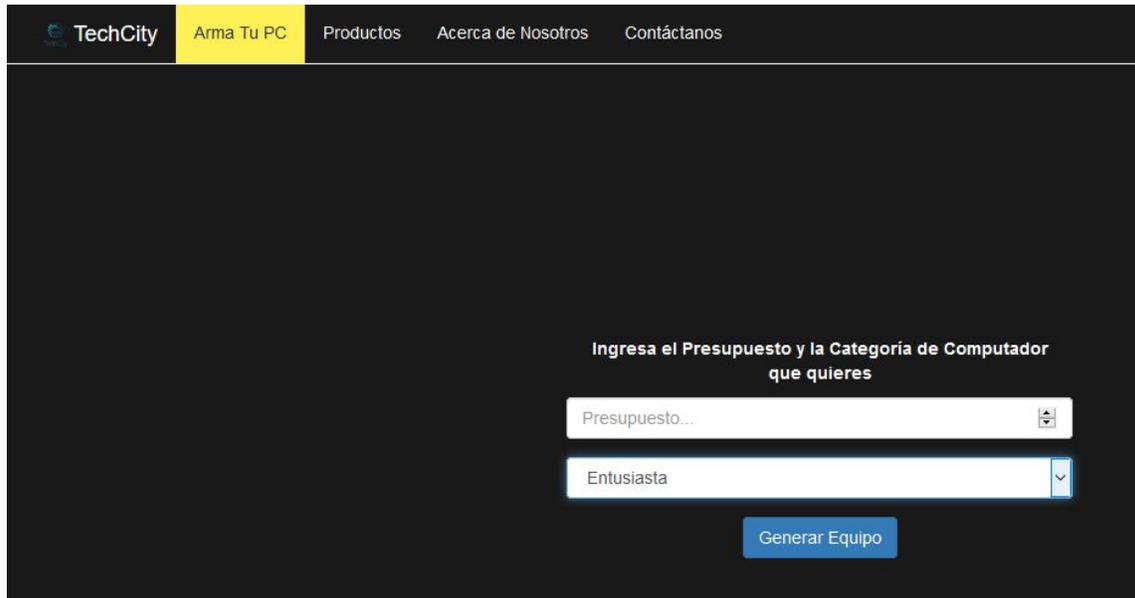


Figura 18. Vista de generador de combinaciones

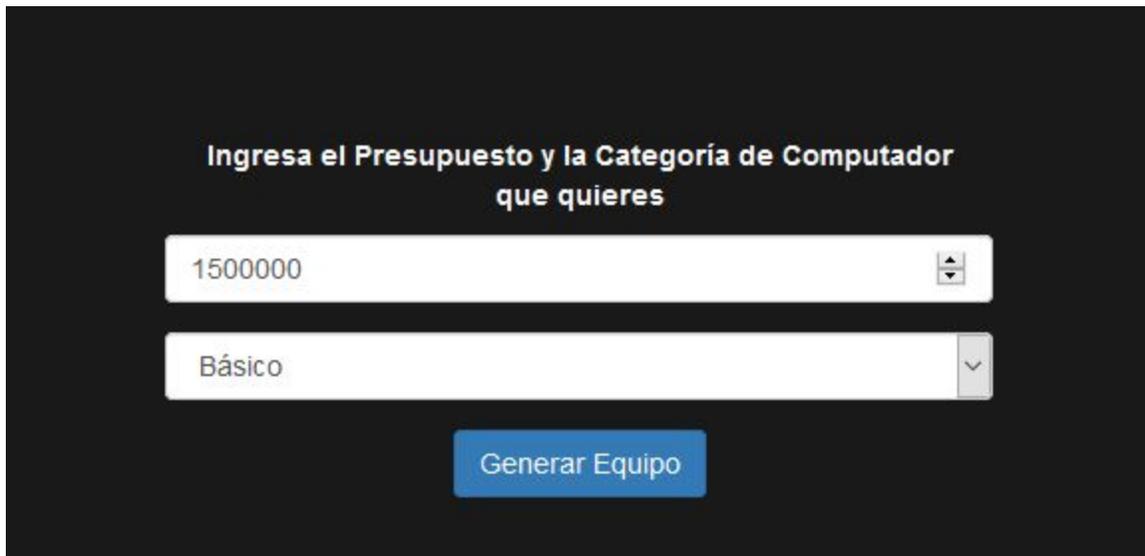


Figura 19. Ingreso de condiciones al generador de combinaciones

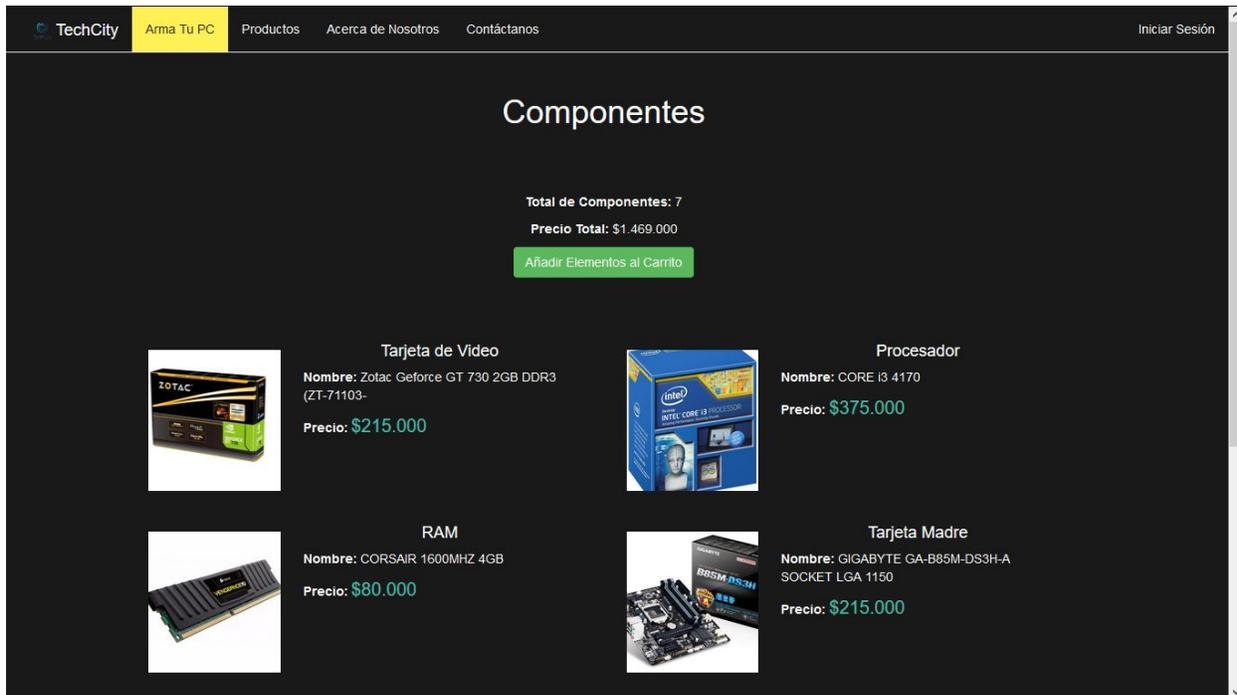
Una vez ingresados estos dos requisitos entra en funcionamiento el algoritmo combinatorio, donde lo que hace es llamar a los componentes que se encuentran agrupados



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.



The screenshot shows a web interface for 'TechCity' with a navigation bar including 'Arma Tu PC', 'Productos', 'Acerca de Nosotros', and 'Contáctanos'. The main content area is titled 'Componentes' and displays a summary: 'Total de Componentes: 7' and 'Precio Total: \$1.469.000'. A green button labeled 'Añadir Elementos al Carrito' is visible. Below the summary, four components are listed in a grid:

| Componente       | Nombre                                    | Precio    |
|------------------|---|-----------|
| Tarjeta de Video | Zotac Geforce GT 730 2GB DDR3 (ZT-71103-) | \$215.000 |
| Procesador       | INTEL CORE I3 PROCESSOR (CORE I3 4170)    | \$375.000 |
| RAM              | CORSAIR 1600MHZ 4GB                       | \$80.000  |
| Tarjeta Madre    | GIGABYTE GA-B85M-DS3H-A SOCKET LGA 1150   | \$215.000 |

Figura 20. 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.

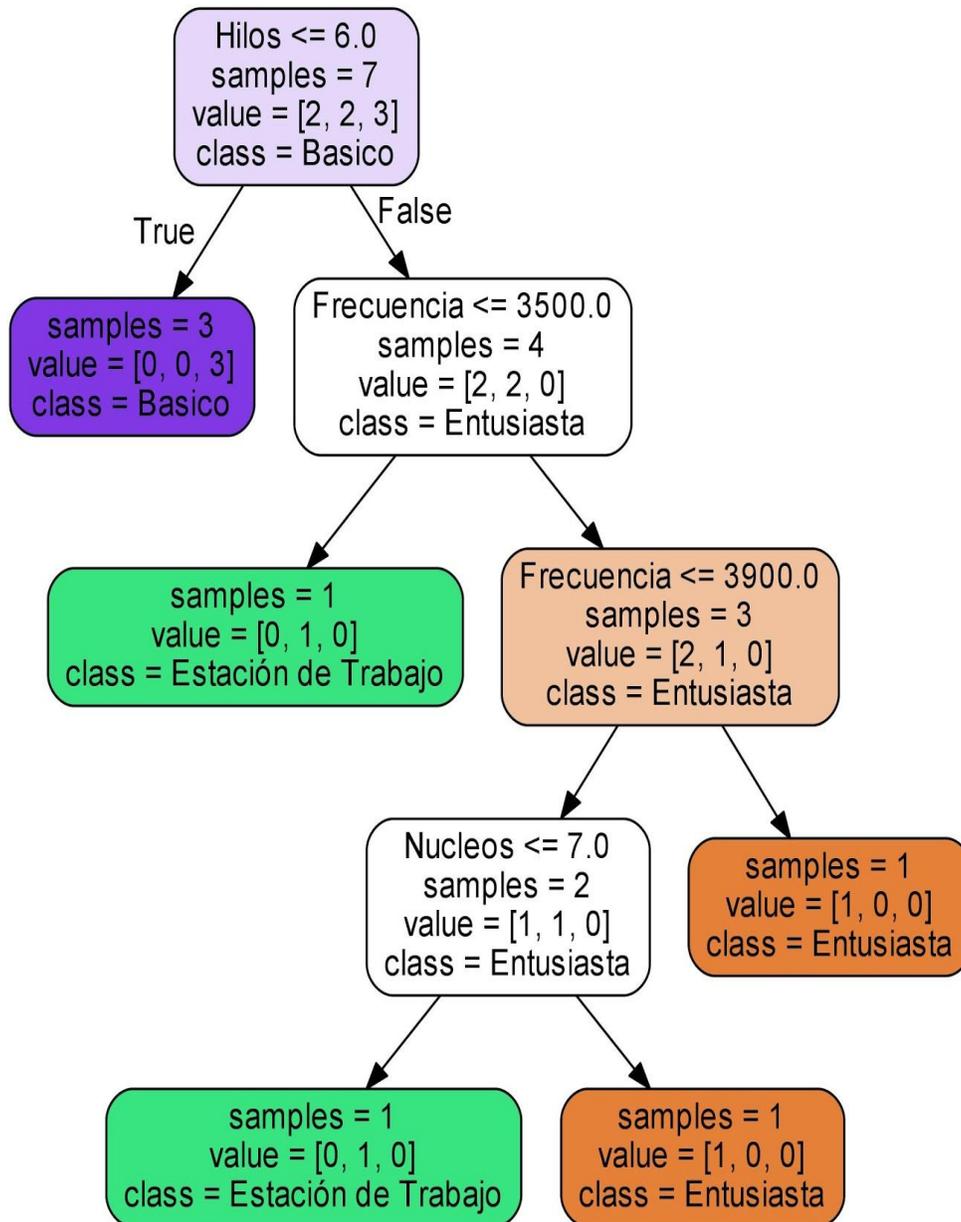


Figura 21. Ejemplo de árbol de decisión aplicado a procesadores

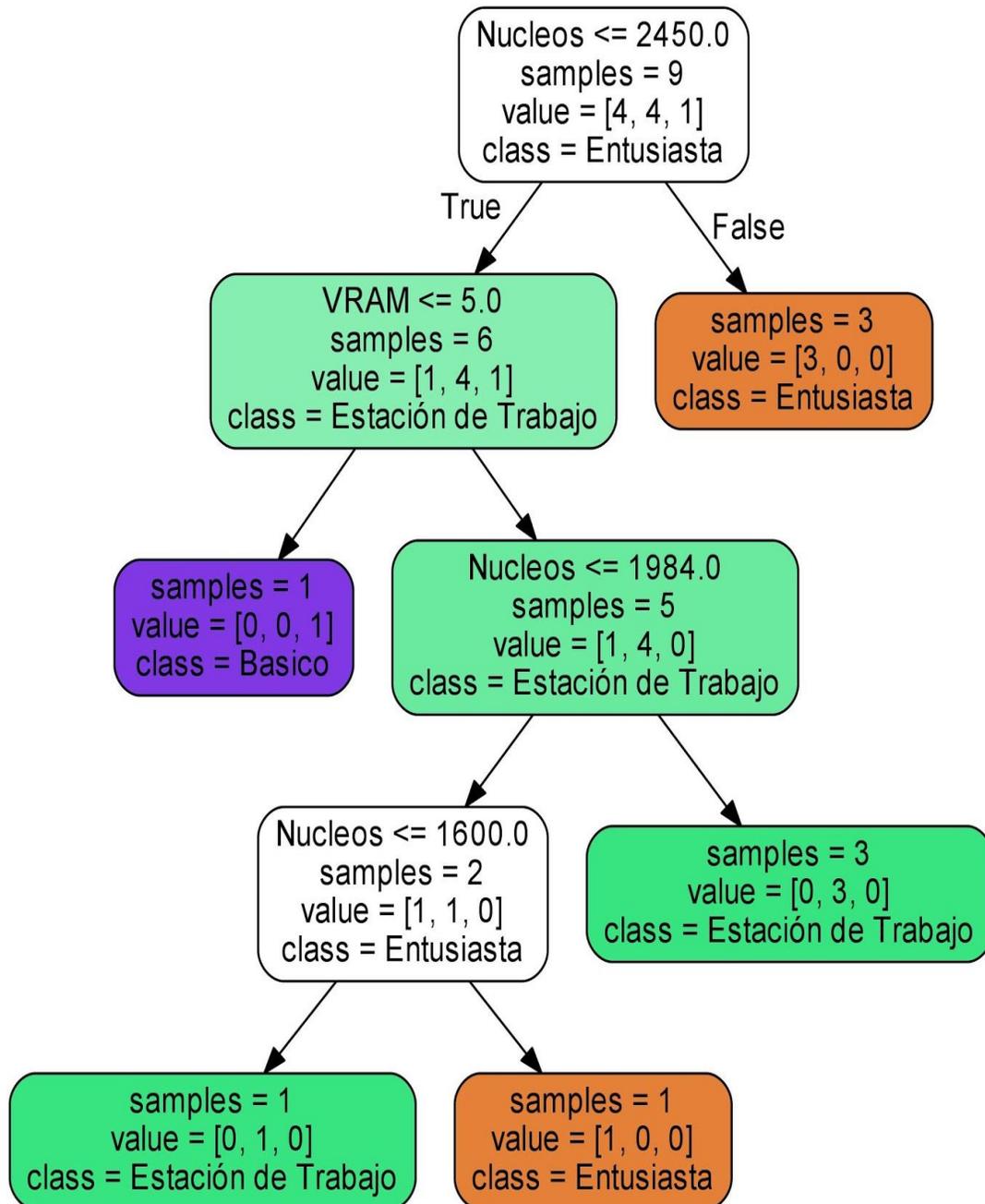


Figura 22. Ejemplo de árbol de decisión aplicado a tarjetas gráficas

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 computo 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.

Sin embargo resulta fundamental eliminar los componentes que ya no estén presentes en los inventarios, con el objetivo de realizar la poda de los nodos que se pudieron haber creado teniendo en cuenta dichos componentes, con el objetivo de minimizar la creación de posibles reglas o nodos que generen resultados incoherentes, dado que los nuevos componentes modificarán la estructura del árbol con su nuevo ingreso, este proceso se realizará en todos los árboles de clasificación, pero la frecuencia variará de unos a otros teniendo en cuenta las características de los elementos de hardware.

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 núcleo 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.



## ANÁLISIS

El sistema de recomendación facilita el proceso de selección de componentes de cómputo debido a que maneja datos reales de dos tiendas comerciales, las cuales presentan una opción que se adapta a las necesidades y presupuesto del usuario, quien finalmente decide comprar basado en la opción ideal que le proporciona el sistema en términos de presupuesto y de los componentes que puede comprar, junto con las referencias que también puede encontrar en el aplicativo.

El aplicativo Web funciona de manera local por tanto está limitado sólo a pruebas por medio de simulación con usuarios ficticios o personas que conocen cómo seleccionar elementos de cómputo. Lo anterior, para corroborar el funcionamiento de los árboles de decisión en conjunto con el algoritmo combinatorio, que estén bien las etiquetas con la base de datos, la cual debe ser actualizada con el inventario de la tienda.



## ANEXOS

Basado en los datos anteriores, el siguiente vínculo conduce a la lista de precios de productos de la tienda Smartgamer.

**[http://smartgamer.co/wp-content/uploads/descripciones/publico/precios.pdf?utm\\_source=sgw&utm\\_medium=side&utm\\_campaign=lprecios](http://smartgamer.co/wp-content/uploads/descripciones/publico/precios.pdf?utm_source=sgw&utm_medium=side&utm_campaign=lprecios)**

*Anexo A-1.* Lista de precios de los componentes de la tienda Smartgamer

Basado en los datos anteriores, el siguiente vínculo conduce a la lista de precios de productos de la tienda Tauret Computadores.

**<https://tauretcomputadores.com/lista-de-precios.php>**

*Anexo B-1.* Lista de precios de los componentes de la tienda Tauret Computadores

## 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 el servicio prestado.



### Lista de referencias

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

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.

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>

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