ISSN 2531-2197

Volumen I, Número 2 — Julio — Septiembre - 2017

Revista de Tecnología

Informática

ECORFAN®

ECORFAN-Spain

Indización

Google Scholar

Research Gate

REBID

Mendeley

Revista de Tecnología Informática, Volumen 1, Número 2, de Julio a Septiembre - 2017, es una revista editada trimestralmente por ECORFAN-Spain. Calle Matacerquillas 38, CP: 28411. Moralzarzal - Madrid. WEB: www.ecorfan.org/spain, revista@ecorfan.org. Editora en Jefe: RAMOS– ESCAMILLA, María. Co-Editor: MIRANDA-GARCÍA, Marta. PhD. ISSN-2531-2197. Responsables de la última actualización de este número de la Unidad de Informática ECORFAN. ESCAMILLA-BOUCHÁN, Imelda, LUNA-SOTO, Vladimir, actualizado al 30 de Septiembre 2017.

Presentación

ECORFAN, es una revista de investigación que pública artículos en el área de: Tecnología Informática

En Pro de la Investigación, Docencia, y Formación de los recursos humanos comprometidos con la Ciencia. El contenido de los artículos y opiniones que aparecen en cada número son de los autores y no necesariamente la opinión del Editor en Jefe.

El artículo *Reconocimiento de patrones en gráficos de control utilizando una red neuronal* por GUARNEROS-RIVERA, Manuel, DÍAZ, LÓPEZ-CHAU, Asdrúbal, MUÑOZ-CONTRERAS, Hilarion y PELÁEZ-CAMARENA, Silvestre Gustavo con adscripción en el Instituto Tecnológico de Orizaba, como siguiente artículo está *Estudio del comportamiento de algoritmos de clasificación según la naturaleza de los datos* por A CRUZ-GUERRERO, René, ALONSO-LAVERNIA, Ma. de los Ángeles, FRANCO-ARCEGA, Anilú, SIMÓN-MARMOLEJO, Isaías con adscripción en la Universidad Autónoma del Estado de Hidalgo y el Instituto Tecnológico Superior del Oriente del Estado de Hidalgo, como siguiente artículo está *Sistema de apoyo para la detección de entropía económica en municipios vulnerables* por CONTRERAS-Meliza, BELLO, Pedro, CERVANTES, Ana y MENDIETA, Roque con adscripción en la Benemérita Universidad Autónoma de Puebla, como siguiente artículo está *Clúster de computadoras de alto rendimiento usando raspberry Pi 3, para mejorar prácticas educativas* por SALAZAR, Pedro, SOTO, Saúl y HERNÁNDEZ, Talhia, como siguiente artículo está *Análisis de vulnerabilidades en redes inalámbricas instaladas en diversos municipios del Estado de Hidalgo* por GONZÁLEZ-MARRÓN, David, PÉREZ-HERNÁNDEZ, Iridian, MARQUÉZ-CALLEJAS, Alejandro y BADILLO-PAREDES, Leonardo, como siguiente artículos está *Determinación de parámetros que impiden una implementación eficiente de algoritmos criptográficos en ambiente multiplataforma* por GONZÁLEZ-MARRÓN, David, GAMERO-PLAFOX, Benito, LÓPEZ-MELO, Eduardo y AGUILAR-GÓMEZ, José con adscripción en el Instituto Tecnológico de Pachuca.

Contenido

Artículo **Revista de Tecnología Informática**
Septiembre 2017 Vol.1 No.2 1-8

Reconocimiento de patrones en gráficos de control utilizando una red neuronal

GUARNEROS-RIVERA, Manuel*†, DÍAZ, LÓPEZ-CHAU, Asdrúbal, MUÑOZ-CONTRERAS, Hilarion y PELÁEZ-CAMARENA, Silvestre Gustavo

Instituto Tecnológico de Orizaba, Oriente 9, Emiliano Zapata Sur, C.P. 94320 Orizaba, Veracruz, México

Recibido Julio 5, 2017; Aceptado Septiembre 15, 2017

Resumen

Los gráficos de control son una herramienta importante en el control de procesos estadístico para mejorar la calidad de los productos mediante estabilidad y reducción de la variabilidad. Los patrones no naturales en los gráficos de control suponen que existe una causa asignable que afecta al proceso y que se deben tomar acciones para solucionar el problema. Debido a su capacidad y rapidez de reconocimiento, las redes neuronales proporcionan gran rendimiento para reconocer tendencias en los procesos. En este artículo, se describe un modelo de red neuronal para el reconocimiento de patrones en gráficos de control. Los resultados señalan una configuración de red que lleva a una buena calidad de reconocimiento.

Perceptron multicapa, Retropropagación, Reconocimiento de patrones de gráficos de control

Abstract

Control charts are an important tool in statistical process control to improve product quality through stability and variability reduction. Unnatural patterns in control charts assume that there is an assignable cause that affects the process and that actions must be taken to solve the problem. Because of their ability and speed of recognition, neural networks provide great performance to recognize process trends. In this paper, we describe a neural network model for pattern recognition in control charts. The results indicate a network configuration leading to good recognition quality.

Backpropagation, control chart pattern recognition, multilayered perceptron

Citación: GUARNEROS-RIVERA, Manuel, DÍAZ, LÓPEZ-CHAU, Asdrúbal, MUÑOZ-CONTRERAS, Hilarion y PELÁEZ-CAMARENA, Silvestre Gustavo. Reconocimiento de patrones en gráficos de control utilizando una red neuronal. Revista de Tecnología Informática 2017, 1-2: 1-8

* Correspondencia al Autor (Correo Electrónico: mrivera@acm.org)
† Investigador contribuyendo como primer autor.

Introducción

El Control estadístico de Procesos (CEP), es un concepto que está ligado con la calidad, es una herramienta que muestra el estado de un proceso de transformación en términos estadísticos, lo cual permite monitorear y establecer parámetros para su mejor control, además es útil para conseguir estabilidad y mejorar la capacidad del proceso mediante la reducción de variabilidad [1]. Los gráficos de control son elaborados a partir de los valores medidos de muestras tomadas del proceso, los gráficos son una herramienta que se utiliza para analizar datos estadísticos de manera sofisticada, muestran la cantidad y la naturaleza de la variación de un proceso, indican el control estadístico o la falta de él y permiten la interpretación y detección de patrones de cambio en el proceso de estudio [2]. El reconocimiento exacto y rápido de los patrones de cambio en gráficos de control es esencial para mantener productos de alta calidad.

Varios enfoques se han propuesto para el reconocimiento de patrones en gráficos de control, incluidos sistemas basados en reglas [3], sistemas expertos [4] y con redes neuronales artificiales [5-19]. La ventaja de un sistema experto o basado en reglas es que contiene la información explícitamente, si es necesario, las reglas pueden ser modificadas y actualizadas fácilmente, sin embargo, el uso de las reglas basadas en propiedades estadísticas tiene la dificultad de que pueden derivar en propiedades estadísticas similares para algunos patrones de diferentes clases, lo que puede crear problemas de reconocimiento incorrecto.

Las redes neuronales artificiales han sido ampliamente aplicadas en el reconocimiento de patrones.

Estas redes han demostrado ser buenas alternativas a los sistemas tradicionales de reconocimiento de patrones en gráficos de control, debido a sus características para generalizar, su facilidad de implementación y la capacidad de manejar medidas ruidosas que no requieren ninguna suposición acerca de la distribución estadística de los datos monitorizados.

La mayoría de los investigadores han utilizado redes neuronales artificiales supervisadas, tales como perceptrón multicapa (MLP), función de base radial (RBF) [20] y cuantificación del vector de aprendizaje (LVQ) [21] para clasificar diferentes tipos de patrones en gráficos de control. El perceptrón multicapa, con algoritmo de aprendizaje de retro propagación, es quizás el modelo de red neuronal más utilizado, siendo fácil de entender y de implementar. Algunos otros investigadores han utilizado fuzzy clustering [22] para el reconocimiento de patrones. Un clasificador basado en árbol de decisiones (DT) [23] también es popular para el problema del reconocimiento de patrones en gráficos de control. Para el aprendizaje de redes neuronales, es necesario encontrar un algoritmo que aprenda bien y rápidamente. Hay algunas comparaciones disponibles en la literatura, pero no dan un concepto preciso para determinar si un algoritmo es mejor para una aplicación específica.

Este artículo describe una red neuronal artificial multicapa para identificar situaciones fuera de control en gráficos de control con el fin de mejorar la capacidad de detección de patrones. La red esta entrenada para diferentes estructuras y se han evaluado varias reglas de aprendizaje usadas para ajustar los pesos de la red neuronal. Se conserva la mejor configuración y el algoritmo más preciso.

ISSN-2531-2197

En resto del artículo está organizado en 5 secciones. En la Sección 2, se presenta una breve revisión del perceptrón multicapa MLP y el algoritmo para su entrenamiento. En la Sección 3 se muestra la aplicación de MLP al problema de reconocimiento de patrones en gráficos de control. Los resultados obtenidos con datos obtenidos de la replicación de registros de una empresa alimentaria se presentan en la Sección 4. En la Sección 5 se encuentran las conclusiones de este trabajo. Las referencias utilizadas están al final del presente artículo.

Perceptrón Multicapa (MLP)

Como se muestra en la Figura 1, MLP consta de tres tipos de capas: una capa de entrada, una capa de salida y una o más capas ocultas. Las neuronas en la capa de entrada actúan solamente como reguladores para distribuir la señal de entrada x_i a las neuronas en la capa oculta. Cada neurona j en la capa oculta añade sus señales de entrada x_i después de multiplicarlas por las resistencias de los respectivos pesos de conexión w_{ji} y calcula su salida y_j como una función de la suma, es decir;

$$y_j = f\sum(w_{ji}x_i) \tag{1}$$

f es generalmente una función tangente sigmoidal o hiperbólica. Las salidas de las neuronas en la capa de salida se calculan de manera similar.

El entrenamiento de una red consiste en ajustar sus pesos usando un algoritmo de entrenamiento. El algoritmo de entrenamiento adoptado en este estudio optimiza los pesos intentando minimizar la suma de las diferencias cuadradas entre los valores deseados y reales de las neuronas de salida, es decir;

$$E = \frac{1}{2}\sum_j(y_{dj} - y_j)^2 \tag{2}$$

Donde y_{dj} es el valor deseado de la neurona de salida j y y_j es la salida real de esa neurona.

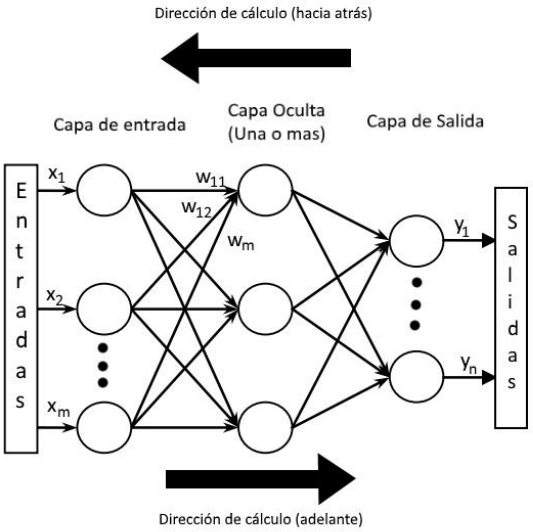


Figura 1 Estructura básica de un perceptrón multicapa

Cada peso w_{ji} se ajusta añadieno un incremento w_{ji} a él. w_{ji} se selecciona para reducir E tan rapidamente como sea posible. El ajuste se realiza a lo largo de varias iteraciones de entrenamiento hata que se obtiene un valor satisfactorio pequeño de E o se alcanza un número dado de iteraciones. La forma en que w_{ji} se calcula depende del algoritmo de entrenamiento adoptado. El algoritmo adoptado en este trabajo se describe brevemente a continuación.

Algoritmo de Retropropagación "Backpropagation" (BP)

El algoritmo de retropropagación (BP) da el cambio $\Delta w_{ji}(k)$ en el peso de la conexión entre las neuronas i y j en la iteración k como

$$\Delta w_{ji}(k) = -\alpha\frac{\delta E}{\delta w_{ji}(k)} + \mu\Delta w_{ji}(k-1) \tag{3}$$

Donde α corresponde al coeficiente de aprendizaje, μ el coeficiente de momento y $\Delta w_{ji}(k-1)$ el cambio de peso en la iteración inmediata anterior.

El entrenamiento de un MLP por BP implica presentarlo secuencialmente con todas las tuplas de entrenamiento. Las diferencias entre la salida objetivo $y_d(k)$ y la salida real $y(k)$ del MLP se propagan de nuevo a través de la red para adaptar sus pesos. Una iteración de entrenamiento se completa después de que una tupla en el conjunto de entrenamiento ha sido presentada a la red y los pesos actualizados.

Reconocimiento de Patrones en gráficos de control

En este trabajo, se usa un MLP para reconocimiento de patrónes en gráficos de control. Los gráficos de control se utilizan para supervisar el comportamiento de un proceso.
La Figura 2 muestra los 6 tipos principales de patrones que se pueden observar en un gráfico de control: normal (NR), cíclico (CC), desplazamiento hacia abajo (DS), desplazamiento hacia arriba (US), tendencia creciente (UT) y tendencia decreciente (DT). Todos los patrones, excepto el patrón normal, indican que el proceso que se está monitoreando no está funcionando correctamente y requiere ajuste.

Para este trabajo, los patrones de los seis diferentes tipos fueron generados replicando datos registrados de una empresa alimentaria, obtenidos en un estudio para ver el estado de control de sus procesos. Cada patrón se tomó como una serie de tiempo de 24 datos. Cuatrocientos veinte patrones, 70 de cada tipo se generaron en total. Doscientos cuarenta patrones se utilizaron para la formación del clasificador perceptrón multicapa y el resto para la prueba del clasificador entrenado.

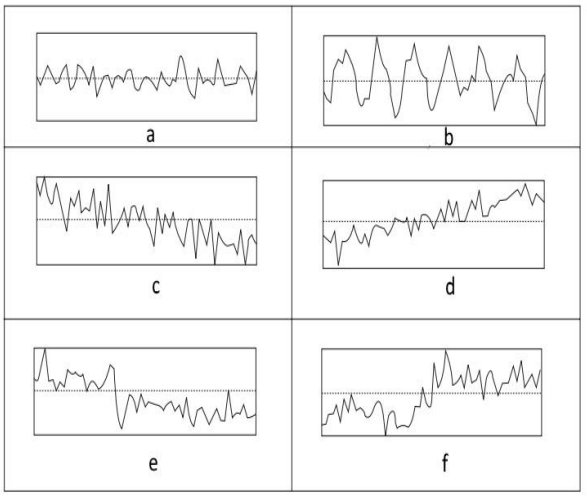


Figura 2 Patrones básicos en gráficos de control: (a) Patrón normal, (b) Patrón Cíclico, (c) Tendencia decreciente, (d) Tendencia creciente, (e) Desplazamiento hacia abajo, (f) Desplazamiento hacia arriba

Antes de que los datos fueran presentados a la red, se implementaron dos etapas de preprocesamiento de datos: normalización y codificación. La normalización es un proceso de transformación lineal: a través del cual la variable de datos brutos X se transforma en una nueva variable Z, es decir;

$$z(t) = (x(t) - \mu)/\sigma \tag{4}$$

La variable $z(t)$ seguirá una distribución normal estándar con una media cero y una desviación estándar de la unidad siempre que $x(t)$ siga una distribución normal.

El proceso de normalización reduce los datos a un rango constante, aproximadamente [-3, +3], independientemente de los valores que tomaron los datos antes de la normalización. Esta transformación es necesaria porque una red neuronal entrenada sólo puede aceptar un cierto rango de datos de entrada.

ISSN-2531-2197

Un proceso de codificación se implementó después de la normalización para reducir el efecto del ruido en los datos de entrada antes de que los datos fueran presentados a la red. El esquema de codificación funciona como una operación de suavizado para filtrar las pequeñas variaciones aleatorias mientras se conservan las características principales de los datos. Esta codificación permitió que la convergencia se produjera más fácilmente.

Cada red tenía 24 neuronas de entrada, una para cada dato en la serie temporal y 6 neuronas de salida, una para cada tipo de patrón de gráfico de control. La tabla 1 representa los patrones de gráficos de control y la representación de las salidas de la red neuronal deseadas.

La estructura del clasificador propuesto se muestra en la Figura 3. Se observa que este sistema se compone de una capa oculta de tres neuronas. El número de capas ocultas fue elegido ya que se había encontrado adecuado para la mayoría de los problemas de clasificación.

El clasificador fue entrenado utilizando el algoritmo BP. Los valores de los parámetros de entrenamiento adoptados para los algoritmos se determinaron empíricamente. Fueron los siguientes: para BP $\alpha = 0.9$ y $\mu = 0.5$.

Clase	**Descripción**	**Salida de la Red neuronal**					
1	Tendencia creciente (UT)	1	0	0	0	0	0
2	Tendencia decreciente (DT)	0	1	0	0	0	0
3	Normal (NR)	0	0	1	0	0	0
4	Cíclico (CC)	0	0	0	1	0	0
5	Desplazamien-to hacia arriba (US)	0	0	0	0	1	0
6	Desplazamien-to hacia abajo (DS)	0	0	0	0	0	1

Tabla 1 Patrones de gráficos de control y representación de las salidas deseadas de la red neuronal

Se registró el número de iteraciones de entrenamiento requeridas para lograr el valor de E. El criterio de parada real empleado fue la suma de los cuadrados del error, que se define como:

$$SSE = \sum_{i=j}^{M} \sum_{j=1}^{N} \left(y_{dj}^{(i)} - y_{j}^{(i)} \right)^2 \quad (5)$$

Donde N es el número de salidas y M es el número de patrones en el conjunto de entrenamiento. El clasificador fue entrenado hasta que se definió el número de iteraciones fijado en 1000.

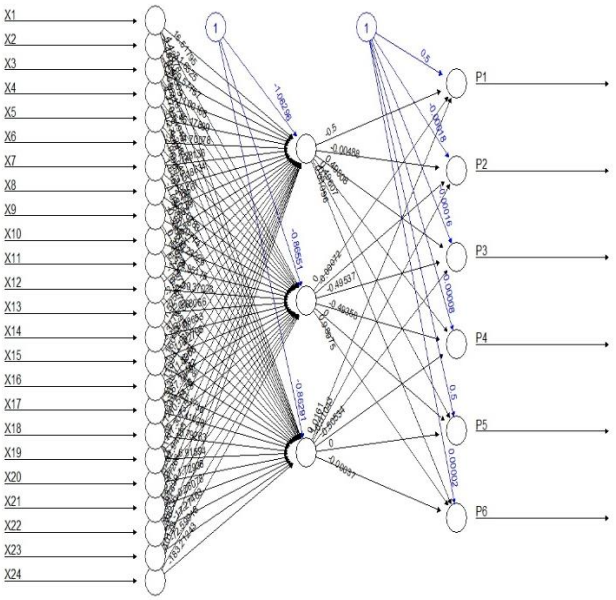


Figura 3 Estructura del clasificador propuesto

Resultados

Para el estudio, se utilizó el 60% de los datos para la formación del clasificador y el resto para la prueba. La manera típica de evaluar la tasa de rendimiento es elegir un conjunto de pruebas independiente del conjunto de entrenamiento para clasificar sus elementos, contar los elementos que se han clasificado correctamente y dividir por el tamaño del conjunto de pruebas.

ISSN-2531-2197

La proporción de elementos del conjunto de pruebas que clasifican correctamente del total de elementos estima el rendimiento del clasificador para cada patrón. Los resultados preliminares se pueden apreciar en la Tabla 2, como puede observarse el clasificador reconoce los seis tipos de patrones con una media de precisión de 94.445%.

Clase	**Descripción**	**Precisión de clasificación (%)**
1	Tendencia creciente (UT)	96.67
2	Tendencia decreciente (DT)	93.33
3	Normal (NR)	96.67
4	Cíclico (CC)	90.00
5	Desplazamiento hacia arriba (US)	93.33
6	Desplazamiento hacia abajo (DS)	96.67

Tabla 2 Precisión de Reconocimiento del clasificador

Los valores en la matriz diagonal de confusión muestran el desempeño correcto del clasificador para cada patrón. Estos valores muestran que varios de los patrones considerados son reconocidos correctamente. Por ejemplo, en la primera fila de la matriz, en la Tabla 3, el valor 96.67% muestra el porcentaje de reconocimiento correcto del patrón de tendencia creciente y el valor 3.33% muestra que este tipo de patrón se reconoce erróneamente con el patrón de desplazamiento hacia arriba.

	UT	DT	NR	CC	US	DS
UT	96.67	0	0	0	3.33	0
DT	0	93.33	0	0	0	6.67
NR	0	0	96.67	3.33	0	0
CC	3.33	3.33	3.33	90.0	0	0
US	6.67	0	0	0	93.33	0
DS		3.33				96.67

Tabla 3 Matriz de confusión del clasificador (%)

Conclusiones

Los gráficos de control son importantes herramientas estadísticas de control de procesos para determinar si un proceso se ejecuta en su modo deseado o existe presencia de patrones no naturales al proceso. Este estudio abordó el diseño de un clasificador para el reconocimiento de patrones en gráficos de control. Se propone un clasificador MLP (perceptrón multicapa) que se compone de una capa oculta con tres neuronas, entrenado con el algoritmo de retropropagación para el reconocimiento de los seis patrones básicos en gráficos de control. La complejidad de este clasificador propuesto es menor a otras obras no obstante la precisión más alta obtenida de manera preliminar es de 96.67%.

Con el fin de aumentar la precisión de clasificación se plantea una optimización de la estructura de la red neuronal, la integración con otras técnicas que permitan un mejor razonamiento de la red neuronal en la etapa del entrenamiento así como una mejor clasificación de los elementos a clasificar. Es posible combinar el modelo propuesto con otras redes neuronales o sistemas expertos para inferir las causas relevantes de las variaciones y facilitar el control automático de la calidad.

Referencias

Montgomery, D. (2001). Introduction to statistical quality control., 5th edition, John Wiley, Hoboken.

Amitava, M. (2006). Fundamentals of Quality Control and Improvement, 2nd edition, Thompson.

Ataollah, E. y Vahid, R. (2009). A hybrid intelligent technique for recognition of control chart patterns, 2009 First International Conference on Networked Digital Technologies, Ostrava, pp. 32-36.

ISSN-2531-2197

Jenn, H.Y. y Miin-Shen, Y. (2005).A control chart pattern recognition system using a statistical correlation coefficient method, Computers & Industrial Engineering, vol. 48, pp 205-221.

Cheng, C. (1997). A neural network approach for the analysis of control chart patterns, International Journal of Production Research, vol. 35, pp. 667-697.

Ruey-Shy, G. y Yi-Chih, H. (1999). A neural network based model for abnormal pattern recognition of control charts, Computers & Industrial Engineering, vol. 36, pp. 97-108.

Jianbo, Y. y Lifeng, X. y Bin, W. (2007). A Neural Network Ensemble Approach for the Recognition of SPC Chart Patterns, Third International Conference on Natural Computation (ICNC 2007), pp. 575-579.

Zhiqiang, C. y YiZhong, M. (2008). A Research about Pattern Recognition of Control Chart Using Probability Neural Network, 2008 ISECS International Colloquium on Computing, Communication, Control, and Management, pp. 140-145.

Mahmoud, B. (2015). An Effective and Novel Neural Network Ensemble for Shift Pattern Detection in Control Charts, Computational Intelligence and Neuroscience, vol. 2015.

Ataollah, E. y Vahid R. (2010). Control chart pattern recognition using an optimized neural network and efficient features, ISA Transactions, vol. 49, pp. 387-393.

Ruey-Shiang, G. (2004). Optimizing feedforward neural networks for control chart pattern recognition through genetic algorithms. International Journal of Pattern Recognition and Artificial Intelligence, vol. 18, p. 75-99.

Marcus, P. y Julie, K. y Tomas V. (2004). Control chart pattern recognition using back propagation artificial neural networks, International Journal of Production Research, vol. 39, pp. 3399-3418.

Ataollah, E. y Jalil A. y Zahra R. (2011). Control chart pattern recognition using K-MICA clustering and neural networks, ISA Transactions, vol. 51, pp. 111-119.

Lesany, S. y Koochakzadeh, A. y Fatemi G. (2014). Recognition and classification of single and concurrent unnatural patterns in control charts via neural networks and fitted line of samples, International Journal of Production Research, vol. 52, pp. 1771-1786.

Ruey-Shiang, G. (1999). Recognition of control chart concurrent patterns using a neural network approach, International Journal of Production Research, vol. 37, pp. 1743-1765.

Yousef, A. (2004). Recognition of control chart patterns using multiresolution wavelets analysis and neural networks, Computers & Industrial Engineering, vol. 47, pp. 17-29.

Bidyut B. (2009). Recognition of quality control chart patterns based on back propagation neural network, 2009 16th International Conference on Industrial Engineering and Engineering Management, pp. 1124-1128.

Duc, P. y Seref, S. (2000). Training multilayered perceptrons for pattern recognition: a comparative study of four training algorithms, International Journal of Machine Tools and Manufacture, vol. 41, pp. 419-430.

Omar, E. y Ali, M. y Said, B. (2015). Using Artificial Neural Networks for Recognition of Control Chart Pattern, International Journal of Computer Applications, vol. 116.

ISSN-2531-2197

Jalil, A. y Ata, E. y Milad, A. y Vahid, R. (2014). Statistical process control using optimized neural networks: A case study, ISA Transactions, vol. 53, pp. 1489-1499.

Ruey-Shiang, G. (2005). A hybrid learning based model for online detection and analysis of control chart patterns, Computers & Industrial Engineering, vol. 49, pp. 35-62.

Amir, B. y Abdolreza, G. y Masoud, A. y Jalil A. (2013). Control Chart Patterns Recognition Using Optimized Adaptive Neuro-Fuzzy Inference System and Wavelet Analysis, Journal of Engineering and Technology, vol. 3.

Chih-Hsuan, W. y Ruey-Shan, G. y Ming-Huang, C. y Jehn-Yih, W. (2007). Decision tree based control chart pattern recognition, International Journal of Production Research, vol.46, pp. 4889-4901.

ISSN-2531-2197

Artículo **Revista de Tecnología Informática**
Septiembre 2017 Vol.1 No.2 9-18

Estudio del comportamiento de algoritmos de clasificación según la naturaleza de los datos

CRUZ-GUERRERO, René*†, ALONSO-LAVERNIA, Ma. de los Ángeles', FRANCO-ARCEGA, Anilú', SIMÓN-MARMOLEJO, Isaías''

Universidad Autónoma del Estado de Hidalgo, Carretera Pachuca-Tulancingo, Km 4.5, CP. 42186, Mineral de la Reforma, Hidalgo, México.
'Instituto Tecnológico Superior del Oriente del Estado de Hidalgo, Carretera Apan-Tepeapulco Km 3.5, Colonia Las Peñitas, C.P. 43900, Apan Hidalgo.
''ESSAH, Universidad Autónoma del Estado de Hidalgo, Carretera Cd. Sahagún-Otumba s/n. Zona Industrial, CP. 43990, Tepeapulco, Hidalgo, México.

Recibido Julio 3, 2017; Aceptado Septiembre 14, 2017

Resumen

La clasificación es una técnica de Minería de Datos que se utiliza para averiguar con qué grupo una instancia de datos está relacionada dentro de un determinado conjunto de datos. Actualmente, existe una gran diversidad de algoritmos que ejecutan esta tarea, teniendo cada uno una base teórica distinta. El dilema con el que se enfrenta un usuario al realizar la tarea de clasificación es la de seleccionar el algoritmo que responda con mayor eficacia. Este trabajo presenta un estudio de la aplicación de los algoritmos Naive Bayes, C4.5, Perceptrón multicapa y K-vecinos a 38 conjuntos de datos con diferentes características, de lo cual resultaron algunas reglas que describen patrones de comportamiento en correspondencia con la población tratada. Los resultados de este trabajo proporcionaron un conjunto de criterios, los cuáles son un recurso útil que permite reducir el tiempo dedicado a la selección del clasificador, sobre todo para aquellos usuarios que no tienen dominio sobre cómo trabajan los diferentes algoritmos.

Minería de Datos, clasificación supervisada, eficacia de algoritmos de clasificación

Abstract

Classification is a Data Mining technique that is used to find out with which group an instance of data is related within a given set of data. Currently, there is a great diversity of algorithms that execute this task, each having a different theoretical base. The dilemma faced by a user when performing the classification task is to select the algorithm that responds most effectively. This paper presents a study of the application of the algorithms Naive Bayes, C4.5, Perceptron multi-layer and K-neighbors to 38 data sets with different characteristics, resulting in some rules that describe behavior patterns in correspondence with the treated population. The results of this work provided a set of criteria, which are a useful resource that allows reducing the time devoted to the selection of the classifier, especially for those users who do not have control over how the different algorithms work.

Data Mining, Sspervised classification, efficiency of classification algorithms

Citación: CRUZ-GUERRERO, René, ALONSO-LAVERNIA, Ma. de los Ángeles, FRANCO-ARCEGA, Anilú, SIMÓN-MARMOLEJO, Isaías. Estudio del comportamiento de algoritmos de clasificación según la naturaleza de los datos. Revista de Tecnología Informática 2017, 1-2: 9-18

* Correspondencia al Autor (Correo Electrónico: rencrug@gmail.com)
† Investigador contribuyendo como primer autor.

Introducción

Dentro de las técnicas predictivas, la clasificación es una tarea muy socorrida en cualquier área del conocimiento por su capacidad de identificar automáticamente para un objeto, la clase a la cual está asociado, utilizando para ello el conocimiento que relaciona las características de las instancias con sus respectivas clases (Hernández, Ramírez, & Ferri, 2008).

Para llevar a cabo la tarea de clasificación se han desarrollado diversos algoritmos, los cuales ofrecen soluciones bajo diferentes enfoques como son: árboles de decisión, basados en vecindad, probabilísticos, redes neuronales, entre otros. Sin embargo, la diversidad de clasificadores y la versatilidad de las poblaciones que se estudian en el ámbito real, hace compleja la tarea de seleccionar un algoritmo en específico que se ejecute con mayor eficacia.

En investigaciones realizadas, con la intención de identificar relaciones entre clasificadores y conjuntos de datos, se ha encontrado que algunos algoritmos funcionan mejor que otros para ciertos conjuntos de datos. El presente trabajo tiene el objetivo de estudiar diversos algoritmos y poblaciones de datos para analizar su comportamiento y encontrar relaciones entre ambos aspectos, con el propósito de poder proponer algunas reglas que aunque no sean absolutas permitan realizar una selección del clasificador lo más acertada posible en función de las características de la base de datos que se analiza. Para este estudio se utilizan los algoritmos de clasificación más utilizados en la literatura, mismos que fueron probados con diversas bases de datos de características diferentes, considerando la cantidad de instancias, número de clases, número de atributos, tipo de datos, si es o no balanceada, entre otras.

El resto del documento presenta en la Sección 2 algunos trabajos relacionados a estudios de algoritmos de clasificación, en la Sección 3 se describen los algoritmos utilizados en la experimentación, en la Sección 4 se muestran los resultados obtenidos y en la Sección 5, la discusión de estos. Finalmente, se presentan las conclusiones y trabajo futuro.

Antecedentes

En los últimos años, se han desarrollado algunos trabajos dirigidos al análisis comparativo de algoritmos de clasificación, considerando aspectos como velocidad de ejecución, precisión y tipos de datos tratados.

Akinola y Oyabugbe (2015) realizaron un estudio con los algoritmos Árboles de Decisión (AD), Naive Bayes y Perceptrón multicapa respecto a su velocidad de ejecución, utilizando un solo conjunto de datos. En este estudio se pudo observar que el algoritmo Naive Bayes consumía menos tiempo en ejecutar el proceso de clasificación, sin embargo el haber considerado una sola población, no permitió identificar alguna dependencia entre la velocidad y algunas otras características de la población como por ejemplo la cantidad de instancias o los tipos de atributos. Otro trabajo similar lo realizaron Ashari et al. (2013) con los algoritmos AD, Naive Bayes y k-Vecinos más cercanos (KNN), probandolos con cinco bases de datos. Después de realizar su trabajo experimental, mostraron en sus resultados que el algoritmo AD fue el más rápido, sin embargo estos resultados requieren constatarse con una mayor cantidad de conjuntos de datos, porque con mayores cantidades de instancias o atributos los resultados pueden cambiar.

ISSN-2531-2197

Respecto a estudios realizados para comparar la precisión de los clasificadores dependiendo del conjunto de datos tratados, Entezari et al. (2009) compararon los algoritmos KNN, LogR, Naive Bayes, AD, C4.5, Máquinas de Vectores de Soporte (SVM por sus siglas en inglés Support Vector Machine) y Linear Classifier (LC), tomando en cuenta características como: cantidad de instancias, tipo de atributos y número de atributos discretos o continuos. Para realizar este estudio, los autores generaron 29 conjuntos de datos sintéticos, mismos que se organizaron en cuatro grupos de acuerdo al número de atributos que contenían (3, 5, 7 y 10, respectivamente). Con el fin de tener una diversidad de poblaciones, de cada grupo de datos se consideraron para la creación de los conjuntos distintas cantidades de atributos numéricos y discretos y distintas cantidades de instancias (200, 500, 1000, 3000 y 5000). Los algoritmos que mejor desempeño mostraron fueron KNN, SVM, AD y C4.5, no importando si el número de instancias o de atributos aumentaba. Un comportamiento particular se observó con SVM, quien obtuvo mejor precisión que KNN cuando la cantidad de atributos numéricos es mayor y en caso contrario, KNN trabaja mejor con los discretos. Sólo se analizan dos características (número de instancias y tipo de atributo), omitiendo otras que pueden incidir en los resultados que proporciona el clasificador sobre determinada población.

Otro estudio lo desarrollaron Moran et al. (2009) en donde se procesaron 39 bases de datos, las cuales se agruparon en un total de 12 conjuntos mediante la combinación de tres de sus características: número de instancias, total de atributos y porcentaje de atributos categóricos. Tomando la cantidad promedio del número de instancias de las poblaciones tratadas (286 instancias), los conjuntos se separaron en dos grupos, los que tenían más de esta cantidad de instancias de los que tienen menos o igual.

Cada uno de estos grupos se subdividió en dos grupos de acuerdo al número de atributos, en los que tenían más de 16 y los que tenían menos o igual que 16. Por último, se subdividió cada subgrupo anterior en tres, ahora de acuerdo al número de atributos nominales, con el 100% de atributos de este tipo, con más del 50% y con menos o igual que el 50%. En este estudio sólo se utilizaron clasificadores que se derivan o son variantes del algoritmo Naive Bayes, siendo estos: Averaged One Dependence Estimator (AODE), Tree Augmented Naive Bayes (TAN), BN K2, Genetic Search (BN-GS) y Simulated Annealing (BN-SA). El trabajo se divide en dos etapas, primero se probaron los algoritmos con los distintos conjuntos de datos para verificar cuál funciona mejor y posteriormente, considerando los resultados y con el uso del algoritmo J48 se generó un conjunto de reglas de clasificación que ayudan a seleccionar el mejor clasificador para un conjunto de datos particular. En los resultados de la primera etapa detectaron que el clasificador que brindó mejores porcentajes de precisión fue Tree Augmented Naive Bayes (TAN) para bases de datos con atributos numéricos o combinados y el algoritmo ODE para datos 100% nominales. Respecto a las reglas obtenidas en la segunda etapa, se comprobó con los conjuntos de datos iniciales obteniéndose un 78% de efectividad en su aplicación sobre dichos conjuntos. A pesar de los resultados alcanzados, no se efectúa una validación con nuevos conjuntos de datos y se utilizan solo clasificadores de enfoque probabilístico.

Algoritmos de clasificación

El objetivo de la técnica de clasificación es obtener un valor particular de un atributo a partir de los valores de otros atributos. El atributo a obtener es comúnmente llamado Clase o variable dependiente, mientras que los atributos usados para hacer la predicción se llaman Variables Independientes. Para llevar a cabo esta tarea existen diversos algoritmos de clasificación, a continuación se describen los utilizados en el desarrollo de este trabajo.

ISSN-2531-2197

C4.5

El método C4.5 está basado en árboles de decisión, el cual utiliza la ganancia de información por medio del cálculo de la entropía para medir qué tan bien un atributo separa el conjunto de instancias de acuerdo a sus clases (Quinlan, 1986). A continuación, se explica el algoritmo de dicho método.

La construcción del árbol inicia con un nodo raíz vacío. Después de haberse creado la raíz, se comprueba si las diferentes instancias tienen el mismo valor para el atributo clase, de ser así se obtiene solo un nodo, posteriormente para todos los atributos no clase se verifica mediante el cálculo de entropías cuál es el que proporciona mayor ganancia para ubicarlo en el nodo del nivel más alto del árbol, asignando al nodo el nombre del atributo ganador y a los arcos los valores de dicho atributo. Para el resto de los atributos no clase, esto se realiza de forma iterativa hasta llegar a los nodos hoja.

Cuando se quiere clasificar una nueva instancia, se utilizan sus diferentes atributos (incluyendo sus valores) para recorrer el árbol creado (modelo de clasificación) iniciando por el nodo raíz. El recorrido de los nodos y arcos se efectúan de forma descendente hasta encontrar la hoja buscada (clase).

KNN

El método KNN (por sus siglas en inglés, K Nearest Neighbors) se basa en efectuar aprendizaje por analogía, consiste en realizar una serie de comparaciones para que a una nueva instancia que contiene n atributos se le asigne la clase mayoritaria de sus k vecinos más cercanos (Coomans & Massart, 1982).

Las comparaciones entre las instancias se realizan mediante algún criterio de vecindad definida en términos de algún tipo de métrica, cuando todas las variables son numéricas se aplican métricas como la distancia Euclidiana, de Manhattan, de Chebyshev, entre otras, por otra parte, cuando se tienen tanto variables numéricas como categóricas, se puede aplicar la métrica de Gower (Deza & Deza, 2009).

El proceso inicia especificando los datos de la nueva instancia, posteriormente es importante indicar el número de vecinos a evaluar, así como seleccionar la métrica con la que se calculará la distancia entre las instancias. El siguiente paso consiste en calcular la distancia que existe entre la nueva instancia y las demás, posteriormente se ordenan los resultados de manera ascendente y de sus vecinos más cercanos se verifica cuál es la clase más frecuente para asignarla a la nueva instancia.

Perceptrón multicapa

El Perceptrón Multicapa (MLP por sus siglas en inglés MultiLayer Perceptron) se basa en un algoritmo de propagación hacia atrás para clasificar nuevas instancias con el uso de redes neuronales (Lu & Setiono, 1997). La red neuronal de retro propagación es esencialmente una interconexión de elementos simples de procesamiento que trabajan juntos para producir una salida. El entrenamiento de la red consiste en calcular de manera iterativa un conjunto de pesos para la predicción de la etiqueta de la clase de las instancias analizadas.

Una red neuronal de tipo MLP está formada de una capa de entrada, una o más capas ocultas y una capa de salida. Cada capa de una red neuronal está compuesta por un conjunto de unidades (neuronas). Las entradas a la red corresponden a valores de los atributos de cada instancia usada para entrenar.

Las entradas son alimentadas simultáneamente en las neuronas que forman la capa de entrada y luego se ponderan y alimentan a una segunda capa llamada oculta. Las salidas de las neuronas de la capa oculta se pueden introducir a otra capa oculta, la capa de salida corresponde al resultado esperado.

Naive Bayes

El método Naive Bayes permite usar el conocimiento apriori para predecir una suposición mediante el cálculo de probabilidades. El uso del teorema de Bayes en la tarea de clasificación se debe a que permite calcular las probabilidades a posteriori de cualquier hipótesis consistente con el conjunto de datos de entrenamiento para así escoger la hipótesis más probable (Miquelez, Bengoetxea, & Larranaga, 2004).

El método consiste en calcular tanto la probabilidad de cada clase como la de los diferentes valores de cada variable independiente. Debido a que este clasificador asume que las características son condicionalmente independientes, evita comparaciones de ganancia de información entre combinaciones de variables o atributos.

Máquinas de Soporte Vectorial

El método de Máquinas de Soporte Vectorial (SVM por sus siglas en inglés, Support Vector Machines) se centra en lo que se conoce como Teoría del Aprendizaje Estadístico donde se tiene el objetivo de establecer un margen máximo de separación entre clases (Taylor, 2004).

El método consiste en crear un modelo que permita separar instancias de una clase de otra clase. Cuando las instancias se pueden representar a través de vectores, lo más fácil para separar dos clases es utilizar una línea. Para saber la clase de una nueva instancia dependerá saber de qué lado queda de la línea.

Las SVM tienen como meta encontrar la línea que maximiza la distancia entre dos instancias de cada clase, las instancias usadas para definir la línea se conocen como vectores de soporte. Si el problema se puede separar usando líneas se dice que es linealmente separable, de lo contrario, para resolver el problema MSV lo traslada a una dimensión mayor en la que sí es separable, para lo cual se utiliza una función llamada Núcleo o Kernel. En el algoritmo cinco se muestran de forma resumida los pasos del método.

Trabajo experimental

Con la finalidad de encontrar las particularidades que influyen en la eficacia de los algoritmos de clasificación, se llevó a cabo la ejecución de cuatro algoritmos de este tipo sobre un conjunto de 38 bases de datos y se realizó un análisis de los resultados obtenidos, identificándose comportamientos particulares de desempeño relacionados con las poblaciones en estudio mediante obtención de reglas de clasificación. Posteriormente, se validaron los patrones encontrados con 8 bases de datos distintas.

Aspectos a considerar en el estudio

Para elegir los conjuntos de datos a utilizar en este estudio se consideraron características básicas de las bases de datos como: cantidad de instancias, número de atributos, número de clases y tipos de datos. Adicionalmente, se consideró que existían otras características de las bases de datos que también podrían influir en la elección de un algoritmo de clasificación. Estas características son:

Estructura.- Propiedad de la base de datos que permite especificar si tiene una forma vertical, horizontal o mixta. Vertical se refiere a que tiene muchas instancias y pocos atributos, Horizontal indica lo contrario. Por otra parte, cuando tiene pocos atributos y pocas instancias o muchos atributos y muchas instancias se le denomina Mixta.

ISSN-2531-2197

Multivaluada.- Permite especificar si la mayoría de los atributos nominales tiene más de dos valores.

Completa.- Permite especificar si una base de datos tiene o no datos faltantes.
Balanceada.- Especifica si el conjunto de datos contiene el mismo número de instancias en todas las clases.

Cantidad de atributos numéricos o nominales.- Se detalla la cantidad de variables numéricas o nominales.

Bases de datos y clasificadores utilizados

Se utilizaron 30 bases de datos para la etapa de experimentación y 8 para la de validación, obtenidas de los repositorios de UCI Machine Learning (Lichman, 2013.), portal de WEKA (Hall M. , Frank, Holmes, & Pfahringer, 2009) y portal de Promise (Zwirk & Wigury, 2013). Los algoritmos de clasificación elegidos para este estudio son los descritos en la sección anterior, Árboles de Decisión (C4.5), Naive Bayes, Perceptrón Multicapa y KNN, por ser ellos de los más utilizados en la literatura.

La Tabla 1 muestra los conjuntos de datos utilizados en la experimentación para analizar el comportamiento de los algoritmos elegidos. El valor NumNom, de la característica Tipo de datos, significa que el conjunto de datos tiene atributos de ambos tipos (Numérico y Nominal).

Base de datos	Tipo de datos	Atributos	Numéricos	Nominales	Instancias	Clases	Balanceada	Incompleta	Estructura
Weather_Nom	Nominal	4	0	4	14	2	No	No	Mixta
Car	Nominal	6	0	6	1728	4	No	No	Vertical
Nursery	Nominal	8	0	8	12960	5	No	No	Vertical
Primary_tumor	Nominal	17	0	17	339	22	No	No	Horizontal
Led7	Nominal	7	0	7	200	10	No	No	Vertical
lymphography	Nominal	18	0	18	148	4	No	No	Mixta
Splice	Nominal	60	0	60	3190	3	No	No	Mixta
Breast_Cancer	Nominal	9	0	9	286	2	No	No	Vertical
Spect	Nominal	23	0	23	80	2	No	No	Horizontal
Balance_Scale	Numérico	4	4	0	625	3	No	No	Vertical
Diabetes	Numérico	8	8	0	768	2	No	No	Vertical
Glass	Numérico	9	9	0	214	7	No	No	Vertical
Breast_w	Numérico	9	9	0	699	2	No	No	Vertical
Messidor_features	Numérico	19	19	0	1151	2	Si	No	Mixta
Sonar	Numérico	60	60	0	208	2	Si	No	Horizontal
Iris	Numérico	4	4	0	150	3	Si	No	Vertical
Vehicle	Numérico	18	18	0	846	4	Si	No	Mixta
Waveform_5000	Numérico	40	40	0	5000	3	Si	No	Mixta
Column3	Numérico	7	7	0	310	3	No	No	Horizontal
Ecoli	Numérico	7	7	0	336	6	No	No	Horizontal
Weather_Num	NumNom	4	2	2	14	2	No	No	Mixta
Vowel	NumNom	13	10	3	990	11	Si	No	Mixta
Zoo	NumNom	17	1	16	121	7	No	No	Horizontal
Tae	NumNom	5	3	2	151	3	Si	No	Mixta
Zoo	NumNom	17	1	16	121	7	No	No	Horizontal
Credit-g	NumNom	20	7	13	1000	2	No	No	Horizontal
Bands	NumNom	39	22	17	539	2	No	Si	Horizontal
Credit_Aproval	NumNom	15	6	9	690	2	No	Si	Vertical
Autos	NumNom	25	15	10	205	7	No	Si	Horizontal
Colic	NumNom	22	7	15	368	2	No	Si	Horizontal

Tabla 1 Bases de datos para pruebas

Base de datos	Tipo datos	Atributos	Numéricos	Nominales	Instancias	Clases	Balanceada	Incompleta	Estructura
Vote	Nominal	16	0	16	435	2	No	Si	Horizontal
Monk	Nominal	6	0	6	124	2	Si	No	Horizontal
Letter	Numérico	16	16	0	20000	26	Si	No	Mixta
Sick	NumNom	29	7	22	3772	2	No	Si	Horizontal
Ozone	Numérico	72	72	0	2536	2	No	Si	Horizontal
Segment	Numérico	19	19	0	2310	7	Si	No	Horizontal
Page_Blocks	Numérico	10	10	0	5473	5	No	No	Vertical
Squash_Stored	NumNom	24	21	3	52	3	No	No	Horizontal

Tabla 2 Bases de datos para validación

Etapa de experimentación

Para ejecutar los algoritmos de clasificación se utilizó la herramienta Weka versión 3.6, en un procesador Core i7 de 8 GB de memoria RAM con Windows 7.

Resultados obtenidos

La evaluación de los resultados se hizo usando validación cruzada con 10 particiones, obteniéndose los resultados que se muestran en la Tabla 3, en donde se puede observar de manera resaltada quien de los cuatro algoritmos obtuvo un mejor desempeño para cada conjunto de datos. Resultando AD mejor en 3 conjuntos, Naive Bayes en 6, KNN en 10 y MLP en 11.

ISSN-2531-2197

Base de datos	C4.5 (J48)	Naive Bayes	KNN	MLP
Weather_Nom	50%	57.10%	57.10%	71.42%
Car	92.30%	85.70%	93.50%	99.53%
Nursery	97%	90.30%	98.37%	99.72%
Primary_tumor	39.80%	50.10%	39.20%	38.30%
Led7	71.50%	74%	70%	69.50%
lymphography	77.02%	83.10%	82.40%	84.40%
Splice	94.35%	95.36%	74.67%	95.55%
Breast_Cancer	75.52%	76.67%	72.37%	64.68%
Spect	71.05%	71.25%	53.75%	63.75%
Balance_Scale	76.64%	90.40%	86.56%	90.72%
Diabetes	73.80%	76.30%	70.18%	75.39%
Glass	66.82%	48.59%	71.96%	67.75%
Breast_w	94.56%	95.99%	96.85%	95.27%
Messidor_features	64.37%	77.68%	81.15%	72.02%
Sonar	71.15%	67.78%	86.53%	82.21%
Iris	96%	96%	95.30%	97.33%
Vehicle	72.45%	44.79%	69.85%	81.67%
Waveform_5000	75.08%	80%	73.62%	83.56%
Column3	81.61%	83.22%	78.38%	85.48%
Ecoli	84.22%	85.41%	80.35%	86.01%
Weather_Num	64.28%	64.28%	78.57%	78.37%
Vowel	81.51%	63.73%	99.29%	92.82%
Zoo	92.07%	95.04%	96.03%	95.90%
Tae	59.60%	54.30%	62.25%	54.30%
Zoo	92.07%	95.04%	96.03%	95.90%
Credit-g	70.50%	75.40%	72.00%	71.50%
Bands	64.74%	73.28%	78.29%	77%
Credit_Aproval	86.08%	77.60%	81.10%	83.18%
Autos	81.95%	56.09%	76.09%	80%
Colic	85.32 %	77.98 %	81.25 %	80.43 %

Tabla 2 Porcentajes de precisión obtenidos por método

Recomendación de uso de algoritmos de clasificación

Una vez obtenidos los resultados de los algoritmos, se creó una base de datos con la información que se muestra en la Tabla 1, eliminando la columna del nombre de la base de datos y agregando en una última columna como atributo clase o variable dependiente el nombre del clasificador que obtuvo el mejor porcentaje de precisión, obtenido de los resultados que se mostraron en la Tabla 3.

Este conjunto de datos se creó con el objetivo de aplicarle algoritmos de reglas de clasificación para generar un conjunto de patrones o reglas que permitan saber qué clasificador utilizar bajo ciertas características de una población de datos. Se aplicaron los algoritmos Apriori, BFTree, JRIP rules, J48 y Ridor, obteniéndose los siguientes resultados:

Algoritmo A priori

```
 1. Tipo_Datos=Nominal Num_Clases=3_7 4 ==> Clase=MLP 4 conf:(1)
 2. Tipo_Datos=Nominal Num_Clases=3_7 Completa=No 4 ==> Clase=MLP 4  conf:(1)
 3. Tipo_Datos=Numérico Num_Clases=3_7 7 ==> Clase=MLP 6 conf:(0.86)
 4. Tipo_Datos=Numérico Num_Clases=3_7 Completa=No 7 ==> Clase=MLP 6  conf:(0.86)
 5. Completa=Si 5 ==> Clase=J48 4  conf:(0.8)
 6. Tipo_Datos=NumNom Completa=No 5 ==> Clase=KNN 4 conf:(0.8)
 7. Tipo_Datos=NumNom Completa=Si 5 ==> Clase=J48 4  conf:(0.8)
 8. Balanceada=No Completa=Si 5 ==> Clase=J48 4  conf:(0.8)
 9. Tipo_Datos=NumNom Balanceada=No Completa=Si 5 ==> Clase=J48 4  conf:(0.8)
Algoritmo BFTree
Tipo_Datos = (NumNom)
| Completa = (No): KNN(4.0/1.0)
| Completa != (No): J48(3.0/1.0)
Tipo_Datos != (NumNom)
| Num_Clases < 2.5: NB(3.0/4.0)
| Num_Clases >= 2.5
| | Num_Clases < 6.5: MLP(10.0/0.0)
| | Num_Clases >= 6.5: NB(2.0/1.0)
Algoritmo JRIP rules
(Completa = Si) => Clase=J48 (4.0/1.0)
(Tipo_Datos = NumNom) => Clase=KNN (5.0/1.0)
 => Clase=MLP (20.0/9.0)
```

Algoritmo J48

```
Completa = Si: J48 (4.0/1.0)
Completa = No
|  Tipo_Datos = Nominal
|  |  Num_Clases <= 7
|  |  |  Num_Clases <= 2: NB (3.0/1.0)
|  |  |  Num_Clases > 2: MLP (4.0)
|  |  Num_Clases > 7: NB (2.0)
|  Tipo_Datos = Numérico
|  |  Num_Clases <= 2: KNN (4.0/1.0)
|  |  Num_Clases > 2: MLP (7.0/1.0)
|  Tipo_Datos = NumNom: KNN (5.0/1.0)
Algoritmo Ridor
Clase = J48  (29.0/26.0)
     Except (Completa = No) => Clase = NB  (17.0/0.0) [8.0/0.0]
          Except (Num_Clases > 2.5) and (Num_Clases <= 8.5) => Clase = KNN
                    (10.0/0.0) [3.0/0.0]
               Except (Num_Clases <= 6.5) => Clase = MLP  (7.0/0.0) [4.0/1.0]
               Except (Balanceada = Si) => Clase = KNN  (2.0/0.0) [1.0/0.0]
               Except (Tipo_Datos = Numérico) => Clase = MLP (3.0/1.0) [2.0/1.0]
```

A partir de estos resultados, se verificó qué reglas coinciden, obteniendo en común las que se muestran en la Tabla 4.
A partir de estos resultados, se verificó qué reglas coinciden, obteniendo en común las o criterios que se muestran en la Tabla 4.

No.	Regla	Algoritmo	Porcentaje
1	BD_Incompleta = Si	C4.5	75%
2	((Num_Clases >= 3 and Num_Clases < 7) and (Tipo_Dato = Numérico)) or ((Num_Clases >= 3 and Num_Clases < 7) and (Tipo_Dato=Nominal))	MLP	90%
3	((Num_Clases < 3 or Num_Clases >= 7) and Tipo_Datos = Numérico)) OR (Si BD=Mixta)	KNN	81%
4	(Tipo_Datos = Nominal) and (Num_Clases < 3 or Num_Clases >= 7)	Naive Bayes	100%

Tabla 3 Reglas obtenidas por método de clasificación

Como puede observarse en la Tabla 4, las características que más influyeron en la creación de estas reglas fueron: número de clases, tipos de atributos y si la base de datos tiene valores faltantes o no. La misma tabla presenta a qué algoritmo corresponde cada una y con qué precisión son creadas. Esta precisión corresponde a los 30 conjuntos de datos usados en la experimentación.

Para probar el funcionamiento de las reglas o criterios obtenidos anteriormente, fue necesario desarrollar un framework en Java, donde se solicitan las características de la base de datos y el sistema automáticamente sugiera el clasificador a utilizar, permtiendole al usuario cambiarlo si asi lo desea. En la Figura 1 se muestra la interface principal.

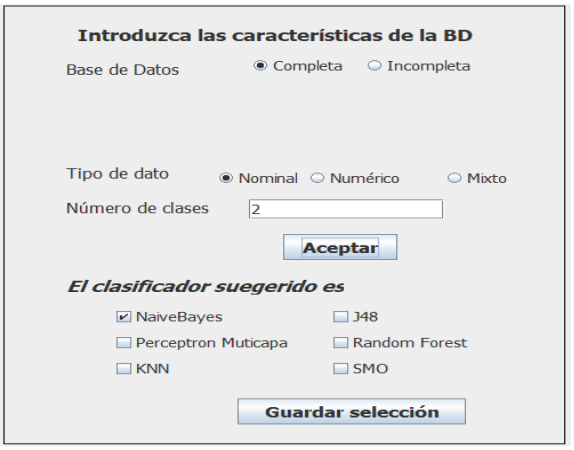


Figura 1 Sistema para seleccionar el clasificador

Otros comportamientos que se detectaron en la etapa de experimentación con los diferentes algoritmos fueron los siguientes:

- El algoritmo que mostró en promedio ser más rápido fue Naive Bayes y el que se mostró más lento en generar el modelo de clasificación fue Perceptrón Multicapa.

- El método que mostró mejor desempeño ante datos con ruido fue J48.

Etapa de validación

Con la finalidad de validar las reglas obtenidas en la sección anterior, se utilizaron las bases de datos mostradas en la Tabla 2. En esta etapa, se aplicaron los mismos métodos que en la etapa de experimentación. En la Tabla 5 se muestran los porcentajes de precisión obtenidos, indicando en la última columna si el resultado coincide con la recomendación emitida por la regla correspondiente.

ISSN-2531-2197

Base de datos	C4.5 J48	Naive Bayes	KNN	MLP
Vote	96.30%	90.10%	92.40%	94.71%
Letter	87.98%	64.11%	96.03%	82.08%
Monk	82.25%	77.41%	76.61%	96.36%
Sick	98.80%	92.60%	96.18%	97.24%
Ozone	96.33%	70.78%	95.26%	95.67%
Page_Blocks	96.87%	90.84%	96.01%	96.23%
Segment	96.92%	80.21%	97.14%	96.14%
Squash_Stored	65.38%	61.53%	73.07%	63.46%

Tabla 4 Porcentajes de precisión en la validación

Discusión

Los resultados generados en la etapa de experimentación con el uso de 30 bases de datos arrojaron en promedio un 86% de efectividad en las reglas obtenidas, mientras que en la etapa de validación se obtuvo un 87%. En particular, donde no se cumplió (conjunto de datos Page_Blocks de la Tabla 5), el algoritmo que debió ser seleccionado según la regla (Perceptrón Multicapa) quedó en segundo lugar y con una diferencia no significativa, lo cual indica que se podría usar este último sin perder precisión en su ejecución sobre los datos tratados. Los porcentajes de efectividad obtenidos en ambas etapas muestran un comportamiento bastante estable en la recomendación brindada por la regla, ya que la variación mínima en la etapa de validación mostró consistencia en el uso de las reglas.

Para cada algoritmo se detectaron características que influyen en los resultados: el método AD mostró una ventaja sobre el resto de los algoritmos cuando trata con bases de datos con valores faltantes, con Perceptrón Multicapa el factor que más influye es el número de clases tratadas y por último, con KNN y Naive Bayes la característica que más influyó fue el tipo de datos. Naive Bayes se comportó mejor con atributos nominales y KNN con numéricos y combinados.

Otro resultado no menos importante en este estudio es que algunas de las características consideradas no influyeron en los resultados alcanzados, estas son: la estructura de la base de datos, si es o no balanceada, número de atributos y número de instancias.

Conclusiones y trabajo futuro

El presente trabajo permitió comprobar que las características del conjunto de datos bajo estudio pueden influir en los resultados que se obtienen al aplicar un determinado algoritmo de clasificación, obteniéndose en la aplicación de las reglas generadas un buen porcentaje de efectividad.

Los resultados de este trabajo proporcionan una alternativa para decidir qué clasificadores son los mejores para ser utilizados para un conjunto de datos con unas características particulares. Por lo tanto, las reglas propuestas en este trabajo son un recurso útil que permite reducir el tiempo dedicado a la selección del clasificador, sobre todo para aquellos usuarios que no tienen dominio sobre cómo trabajan los diferentes algoritmos y/o cómo influye la naturaleza de los datos en esta tarea. Como trabajo futuro se propone extender el estudio realizando experimentos con más bases de datos y proporcionando criterios más comprensibles para los usuarios.

Referencias

Akinola S. y Oyabugbe O. (2015). Accuracies and Training Times of Data Mining Classification Algorithms: An Empirical Comparative Study, *Indian Journal of Science and Technology*, Vol 8, No. 15, 440-447.

Ashari A., Paryudi I., Tjoa, A. (2013). Performance Comparison between Naive Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool, *International Journal of Advanced Computer Science and Applications* (IJACSA), Vol. 4, No. 11, 33-39.

Coomans D., Massart D. (1982). Alternative k-nearest neighbour rules in supervised pattern recognition: k-Nearest neighbour classification by using alternative voting rules, *Analytica Chimica Acta*, Vol. 136, 15-27.

Deza E., Deza, M. (2009). Encyclopedia of Distances, *Springer*, 94-105.

Entezari, R., Rezaei, A., Minaei, B. (2009). Comparison of Classification Methods Based on the Type of Attributes and Sample Size, *Journal of Convergence Information Technology*, Vol. 4, No. 3, 94-102.

Miquelez, T., Bengoetxea E., Larranaga P. (2004). Evolutionary Computation based on Bayesian Classifier, *International Journal Application Mathematics Computation*. Vol. No. 3, pp. 335 – 349.

Moran, S., He Y., Liu, K. (2009). Choosing the Best Bayesian Classifier: An Empirical Study, *International Journal of Computer Science* (IJCS), Vol. 36, No. 4, 25-34.

Hernández, J. Ramírez, M., y Ferri, C. (2006): Introducción a la Minería de Datos, *Ed. Pearson* Prentice Hall, 25-28.

Lichman N., (2013). UCI Machine Learning Repository, [http://archive.ics.uci. edu/ml], Irvine, CA: University of California, *School of Information and Computer Sci-ence*.

Lu, H., Setiono, R. (1997). Effective Data Mining Using Neural Networks, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 1, 957-961.

Quinlan, J. (1986). Induction of decision trees. Machine Learning, *Academic Publishers Boston* Vol. 1, 81-106.

Taylor, N. y Shawe J., (2004), An Introduction to Support Vector Machines and other kernel based learning methods, *Cambridge University Press*, 15-24.

ISSN-2531-2197

Artículo **Revista de Tecnología Informática**
Septiembre 2017 Vol.1 No.2 19-24

Sistema de apoyo para la detección de entropía económica en municipios vulnerables

CONTRERAS-Meliza*†, BELLO, Pedro, CERVANTES, Ana y MENDIETA, Roque

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computaciòn Avenida San Claudio y 14 Sur C.P. 72570, Puebla, México

Recibido Juliol 12, 2017; Aceptado Septiembre 4, 2017

Resumen

El principal problema para llevar la repartición de apoyos gubernamentales en municipios vulnerables es el desorden y mala administración de los municipios en cada jornada presidencial, lo que ocasiona que los recursos no sean otorgados en su totalidad, por lo que se propone un sistema que administre los apoyos reduciendo la entropía económica generada por el gobierno del municipio. Se generó un diagnóstico de las necesidades en los municipios, se utilizó una metodología de desarrollo de software considerando los riesgos generados por la entropía que impide asignar de forma transparente y correcta los recursos, se generó un repositorio de información en herramientas libres y se integraron varias tecnologías web para la identificación y administración amigable de los expedientes de los candidatos y beneficiarios de los apoyos. Concluimos que este sistema apoyará a la administración transparente y difusión de los apoyos s necesarios para el crecimiento de la población reduciendo las pérdidas de recursos.

Asignación de apoyos, entropía económica, plan de apoyos gubernamentales

Abstract

The main problem for the distribution of governmental support in vulnerable municipalities is the disorder and mismanagement of the municipalities in each day, which causes that the resources are not provided in their entirety, it is proposed that a system that is managing the economic supports reducing the entropy generated by the government of the municipality. A diagnosis of needs in the municipalities was generated, a software development methodology taking into account the risks generated by the entropy that precludes assignment of a transparent and correct gender resources, repository of information was made in free tools and integrated several web technologies for the identification and friendly administration of the records of applicants and beneficiaries of the bearings. We concluded that this system will support the transparent administration and dissemination of the necessary support for the growth of the population by reducing losses of resources.

Allocation of support, economic entropy, plan of government support

Citación: CONTRERAS-Meliza, BELLO, Pedro, CERVANTES, Ana y MENDIETA, Roque. Sistema de apoyo para la detección de entropía económica en municipios vulnerables. Revista de Tecnología Informática 2017, 1-2: 19-24

* Correspondencia al Autor (Correo Electrónico: pb5pbello@gmail.com)
† Investigador contribuyendo como primer autor.

Introducción

El presente trabajo se enfocó en el desarrollo de un sistema web para los municipios, tomando como ejemplo la ciudad de Puebla; con el fin de facilitar las tareas administrativas que se realizan en el mismo.

Para que una persona reciba un apoyo por parte del gobierno de algún municipio, debe ser parte de las zonas que comprende el municipio, contar con la documentación adecuada y en orden como son, comprobante de domicilio e identificación oficial vigente.

Todo esto con el fin de saber sus datos personales como son, dirección, teléfono(s), lugares de referencia, como pueden ser vecinos o algún comercio cerca de la vivienda beneficiada. Se llena el formulario del sistema junto con el tipo de apoyo solicitado.

Posteriormente la administración del departamento de obras públicas se encarga de determinar y realizar los cumplimientos adecuados para que cada solicitud sea revisada y que las personas obtengan su apoyo solicitado.

Después los arquitectos del departamento de obras públicas del ayuntamiento, visitan la vivienda para ser un examen o una investigación para saber si es verdad que necesitan el apoyo solicitado, de ser así la solicitud del apoyo llega al director de obras el cual se encarga de dar orden previa para dar apoyo, en caso contrario si la vivienda no necesita el apoyo solicitado, el arquitecto encargado de hacer el examen a la vivienda, le ofrece otro recurso que el ve adecuado para su vivienda o para apoyar a la familia.

El tiempo en que dura en llegar a la vivienda no precisamente depende del ayuntamiento, si está el recurso en mano se manda inmediatamente pero si no hasta que el recurso federal llegue al ayuntamiento.

Una vivienda beneficiada puede alcanzar 5 apoyos, los cuales no pueden ser repetitivos, en el caso de que la persona que va a pedir el apoyo llegue a fallecer, se toma como referencia un integrante de la familia que resida en el mismo domicilio, en caso de cambio de dirección solo se hace el trámite y en el sistema se hace un cambio.

Para desarrollar el sistema mencionado se analizaron e incluyeron las funciones que son más susceptibles de ser automatizadas, así como las que demandan una mayor atención dentro de los procesos de control administrativo en el H. Ayuntamiento con el fin de que se aprovechen la mayoría de los recursos y se disminuya la entropía.

El trabajo está estructurado primeramente con la introducción para familiarizar al lector con el contexto, posteriormente se define la metodolo´gia de desarrollo, los resultados y finalmente las conclusiones.

Metodología

Se realizaron encuestas en todas las comunidades que existen en el municipio de Acatzingo, Puebla y sus alrededores, con el fin de conocer las necesidades por área de cada zona que conforma al municipio y los posibles candidatos a nuevos apoyos.

Los apoyos que se ofrecen a las diferentes comunidades, se enlistan en las Tablas 1 y 2, en las cuales se pueden ver los apoyos entregados en el año 2016.

ISSN-2531-2197

Municipio	Económico	Estufas Ecológicas	Vivienda
Acatzingo	44	18	2
Villanueva	28	14	14
Actipan	35	13	9

Tabla 1 Entrega de recursos por localidades en 2016

Municipio	Material	Piso digno	Techo digno
Acatzingo	27	22	18
Villanueva	15	13	12
Actipan	9	17	12

Tabla 2 Entrega de recursos por localidades en 2016

Sustentado en una metodología robusta para el desarrollo de software como lo es el proceso unificado, se trazó un diseño que cumpliera con los requisitos y funcionalidades esperadas, perfiladas a partir de casos de uso y escenarios que establecen las características necesarias del sistema para obtener una alternativa de solución. Los casos de uso identificados para el desarrollo de la aplicación se muestran en la Figura 1:

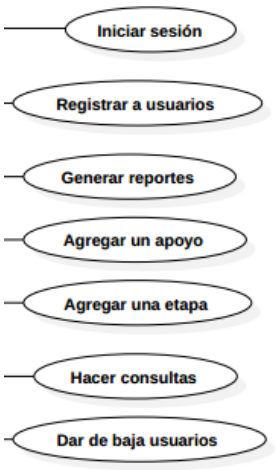


Figura 1 Casos de uso del sistema de apoyo

Así mismo, mediante la aplicación de la teoría de Bases de Datos Relaciones se elaboraron los modelos Entidad-Relación y Relacional que ayudan a implementar la Base de Datos, la cual permite ejecutar las funciones de integridad, persistencia y transacciones del sistema propuesto.

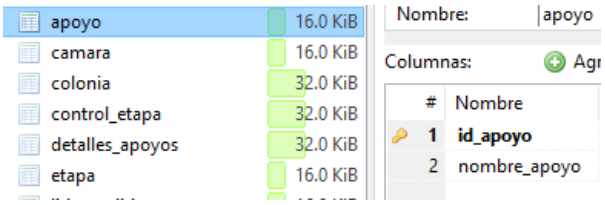


Figura 2 Tabla donde se almacenarán los apoyos

Las tablas principales de la base de datos del sistema son las siguientes como se muestran en las Figuras 2 y 3, donde se almacenan el tipo de apoyo y los detalles del mismo para la posterior obtención de reportes y toma de decisiones sobre a quien otorgarles los recursos.

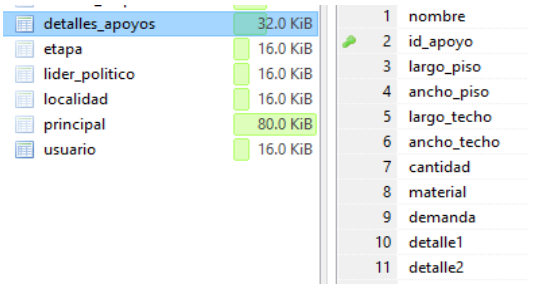


Figura 3 Tabla que almacena los detalles de los apoyos a otorgarse

Por otra parte, se plantea que la naturaleza de la herramienta propuesta, sea utilizando las tecnologías de desarrollo de software más actuales, además de cuidar el aspecto de inversión económica en su elaboración, implantación y manejo, por tanto se eligió la plataforma WAMP (acrónimo de Windows, Apache, MySQL, PHP) para su fabricación.

Se eligió a Windows como Sistema Operativo nativo en la Institución que se desea implementar; Apache como Servidor de páginas WEB; MySQL como Administrador de la base de Datos del Sistema y PHP para la implantación de las reglas de negocio.

ISSN-2531-2197

Adicionalmente se empleó HTML (Lenguaje de Marcado de Hipertexto) y CSS (Hojas de Estilo en Cascada) para complementar el desarrollo de la interfaz del usuario.

Resultados

Como se puede apreciar en la Figura 3, el menú cuenta con 6 opciones para elegir. El primero como lo dice es para Registrar a los usuarios que pidieron los apoyos, la opción de Modificar es donde podemos modificar los datos del usuario registrado más adelante veremos a detalle la pantalla de modificar y las opciones que pueden ser modificadas, en la opción de Baja es donde se da de baja "temporal" esto quiere decir que se da de baja en el sistema más no en la base de datos. En la opción de Consultar se realizarán los reportes.

Figura 4 Pantalla inicial del sistema

En la Figura 4 se muestra la opción de registro del solicitante, aquí es donde se "Registra" un nuevo usuario, ingresamos su nombre completo, los teléfonos de referencia son 3, pero obligatorio sólo es 1, se ingresa la dirección y el lugar de referencia, el tipo de apoyo; aquí hacemos hincapié que dependiendo del tipo de apoyo aparece una opción donde aparecen las especificaciones del apoyo.

Si se elige el apoyo económico, aparece el monto que se dará y la etapa en la que se le dio el apoyo; el año se comprende en 3 etapas las cuales son nombradas por el departamento que tendrá uso del sistema; si se elige techo o piso digno aparecerá la opción para dar el largo y ancho que se donara, si elegimos vivienda, estufa ecológica, material o solicitud aparece el detalle a considerar o si es una persona menos influyente**.** Por último se le toma una fotografía para reconocer quien es la persona apoyada en un futuro.

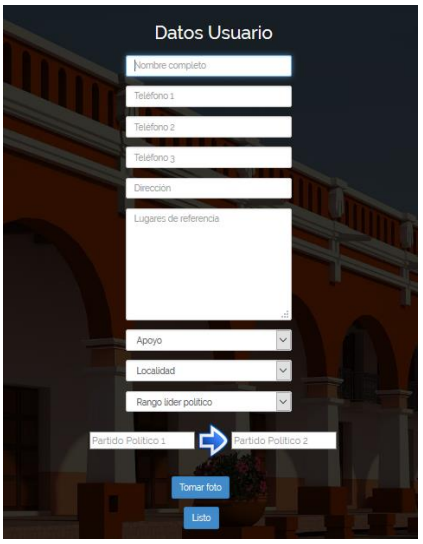


Figura 5 Registro de solicitante

En la Figura 5 se hace una búsqueda; cómo podemos ver en la búsqueda se pueden hacer consultas por nombre de usuario, por localidad, por apoyo o por etapa

Figura 6 Búsqueda de solicitante

Agradecimientos

Agradecemos el apoyo al proyecto VIEP-BUAP Estrategias de teoría de juegos cooperativos y lógica computacional para fortalecer la educación financiera en universitarios para la realización de este trabajo.

Conclusiones

El proyecto presentado proporcionó una visión general de la problemática que se presenta en un ayuntamiento, se analizaron las necesidades más demandantes y se eligieron las funciones principales para desarrollar un software que permita solucionarlas.

Se observó que la aplicación de una metodología para el desarrollo de software, nos proporciona una serie de instrumentos, que nos permiten: crear sistemas de manera más ordenada, estructurar adecuadamente las tareas que se realizan, administrar de forma más eficiente los cambios requeridos y mejorar el control de errores que se presenten; obteniendo un software de mejor calidad y facilidad para su mantenimiento, crecimiento o adaptación.

Analizamos que el paradigma Orientado a Objetos provee los métodos para modelar problemas de la realidad y transformarlos a conceptos más cercanos a la computadora; facilitándonos la labor de análisis y diseño del software; y logrando mejorar los niveles de reutilización y mantenimiento del código.

Las características tanto iterativa como incremental de la metodología del UP, así como las fases que la componen, proporcionaron un esquema de trabajo estructurado que permitió delimitar las tareas y los productos a obtener en cada etapa atendida, funcionando como una excelente guía para el desarrollo del sistema.

La plataforma WAMP para la implementación de un sistema WEB, nos proporcionó una serie de herramientas muy útiles y de fácil manejo, que permitieron transformar los modelos y diseños elaborados en etapas preliminares de la metodología UP, en pequeños sistemas o prototipos de software, permitiéndonos incrementar los mismos hasta lograr el software planificado.

Para el desarrollo de un software de calidad es muy importante que las personas involucradas no escatimen tiempo ni esfuerzos en la selección de una metodología adecuada para el desarrollo del mismo, ya que proporciona un marco de trabajo ampliamente organizado para la consecución de los objetivos planteados.

De las actividades realizadas se concluye que se lograron los objetivos planteados en el proyecto, al obtener un sistema WEB que proporciona las funciones necesarias para lograr que las tareas del departamento de Administración de Apoyos se realicen de manera más eficiente. Así mismo, al emplear tecnologías de bajo costo en su desarrollo, permitió implementar el sistema con un margen de inversión menor a las aplicaciones comerciales ofrecidas en el mercado, beneficiando su implantación en la Institución.

Referencias

Costal Costa D., (2005). Introducción al Diseño de Bases de Datos, Barcelona: Universidad Oberta de Cataluña.

Gutiérrez Ginés A. (2005). PHP5 a través de ejemplos. Madrid: RA-MA.

Kendall & Kendall (2011). Análisis y Diseño de Sistemas. México: Prentice Hall.

Orós Juan C. (2010). Diseño de Páginas Web con XHTML, JavaScript y CSS. Madrid: RA-MA.

Pressman Roger S. (2010). Ingeniería de Software. Un enfoque Práctico. México: Mc Graw Hill.

(2016). Monografias.com: Bases de datos.http://www.monografias.com/trabajos72/base-datos/base-datos2.shtml#ixzz2Lh3ECK9L

(2016). México. Coleg-ERP: Soluciones Globales en Sistemas y Cartera. http://www.coleg-erp.com/

(2016). Lenguaje Unificado de Modelado (UML). http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado

(2016). Metodologías de desarrollo de software.http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software

(2016). México. Mi-escuela.com: Adiminstración y Control Escolar en la Web. http://www.mi-escuela.com/

(2016). Wikipedia La encyclopedia Libre: Proceso Unificado (UP). http://es.wikipedia.org/wiki/Proceso_Unificado

(2016). México. School Manager: SisteMéxico. http://www.sistemexico.net/sistemas-para-escuelas/school-manager-software/

(2016). México. Systems by RR: Web vs Cliente – Servidor. http://www.systemsbyrr.com/aplicaciones-web-vs-cliente-servidor/

(2016). México. WampServer: WampServer, a Windows web development environment. http://www.wampserver.com/en/

(2016). Wikipedia La encyclopedia Libre: Windows.http://es.wikipedia.org/wiki/Microsoft_W indows

(2016). Wikipedia La encyclopedia Libre: Windows.http://definicion.de/windows/#ixzz2Ln3CE0rB

(2016). Wikipedia La encyclopedia Libre: Apache.http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

(2016). Wikipedia La encyclopedia Libre: MySQL.http://es.wikipedia.org/wiki/MySQL

(2016). Wikipedia La encyclopedia Libre: PHP.http://es.wikipedia.org/wiki/PHP

Artículo **Revista de Tecnología Informática**
Septiembre 2017 Vol.1 No.2 25-31

Clúster de computadoras de alto rendimiento usando raspberry Pi 3, para mejorar prácticas educativas

SALAZAR, Pedro*†, SOTO, Saúl y HERNÁNDEZ, Talhia

Recibido Julio 13, 2017; Aceptado Sseptiembre 13, 2017

Resumen

Un clúster de alto rendimiento permite que las aplicaciones trabajen de forma paralela, mejorando el rendimiento de las mismas. Se propone la utilización de un prototipo de clúster, siguiendo la metodología ligera para la implementación de clúster de alta disponibilidad. El clúster implementa un sistema de contenedores virtuales (Docker Swarm), para gestionar el ordenamiento de procesos y la asignación de recursos por contenedor, el clúster está formado por 5 tarjetas Raspberry Pi 3 para concentrar 20 núcleos de 1.2 Ghz con arquitectura ARM Cortex-A53 y 5 Gb de memoria RAM LPDDR2, para conectar las tarjetas se utiliza un Switch CISCO modelo 2900 y un Router CISCO modelo 1800. El clúster tiene como objetivo solucionar el problema de disponibilidad de la plataforma Moodle del Instituto Tecnológico Superior del Occidente del Estado de Hidalgo; el clúster administra y distribuye el servicio para la carrera de Ingeniería en Tecnologías de la Información y Comunicaciones, usando esta herramienta se mejora el desempeño docente e implementación de estrategias de evaluación en línea, así como la apropiación de las competencias del estudiante durante la realización de prácticas de laboratorio.

Clúster, raspberry, docker, swarm, moodle

Abstract

A high performance cluster allows applications to work in parallel, improving performance.It is proposed the use of a cluster prototype, following the light methodology for high availability cluster implementation. The cluster implements a system of virtual containers (Docker Swarm), to manage the ordering of processes and the allocation of resources by container, the cluster consists of 5 Raspberry Pi 3 cards to concentrate 20 cores of 1.2 Ghz with architecture ARM Cortex-A53 And 5 Gb of RAM LPDDR2, to connect the cards a CISCO Switch model 2900 and a router CISCO model 1800 are used. The cluster aims to solve the availability problem of the Moodle platform of the Higher Technological Institute of the West of the State of Hidalgo; The cluster manages and distributes the service for the Engineering of Information Technology and Communications, using this tool improves the teaching performance and implementation of online assessment strategies, as well as the appropriation of the student's competences during the realization of Lab practices.

Clúster, raspberry, docker, swarm, moodle

Citación: SALAZAR, Pedro, SOTO, Saúl y HERNÁNDEZ, Talhia. Clúster de computadoras de alto rendimiento usando raspberry Pi 3, para mejorar prácticas educativas. Revista de Tecnología Informática 2017, 1-2: 25-31

* Correspondencia al Autor (Correo Electrónico: psalazar@itsoeh.edu.mx)
† Investigador contribuyendo como primer autor.

Introducción

Un clúster está formado por dos o más computadoras conectadas por una red, estas computadoras generalmente distribuyen el trabajo (procesos) de manera equitativa, de tal manera que se comportan como si fuese una única computadora, un clúster de cómputo sería muy conveniente para instalar la plataforma Moodle que se necesita en el Instituto Tecnológico Superior del Occidente del Estado de Hidalgo para la impartición de cátedra y el seguimiento en línea de actividades escolares encomendadas a los alumnos.

La desventaja de utilizar un clúster principalmente es que los equipos para la implementación del mismo son muy costosos y la renta de servidores en la nube también tiene un costo elevado además de que restringen ciertas características de configuración.

Con la propuesta del clúster de alto rendimiento utilizando Docker Swarm se reducen costos, el procesamiento de datos utilizando contenedores es más ágil ya que solo utiliza un sistema operativo huésped para todos los contenedores y Docker de manera nativa administra la disponibilidad de los nodos.

En el presente trabajo se describe la propuesta de solución del clúster utilizando hardware y software open source, la metodología utilizada para la construcción física del clúster y la configuración del sistema, en los resultados actuales se describe el avance del proyecto que se tiene hasta el momento y por último se muestran los trabajos futuros para el clúster y la escalabilidad que se pretende.

Estado del arte

Durante años la virtualización de servicios se ha utilizado como la solución más eficiente, según la empresa IBM (IBM, 2017) agiliza la transferencia de información de manera segura, disminuye los costos de procesamiento al utilizar un solo equipo y reduce los riesgos en los proyectos, en la figura 1 se muestra el diagrama de la máquina virtual.

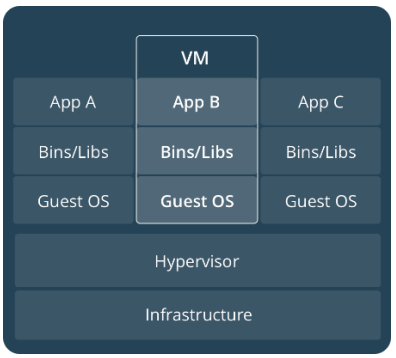


Figura 1 Diagrama de la máquina virtual. Obtenido de https://docs.docker.com/get-started/#virtual-machine-diagram

Las empresas están implementando tecnologías emergentes como Docker Swarm, es una tecnología lanzada en junio de 2016, en su versión 1.12, el portal de documentación oficial de Docker (Docker, 2017) describe que esta versión incorpora las siguientes capacidades: seguridad, actualizaciones, facilidad de uso, el clustering preparado para la producción basado en potentes tecnologías de contenedores y soporte oficial para la arquitectura ARM.

Según la documentación oficial de Docker (Docker Inc, 2017), la tecnología de contenedores ejecuta aplicaciones de forma nativa en el núcleo de la máquina host. Tienen mejores características de rendimiento que las máquinas virtuales que sólo obtienen acceso virtual a los recursos del host a través de un hipervisor. Los contenedores pueden obtener acceso nativo, cada uno ejecutándose en un proceso discreto, no teniendo más memoria que cualquier otro ejecutable, en la figura 2 se muestra el diagrama de los contenedores.

ISSN-2531-2197

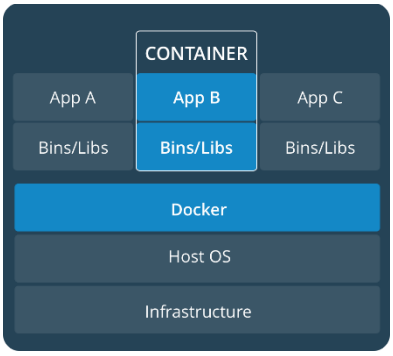


Figura 2 Diagrama de contenedores. Obtenido de https://docs.docker.com/get-started/#virtual-machine-diagram.

La optimización en el uso de recursos es la principal razón por la cual esta tecnología está teniendo gran impacto, un artículo publicado por Großmann titulado "Hypriot Cluster Lab: An ARM-Powered Cloud Solution Utilizing Docker" (Großmann, 2015), propone un clúster con arquitectura ARM y un sistema operativo derivado de debian llamado Hypriot como solución para la virtualización de servicios en la nube, en este artículo se muestra la implementación de Docker para la creación de un laboratorio de virtualización de servicios de red, ejemplifica la secuencia de configuración de un clúster de alto desempeño utilizando arquitectura ARM. McDaniel (McDaniel, 2015) en su artículo titulado "A Two-Tiered Approach to I/O Quality of Service in Docker Containers" asegura que los contenedores Linux permiten que las aplicaciones se ejecuten en completo aislamiento entre sí sin la sobrecarga adicional de ejecutar sistemas operativos totalmente independientes", por último Ismail (Ismail, 2015), en su artículo titulado "Evaluation of Docker as Edge computing platform" hace una evaluación a Docker como plataforma de cómputo, determina que basado en la evaluación y experimentación. Docker proporciona un despliegue rápido, una pequeña huella y un buen rendimiento que lo convierten en una plataforma viable de Edge Computing.

Descripción del problema

En el ámbito educativo y sobre todo a nivel universitario el uso de las tecnologías de la información toma un papel muy importante para la impartición de cátedra, la plataforma Moodle es un apoyo a la docencia presencial, Moodle es una plataforma de aprendizaje diseñada para proporcionarle a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados (Moodle, 2017).

Para operar la plataforma Moodle se recomienda contar con el siguiente hardware: 5 Gb de espacio en disco duro, un procesador de 2 Ghz y una memoria RAM de 1 Gb. El software que necesita tener instalado el servidor es el siguiente: PHP 5.4.4, MySQL 5.5.31 y Mozilla Firefox 25 (Moodle, 2016). Además del hardware y software necesario para la implementación de la plataforma Moodle, es necesario tener alta disponibilidad del servidor, alto rendimiento para el procesamiento de datos y una buena administración de la red.

El problema radica en la dificultad para adquirir un equipo de cómputo especializado con las características necesarias para instalar la plataforma de Moodle que esté siempre disponible y con el rendimiento necesario para atender a los casi 2500 alumnos del Instituto Tecnológico Superior del Occidente del Estado de Hidalgo (ITSOEH), según el portal de DELL México, un equipo con las características necesarias tendría un valor de $149,600.00 pesos MxN (Dell México, 2017), recurso con el que no cuenta el tecnológico y la solicitud podría llevar varios años.

ISSN-2531-2197

Una opción más viable es la renta de hosting en la nube, consultando el hosting de cPanel el costo anual del servicio es de $220 dólares anuales por un servidor privado y de $425 dólares anuales por un servidor dedicado (cPanel & WHM, 2017), la desventaja que tiene este tipo de servicios, son las limitaciones con respecto a la administración del servidor y de la plataforma, el hosting restringe ciertas configuraciones de personalización que podrían ser necesarias y el espacio de almacenamiento generalmente no es suficiente al paso de los años.

Propuesta de solución

Un clúster de alto rendimiento puede solucionar la problemática, para la implementación se utilizó una arquitectura ARM ya que utiliza 5 tarjetas Raspberry Pi 3, este clúster cuenta con 20 núcleos de 1.2 Ghz con arquitectura ARM Cortex-A53 y 5 Gb de memoria RAM LPDDR2, con este hardware podremos instalar un sistema operativo Hypriot y el gestor de contenedores Docker en modo Swarm que permitirá gestionar el clúster y hacer el balanceo de carga. La red contará con un router CISCO 1800 como puerta de salida a la red y como DHCP con asignaciones infinitas a los nodos del clúster, la conmutación de la red se dará por un switch CISCO 2900 con una única VLAN en el segmento de red para la conexión de los nodos del clúster.

Una vez que se ha levantado el clúster se podrá implementar la plataforma de Moodle en un contenedor de servicio web gestionado por Docker Swarm, con esto estará en condiciones de dar servicio a los alumnos del ITSOEH.

Metodología

La metodología utilizada para el desarrollo del proyecto fue una metodología propia denominada “Metodología ligera para la implementación de un clúster de alta disponibilidad”, la cual se describe en la figura 3:

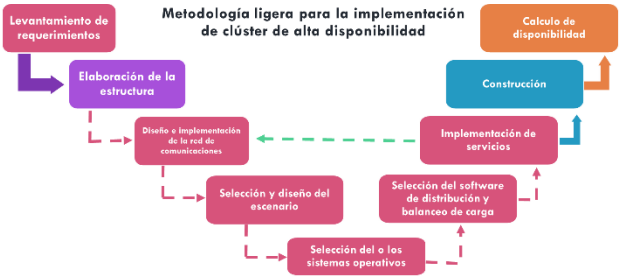


Figura 3 Diagrama de Metodología ligera para la implementación de un clúster de alta disponibilidad. Fuente propia

En la fase de levantamiento de requerimientos se tomó en cuenta la información obtenida de la documentación oficial de la plata forma Moodle y de Docker.

Para la elaboración de la estructura se llevaron a cabo las siguientes actividades:

- **Diseño y elaboración de la red de comunicaciones:** en esta fase se configuró un Router CISCO 1800, en la figura 3 se muestra la configuración del DHCP en R1; dentro del segmento 192.168.0.0/24, con asignación infinita a los clientes, excluyendo la dirección 192.168.0.69 empleada por la interface de R1 para la conexión del SSH.

```
no ip dhcp use vrf connected
ip dhcp excluded-address 192.168.0.69 192.168.0.72
!
ip dhcp pool cluster
   network 192.168.0.0 255.255.255.0
   default-router 192.168.0.69
   lease infinite
```

Figura 3 Configuración del DHCP en el router

En la figura 4 se muestra la Configuración del NAT en Router para la comunicación del clúster con Internet.

```
no ip http secure-server
ip nat inside source list 1 interface FastEthernet0/1 overload
!
access-list 1 permit 192.168.0.0 0.0.0.255
```

Figura 4 Configuración del NAT para la comunicación a internet

ISSN-2531-2197

- **Selección y diseño del escenario:** En esta etapa se determinó que el clúster estaría conformado por un maestro y 4 nodos, aprovechando las características de Docker Swarm el clúster es de alto desempeño, con las características nativas de un clúster de alta disponibilidad.

- **Selección del sistema operativo:** El sistema operativo utilizado para el clúster es Hypriot que como característica especial podemos destacar que tiene instalado por defecto Docker Swarm.

- **Seleccionar el software de distribución y balanceo de carga:** Como se ha mencionado anteriormente, se trabaja con Docker Swarm que permite gestionar, escalar, balancea la carga y hace una detección automática de los servicios en el clúster.

- **Implementación de servicios:** Hasta el momento se tienen implementados los servicios de Portainet y un servicio web para administrar el clúster y hacer algunas pruebas de escalamiento y distribución de la carga de trabajo.

Construcción: Se conectaron las tarjetas Raspberry Pi 3, Router y Switch, en la figura 5 se muestra el clúster ensamblado y conectado.

Figura 5 Conexión del clúster con el Router CISCO 1800 y Switch CISCO 2600

Una vez ensamblado el clúster se instala el sistema operativo Hypriot en las tarjetas SD, se procede a la configuración de Docker Swarm y se agregan los nodos del clúster como se muestra en la figura 6.

```
HypriotOS/armv7: pirate@master in ~
$ docker swarm join-token worker
To add a worker to this swarm, run the following command:

    docker swarm join \
    --token SWMTKN-1-34uvc8f10owkwbc241gd89n40rujqkidsa28mgulkdgy1v1yb9-1407y1w2
2z4h5mlzhhjvoxc6q \
    192.168.0.4:2377
```

Figura 6 Comando para mostrar el token del nodo Maestro; permite agregar nodos

Una vez instalados los nodos se despliegan para saber que están todos agregados al mismo clúster, si todo está bien el clúster está listo para desplegar contenedores de servicios como se muestra en la figura 7.

```
HypriotOS/armv7: pirate@master in ~
$ docker node ls
ID                           HOSTNAME     STATUS  AVAILABILITY  MANAGER STATUS
5offme25k9ghi5mvuy0umwpek    black-pearl  Down    Active
8uspblm6aq0ijcc4s7wur84e7 *  master       Ready   Active        Leader
m5m51do8todlwnpwmnnxw9823    esclavo3     Down    Active
syjbdpqfnp25gz5u28x3njd9o    esclavo2     Down    Active
v5dngaq4ypdt7fulp4g2kv401    esclavo4     Down    Active
```

Figura 7 Despliegue de los nodos agregados al clúster; "Leader" indica quien funge como coordinador del clúster

Cálculo de la disponibilidad: Según el portal de Microsoft (Microsoft, 2005), el cálculo de la disponibilidad está determinado por la siguiente formula:

$$\% \text{ de disp} = (\text{TTO} - \text{STI})/\text{TTO} \qquad (1)$$

En donde TTO es el Tiempo Total de Operación y STI es la Suma de Tiempo Inactivo, para calcular la disponibilidad del clúster propuesto se tomará una base denominada 24x7x365 ya que el clúster debe estar disponible las 24 horas del día, los 7 días de la semana y los 365 días del año. El porcentaje de disponibilidad medido hasta el momento es de 100% a un mes de prueba.

Resultados actuales

Actualmente se tiene un clúster de alto desempeño con un sistema operativo Hypriot que distribuye la carga entre 20 núcleos de 5 tarjetas Raspberry Pi 3 y 5 Gb en RAM con Docker como administrador de contenedores, balanceador de carga y gestor de disponibilidad, como se muestra en las figuras 8, 9 y 10.

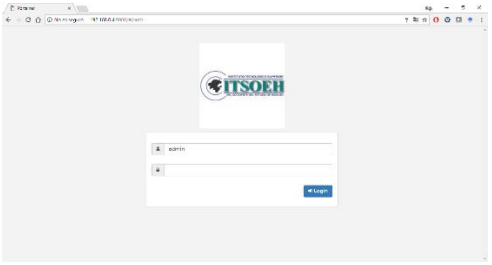


Figura 8 Pantalla de inicio del administrador grafico Portainer

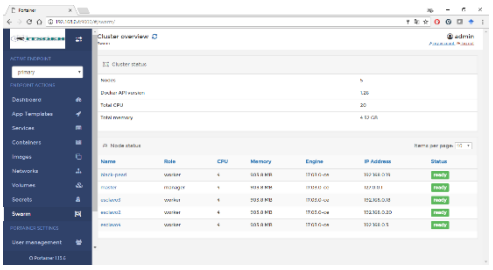


Figura 9 Administrador de nodos (Swarm)

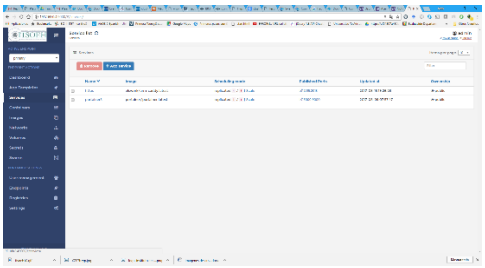


Figura 10 Servicio de servicio HTTPS Canddy corriendo en 4 de los 5 nodos

Trabajos futuros

Como trabajos futuros el clúster será escalado a 10 nodos, migración a la versión 6 de IP, contará con un traductor de IPV4 a IPV6 y se está desarrollando un contenedor de Moodle para arquitecturas ARM.

Agradecimiento

Este proyecto fue desarrollado gracias a la aportación económica del Tecnológico Nacional de México en conjunto con el Gobierno del Estado de Hidalgo, en las Instalaciones del Instituto Tecnológico Superior del Occidente del Estado de Hidalgo.

Conclusiones

En la evaluación del clúster a un mes de su implementación cumple con los requisitos de alta disponibilidad, hasta el momento no se han tenido tiempos inactivos, la disponibilidad del clúster es del 100%, la capacidad de procesamiento es buena, las gráficas indican que solo se usa el 3% de su capacidad de procesamiento y por último el balanceo de carga de trabajo es bueno, distribuye el servicio del contenedor web Canddy en 4 de los 5 nodos disponibles.

El costo del proyecto hasta el momento es de $18,401.67 pesos en comparación con los costos del servidor y de los costos del alojamiento web podemos concluir que un clúster de alto rendimiento con sistema operativo Hypriot que hospeda el gestor de contenedores Docker en modo Swarm tiene mayor viabilidad para su implementación en el ITSOEH.

Referencias

cPanel & WHM. (1 de Julio de 2017). *cPanel*. Obtenido de Plans & Pricing: https://cpanel.com/pricing/

Dell México. (16 de Agosto de 2017). *DELL México*. Obtenido de DELL México: http://www.dell.com/mx/empresas/p/poweredge-r630/pd

Docker. (18 de Junio de 2017). *Docker docs*. Obtenido de Docker docs: https://docs.docker.com/engine/swarm/#feature-highlights

Docker Inc. (22 de Junio de 2017). *Get Started, Part 1: Orientation and setup*. Obtenido de Get Started, Part 1: Orientation and setup: https://docs.docker.com/get-started/#prerequisites

Großmann, M. E. (2015). Hypriot cluster lab: an ARM-powered cloud solution utilizing docker. *23rd International Conference on Telecommunications*, 16-18.

IBM. (1 de Junio de 2017). *Virtualización de servicios*. Obtenido de IBM: https://www-01.ibm.com/software/es/rational/servicevirtualization/

Ismail, B. I. (2015). Evaluation of docker as edge computing platform. *Open Systems (ICOS)*, 130-135.

McDaniel, S. H. (2015). A two-tiered approach to I/O quality of service in Docker containers. *IEEE International Conference*, 490-491.

Microsoft. (20 de Mayo de 2005). *TechNet*. Obtenido de TechNet: https://technet.microsoft.com/es-es/library/aa996704(v=exchg.65).aspx

Moodle. (3 de Diciembre de 2016). *Moodle Docs*. Obtenido de Notas de Moodle 3.0: https://docs.moodle.org/all/es/Notas_de_Moodle_3.0#Requisitos_del_servidor

Moodle. (27 de Julio de 2017). *Moodle Docs*. Obtenido de Acerca de Moodle: https://docs.moodle.org/all/es/Acerca_de_Moodle

Artículo **Revista de Tecnología Informática**
Septiembre 2017 Vol.1 No.2 32-40

Análisis de vulnerabilidades en redes inalámbricas instaladas en diversos municipios del Estado de Hidalgo

GONZÁLEZ-MARRÓN, David†, PÉREZ-HERNÁNDEZ, Iridian, MARQUÉZ-CALLEJAS, Alejandro y BADILLO-PAREDES, Leonardo

Instituto Tecnológico de Pachuca, Felipe Angeles Km. 84.5, Venta Prieta, 42083 Pachuca de Soto, Hgo., México

Recibido Julio 27, 2017; Aceptado Septiembre 21, 2017

Resumen

En este artículo se muestra un análisis de vulnerabilidades con información recolectada en diferentes APs (access points) conectados a una red WiFi localizados en diversos municipios del estado de Hidalgo, México, identificando el nivel de seguridad inalámbrica implementada en los equipos instalados. La recolección de información se realiza utilizando la técnica de Wardriving, la cual nos muestra las características de conexión utilizadas, ubicación física y nombre asignado a cada dispositivo. Se realiza un muestreo en diversas municipios del estado y se seleccionan los atributos relacionados con la seguridad y ubicación mediante el proceso ETL (Extracción, Transformación y Cargado) , se realiza el proceso de minería de datos para obtener estadísticas de seguridad existentes en los diversos municipios analizados, se reportan los hallazgos obtenidos de forma gráfica y tabular, proporcionando el perfil de riesgos de equipos actuales en base a la evolución de las herramientas de análisis de vulnerabilidades actuales, concluyendo con predicciones acerca de la seguridad inalámbrica dentro del Estado de Hidalgo.

Wardriving, seguridad informática, minería de datos, criptografia

Abstract

This article describes a vulnerability analysis with information collected from different access points for Internet interconnection located in different municipalities of the state of Hidalgo, Mexico. The level of wireless security implemented in the installed equipment is identified. The collection of information is done using the Wardriving technique, which shows the connection characteristics used, physical location and name assigned to each device. A sampling is carried out in several municipalities of the state and the attributes related to security and location are selected by means of the ETL process (Extraction, Transformation and Loading), it is realized the data mining process which allows to obtain existing security statistics in the several municipalities analyzed using diverse methods of data analysis, reporting the findings obtained in a graphical and tabular way, providing the risk profile of current equipment based on the evolution of the analysis tools of Current vulnerabilities and concluding with predictions about wireless security within the State of Hidalgo.

Wardriving, information security, data mining, criptography

Citación: GONZÁLEZ-MARRÓN, David, PÉREZ-HERNÁNDEZ, Iridian, MARQUÉZ-CALLEJAS, Alejandro y BADILLO-PAREDES, Leonardo. Análisis de vulnerabilidades en redes inalámbricas instaladas en diversos municipios del Estado de Hidalgo. Revista de Tecnología Informática 2017, 1-2: 32-40

† Investigador contribuyendo como primer autor.

Introducción

Considerando el gran auge que ha tenido internet en los últimos años es notable el incremento y el fácil acceso a éste.

La tecnología Wi-Fi (Wireless Fidelity) es una de las tecnologías líder en la comunicación inalámbrica, incorporandose en cada vez más aparatos portátiles. Pero un aspecto que en ocasiones pasa desapercibido es la seguridad [1].

Tomando en cuentan a la población total en México en el año 2016 se determinó que el 59.5% tiene acceso a internet [2], al ser un número considerable de usuarios conectados, se toma en cuenta cierto tráfico en la red incluyendo información de todo tipo que se transmite al navegar por este medio, en la Gráfica 1 se muestra el aumento que ha tenido internet en los hogares de México desde el año 2013 al 2016.

Gráfico 1 Hogares con conexión a Internet, INEGI. Modulo sobre disponibilidad y uso de tecnologías de la información en los hogares [3]

Debido a este incremento, es cada vez mas importante considerar la seguridad de las comunicaciones, pues los datos al estar transmitiéndose dentro del área de influencia del AP (Access Point) generan diversos tipos de riesgo que el atacante puede explotar, debido a que los datos son transmitidos a través del aire, este es tema abordado por la seguridad informática.

Cabe destacar que existen diferentes métodos de protección de redes que van desde los más simples hasta los más robustos, en la actualidad existen diferentes métodos de protección de redes que van desde las encriptaciones más simples hasta las más robustas, la primer encriptación de WiFi implementada fue la WEP (Wired Equivalent Privacy) la cual fue implementada por el estándar IEEE 802.11 en 1999 [1], sin embargo aunque fue un buen intento para lograr seguridad en las comunicaciones WiFi, su implementación no fue bien realizada, debido a que es vulnerable a ataques.

WPA y WPA2 (Wireless Protected Access), implementación de una versión temprana del estándar 802.11i, basada en el protocolo de encriptación TKIP [1], se basa en la autenticación de usuarios mediante el uso de un servidor, para ello se almacenan credenciales y contraseñas de los usuarios de la red [4];

La diferencia de WPA [5] frente a Wep es que la clave precompartida solo se envía una vez y no como en WEP, donde el envío de la llave es constante.

ISSN-2531-2197

Otro mecanismo de seguridad es el anunciar la existencia de un equipo o no, los equipos que anuncian su existencia lo realizan mediante un SSID, el cual es un acrónimo de (Set Service IDentifier) y permiten que sean vistos por dispositivos que utilizan tarjetas que permitan el uso del WiFi, se considera que aquellos equipos que ocultan su SSID tienen un mecanismo de protección básico ya que la mayor parte de los equipos ignoraran su existencia

Mediante este análisis, se comprueba que existe una muestra considerable de la comunidad de usuarios en el estado de Hidalgo que no cuentan con los conocimientos suficientes para la protección de sus redes inalámbricas.

Para este análisis fue requerido aplicar el método denominado Wardriving, el cual consiste en la detección de redes inalámbricas dentro de una zona geográfica, este es realizado habitualmente con un dispositivo móvil, una laptop, un PDA (Asistente Digital Personal) o por teléfonos celulares [6].

El análisis simplemente se realiza con el dispositivo móvil y en el momento que se detecta la existencia de una red, procede a hacer un estudio de la misma ubicando los puntos de acceso, anidada la información de las características hardware del punto de acceso inalámbrico (AP).

Gracias al método fue posible la obtención de datos proporcionadas por las lecturas de la aplicación "Wigle Wifi Wardriving" disponible para dispositivos con Sistema Operativo Android, las cuales son tratadas en esta investigación con el proceso ETL.

Esta información proporciona a la investigación datos muy relevantes pues a manera estadística podemos determinar las áreas más vulnerables dentro del estado de Hidalgo.

Trabajos relacionados

Existe un artículo realizado en la ciudad de Santiago de Cali en el país de Colombia elaborado por grupo de investigación COMBA I+D [7], en éste se encuentra analizada la seguridad de las redes Wi-Fi, sin embargo no se menciona el proceso de distinción de los datos y se utilizan métodos de análisis convencionales.

Un segundo artículo similar al presente, es llamado "Wardriving: an experience in the city of La Plata" elaborado para LINTI, Facultad de Informática, Universidad Nacional de La Plata, La Plata, Buenos Aires, Argentina [8] en donde también se hace estudio de redes para interpretar la seguridad en dicha ciudad.

También se encuentra un trabajo realizado en Tunja, Boyacá, Colombia para la Universidad Nacional de Colombia, realizando un análisis más a profundidad y dando resultados más gráficos [9].

En este artículo y el objeto diferenciador es que se ha buscado hacer una correcta integración de la información, así como de su adecuada implementación, preservando la integridad, consistencia y disponibilidad la misma.

Pruebas de wardriving a equipos inalámbricos

El propósito es representar los datos esquemáticamente con respecto a la seguridad que se presentan en los equipos inalámbricos además de generar conciencia de los riesgos que representa no tener seguridad en ellos y que en próximos análisis esos resultados mejoren.

Para la realización del análisis, se solicitó apoyo a estudiantes para realizar las pruebas con las técnicas de WarWalking y WarDriving.

ISSN-2531-2197

Fue seleccionada la aplicación para dispositivos móviles. "Wigle Wifi Wardriving" [10], debido a que en su mayoría los estudiantes contaban con celulares con el sistema operativo Android y ésta aplicación permite obtener información de los equipos WiFi y generar mapas de los equipos detectados, en la Figura 1 se muestra una de las diferentes pantallas de las que consta esta aplicación, pudiéndose almacenar los registros obtenidos en diferentes opciones de exportación, en nuestro caso fue solicitada la exportación de los dispositivos detectados en el formato csv (comma sepparated values).

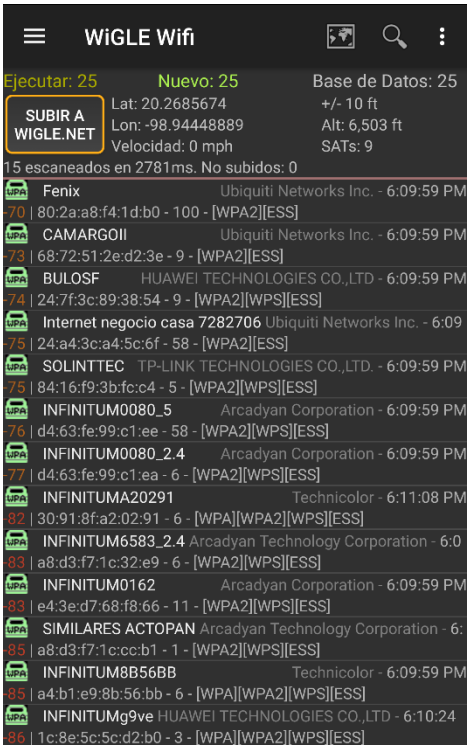


Figura 1 Aplicación WiGLE WIFI

Realización de pruebas de wardriving dentro del estado de Hidalgo

Se recolectó información utilizando wardriving en los distintos municipios donde los estudiantes residen, a fin de conocer el grado de seguridad que se maneja en los equipos existentes con tecnología WiFi, y lograr una conciencia más profunda de la seguridad en redes inalámbricas, se analizaron 60 localidades del estado, pertenecientes al 17.8% de los municipios que existen en el estado de Hidalgo [11], aunque se logró una muestra de los municipios más importantes, faltó el municipio de Tulancingo uno de los más importantes del estado, debido a que de los estudiantes seleccionados ninguno residía en ese municipio.

En la Tabla 1 se muestran los municipios analizados en la prueba del Wardriving, y las colonias pertenecientes a dicho municipios.

Municipios	No. Loc.	Localidades
Actopan	3	Actopan, Cañada Chica Antigua, La Palma
Atotonilco el grande	2	La Puebla, Atotonilco el grande
El Arenal	4	El Jiadi, El Arenal, El Pozo (Santa Ana), San José Tepenene
Huasca de Ocampo	3	Cruz Blanca, Huasca de Ocampo, La mora
Ixmiquilpan	1	Ixmiquilpan
Mineral del monte	2	Barrio del Agua Escondida, Mineral del monte
Pachuca de Soto	15	Cerro de Guadalupe, El Venado, Colonia las Campanitas., Pachuca de Soto, Ejido San Antonio, Ejido San Bartolo, El Huixmí, El Roble, Fraccionamiento Valle del Sol, Hilario Monzalvo Roldán, La Rabia, Los Chávez, Maluco, Pitayas, San Pedro Nopancalco
Zapotlán de Juárez	2	Acayuca, Santa María
Zempoala	2	La Isla, Zempoala
Mineral de la Reforma	19	Azoyatla de Ocampo, Bosques del Mineral, Carboneras, El Popolito, Guadalupe Minerva, La Colonia, San Miguel la Higa, Privada Quinta Bonita, Privadas del Parque, Real de Oriente, Rinconada los Álamos, Rinconadas de San Francisco, Rincones del Paraíso, San Guillermo la Reforma, San José Palma Gorda, Santiago Jaltepec, Unidad Habitacional CTM, Unidad Minera 11 de Julio, Valle Dorado.
San Salvador	2	Caxuxi, San José Doxey
Tepeapulco	2	Fray Bernardino de Sahagún (Ciudad Sahagún), Guadalupe
San Agustín Tlaxiaca	1	San Juan Solís
Cuautepec de Hinojosa	1	Santa Rita
Francisco I. Madero	1	Tepatepec

Tabla 1 Municipios y localidades analizadas

ISSN-2531-2197

Preparación de Datos utilizando ETL

Para el proceso de ETL (Extraction, Transformation and Loading) el equipo que realizó el wardriving, entregó un archivo de cada uno de los sitios analizados en formato CSV, al haber utilizado todos el mismo software y la misma opción de almacenamiento, hubo una estandarización en los datos, lo que facilitó el proceso de integración de la información recolectada, sin embargo aún así se encontraron archivos con datos corruptos u opciones que se salieron de lo especificado, sin embargo fueron muy pocos estos casos. Una vez que se tuvieron los archivos corréctos se unieron estos registros en uno solo para hacer el proceso de análisis. Las principales actividades llevadas en la realización del proceso ETL fueron las siguientes:

- Identificar y eliminar archivos fuera de lo solicitado o corruptos.

- Eliminar registros de sitios que habían sido analizados por otra persona, este caso fue muy frecuente, debido a que, aunque cada persona tenía una ruta diferente asignada, había traslape en algunas zonas, principalmente en los municipios con más habitantes, de maś de 14000 equipos analizados, se redujo la cantidad a 7562 equipos.

- Se eliminaron columnas que eran innecesarias para el análisis de vulnerabilidades.

- Se eliminaron datos de equipos que no fueran de tipo WiFi.

- Se procedieron a identificar las localidades analizadas en base a sus coordenadas proporcionadas por el GPS, este proceso requirió hacer una búsqueda de una aplicación que nos diera esta posibilidad, utilizando para esto Maplarge [12].

- Posteriormente se requirió que las localidades proporcionadas se ubicaran a los municipios del estado.

Uno de los procesos mas consumidores de tiempo fue la identificación mediante la posición absoluta de las localidades y su posterior ubicación a uno de los 60 municipios del estado, este proceso se debió relizar manualmente debido a que no se encontró una aplicación gratuita que nos realizara esta operación de manera automática, el proceso se puede apreciar en la Figura 2 mostrada a continuación.

Figura 2 Proceso ETL

Otra herramienta utilizada fue el software Pentaho [13] para poder realizar la transformación de nuestro archivo CSV a un archivo ARFF del acrónimo en ingles (Attribute-Relation File Format) utilizado por el software de Weka [14], de donde se procedió a generar los resultados reportados en este trabajo, en la Figura 3 se muestra el proceso realizado con Pentaho y su correcta transformación de 7562 registros.

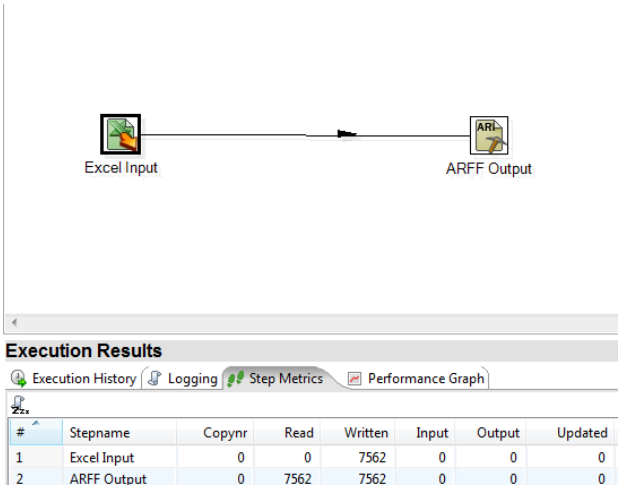


Figura 3 Proceso de transformación a un archivo ARFF

Resultados Obtenidos

En la Tabla 2 se muestra la clasificación utilizada para ubicar los equipos analizados con respecto a su seguridad.

Seguro Equipos con cifrado WPA o WPA/2.
Inseguro Oculto Equipos con SSID Oculto con cifrado WEP
Seguro Oculto Equipos con SSID Oculto y cifrado WPA y WPA2
Inseguro Equipos con cifrado WEP o ESS

Tabla 2 Clasificación de tipos de vulnerabilidades utilizadas para análisis de APs

En la Tabla 3 se muestran los resultados obtenidos por municipio de la clasificación de seguridad realizada:

– Pachuca la capital de Hidalgo, de 3771 AP's, el 65.23% son seguros, el 24.48% son seguros con el SSID oculto, el 9.25% son inseguros y el 1.03 son inseguros con SSID oculto.

– Mineral de la Reforma los 1240 AP's el 70.48% son seguros, el 20.81% son seguros con el SSID oculto, el 8.15% son inseguros, y el 0.56% son son inseguros con SSID oculto.

– Tepeapulco con 1207 AP's el 47.39% son seguros, el 29.41% son seguros con el SSID oculto, el 6.88% son inseguros, y el 16.32% son inseguros con SSID oculto.

Grado de Seguridad por Municipio					
	INSEGUROS	INSEGUROS OCULTOS	SEGUROS	SEGUROS OCULTOS	Total
Actopan	42	11	269	120	442
Atotonilco el Grande	24	2	154	81	261
Cuautepec de Hinojosa	1	0	15	8	24
El Arenal	3	3	71	41	118
Francisco I. Madero	17	3	105	39	164
Huasca de Ocampo	7	0	35	10	52
Ixmiquilpan	4	0	23	7	34
Mineral de la Reforma	101	7	874	258	1240
Mineral del Monte	3	1	11	0	15
Pachuca de Soto	349	39	2460	923	3771
San Agustín Tlaxiaca	0	0	5	1	6
San Salvador	1	1	53	15	70
Tepeapulco	83	197	572	355	1207
Zapotlán de Juarez	1	0	1	0	2
Zapotlán de Juárez	5	0	74	40	119
Zempoala	0	1	19	17	37
Total	**641**	**265**	**4741**	**1915**	**7562**

Tabla 3 Valoración del nivel seguridad de equipos WiFi en municipios del estado de Hidalgo

Posteriormente con los datos obtenidos, se utilizó una plataforma web llamada CARTO, que nos permite subir las coordenadas obtenidas y observarlas como puntos geográficos en un mapa [15].

Se utiliza la siguiente convención para mostrar la seguridad para el tipo de equipos analizados

- Equipos Seguros (color verde)
- Equipos Seguros Ocultos (color azul)
- Equipos Inseguros (color rojo)
- Equipos Inseguros Ocultos (color amarillo)

En la Figura 4 se muestra un mapa del Estado de Hidalgo, donde se reflejan principalmente los sitios donde se encuentran equipos mal configurados (inseguros) que los hacen vulnerables a ataques por parte de usuarios maliciosos.

ISSN-2531-2197

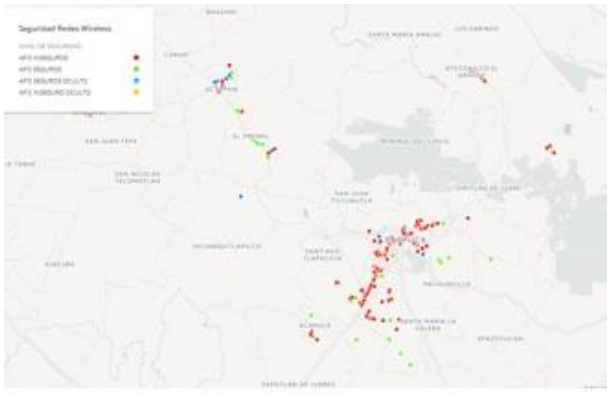


Figura 4 Mapa de seguridad en equipos WiFi analizados dentro del estado de Hidalgo (Enfatizando equipos inseguros)

En la Figura 5 se muestran los equipos que se encuentran adecuadamente configurados en el Municipio de Pachuca

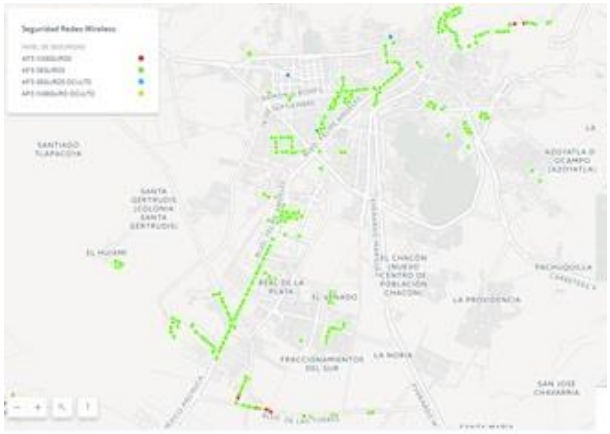
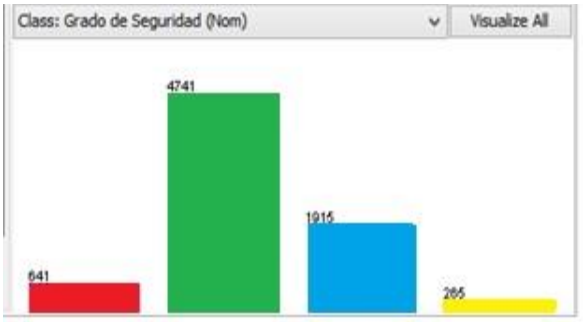


Figura 5 Mapa de seguridad en equipos WiFi analizados dentro de Pachuca (Enfatizando equipos seguros)

En la Gráfica 2 obtenida con el software de Weka pueden verse reflejados los resultados de todos los municipios del estado de Hidalgo encontrándose redes inseguras con un 8.48%, redes seguras con un 62.70%, redes seguras ocultas con el 25.32% y redes inseguras ocultas con un 3.5%.



Gráfica 2 Grado de seguridad en WiFi de municipios analizados en el estado de Hidalgo

Para la realización de minería se procedió a utilizar una clasificación utilizando algoritmos de conjuntos de datos disjuntos, probabilísticos y jerárquicos, habiéndose utilizando los métodos de agrupamiento (clustering) siguientes: (Kmeans, Xmeans y Cobweb). Para la realización de éstos métodos se utilizaron datos nominales, removiendo de los datos el grado de seguridad asignado de manera manual al momento de hacer el proceso ETL. Se analizaron diferentes grupos de datos, obteniéndose un mejor resultado con los atributos nominales (Authmode, Localidad y Ciudad-Municipio). A fin de validar con cual de los métodos se logra una mejor clasificación, se muestra en la Tabla 4 una comparación de los resultados obtenidos con el proceso de minería con respecto a la clasificación realizada manualmente. Como puede ser visto se logró una clasificación muy similar con el método SimpleKMeans utilizando Weka con 11 semillas (seeds) que son utilizadas para inicializar los clusters y que afectan el proceso de clasificación con este método. Asi mismo se establecieron 4 clusters para hacer una clasificación similar a la realizada manualmente, el método de maximización de expectación (EM) reporta una clasificación muy diferente a la obtenida con el SimpleKmeans, entregando resultados poco satisfactorios. El método Cobweb con valores de default genera un número superior a los 1000 clusters, por lo cual no se reportan los resultados.

ISSN-2531-2197

Grado de Seguridad	Manual	Simple-Kmeans (10 seeds)	Simple-Kmeans (11 seeds)	Simple-Kmeans (12 seeds)	Simple-Kmeans (13 seeds)	EM
Inseguro	641	1210	646	1109	2695	1207
Seguro	4741	4292	3488	3009	964	3676
Seguro-Oculto	1915	1975	3177	2566	3527	2561
Inseguro-Oculto	265	85	251	878	376	118
TOTAL	7562	7562	7562	7562	7562	7562

Tabla 4 Comparación de la clasificación realizada manualmente con la obtenida con algoritmos de minería

En la Figura 6 se muestran los resultados obtenidos con el método Kmeans que mejores resultados reportó.

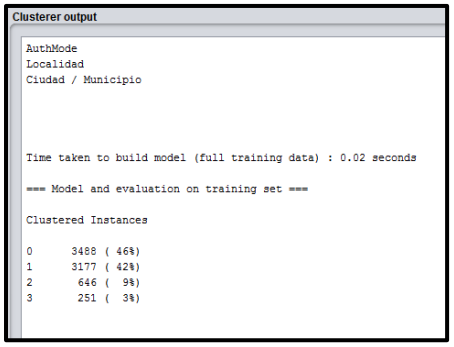


Figura 6 Resultados de minería obtenidos con la clasificación obtenida en SimpleKmeans en Weka

Aunque puede ser visto que existe una variación significativa entre la clasificación realizada automáticamente por SimpleKmeans entre equipos "seguros" y "seguros ocultos", aún así es importante considerar que todos son considerados equipos seguros. Con respecto a la clasificación realizada para equipos inseguros, la diferencia es significativamente menor, encontrándose valores casi similares, en equipos "inseguros" y en equipos "inseguros ocultos".

Puede concluirse que en los equipos analizados predominan los equipos con seguridad con un 88 % y sólo un 12 % con equipos inseguros, durante el análisis pudo ser visto que los equipos recientemente instalados vienen configurados con mejores parámetros de seguridad (encriptación WPA y WPA2).

Que ciertas empresas como Telmex y Totalplay entregan sus nuevos equipos con conexión a internet con configuraciones seguras, mientras que en lugares donde se cuenta con conexiones menos recientes las configuraciones tienden a ser más inseguras. Se encontró igualmente que hay ciertas colonias que tienden a tener configuraciones muy seguras, generalmente en fraccionamientos nuevos con servicios de internet recientemente instalados, predominando el SSID oculto y algoritmos de cifrado robustos.

Referencias

[1] Guillaume Lehembre. (Enero 2006). Seguridad Wi-Fi – WEP, WPA y WPA2. Julio 2017, de hakin9 Sitio web: http://www.zero191513wireless.net/wireless/seguridad/01_2006_wpa_ES.pdf

[2] INEGI De 2013 a 2014: INEGI. Módulo sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares.

[3] Para 2015-2016: INEGI. Encuesta Nacional sobre Disponibilidad y Uso de TIC en Hogares, ENDUTIH.

[4] Lei Z., Jiang Y., Zugao D. and Renfe Z.(2012), The security analysis of WPA encryption in wireless network, Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference. sitioweb:http://doi:10.1109/CECNet.2012.6202145

[5] Lashkari A., Mansoor M. and Danesh (2009), A., Wired EquivalentPrivacy (WEP) versus Wi-Fi Protected Access (WPA), 2009 International Conference on Signal Processing Systems. Sitioweb: http://doi:10.1109/ICSPS.2009.87

ISSN-2531-2197

[6] Universidad Central de Venezuela. (2005). Seguridad en Redes Inalámbricas 802.11. 10/08/2017, de Universidad Central de Venezuela Sitio web: http://www.ciens.ucv.ve:8080/genasig/sites/redesmov/archivos/Seguridad%20en%20Redes%20Inalambricas%20802.pdf

[7] Millán A.; Daza R.; Campiño J. (2006). Estudio de los puntos de acceso inalámbricos 802.11 en la ciudad de Cali usando las técnicas WAR-X. Sistemas & Telemática, Enero-Junio, 35-42.

[8] Díaz J., M., Venosa P., Macia N. (2017). Wardriving: an experience in the city of La Plata. 8/2017, de LINTI, Facultad de Informática, Universidad Nacional de La Plata, La Plata, Buenos Aires, Argentina Sitio web: http://sedici.unlp.edu.ar/bitstream/handle/10915/21678/Documento_completo.pdf?sequence=1.

[9] Julián Alberto Monsalve-Pulido a, Fredy Andrés Aponte-Novoa b & Fabián Chaparro-Becerra c. (November 19th, 2014). Security analysis of a WLAN network sample in Tunja, Boyacá, Colombia. DYNA, 1.

[10] Wigle Wifi Wardriving (2010).Consultado 8/Junio/2017,WiGLE.net.sitioweb:https://wigle.net/

[11] Municipios de México (2017).consultado 8/Agosto/2017,MUNICIPIOS.sitioweb:https://www.municipios.com.mx/hidalgo.

[12] MapLarge (2017).Consultado 7/Agosto/2017,MAPLARGE.sitioweb:https://www.maplarge.com.

[13] Pentaho (Septiembre 2014).Consultado 7/Agosto/2017,Pentaho A Hitachi Group Company.sitioweb: http://www.pentaho.com/

[14] Weka (1993).Consultado 10/Agosto/2017, Universidad de Waikato de Nueva Zelanda.sitioweb: http://www.cs.waikato.ac.nz/ml/weka/

[15] JAVIER DE LA TORRE (2012).Consultado 17/Agosto/2017,CARTO, sitioweb:https://www.carto.com

ISSN-2531-2197

Artículo **Revista de Tecnología Informática**
Septiembre 2017 Vol.1 No.2 41-50

Determinación de parámetros que impiden una implementación eficiente de algoritmos criptográficos en ambiente multiplataforma

GONZÁLEZ-MARRÓN, David†, GAMERO-PLAFOX, Benito, LÓPEZ-MELO, Eduardo y AGUILAR-GÓMEZ, José

Instituto Tecnológico de Pachuca, Felipe Angeles Km. 84.5, Venta Prieta, 42083 Pachuca de Soto, Hgo., México

Recibido Julio 4, 2017; Aceptado Septiembre 7, 2017

Resumen

En este artículo se analizan los problemas que se presentan en el desarrollo de un algoritmo criptográfico simétrico por bloques con un tamaño de llave máxima de 16 caracteres ASCII, que realiza el cifrado de textos de longitud variable en diferentes lenguajes y plataformas, los lenguajes seleccionados para el desarrollo son C++, Java y C#, el algoritmo es probado en los sistemas operativos de Windows y Linux, se analizan los problemas de compatibilidad que se generan al realizar el proceso de sustitución de símbolos, así como los convenientes e inconvenientes que se presentan entre lenguajes. Se analizan los aspectos relativos a especificaciones funcionales requeridas para trabajar en un ambiente de multiplataforma. El algoritmo parte de una especificación general que los desarrolladores deben interpretar para su implementación en cada lenguaje, lo que conlleva cambios significativos en su implementación. Se detalla el desempeño obtenido en cada implementación realizada en los lenguajes de programación utilizados, así como las pruebas utilizadas para verificar el comportamiento del algoritmo bajo diferentes situaciones.

Ingeniería de Software, criptografía, ambiente multiplataforma

Abstract

Throughout this article are analyzed the problems presented in the development of a symmetric cryptographic block algorithm with 16 ASCII characters maximum key, the algorithm realizes the the encryption of variable length texts in different languages and platforms, the languages selected for development are C ++, Java and C #. The algorithm is tested in Windows and Linux operating systems, there are analyzed the compatibility problems generated when performing the process of symbols substitution as well as the conveniences and inconveniences presented between languages. Aspects related to functional specifications required to work in a multiplatform environment are analyzed. The algorithm starts from a general specification that developers must interpret for their implementation in each language, which entails significant changes in its implementation. It is detailed the performance obtained in each implementation performed in the programming languages used, as well as, the tests used to verify the behavior of the algorithm under different situations and the results of the implementation.

Software engineering, criptography, multiplatform environment

Citación: GONZÁLEZ-MARRÓN, David, GAMERO-PLAFOX, Benito, LÓPEZ-MELO, Eduardo y AGUILAR-GÓMEZ, José. Determinación de parámetros que impiden una implementación eficiente de algoritmos criptográficos en ambiente multiplataforma. Revista de Tecnología Informática 2017, 1-2: 41-50

† Investigador contribuyendo como primer autor.

Introducción

Actualmente la cantidad de información que se está generando, está alcanzado niveles muy altos, como puede ser visto en la Figura 1, además de que ésta información es almacenada y transmitida en diversos dispositivos que se encuentran diseminados en el internet.

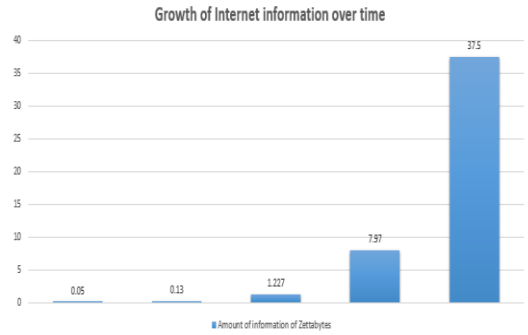


Figura 1 Crecimiento de información en Zbytes

Es por eso que se requiere de uso de mecanismos de protección que preserven las propiedades de Confidencialidad, Integridad y Accesibilidad de la información (CIA), el uso de las técnicas de encriptación nos ayuda a conservar dos de estas propiedades (Confidencialidad e Integridad) de allí la enorme importancia de la encriptación, los desarrolladores de implementaciones de algoritmos de criptografía tienen un reto significativo para hacer algoritmos robustos y rápidos en diversas plataformas. El objetivo del presente trabajo es la identificación de los principales parámetros que pueden llevar a una implementación deficiente de un algoritmo criptográfico.

El análisis de dichos parámetros se ha hecho desde el punto de vista técnico, evaluando el rendimiento de la implementación en diversos lenguajes de programación y bajo diversos sistemas operativos; y desde el punto de vista humano, identificando los retos cuyo origen son deficiencias de ingeniería de software.

"La ingeniería de software comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de estos después de que se utiliza" [1] partiendo de esa definición, resulta evidente que la implementación de algoritmos criptográficos en un marco de calidad se verá afectado por aspectos netamente humanos.

Se seleccionaron tres de los lenguajes de programación más conocidos, como son (Java, C# y C++) a fin de elegir el lenguaje más adecuado para el problema de desarrollo de algoritmos de encriptación.

Los lenguajes de programación basados en el lenguaje de programación ANSI C permiten que el software desarrollado sea compatible con diversos Sistemas Operativos. Pero en cuestiones de rendimiento, los lenguajes de programación son totalmente distintos porque algunos se ejecutan bajo el uso de máquinas virtuales, algunos otros lenguajes son compilados o interpretados, incluso las librerías con las que el lenguaje funciona afectan su rendimiento.

La encriptación puede conceptualizarse como una forma de proteger la información a aquellos que no deben tener acceso. El arte de encriptar vio su nacimiento al tiempo que lo hacia la escritura [2], sin embargo, no ha sido hasta épocas más recientes en que la encriptación informática se ha convertido en la forma predilecta de asegurar las transacciones electrónicas.

A lo largo de los años se han desarrollado múltiples algoritmos de cifrado con características específicas que los diferencian. Una de las características más notables en un algoritmo de cifrado es la clave, de hecho, partiendo de la clave es cómo se han podido clasificar los algoritmos en simétricos, aquellos que emplean la misma clave para cifrar y descifrar, y algoritmos asimétricos, aquellos que emplean una clave pública y una privada.

ISSN-2531-2197

El algoritmo seleccionado para realizar esta investigación es de tipo simétrico por bloques y se detalla posteriormente.

Trabajos relacionados

En 2014 Philipp Holtkamp et al [3], realizaron un estudio partiendo del hecho de que el factor humano es el origen de la mayoría de retos en los proyectos de software. Resultado de dicho trabajo se concluyó que todas las competencias de internacionalización son importantes para el desarrollo de software, sin embargo, se ha encontrado que tienen especial importancia en las fases con elementos colaborativos y creativos. En ese trabajo se evaluó el efecto que tienen las deficiencias en las competencias analizadas en la implementación colaborativa de un algoritmo criptográfico.

Encriptación simétrica por bloque

Los algoritmos de cifrado simétrico por bloques se basan en dividir el mensaje en bloques de tamaño fijo y a partir de aquí poder realizar operaciones matemáticas u operaciones lógicas (principalmente sumas utilizando la función XOR) utilizando una clave secreta, con la cual se realizan dichas operaciones. Se recomienda que el tamaño de bloque sea considerablemente grande para poder evitar algún tipo de ataque. Se tiene que verificar que, durante el proceso de cifrado, el tamaño de bloque no cambie, para que este pueda ser reversible y no se tenga problemas a la hora de descifrarlo.

Codificación de caracteres en los lenguajes de programación

Las computadoras manejan internamente toda la información como dígitos binarios por lo que la representación de caracteres se hace mediante números [4]. La codificación de caracteres se basa en establecer una relación uno a uno entre un carácter y un valor numérico.

La evolución de la codificación en los lenguajes de programación se vio impulsada por el surgimiento de los GSD (Global Software Developments) Desarrollos de Software Global, por lo que los lenguajes de programación han empleado diversas codificaciones a lo largo de los años.

El lenguaje Java emplea el estándar Unicode y define el tipo primitivo char como un tipo de dato de 16 bits, con caracteres entre el rango hexadecimal de 0x0000 a 0XFFFF [5].

Visual C# emplea Unicode para almacenar los caracteres y cadenas, sin embargo también puede manejar caracteres en codificación ANSI, que usan un byte para representar un carácter y se limita a 256 caracteres, o ASCII [6].

C++ emplea el código ASCII que proporciona códigos para representar el idioma inglés [7]. Adicionalmente, C++ permite emplear Unicode para aplicaciones internacionales mediante el tipo wchar_t.

Realización de un algoritmo de encriptación por bloque

La especificación original del algoritmo de encriptación requerido se muestra a continuación.

Desarrollar un algoritmo que haga una transposición de valores, utilizando el último elemento como el primero, en base al tamaño seleccionado, una vez realizado esto, hacer una operación de suma con una llave de hasta 16 caracteres (ASCII), la llave será particionada en base al tamaño del bloque seleccionado y en caso de ser una llave menor que el tamaño del bloque deberá ser completada por blancos, los valores válidos a encriptar estarán dados por la tabla ASCII, con un máximo de 255 valores.

ISSN-2531-2197

El algoritmo deberá ser válido para cualquier tamaño de archivo o string. Deberá contar con una interface para hacer el cifrado y descifrado del mensaje y deberán ser programas que puedan ser independientes uno de otro.

Dicha especificación al ser tan genérica dio origen a diversas interpretaciones de parte de los desarrolladores, a continuación, se detallan dos diferentes diagramas de flujo a fin de ver las diferencias que se presentan.

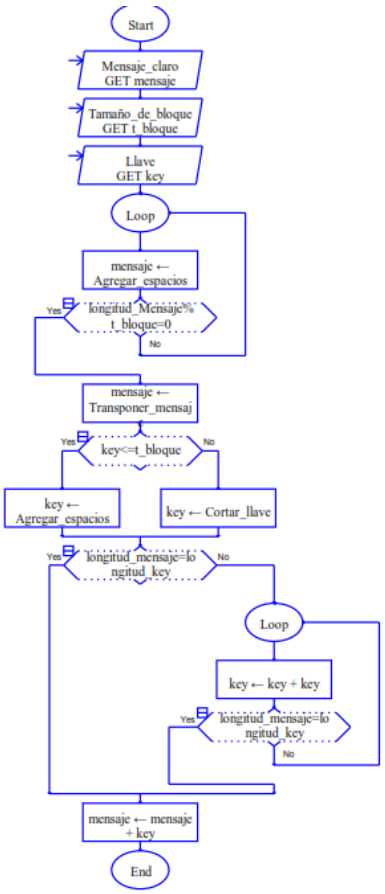


Figura 2 Diseño de algoritmo por programador 1

La primera interpretación que se ilustra en la Figura 2. hace un ajuste al mensaje y a la llave para que su longitud sea múltiplo del tamaño de bloque. Posteriormente se repite la llave hasta que la longitud es igual a la longitud del mensaje, se suma la llave con el mensaje transpuesto y se obtiene el mensaje cifrado.

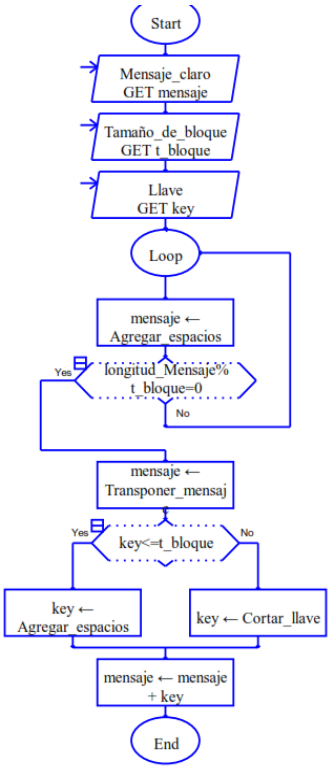


Figura 3 Diseño de algoritmo por programador 2

La segunda interpretación se ilustra en el la Figura 3., la cual consiste en ajustar el mensaje para obtener una longitud de mensaje que sea múltiplo del tamaño de bloque, transponer el mensaje, ajustar la clave según su tamaño respecto al tamaño de bloque y sumar la llave con el mensaje.

Derivado de los problemas de interpretación del algoritmo fue requerido replantear el mismo, resultando el proceso de detalle con el propósito de que las implementaciones fueran equivalentes.

1. Se ajusta el mensaje agregando espacios hasta que su longitud sea un múltiplo del tamaño del bloque.

M = MENSAJE b = 3 K = CLAVE

M	E	N	S	A	J	E		

2. Se divide el mensaje en segmentos del tamaño del bloque.

M	E	N	S	A	J	E		

3. Se transpone cada uno de los segmentos.

N	E	M	J	A	S			E

4. Se unen los segmentos para originar el mensaje.

N	E	M	J	A	S			E

5. Se repite la contraseña hasta que su longitud sea igual a la longitud del mensaje.

C	L	A	V	E	C	L	A	V

6. Se suma el valor decimal del código ASCII vinculado a los caracteres ubicados en la posición n del mensaje y de la clave.

	N_{78}	E_{69}	M_{77}	J_{74}	A_{65}	S_{83}	32	32	E_{69}
+	C_{67}	L_{76}	A_{65}	V_{86}	E_{69}	C_{67}	L_{76}	A_{65}	V_{86}
	145	145	142	160	134	150	108	097	155

C = 145145142160134150108097155

Consideraciones para verificar el correcto funcionamiento del algoritmo en diversos lenguajes

El set de pruebas se basó en pruebas de caja negra, eligiendo los casos de prueba en función de las especificaciones funcionales del software [8], cuyo único interés es determinar si las salidas son correctas en función de las entradas.

Las pruebas realizadas verifican la compatibilidad, robustez y estrés al que se pueden someter las diferentes implementaciones en lenguajes de programación diferentes del algoritmo de cifrado.

En el aspecto de compatibilidad se realizaron pruebas que validan el aspecto funcional del mensaje, que el mensaje cifrado por cualquier implementación pueda ser descifrado por las implementaciones restantes y viceversa.

En el aspecto de robustez se verificó que las diversas implementaciones se comportaran correctamente a valores de entrada incorrectos, evaluando la forma en que el software debería responder

Las pruebas de estrés consistieron en someter los programas implementados en diferentes lenguajes a valores extremos de tamaño de bloque y longitud de mensaje.

Performance de algoritmo en diversos lenguajes

Los aspectos considerados para determinar el desempeño del algoritmo en los diversos lenguajes de programación fueron el tiempo que le toma al software cifrar o descifrar el mensaje, el porcentaje de uso de CPU y la cantidad de memoria RAM.

Se diseñaron un total de 30 pruebas, para los programas desarrollados en los diversos lenguajes que consideraron validaciones de entrada en cada una de las partes componentes, solicitudes de encriptación con incongruencia en parámetros solicitados y pruebas de estrés, en este trabajo se describen las principales pruebas que afectan el performance de los programas. Las pruebas se realizaron en el mismo hardware y sistema operativo, midiendo el consumo de recursos con el software Process Explorer de Mark Russinovich [9].

La metodología de medición del tiempo de ejecución consistió en imprimir la hora en el momento en que se presiona el botón de cifrar/descifrar e imprimir la hora antes de enviar el mensaje cifrado o descifrado a pantalla o archivo.

ISSN-2531-2197

Posteriormente se restó el tiempo inicial al tiempo final, obteniendo así el tiempo de procesamiento.

Resultados Obtenidos

Se muestran mediante gráficas los resultados obtenidos., puntualizando el desempeño en cada lenguaje, explicando el porqué de las diferencias. La prueba número uno se diseñó con el objetivo de someter a las implementaciones del algoritmo de cifrado a lo que se consideró un caso de uso representativo del lenguaje común, sometiendo al software a una prueba de estrés, cifrando un texto extenso de 450 mil caracteres.

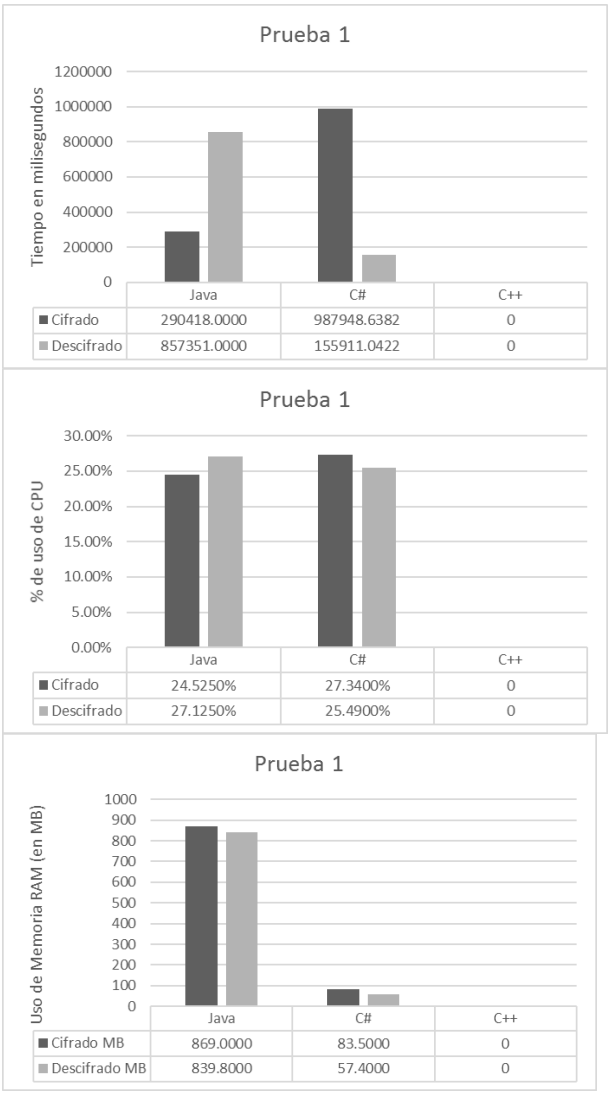


Gráfico 1

Se detectó que el cifrado y descifrado programado en C++ falló debido al deficiente manejo de los caracteres especiales, tales como el salto de línea y la letra "ñ". Verificándose además de que el algoritmo en C++ presentó problemas para manejar el código ASCII extendido.

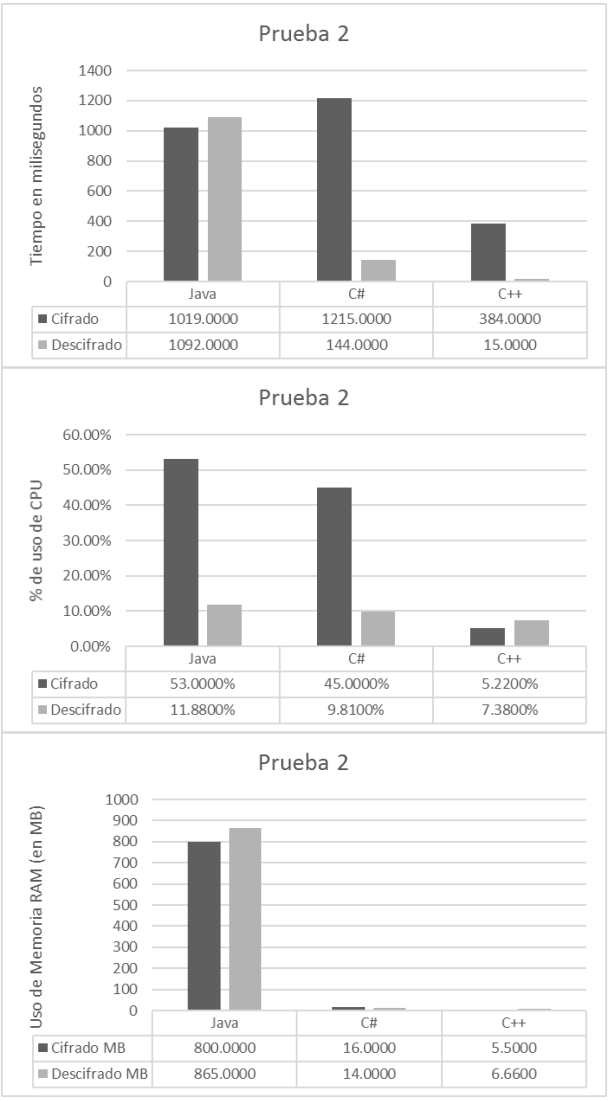


Gráfico 2

La prueba 2, basada en un análisis de rendimiento se basó en encriptar un mensaje pequeño en un bloque de mayor tamaño al del mensaje, se cifró un texto de 45 caracteres y un tamaño de bloque de 10 mil unidades, en esta prueba todos los lenguajes cumplieron la tarea, encontrándose que el programa desarrollado en Java tuvo un consumo mayor en memoria y tiempo de uso del CPU, en comparación con los programas desarrollados en C# y C++.

En la prueba 3 se repite el caso de un mensaje pequeño en un bloque excesivamente mayor, se utilizó un mensaje de 45 caracteres, pero con un tamaño de bloque de 100 mil unidades, con ello se comprobó que el programa en C++ pudo cifrar el mensaje sin problemas, pero surgió el problema en la consola, ya que la consola limita la cantidad de caracteres que pueden ser escritos en esta, por lo tanto, no se pudo llevar a cabo el descifrado en el software desarrollado en C++.

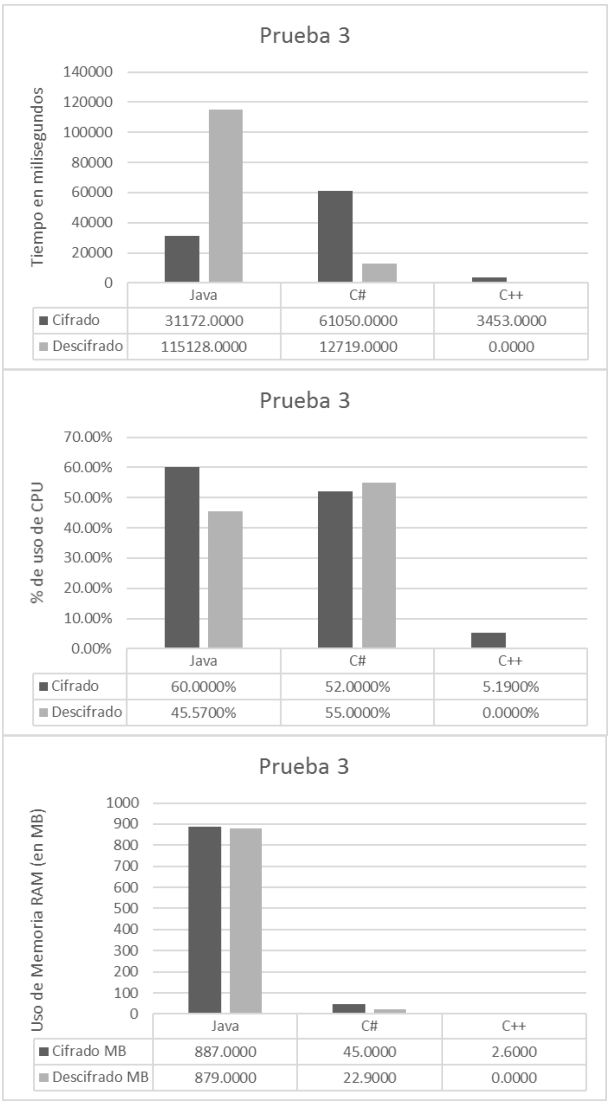


Gráfico 3

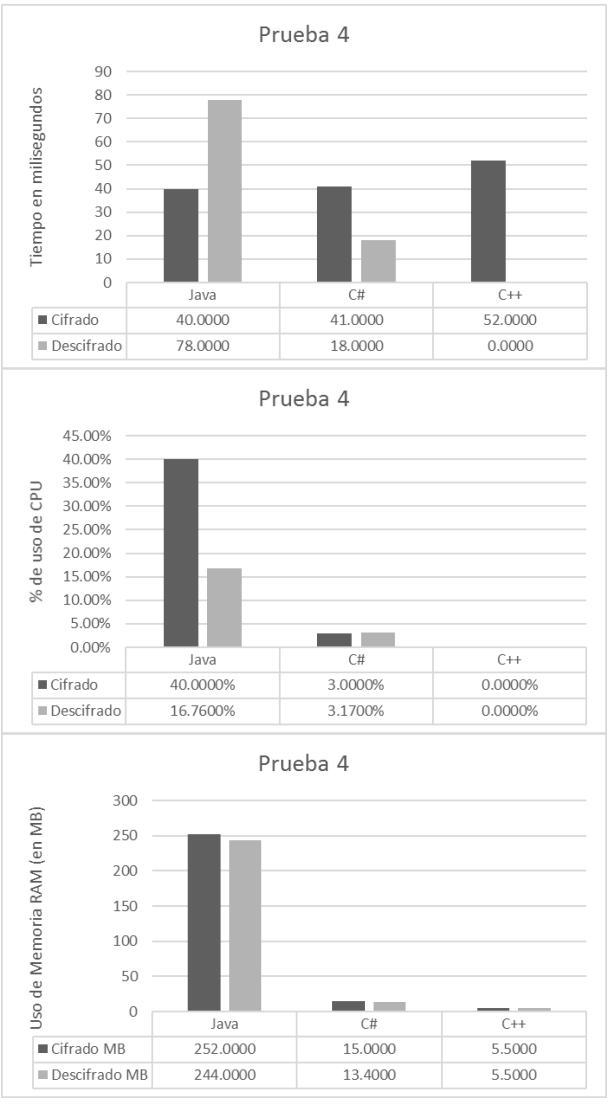


Gráfico 4

ISSN-2531-2197

En la prueba 4, se cifró el mismo mensaje, pero con un tamaño de bloque de mil unidades, y se demostró que el programa en C++ es muy apto en cuestiones de rendimiento, dado que empleó un bajo consumo de CPU y muy poco uso de la Memoria RAM. El diseño de las pruebas número dos, tres y cuatro se hizo de forma que se esperaba que el tiempo de ejecución, consumo de CPU y memoria fuera diez veces mayor en la prueba dos y cien veces mayor en la prueba tres respecto a la prueba cuatro. En la práctica no ha sido posible observar una relación entre los tiempos y consumo de las diversas pruebas por lo que se asume que existen factores cuyo nivel relevancia es más significativa para el desempeño.

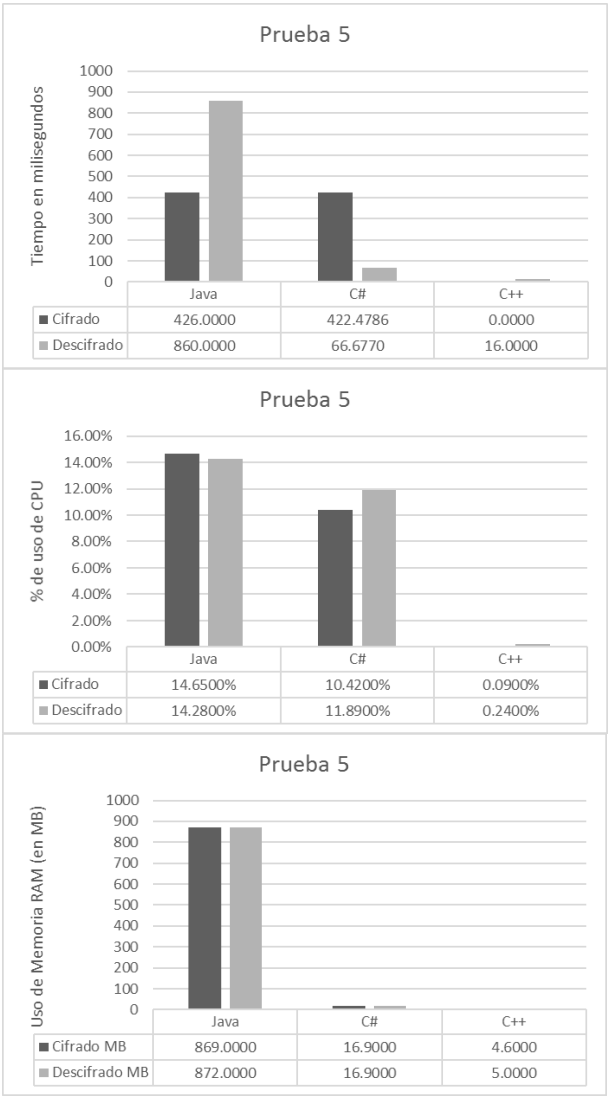


Gráfico 5

La prueba número cinco consistió en cifrar un mensaje de mil caracteres con un tamaño de bloque de 10 mil unidades. Se corroboró el excelente manejo de recursos de la implementación de C++. Además, se constató que el consumo de memoria para los procesos de cifrado y descifrado es muy similar dentro del mismo lenguaje.

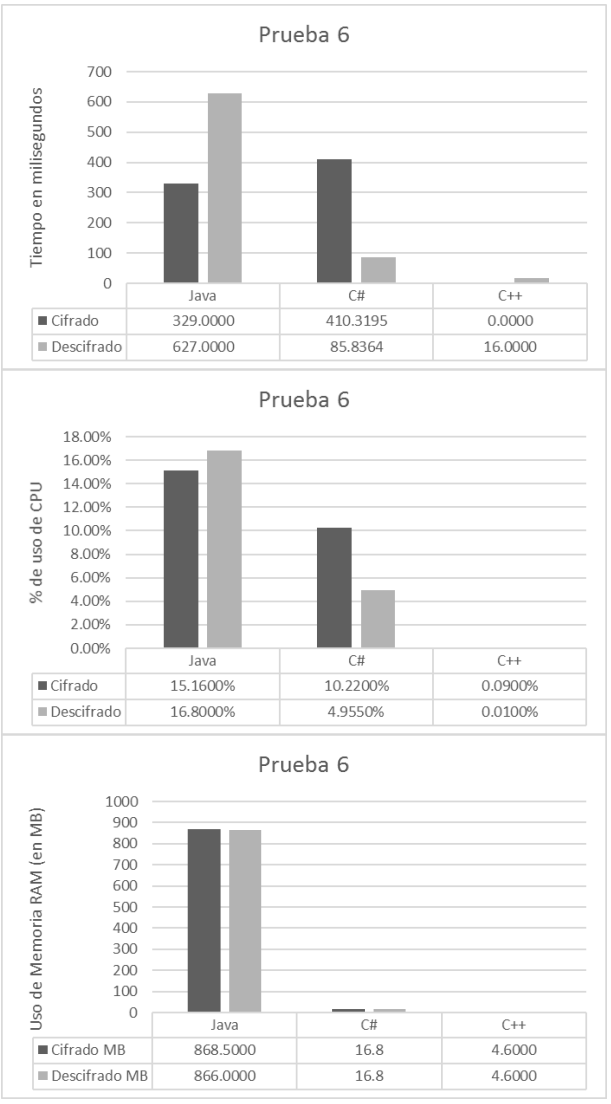


Gráfico 6

La prueba seis consistió en invertir los valores de longitud del mensaje y tamaño de bloque de la prueba cinco, se empleó la misma contraseña y se observó un rendimiento muy similar entre ambas pruebas. A continuación, se contrastan los resultados de la prueba número cinco y seis para conocer las similitudes y diferencias de ambas pruebas.

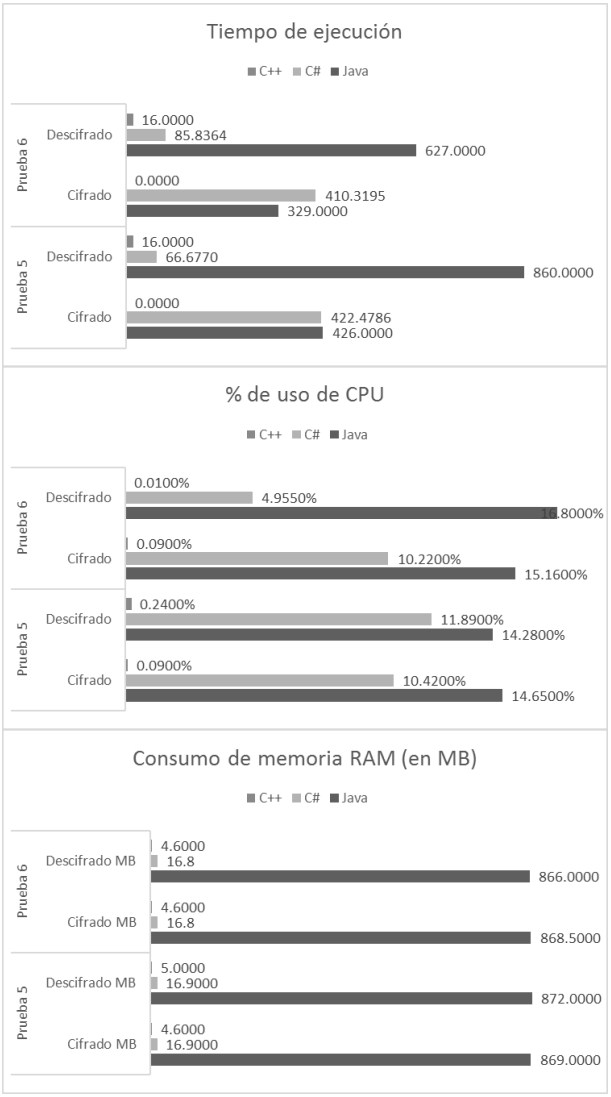


Gráfico 6

Al comparar los resultados de tiempo de ejecución y uso de CPU no se observa alguna similitud aparente por lo que se corrobora lo observado en el set de pruebas dos, tres y cuatro. Por otra parte, el consumo de memoria RAM es muy similar en el caso de los tres lenguajes de programación por lo que se asume que mientras la cantidad de información a cifrar o descifrar sea similar y no existan factores que puedan perjudicar el rendimiento, se obtendrá un consumo de memoria similar.

Conclusiones

Durante el desarrollo del software de cifrado, diferentes experiencias fueron obtenidas requeridas para su realización.

Se requiere una especificación a detalle del algoritmo, como pudo ser visto en el proceso a detalle de desarrollo del algoritmo especificado, principalmente si es un software que será realizado por diferentes programadores en diferentes lenguajes y plataformas, porque esto puede generar una implementación deficiente

Existen detalles técnicos de cada lenguaje que deben ser considerados por los desarrolladores de código, en este caso se debieron realizar juntas entre los diferentes programadores a fin de solucionar problemas existentes en cada desarrollo, como se recomienda en [10].

El funcionamiento de los algoritmos debió ser comprobado entre los diferentes programas desarrollados usando el concepto de sistemas débilmente acoplados, utilizando para esto archivos de paso donde se almacenaban los textos planos y cifrados para verificar su correcto funcionamiento

La transformación de datos al ser generada por un proceso aritmético, causó problemas debiéndose restringir los valores obtenidos a un rango definido que no afectara la reconversión de datos
Pudo ser constatada una mejor eficiencia en el lenguaje C++ con respecto a los otros lenguajes, lo que era de esperarse, sin embargo, las interfaces de los otros lenguajes son claramente superiores, facilitando su desarrollo

ISSN-2531-2197

En general el proceso de cifrado y descifrado es muy similar en condiciones normales debido a que es un procedimiento matemático inverso, la selección inapropiada del lenguaje de programación es un factor importante en el desarrollo de software, pues la implementación de un algoritmo criptográfico requiere de un lenguaje de programación exacto, con un consumo que no afecte el rendimiento del equipo y tenga un amplio soporte de los caracteres necesarios para comunicar mensajes en diferentes idiomas.

Referencias

Sommerville, I., & Alfonso Galipienso, M. (2005). Ingeniería del software. Madrid: Pearson Educación.

Orozco, G., & Nuñez, J. (2017). Introducción a la Criptografía. Recuperado de http://patux.net/downloads/crypto/crypto.pdf

Holtkamp, P., Jokinen, J., & Pawlowski, J. (2015). Soft competency requirements in requirements engineering, software design, implementation, and testing. Journal Of Systems And Software, 101, 136-146. http://dx.doi.org/10.1016/j.jss.2014.12.010

Garrido A (2006). Fundamentos de programación en C++. Madrid: Delta Publicaciones.

Unicode (The Java™ Tutorials > Internationalization > Working with Text). (2017). Docs.oracle.com. Recuperado 10 Agosto 2017, Sitio web: https://docs.oracle.com/javase/tutorial/i18n/text/unicode.html

Ceballos Sierra F. (2007). Microsoft C#. Madrid: Ra-Ma.

Bronson, G., Borse, G., & Velázquez Arellano, J. (2007). C++ para ingeniería y ciencias. México: Thomson.

Beizer, B. (1990). Software testing techniques. London: International Thomson computer Press.

Process Explorer. (2017). Docs.microsoft.com. Recuperado 10 de Agosto de 2017, from https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer

Tuya, J., Ramos Román, I., & Dolado Cosín, J. (2007). Técnicas cuantitativas para la gestión en la ingeniería del software. Oleiros, La Coruña: Netbiblo.

[Titulo en Times New Roman y Negritas No.14]

Apellidos en Mayusculas -1er Nombre de Autor †, Apellidos en Mayusculas -2do Nombre de Autor
Correo institucional en Times New Roman No.10 y Cursiva

(Indicar Fecha de Envio:Mes,Dia, Año); Aceptado(Indicar Fecha de Aceptación: Uso Exclusivo de ECORFAN)

Resumen

Titulo

Objetivos, metodología

Contribución

(150-200 palabras)

Indicar (3-5) palabras clave en Times New Roman y Negritas No.11

Abstract

Title

Objectives, methodology

Contribution

(150-200 words)

Keyword

Cita: Apellidos en Mayúsculas -1er Nombre de Autor †, ApellidosenMayusculas -2do Nombre de Autor. Titulo del Paper. Título de la Revista. 2015, 1-1: 1-11 – [Todo en Times New Roman No.10]

† Investigador contribuyendo como primer autor.

Instrucciones para Autores

Introducción

Texto redactado en Times New Roman No.12, espacio sencillo.

Explicación del tema en general y explicar porque es importante.

¿Cuál es su valor agregado respecto de las demás técnicas?

Enfocar claramente cada una de sus características

Explicar con claridad el problema a solucionar y la hipótesis central.

Explicación de las secciones del artículo

Desarrollo de Secciones y Apartados del Artículo con numeración subsecuente

[Título en Times New Roman No.12, espacio sencillo y Negrita]

Desarrollo de Artículos en Times New Roman No.12, espacio sencillo.

Inclusión de Gráficos, Figuras y Tablas-Editables

En el *contenido del artículo* todo gráfico, tabla y figura debe ser editable en formatos que permitan modificar tamaño, tipo y número de letra, a efectos de edición, estas deberán estar en alta calidad, no pixeladas y deben ser notables aun reduciendo la imagen a escala.

[Indicando el título en la parte inferior con Times New Roman No.10 y Negrita]

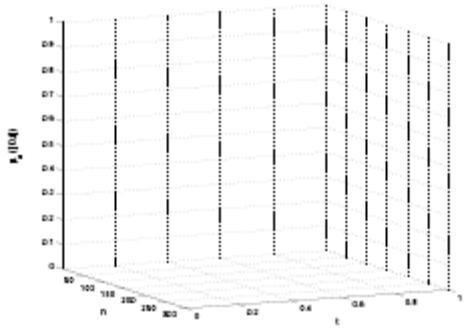


Grafico 1 Titulo y Fuente (en cursiva).

No deberán ser imágenes- todo debe ser editable.

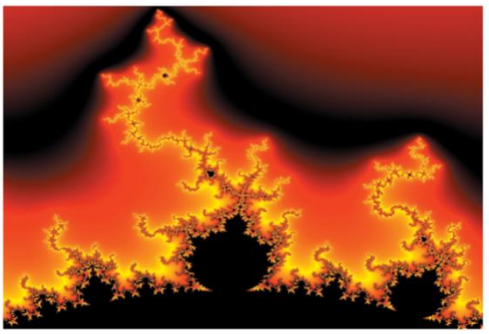


Figura 1 Titulo y Fuente (en cursiva).

No deberán ser imágenes- todo debe ser editable.

Tabla 1 Titulo y Fuente (en cursiva).

No deberán ser imágenes- todo debe ser editable.

Cada artículo deberá presentar de manera separada en **3 Carpetas**: a) Figuras, b) Gráficos y c) Tablas en formato .JPG, indicando el número en Negrita y el Titulo secuencial.

Instrucciones para Autores

Para el uso de Ecuaciones, señalar de la siguiente forma:

$$Y_{ij} = \alpha + \sum_{h=1}^{r} \beta_h X_{hij} + u_j + e_{ij} \qquad (1)$$

Deberán ser editables y con numeración alineada en el extremo derecho.

Metodología a desarrollar

Dar el significado de las variables en redacción lineal y es importante la comparación de los criterios usados

Resultados

Los resultados deberán ser por sección del artículo.

Anexos

Tablas y fuentes adecuadas.

Agradecimiento

Indicar si fueron financiados por alguna Institución, Universidad o Empresa.

Conclusiones

Explicar con claridad los resultados obtenidos y las posiblidades de mejora.

Referencias

Utilizar sistema APA. **No** deben estar numerados, tampoco con viñetas, sin embargo
en caso necesario de numerar será porque se hace referencia o mención en alguna parte del artículo.

Ficha Técnica

Cada artículo deberá presentar un documento Word (.docx):

Nombre de la Revista
Título del Artículo
Abstract
Keywords
Secciones del Artículo, por ejemplo:

1. *Introducción*
2. *Descripción del método*
3. *Análisis a partir de la regresión por curva de demanda*
4. *Resultados*
5. *Agradecimiento*
6. *Conclusiones*
7. *Referencias*

Nombre de Autor (es)
Correo Electrónico de Correspondencia al Autor
Referencia

Formato de Originalidad

Madrid, España a ____ de ____ del 20_____

Entiendo y acepto que los resultados de la dictaminación son inapelables por lo que deberán firmar los autores antes de iniciar el proceso de revisión por pares con la reivindicación de ORIGINALIDAD de la siguiente Obra.

Artículo (Article):

_____

Firma (Signature):

_____

Nombre (Name)

Formato de Autorización

Madrid, España a ____ de ____ del 20____

Entiendo y acepto que los resultados de la dictaminación son inapelables. En caso de ser aceptado para su publicación, autorizo a ECORFAN-Spain difundir mi trabajo en las redes electrónicas, reimpresiones, colecciones de artículos, antologías y cualquier otro medio utilizado por él para alcanzar un mayor auditorio.

I understand and accept that the results of evaluation are inappealable. If my article is accepted for publication, I authorize ECORFAN-Spain to reproduce it in electronic data bases, reprints, anthologies or any other media in order to reach a wider audience.

Artículo (Article):

_____

Firma (Signature)

_____

Nombre (Name)

Revista de Tecnología Informática

2531 2197
ISSN 2531 - 2197