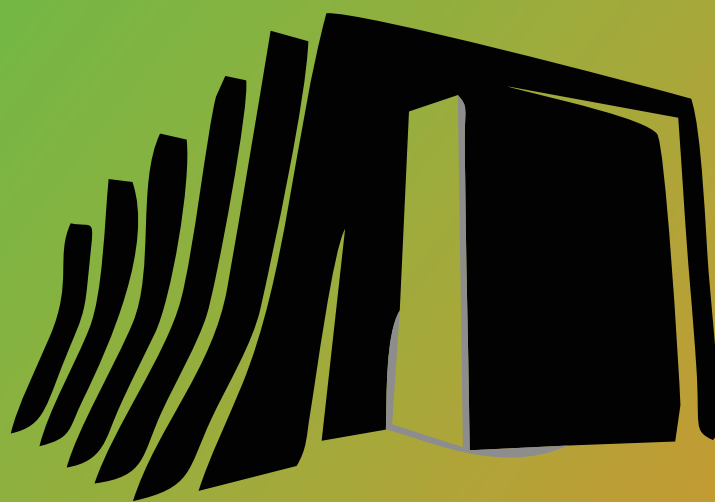


Avances en Sistemas y Computación



Centro Universitario UAEM Valle de México



Cuerpo Académico de Inteligencia Computacional
mayo 2015

Avances en Sistemas y Computación

COMPILADORES:

Dra. en C. Maricela Quintana López

Dr. en C. José Martín Flores Albino

Dr. en C. Víctor Manuel Landassuri Moreno

Dr. en C. Saúl Lazcano Salas

M. en C. Saturnino Job Morales Escobar

Dr. en C. Héctor Rafel Orozco Aguirre

Centro Universitario UAEM Valle de México

2015

Coordinador Editorial: Dr. José Martín Flores Albino
Asistente Editorial: Dra. Maricela Quintana López
Editor de Desarrollo: Dr. Víctor Manuel Landassuri Moreno
Supervisor de Producción: Dr. Saúl Lazcano Salas
Avances en Sistemas y Computación

Primera edición: mayo, 2015
Universidad Autónoma del Estado de México

Centro Universitario UAEM Valle de México
Boulevard Universitario S/N Predio San Javier
Km. 11.5, Carr. Atizapán - Nicolás Romero,
Atizapán de Zaragoza, Estado de México, C.P. 52916
Tel: 5827-0361, 5827-0579, Ext. 1183

Prohibida su Reproducción Total y Parcial por cualquier medio sin autorización escrita del editor

DERECHOS RESERVADOS, Respecto a la primera edición por

Universidad Autónoma del Estado de México
Instituto Literario #100 Col Centro
Toluca, Estado de México
C.P.: 50000

Libro Electrónico
ISBN: 978-607-422-678-2
México.

Todo la información vertida en este libro es responsabilidad única y exclusiva de los autores.
Los editores, compiladores, Centros Universitarios y la UAEM no se hace responsables de la información vertida aquí.

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

Dr. en D. Jorge Olvera García
Rector

M. en C. A. Des. Ed. María Laura González Santos
Directora del Centro Universitario UAEM Valle de México

Con la colaboración de:

Cuerpo Académico Inteligencia Computacional
Coordinación de la Carrera Ingeniería en Sistemas y Comunicaciones
Coordinación de la Carrera Ingeniería en Computación

Índice general

1	Prólogo	7
2	Lista de Autores	9
	Comparación de Algoritmos de Agrupamiento K-means, Suffix Tree Clustering y Lingo para la Tarea de Recuperación de Información. <i>Vázquez Vázquez, E., Ledeneva Y. y García Hernández, R. A.</i>	11
	La informática en los Delitos, un Panorama en la Perspectiva del Cómputo Forense (CF). <i>García Garduño, E., Salgado Gallegos, M., Albarran Trujillo, S. E., Pérez Merlos, J. C. y Rodríguez Ángeles, E.</i>	18
	Configuración PIC-Raspberry Pi® Modelo B, con Fines de Monitoreo de Señales Analógicas. <i>Morelos Romero, J. C., Delgado Reyes, G. y Asdrúbal López, C.</i>	25
	Efecto del Overclocking sobre los Tiempos de Ejecución en una Computadora Embebida con Diferentes Niveles de Carga. <i>Antonio Aguilar, E., Cano Rosas, L. y Guevara López, P.</i>	31
	Técnicas de Generación y Solución de Laberintos. <i>Requena Vega, E. A., Landassuri Moreno, V. M. y Orozco Aguirre, H. R.</i>	37
	Arquitectura de un Sistema de Control para una Fuente Mágica. <i>Flores Albino, J. M., Landassuri Moreno, V. M. y Romero Morales, L. F.</i>	43
	Sistema Integral para el Uso y Control de Dispositivos en el Hogar Utilizando Tecnologías de la Información y Comunicación. <i>Gómez Simón, C., Ramírez López, M. y González Morán, C. O.</i>	48
	Vuelo Autónomo con Programación Visual y Redes Neuronales Artificiales de un Parrot AR-Drone. <i>Rodríguez Sánchez, J. C., Moreno Martínez, M. A., Landassuri Moreno, V. M. y Flores Albino, J. M.</i>	55
	Revisión de la transición del protocolo IPv4 a IPv6 en Latinoamérica. <i>Olvera Cueyar, M., Vite Rojo, A. D. y Orozón Ramírez, L. B.</i>	62
	Situación Actual de los Repositorios Abiertos en México. <i>González Mendoza, A. M., y Aguilar Juárez, I.</i>	71
	Clasificación de Comportamientos Riesgosos entre Peatón y Conductor Utilizando Algoritmos de Minería de Datos. <i>Roldán López, R. y Quintana López, M.</i>	77
	Segmentación de Imágenes Aplicado a Visión Artificial con Drones. <i>Moreno Martínez, M. A., Rodríguez Sánchez, J. C., Landassuri Moreno, V. M., Flores Albino, J. M. y Lázcana Salas, S.</i>	79
	Sistema de Reconocimiento de Caracteres de Placas de Automóviles en México. <i>Palacios Rojas, E. R., Landassuri Moreno, V. M., y Lazcano Salas, S.</i>	80

El pasado mes de mayo en el Centro Universitario UAEM Valle de México, se convocó a diversos autores a que nos enviaran sus trabajos de investigación. Como resultado se creó este volumen, en el que se presentan avances en las áreas de ciencias de la Computación y de Sistemas de Comunicación. Se recopilan trabajos sobre: Electrónica, Inteligencia Artificial, Procesamiento Digital de Imágenes, Minería de Datos, entre otras. El desarrollo e investigación tecnológica es primordial para el avance de una institución de educación superior, pues impulsa a estar actualizados y que los estudiantes tengan oportunidad de realizar investigación con el apoyo de expertos.

El presente volumen es un ejemplo de un medio que busca concentrar los resultados de investigación, para hacerlos disponibles a la comunidad de expertos en el campo de las ciencias de la computación. La recopilación de los trabajos se realizó con apertura a las instituciones y grupos de investigación interesados en exponer su trabajo y así participar en el avance de la tecnología.

Esta recopilación se presenta para dar espacio de divulgación a resultados de investigación que se realizan en diversas instituciones. Se buscó seleccionarlos para que cumplieran con requerimientos de calidad y profundidad que aportaran en sus distantes áreas. Se seleccionaron trabajos que tratan temas que son de interés actual. Como objetivo se planteó que todos los artículos dieran resultados o propuestas originales. Que den al lector un panorama amplio y claro sobre los temas.

Los compiladores de la obra somos miembros del cuerpo académico de “Inteligencia Computacional” que integra profesores-investigadores que estamos interesados en la divulgación y desarrollo del campo de las ciencias de la computación, y que con esta obra esperamos aportar en su desarrollo.

Cuerpo Académico de Inteligencia Computacional
COMPILADORES

Lista de Autores

Núm	Nombre	Procedencia
1	Irene Aguilar Juarez	Centro Universitario UAEM Texcoco
2	Silvia Edith Albarrán Trujillo	Ingeniería en Computación e Ingeniería Electrónica, FI, UAEMex
3	Erick Antonio Aguilar	IPN-ESIME Culhuacán
4	José Luis Cano Rosas	IPN-ESIME Culhuacán
5	Omar Carlos Romero	UAEM
6	Gustavo Delgado Reyes	IPN-ESIME Culhuacán
7	José Martín Flores Albino	CU UAEM VM
8	Eladia García Garduño	Ingeniería en Computación e Ingeniería Electrónica, FI, UAEMex
9	René Armulfo García Hernández	Unidad Académica Profesional Tianguistenco, UAEMex
10	Christian Gómez Simón	CU UAEM VM
11	Ana María González Mendoza	Centro Universitario UAEM Texcoco, UAEMex
12	Carlos Omar Gonzales Morán	CU UAEM VM
13	Pedro Guevara López	IPN-ESIME-Culhuacán
14	Victor Manuel Landassuri Moreno	CU UAEM VM
15	Saúl Lazcano Salas	CU UAEM VM
16	Yulia Ledeneva	Unidad Académica Profesional Tianguistenco, UAEMex
17	Asdrúbal López Chau	Centro Universitario UAEM Zumpango
18	Juan Carlos Morelos Romero	IPN-ESIME Culhuacán
19	Miguel Ángel Moreno Martínez	CU UAEM VM
20	Miriam Olvera Cueyar	Universidad Politécnica de Tulancingo
21	Luis Bryan Oroxón Ranírez	Universidad Politécnica de Tulancingo
22	Héctor Rafael Orozco Aguirre	CU UAEM VM
23	Edgar Palacio Rojas	CU UAEM VM
24	Juan Carlos Pérez Merlos	Ingeniería en Computación e Ingeniería Electrónica, FI, UAEMex
25	Maricela Quintana López	CU UAEM VM
26	Pedro Guevara López	IPN-ESIME-Culhuacán
27	Maleny Ramírez López	CU UAEM VM
28	Ernesto Requena Vega	CU UAEM VM
29	Eduardo Rodríguez Ángeles	Ingeniería en Computación e Ingeniería Electrónica, FI, UAEMex
30	Juan Carlos Rodríguez Sánchez	CU UAEM VM
31	Rebeca Roldán	CU UAEM VM
32	Luis Felipe Romero Morales	CU UAEM VM
33	Mireya Salgado Gallegos	Ingeniería en Computación e Ingeniería Electrónica, FI, UAEMex
34	Eder Vázquez Vázquez	Unidad Académica Profesional Tianguistenco, UAEMex
35	Alma Delia Vite Rojo	Universidad Politécnica de Tulancingo

Comparación de Algoritmos de Agrupamiento K-means, Suffix Tree Clustering y Lingo para la tarea de Recuperación de Información

Comparing of Clustering Algorithms K-means, Suffix Tree Clustering and Lingo for Information Retrieval Task

Eder Vázquez Vázquez

Yulia Ledeneva

René Arnulfo García Hernández

Unidad Académica Profesional Tianguistenco

Universidad Autónoma del Estado de México

Instituto Literario 100, Toluca, Estado de México, México

eder2v@hotmail.com

yledeneva@yahoo.com, renearnulfo@hotmail.com

Resumen— En la última década, se ha visto un rápido crecimiento de la información en formato electrónico que adquiere nuevas formas de representación que van desde archivos de texto en una computadora hasta archivos de páginas Web. Ante esta situación, el problema reside en encontrar información de interés entre todos esos datos. Los motores de búsqueda se encargan de realizar esta tarea, i.e., buscan información relevante en un gran conjunto de datos a partir de una consulta ingresada, pero estos sistemas solo informan de la existencia de documentos que pueden ser relevantes a la consulta. En este artículo se comparan los resultados de los algoritmos de agrupamiento los cuales se aplican al conjunto de datos MED (1033 documentos) para la tarea de recuperación de información.

Palabras Clave—Recuperación de Información, Algoritmo de Agrupamiento, K-means, Lingo, Suffix Tree Clustering, Carrot2.

I. INTRODUCCIÓN

Con el desarrollo de internet, la información de la Web ha estado creciendo de manera explosiva, generando así una gran cantidad de documentos textuales, desde archivos en computadoras hasta páginas Web [1]. Cada día millones de usuarios acceden a la Web para trabajar, estudiar y entretenerse, cuya actividad principal comienza regularmente con búsquedas [2], de diferentes tipos y objetivos.

Hoy en día, los motores de búsqueda se han convertido en las herramientas más populares e importantes para los usuarios de la web para encontrar la información de interés [3]. Los motores de búsqueda se encargan de resolver consultas sobre tales documentos, pero a medida que el número de páginas web aumenta, encontrar información relevante se vuelve una tarea compleja. Un sistema de recuperación de información se encarga de encontrar los documentos relevantes a una consulta generada [4], pero la salida de este tipo de sistemas solo

informa la existencia de documentos que son relevantes de acuerdo a la consulta, el resto del trabajo lo hace el usuario quien se encarga de revisar todos (o muchos) de los documentos.

Ante esta situación resulta apropiado agrupar documentos de contenidos similares para facilitar el acceso a esta información así como permitir la visualización y navegación a través de ella [5]. Los algoritmos de agrupamiento de resultados de búsqueda (Search Result Clustering, SRC) y los algoritmos de etiquetado proporcionan fácil acceso a resultados de búsqueda, este tipo de algoritmos involucra una clase de algoritmos llamados Algoritmos de Agrupamiento (Clustering) [6]. La agrupación de documentos post-recuperación se basa en la idea de que documentos similares pueden ser relevantes a la misma consulta y por lo tanto ser agrupados bajo una etiqueta representativa. Por esta razón, la agrupación post-recuperación ha sido una técnica que se ha estudiado por al menos una década [7].

En este trabajo se expone una comparación de tres técnicas de agrupamiento las cuales serán utilizadas para una agrupación post-recuperación, de tal manera que se permita una mejor navegación entre los resultados devueltos. Se utilizó el sistema Carrot2¹ como herramienta central, ya que permite la búsqueda de documentos de varios sitios (Web o repositorios) y realiza la agrupación de los mismos bajo diversos algoritmos de agrupamiento.

II. TRABAJO RELACIONADO

Las técnicas de agrupamiento han sido abordadas desde diferentes ángulos, ya sea para agrupamiento de páginas web (devueltas por un sistema de recuperación de información), o

¹ Carrot2: <http://search.carrot2.org/stable/search>

para subdividir y agrupar un conjunto de datos. A continuación se mencionan algunos trabajos que hacen uso de técnicas de agrupamiento para recuperación de información.

En [8] se presenta un software que permite a un usuario plantear una consulta general y obtener los resultados de su consulta en forma agrupada. El usuario define como un parámetro de entrada adicional el número de grupos que desea obtener. Cada grupo resultante consta de una identificación del grupo, una relación con los títulos pertenecientes al grupo y los términos principales que caracterizan al grupo. Cada título, a su vez, es un enlace al registro completo del título dentro de la colección o ficha bibliográfica. El sistema extrae las raíces de los términos de búsqueda y elimina los términos irrelevantes, encuentra los títulos de la colección que contienen dichos términos de búsqueda y genera una matriz que representa el modelo de espacio vectorial en el cual las filas contendrán todos los términos asociados a los resultados obtenidos y en las columnas los títulos encontrados. El sistema invoca el algoritmo Divisive-Binary-K-means. Posteriormente, los resultados son mostrados al usuario en forma de grupos; a cada grupo se le asigna un identificador, los títulos que pertenecen al grupo y los términos que caracterizan al grupo.

En [7] se muestra un sistema de recuperación de información para el dominio biomédico con agrupamiento post-recuperación que se llama Clummed (CLUstering on Mobile Medical Devices). Clummed realiza dos tareas principales: la indexación y la recuperación. En la primera fase realiza el pre-procesamiento de los documentos y posteriormente la indexación mediante la herramienta Lucene. La fase de recuperación empieza con el ingreso de una consulta a partir de la cual se busca en una colección de documentos que términos de la consulta se encuentran en el índice. Para realizar el agrupamiento de los resultados se utilizó el algoritmo K-means con $k=4$, el cual mostró mejores resultados. La salida final del sistema mostró una organización de los resultados de la búsqueda en carpetas donde se almacenan los documentos que son semánticamente relacionados, lo cual facilita al usuario la elección de los temas de su interés.

Se introduce un nuevo algoritmo de clustering para texto Chino. El algoritmo C-Lingo (Chinese Lingo) [9] muestra una mejora frente al algoritmo Lingo original reemplazando la descomposición de valores por una Matriz de Factorización No Negativa (NMF). Las mejoras que explica son las siguientes, la primera es que la matriz de descomposición se mejora debido a que de esta forma es más efectivo para el texto chino. Se seleccionó una herramienta apropiada para la segmentación de las palabras chinas. Por último, se construyó una lista de palabras vacías en el orden de eliminado en la etapa de pre procesamiento. La evaluación del nuevo algoritmo C-Lingo se realizó usando datos de Open Directory Project el cual es un directorio jerárquico en la Web, donde cada rama de la jerarquía corresponde a distintos tópicos. Generaron 20 grupos de datos sobre los cuales el algoritmo trabajó. El trabajo ahí realizado muestra los resultados de la aplicación del algoritmo C-Lingo para agrupamiento de texto en lenguaje chino, los cuales no fueron suficientes y se plantean la idea de mejoras futuras.

III. RECUPERACIÓN DE INFORMACIÓN

Según Baeza [10], la recuperación de la información trabaja con la representación, almacenamiento, organización y acceso a los elementos de información; tales como documentos. La representación y la organización de los elementos de información deben ser tal que proporciona a los usuarios un acceso fácil a la información de su interés. Por otro lado, Van Rijsbergen [11], la recuperación de información está relacionada con la recuperación de aquellos documentos que sean probablemente relevantes para la necesidad de información del usuario expresada en una consulta.

Los sistemas de recuperación de información devuelven resultados tanto relevantes como no relevantes, por lo que es conveniente contar con una técnica que permita de alguna manera dividir la información devuelta. Muchas técnicas han sido desarrolladas cuyo fin es tratar de que las respuestas generadas por un sistema de recuperación de información sean las más relevantes para el usuario, o en otro caso, lograr que las respuestas queden organizadas en conjuntos distinguidos por temas o relaciones. Estas técnicas pueden ser: la representación de relaciones entre los documentos, filtrado, clasificación, generación automática de resúmenes, búsqueda de respuestas y agrupamiento [12].

El agrupamiento de documentos para la recuperación de información ha sido un campo muy estudiado por muchos años. La idea descrita en [13] se basa en la *hipótesis de grupos*, la cual dice que “los documentos estrechamente asociados tienden a ser relevantes para las mismas peticiones”. Hearst [14] agrega que esta hipótesis también aplica para un conjunto de documentos devueltos por un sistema de recuperación de información, i.e., si se realiza un agrupamiento correcto a los documentos recuperados, es posible separar los documentos relevantes de los no relevantes y así encontrar información más rápidamente.

IV. ALGORITMOS DE AGRUPAMIENTO

Algoritmos de agrupamiento son una tarea descriptiva de Minería de Datos, los cuales se encargan de dividir automáticamente los datos de entrada en grupos significativos (o representativos) basados en las características de los datos, donde los objetos similares se colocan dentro del mismo grupo (cluster) y los objetos diferentes se colocan en otro grupo [15]. El agrupamiento es una forma de clasificación no supervisada de patrones (i.e., los datos de formación no incluyen una muestra de cómo realizar la agrupación) que crea grupos basados en la similitud de sus elementos [16]. Existen dos categorías principales de algoritmos de agrupamiento: jerárquicos y no jerárquicos [15]. Los métodos jerárquicos producen una serie de particiones por niveles basándose en un criterio para la fusión o división de los grupos. Los algoritmos más conocidos de este tipo son single-link, complete-link y minimum-variance [16].

Los métodos no jerárquicos obtienen una partición simple de los datos de entrada en una estructura de grupos, este tipo de algoritmos necesita como entrada el número k de grupos deseados bajo los cuales se agruparan los datos. El algoritmo más popular de estos métodos es el algoritmo K-means. Este algoritmo es iterativo y se basa en asignar cada elemento al

grupo cuyo centroide sea más cercano al elemento, después se recalcula el centroide. El ciclo continua hasta que el centroide no sufra cambios.

Una gran variedad de algoritmos de agrupamiento han ido surgiendo con el tiempo [17], definiendo sus propios puntos de inicio y criterios para agrupar los datos. Algunos se basan en la identificación de frases y conceptos que existen en distintos documentos, tal es el caso del Suffix Tree Clustering (STC), Semantic Online Hierarchical Clustering y Lingo. El último se basa en los descriptores de los documentos.

En este trabajo se realiza la comparación de los algoritmos K-means, STC y Lingo haciendo uso de la herramienta Carrot2, la cual incluye entre sus algoritmos de agrupamiento estos tres algoritmos.

A. Algoritmo K-means

El algoritmo K-means, fue creado por MacQueen en 1967, es el algoritmo de agrupamiento más conocido y utilizado ya que es de una aplicación simple y eficaz [18]. Este algoritmo empieza por asignar aleatoriamente cada objeto a uno de los k grupos y después calcula la media, i.e., el centro (centroide) del grupo, después cada objeto es reasignado al grupo cuyo centroide este más cercano a dicho objeto, luego los centroides se vuelven a calcular. Este ciclo termina hasta que los centroides no tengan cambios o cuando se alcance una función de término [15].

El problema del empleo de estos esquemas es que fallan cuando los puntos de un grupo están muy cerca del centroide de otro grupo o cuando los grupos tienen diferentes tamaños y formas. Algunas de sus ventajas son que a menudo termina en un óptimo local, es escalable y eficiente en procesos que involucran grandes conjuntos de datos, su complejidad es de $O(nt)$, donde t es el número de iteraciones y n el número de objetos [19].

B. Algoritmo Suffix Tree Clustering

El algoritmo Suffix Tree Clustering (STC) fue creado por Oren Zamir [20-21]. Es un algoritmo incremental y de tiempo lineal basado en arboles diseñado para agrupamiento de documentos o de resultados de búsqueda a partir de frases idénticas que éstos contienen. Este algoritmo parte de un árbol de sufijos, el cual contiene todos los sufijos del conjunto de documentos a agrupar, si existen dos o más elementos en un nodo, éste se toma como grupo base. A partir de aquí agrupa los demás elementos de manera iterativa uniendo a los grupos base que sean semejantes [22].

Este algoritmo fue creado para el agrupamiento de snippets (los cuales son los pequeños textos devueltos por motores de búsqueda para describir los resultados de la consulta) [22], la razón de usar snippets es que la construcción del árbol de sufijos puede ser muy costosa mientras mayor sea la dimensión del documento.

Las fases que debe seguir el algoritmo STC son las siguientes [23]: 1) Preparación de los documentos; recupera los snippets de un buscador (o documentos) y se aplica análisis y stemming. 2) Construcción del árbol de sufijos: Inserta las palabras asociadas a cada documento en el árbol de sufijos. 3)

Fusión de grupos: combina nodos similares del árbol de sufijos. 4) Etiquetado de grupo: se genera una etiqueta (a partir del nodo en el árbol de sufijos) para cada grupo. 5) Puntuación de grupo.

Entre las ventajas que ofrece este algoritmo destacan la fácil adición de nuevos objetos al árbol de sufijos, poder descriptivo para identificar conjuntos y es de una complejidad $O(n)$ (donde n es el número total de documentos). Su principal limitación es que solo es aplicable a documentos [22].

C. Algoritmo Lingo

El algoritmo Lingo fue creado por Osiński [25] combina varios métodos de agrupamiento existentes para poner especial énfasis en las descripciones de cada grupo, además de descubrir las similitudes entre los documentos. Este algoritmo primero intenta descubrir los nombres para los futuros grupos y solo entonces asigna cada documento a los clúster correspondientes lo cual le permite esquivar el problema de grupos verbalmente inexplicables. Es posiblemente capaz de capturar la temática central en un resultado de búsqueda, i.e., puede descubrir grupos de documentos y describir el tema central de estos [25-26].

Las fases principales [27] de Lingo son: 1) Pre-procesamiento: uso de stemming, eliminación de stopwords, identificación del lenguaje. 2) Extracción de frases: descubre términos y frases representativas de los documentos. 3) Inducción de la etiqueta al grupo: identifica los conceptos abstractos que mejor describan los documentos de entrada, y de estos selecciona la mejor frase. 4) Descubrimiento del contenido de los grupos: se usó una Máquina de Soporte Vectorial (MSV) para determinar el contenido de cada grupo a partir de su etiqueta. 5) Formación final de los grupos: calcula la puntuación de los grupos y de ser necesario los fusiona.

Lingo está diseñado para agrupar documentos de poca longitud tales como los snippets.

V. DISEÑO EXPERIMENTAL Y ANÁLISIS DE RESULTADOS

Para el trabajo experimental se utilizó el sistema Carrot2, el cual es un sistema de recuperación basado en técnicas de agrupación de documentos y contenidos web, además de facilitar, ampliar o modificar la estructura de la búsqueda, comprender mejor tema, y favorece la exploración sistemática de los resultados [28]. Carrot2 proporciona seis diferentes versiones de su aplicación: Workbench, Java API, C# API, Clustering Server, Command Line Interface y Web Application.

C# API y Java API proporcionan librerías y dependencia para integrar el paquete a un software desarrollado en alguno de estos lenguajes. Workbench, Server y Web Application son distribuciones que permiten realizar agrupamiento desde un sistema fijo a un servidor, a una aplicación web o un conjunto de archivos, además permiten configurar las búsquedas con el fin de restringir los resultados a un contexto, ya sea por número de respuestas, idioma, ciudad. Command Line Interface es una aplicación que permite invocar a Carrot2 desde la línea de comandos, en esta distribución solo se permite configurar los archivos de entrada, de salida y el

algoritmo a usar, también permite ejecutar el algoritmo de agrupamiento a un conjunto de consultas almacenadas en formato XML.

Los algoritmos de agrupamiento disponibles en Carrot2 son Lingo, STC y K-means. La implementación del algoritmo STC en Carrot2 fue realizada por Dawid Weiss [24], mientras que la implementación de Lingo fue por Osiński [29].

A. Datos de prueba

Para la implementación se utilizó la colección MED, la cual es una colección destinada para la tarea de recuperación de información. Esta colección cuenta con 1033 documentos y 30 consultas, con promedio de 51.6 términos por documento. Se decidió utilizar la colección MED debido a la cantidad de documentos que contiene, además como se describió antes, los algoritmos Lingo y STC están diseñados para el manejo de pequeños textos. Se realizó el experimento con los 1033 documentos disponibles en esta colección utilizando las 30 consultas de la colección. Para cada consulta se aplicaron el algoritmo Lingo, STC y Kmeans.

Se utilizó la distribución de Command Line Interface ofrecida por Carrot2, debido a que esta aplicación nos permite ingresar consultas y realizar el agrupamiento desde la consola de comandos. En la entrada recibe el conjunto de consultas y documentos en un formato XML, y en la salida produce un conjunto de archivos XML, los cuales contienen la consulta realizada, el número de grupos realizados y el contenido de cada uno, así como el nombre colocado a cada grupo.

Para adaptar la colección de prueba al formato requerido por Lingo se realizó lo siguiente: primero se eliminaron las palabras vacías contenidas en los documentos de la colección (esto con el fin de reducir el tamaño de los documentos, ya que se intentó procesar los documentos completos y la aplicación no lo permitía). Como segundo paso, se separaron los archivos de la colección para su procesamiento. Después se extrajeron las consultas del archivo proporcionado en la colección. Como último paso, para cada consulta de la colección, se construyó el archivo en formato XML adecuado para Carrot2; este archivo contiene una consulta con todos los documentos a agrupar bajo la consulta.

B. Resultados

Al realizar las consultas, la aplicación regresó un conjunto de archivos en formato XML los cuales contenían la información de los grupos formados y su relación con la consulta, el formato de los archivos devueltos como se ilustra en la Fig. 1.

En el formato del archivo XML obtenido se indica la consulta realizada entre la etiqueta <query>, dentro de la etiqueta <group> se indica la cantidad de elementos en el grupo, el identificador del mismo y la puntuación (score) calculada por el algoritmo implementado, el cual nos indica la relevancia que tiene el grupo frente a la consulta ingresada. Dentro de la etiqueta <title> se encuentra el nombre asignado por el algoritmo al grupo, i.e., es el identificador textual del grupo formado. Y dentro de la etiqueta <document refid> se indica el número de los documentos que fueron agrupados en

el mismo grupo. La cantidad de documentos generados por los tres algoritmos (Lingo, STC y Kmeans) se indica en la Tabla I.

Fig. 1. Formato del archivo XML obtenido

```
<?xml version="1.0"?>
- <searchresult>
  <query>seattle</query>
  - <group score="42.884137906028606" size="23" id="0">
    - <title>
      <phrase>Seattle News</phrase>
    </title>
    <document refid="2"/>
    <document refid="3"/>
    <document refid="6"/>
    <document refid="16"/>
    <document refid="20"/>
    <document refid="22"/>
    <document refid="26"/>
    <document refid="34"/>
    <document refid="37"/>
    <document refid="40"/>
    <document refid="64"/>
    <document refid="72"/>
    <document refid="78"/>
    <document refid="94"/>
    <document refid="108"/>
    <document refid="111"/>
    <document refid="140"/>
    <document refid="144"/>
    <document refid="152"/>
    <document refid="154"/>
    <document refid="162"/>
    <document refid="164"/>
    <document refid="199"/>
  </group>
```

TABLA I. CANTIDAD DE GRUPOS GENERADOS POR LOS ALGORITMOS LINGO, STC Y K-MEANS A PARTIR DE LA COLECCION MED

	Lingo	STC	Kmeans
Clusters	37	16	25

Se esperaba que al utilizar la colección de documentos bajo el mismo algoritmo aplicado a cada una de las consultas, los grupos formados por el algoritmo utilizado resultaran iguales, i.e., si por ejemplo, la consulta de entrada era "ACID" utilizando el algoritmo STC, los grupos generados por esta consulta deberían ser los mismos que los grupos generados por el algoritmo STC bajo la consulta "BLOOD". Esta suposición fue correcta, debido a que la colección de documentos de entrada fue la misma para todas las consultas, era lógico que al aplicar un algoritmo de agrupamiento se obtuvieran los mismos grupos.

Como todos los grupos generados por algún algoritmo (ya sea Lingo, STC o K-means) son los mismos frente a cualquier consulta de la colección, era necesario saber qué grupo era más relevante frente a la consulta ingresada. Para esto, los archivos XML que genera Carrot2 como salida contienen un tributo llamado score, el cual indica la relevancia del grupo frente a la consulta bajo la cual se agrupo. El atributo score solo está disponible cuando se hace uso de los algoritmos Lingo y STC.

En la Tabla II, Tabla III y Tabla IV se muestran los grupos generados por los algoritmos Lingo, STC y K-means respectivamente para la consulta "Tissue culture of lung or bronchial neoplasms", en estas tablas se muestran la cantidad de grupos formados por cada algoritmo; en el caso de los algoritmos Lingo y STC se muestra la relevancia del grupo frente a la consulta de entrada, el algoritmo K-means no muestra la relevancia del grupo frente a la consulta.

TABLA II. GRUPOS GENERADOS POR LINGO PARA LA CONSULTA
"TISSUE CULTURE OF LUNG OR BRONCHIAL NEOPLASMS"

Nombre del grupo	SCORE	SIZE
ACID	37.7270986	111
ACTIVITY	35.5574484	156
RESPONSE	35.4454993	148
GROWTH	34.5110126	113
HUMAN	34.3565733	114
CHILDREN	34.1003584	111
RATS	33.5140847	115
OBSERVED	33.2257539	197
TREATMENT	32.9344083	173
TUMOR	32.8679752	76
CONCENTRATION	32.2932417	91
INFECTION	32.1327216	70
HORMONE	32.0343654	82
PROTEIN	31.8557631	75
BLOOD	31.3980745	149
PLASMA	30.8746602	71
KIDNEY	30.2051318	80
CANCER	30.0964811	80
LIVER	28.8282529	85
VENTRICULAR	28.5166451	54
SIGNIFICANT	28.3540952	166
RENAL	27.0714187	76
ANTIGEN	26.8507789	48
INCREASED	26.4185143	248
NORMAL	25.975988	202
CELLS	25.0215996	215
MARROW	24.226932	37
EFFECT	23.3909054	246
DNA	22.6381121	47
LEVEL	22.5399999	143
DEFECT	22.1055967	64
DISEASE	21.516831	145
RESULTS	19.7510044	278
PATIENTS	14.6221332	301
CASES	14.388301	253
STUDY	12.3744115	356
OTHER TOPICS	0	18

Todos los experimentos realizados fueron hechos bajo la misma colección de documentos, por lo tanto se esperaba que se crearan grupos similares al utilizar los tres diferentes algoritmos, por ejemplo, para la consulta anterior "*Tissue culture of lung or bronchial neoplasms*" el agrupamiento con el algoritmo STC se obtuvieron 16 grupos, de los cuales, 12 de estos grupos fueron nombrados exactamente igual que 12 grupos del algoritmo Lingo.

TABLA III. GRUPOS GENERADOS POR EL ALGORITMO STC PARA LA CONSULTA "TISSUE CULTURE OF LUNG OR BRONCHIAL NEOPLASMS"

Nombre del grupo	SCORE	SIZE
STUDY	177.5	355
PATIENTS	150.5	301
RESULTS	139	278
CASES	126.5	253
INCREASED	124	248
EFFECT	123	246
PRESENT	118	236
CELLS	107.5	215
NORMAL	100.5	201
OBSERVED	98.5	197
TREATMENT	86.5	173
SHOWED	84	168
SIGNIFICANT	83	166
ACTIVITY	78	156
SUGGESTED	78	156
OTHER TOPICS	0	54

Se destaca que los nombres de los grupos generados por los algoritmos STC y Lingo son de un solo término, este término indica el contenido de cada grupo, por lo tanto debe ser representativo. Por otro lado, el algoritmo K-means genera los nombres de los grupos con tres términos, los cuales indican de manera más descriptiva el contenido de los grupos.

VI. CONCLUSIONES Y TRABAJO FUTURO

Se compararon los algoritmos K-means, Suffix Tree Clustering y Lingo utilizando la colección MED con 1033 documentos y 30 consultas, con promedio de 51.6 términos por documento. Se obtuvieron 37 grupos generados por el algoritmo Lingo, 16 grupos generados por el algoritmo STC y 25 grupos generados por el algoritmo K-means.

Las descripciones de valor para los grupos tales como "bueno", "preciso", "aceptable" son difíciles de expresar numéricamente, aun cuando la evaluación de los grupos se realiza por un experto humano [26]. Los algoritmos Lingo y STC aplicados bajo Carrot2 ofrecen una descripción numérica de la relevancia de los grupos generados frente a la consulta de entrada.

Los algoritmos Lingo y STC proporcionan una puntuación (score) a los grupos generados. En el caso de Lingo, esta puntuación refleja si un grupo es relevante o no a la consulta ingresada. Para el algoritmo STC, la puntuación de cada grupo no tiene relevancia frente a la consulta, pues para todas las consultas realizadas se generaron los mismos grupos exactamente con la misma puntuación, por lo tanto, de nada sirve el atributo score en este algoritmo. Los grupos generados por el algoritmo K-means no proporcionan una puntuación de relevancia de los grupos frente a la consulta ingresada.

Por otro lado se descubrió que el algoritmo Lingo realiza la agrupación de forma tal que el nombre de los grupos generados no contengan alguna palabra de la consulta ingresada, por ejemplo, si la consulta ingresada fue “*Infectious Diseases*”, ningún grupo generados debe contener en su nombre los términos *infectious* o *diseases*.

Como trabajo futuro se pretende el uso de algoritmos de agrupamiento a resultados generados a partir de un Modelo de Recuperación de Información aplicado a una colección de datos, con el fin de averiguar si los grupos generados por los algoritmos a los resultados del modelo permiten una separación más precisa de los documentos relevantes y no relevantes.

TABLA IV. GRUPOS GENERADOS POR EL ALGORITMO KMEANS PARA LA CONSULTA “TISSUE CULTURE OF LUNG OR BRONCHIAL NEOPLASMS”

Nombre del grupo	SIZE
PATIENTS, CANCER, VENTRICULAR	99
CHILDREN, DISEASE, DEVELOPMENT	89
CELLS, NORMAL, STUDY	87
PATIENTS, CASES, STUDY	83
PATIENTS, TREATMENT, RESPONSE	77
CELLS, DNA, MARROW	51
PROTEIN, STUDY, ACTIVITY	50
RATS, INCREASED, ACID	47
CASES, RESULTS, RENAL	45
RATS, KIDNEY, RENAL	42
BLOOD, INFECTION, RESULTS	41
RATS, EFFECT, DAYS	41
PROTEIN, TUMOR, ANTIGEN	40
PATIENTS, DISEASE, RENAL	37
GROWTH, HORMONE, HUMAN	32
GROUP, CHILDREN, NORMAL	31
INCREASED, BLOOD, LEVEL	31
CELLS, TISSUE, TUMOR	27
INCREASED, EFFECT, STUDY	24
PATIENTS, CASES, DEVELOPMENT	21
STUDY, HUMAN, RESULTS	21
ACID, NORMAL, PLASMA	20
HORMONE, EFFECT, PLASMA	11
GROWTH, HORMONE, LEVEL	5
GROWTH, TUMOR, TREATMENT	4

REFERENCIAS

- [1] Dequan Zheng; Haibo Liu; Tiejun Zhao, "Search Results Clustering Based on a Linear Weighting Method of Similarity," *Asian Language Processing (IALP)*, 2011 International Conference on , vol., no., pp.123,126, 15-17 Nov. 2011.
- [2] Jaffri, A.: Knowledge Enhanced Searching on the Web. *Lecture Notes in Computer Science*, Volume 4825., pp 921-925 (2007)
- [3] Shunlai Bai; Wenhao Zhu; Bofeng Zhang; Jianhua Ma, "Search Results Clustering Based on Suffix Array and VSM," *Green Computing and Communications (GreenCom)*, 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom) , vol., no., pp.852,857, 18-20 Dec. 2010.
- [4] Llopis, F.; Muñoz, R.; Terol, R. M.; Noguera, E.: IR-n r2: Using normalized passages. Working Notes for the CLEF 2004 *Workshop*, 15-17 September (2004)
- [5] García-Plaza, A. P.; Fresno, V.; Martínez, R.: Una Representación Basada en Lógica Borrosa para el Clustering de páginas web con Mapas Auto-Organizativos. *Procesamiento del Lenguaje Natural*, Revista n° 42, pp. 79-86, Marzo (2009)
- [6] Anil Turel, Fazli Can, "A New Approach to Search Result Clustering and Labeling", *Information Retrieval Technology*, Lecture Notes in Computer Science Volume 7097, pp . 283-292, 2011. Springer-Verlag Berlin Heidelberg 2011.
- [7] Millán, M.; Muñoz, A.; Villa, M.; Maña, M. J.: A Biomedical Information Retrieval System based on Clustering for Mobile Devices. *Procesamiento del Lenguaje Natural*, Revista n° 45, septiembre, pp 255-258 (2010)
- [8] Marín, A.; Brach, J. W.: Aplicacion de dos nuevos algoritmos para agrupar resultados de búsquedas en sistemas de catálogos públicos en línea (OPAC). *Revista Interamericana de Bibliotecología. Escuela Interamericana de Bibliotecología*, Medellín, Colombia, Vol. 31 No. 1, enero-junio (2008).
- [9] Xiuqin Lin; Qianhao Zhang; Gengyu Wei, "The clustering algorithm for Chinese texts based on Lingo," *Fuzzy Systems and Knowledge Discovery (FSKD)*, 2011 Eighth International Conference on , vol.2, no., pp.1187,1190, 26-28 July 2011.
- [10] Baeza-Yates, R.; Ribeiro-Nieto, B.: *Modern Information Retrieval: The concepts and technology behind search*, Second edition. Pearson Education Limited, pp. 1-2 (2011)
- [11] Van Rijsbergen, C.J.: *A non-classical logic for information retrieval*. Sparck Jones, K; Willett, P.(Eds.). Readings in Information Retrieval, San Francisco: Morgan Kaufman Publisher, Inc. 1997, pp. 268-272
- [12] Leuski Venue, A.: Evaluating Document Clustering for Interactive Information Retrieval. *Proceeding CIKM '01 Proceedings of the tenth international conference on Information and knowledge management*, Pages 33-40, ACM New York, NY, USA 2001.
- [13] C.J.van Rijsbergen. *Information Retrieval*. Butterworths, London, second edition (1979)
- [14] Hearst, M. A.; Pedersen, J. O.: Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. *In Proceedings of ACM SIGIR*, pages 76-84, Aug. (1996)
- [15] Weiss, G.M.; Davison, B.D.: *Data Mining. To appear in the Handbook of Technology Management*, H. Bidgoli (Ed.), John Wiley and Sons, (2010)
- [16] Jain, A. K.; Murty, M. N.; Flynn, P. J.: *Data Clustering: A Review. ACM Computing Surveys*, Volume 31 Issue 3, Pages 264-323. (1999)
- [17] Jain, R.: A hybrid clustering algorithm for data mining. *CoRR abs/1205.5353* (2012).
- [18] Pavel Berkhin, P.: *Survey of Clustering Data Mining Techniques*. Accrue Software, Inc. (2002)
- [19] Leiva-Valdebenito, S. A.; Torres-Avilés, F. J.: Una revisión de los algoritmos de partición más comunes en el análisis de conglomerados: un estudio Comparativo. *Revista Colombiana de Estadística*, Universidad Nacional de Colombia, Vol. 33, núm. 2, pp. 321-339, Diciembre (2010)

- [20] Zamir, O.; Etzioni, O.: Web Document Clustering: A Feasibility Demonstration. *In Research and Development in Information Retrieval*, pages 46–54, 1998
- [21] Zamir, O.: Clustering Web Documents: A Phrase-Based Method for Grouping Search Engine Results. *PhD thesis*, University of Washington, 1999.
- [22] Pérez-Suárez, A.; Martínez-Trinidad, J.F.; Medina-Pagola, J.E.; Carrasco-Ochoa, J.A.: Algoritmos dinámicos para el agrupamiento con traslape. *Reporte Técnico No. CCC-10-001*, Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, México, 2010.
- [23] Branson, S.; Greenberg, A.: Clustering Web search results using suffix tree methods. Stanford University, *Tech. Rep. CS276A*, Final Project (2002)
- [24] Weiss, D.: Descriptive Clustering as a Method for Exploring Text Collections. *PhD thesis*, Poznań University of Technology, Poznań, Poland, 2006
- [25] Osiński, S.; Stefanowski, J.; Weiss, D.: Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition. *Intelligent Information Processing and Web Mining, Advances in Soft Computing*, Volume 25, pp 359-368 (2004)
- [26] Osiński, S.; Weiss, D.: Conceptual Clustering Using Lingo Algorithm: Evaluation on Open Directory Project Data. *Intelligent Information Processing and Web Mining, Advances in Soft Computing*, Volume 25, pp 369-377(2004)
- [27] Fafat, P. C.; Sikchi, S.S.: Lingo an approach for Clustering. Poonam C. Fafat, S.S.Sikchi. *International Journal of Engineering Research & Technology (IJERT)*, Vol. 1, Issue 3, May (2012)
- [28] Dini, M. A.; Varela, M. A.; Antúnez, V.; Maguitman, A.; Herrera, L.: Soporte Inteligente para el Mantenimiento y Acceso Contextualizado a Repositorios Institucionales. *8ª Jornada sobre la Biblioteca Digital Universitaria - JBUDU2010. "Las redes sociales y las bibliotecas universitarias"*. Buenos Aires, Argentina, Noviembre (2010).
- [29] Osiński, S.; Weiss, D.: Carrot2: Design of a Flexible and Efficient Web Information Retrieval Framework. *Springer Lecture Notes in Computer Science*, vol. 3528, pp. 439-444, Proceedings of the third International Atlantic Web Intelligence Conference (AWIC 2005), Łódź, Poland, (2005)

La informática en los delitos, un panorama en la perspectiva del cómputo forense (CF)

García Garduño E., Salgado Gallegos M., Albarrán Trujillo S. E., Pérez Merlos J.C., Rodríguez Ángeles E.

Universidad Autónoma del Estado de México.
Facultad de Ingeniería

Ingeniería en Computación e Ingeniería Electrónica

Dirección: cerro de Coatepec. s/n Ciudad Universitaria, Toluca Edo. De México.

Teléfono: (01) (722) (2-14-08-55) ext. 111, (2-15-14-53) Fax (2-15-45-12)

Correo electrónico: elygg2002@yahoo.com.mx, msalgadog@uaemex.mx, seat@uaemex.mx,
jccjcc63@yahoo.com, eduroan@yahoo.com

RESUMEN

El Cómputo Forense (CF) o Informática Forense (IF) es una disciplina nueva que permite esclarecer un hecho partiendo de una evidencia digital que, ayudada de herramientas forenses, permite encontrar la prueba de que la información contenida en la evidencia electrónica ha sido alterada o eliminada.

Un análisis forense permite restablecer archivos o documentos que han sido modificados, aun cuando éstos sean eliminados. Las herramientas forenses permiten ver todos los movimientos que se han hecho hasta el momento en que se recogen los dispositivos digitales y analizan las redes a las cuales se conectan.

El propósito de este trabajo es dar a conocer las generalidades del cómputo forense con la finalidad de que el lector pueda ampliar el panorama en una nueva área de conocimiento de la informática. Con base en esto, se abordan conceptos y generalidades propias de la disciplina así como mostrar el estado de arte en que se encuentra en México.

Palabras Claves: Cómputo Forense, Informática Forense, Evidencia digital, Delito informático, Perito forense.

I. INTRODUCCIÓN

El mundo va cambiando día con día, de igual manera evolucionan los avances tecnológicos, la sociedad y también los delitos informáticos. Esto origina nuevas necesidades de fortalecer las normas jurídicas en materia penal, la eficacia y eficiencia de una mejor calidad de vida dentro del ámbito nacional e internacional, de este problema o mal social es de donde surge la Informática o Cómputo Forense.

En la actualidad la vida de las personas ha cambiado así como la forma en la que almacenan la información; los medios digitales tienen un gran auge y la mayoría de la gente cuenta con dispositivos electrónicos que se comunican mediante una red, esto conlleva a una nueva etapa de delitos

informáticos, delitos que deben ser regulados y es en esta regulación donde interviene la informática forense. Su estudio se encamina a esclarecer tales hechos (fraudes electrónicos y telefónicos, robo de identidad, desviaciones de fuertes cantidades de dinero, mal uso de información confidencial, entre otros) afectando los intereses de una organización o de una persona.

El CF esclarece cualquier mal uso de los medios digitales, permitiendo tener un avance en la forma en que los profesionales del derecho llevan a cabo una investigación de esta índole. Mediante el respaldo que se tenga de las leyes, esta disciplina crecerá y no permitirá los supuestos en un juicio debido a que las herramientas forenses, que emplea, son exactas y precisas, por lo tanto no se altera la evidencia que está siendo analizada, son pruebas contundentes y confiables.

Países del mundo, en los últimos años, han realizado esfuerzos sobre este tema, al respecto Instituciones Gubernamentales, Educativas e Investigadores han desarrollado y publicado trabajos y propuestas sobre esta disciplina.

Tal es el caso de Ecuador, por un lado, con “*Retos a superar en la administración de justicia ante los delitos informáticos en el Ecuador*” (2009) [1], esta propuesta de tesis identifica un marco general sobre la conceptualización básica necesaria relativa a los delitos informáticos, tipos de delitos, sus objetivos, importancia, sus principios, la evidencia digital y la informática forense. En conjunto con las regulaciones existentes (leyes) para el manejo de los delitos informáticos, mediante la comprensión de los lineamientos establecidos en la legislación; por otro, en el año 2008, la Policía Nacional de Ecuador, tiene el interés de integrar la IF para encontrar evidencias de los delitos informáticos, crímenes tradicionales y de esta forma ser utilizadas en cualquier proceso judicial como prueba del ilícito mediante la formalización de técnicas y procedimientos para darle valor probatorio a esas evidencias digitales [2].

Argentina, en la Universidad FASTA, un conjunto de investigadores en el 2013 realizaron una propuesta denominada PURI “*Proceso Unificado de Recuperación de la Información*” cuyo objetivo es formalizar un proceso que abarque las fases, tareas, herramientas y actividades aplicables a diversos entornos y dispositivos y que sirva de guía a los expertos forenses [3]. Otra propuesta más, es la presentada en el marco del Simposio Argentino de Informática y Derecho, SID 2013, denominada “*Informática Forense al Servicio de una Justicia Moderna*” [4], ésta propone a la Informática Forense como respuesta a la problemática de los delitos que se presentan por el uso de las tecnologías, trabajo que aborda el camino recorrido por el Poder Judicial de Río Negro para que otros también puedan ahorrar pasos y tiempos en la concreción de un área de investigación forense.

En el 2014, en Colombia, Ariza [5] expone el estado actual de la informática forense en el dispositivo móvil iPhone 3G, resaltando la importancia de contar con un procedimiento estándar para realizar un análisis forense sobre este dispositivo. Asimismo, Rueda y Rico [6], también realizan una investigación para llevar el análisis forense a los terminales móviles y, a través de modelos forenses, las herramientas utilizadas para el manejo de la evidencia que sirva como soporte en un proceso judicial.

Chaves (2011) [7] explica la aplicación que tiene la informática forense en Costa Rica, así como la legislación existente en materia de delitos informáticos en ese país.

Rodríguez y Doménech [8] en su artículo “*La informática forense: El rastro digital del crimen*”, publicado en el 2011, se acerca al proceso técnico y científico para recopilar y analizar pruebas en evidencias digitales de dispositivos informáticos o electrónicos con el objetivo de presentarlas ante un tribunal.

Como es de apreciarse, el cómputo forense cobra una gran importancia en los últimos años dentro de un proceso penal. Con base en lo anterior, este trabajo tiene la finalidad de dar a conocer esta disciplina, así como el estado del arte en México.

II. DESARROLLO

La informática o cómputo forense

El Cómputo Forense está adquiriendo una gran importancia dentro del área de la informática debido a que actualmente la mayoría de las personas almacenan la información en medios digitales, como un disco duro, memorias USB, Ipad, ipod, memorias externas, memorias de cámara, celulares, etc., es decir, todo aquel dispositivo que pueda almacenar información.

Hoy en día el internet se ha involucrado en la vida cotidiana de la mayoría de la gente: el uso de las computadoras por parte de las compañías de negocios, cuentas de correo, transacciones bancarias, etc., lo cual abre paso a un nuevo campo de investigación criminal denominado Cómputo Forense que consiste en recuperar la información de una evidencia de una manera confiable y sobretodo que sirva para sustentar un caso de manera legal, es importante

resaltar que actualmente en México no existen leyes que regulen un procedimiento Informático Forense.

Un análisis forense permite restablecer archivos o documentos que han sido modificados, aún cuando éstos sean eliminados. Las herramientas forenses mediante esta disciplina permiten ver todos los movimientos que se han hecho hasta el momento en que se recogen los dispositivos digitales y analizan las redes a la cuales se conectan.

En la actualidad la mayoría de la gente cuenta con dispositivos electrónicos y se comunican mediante una red, esto conlleva a una nueva etapa de delitos informáticos, delitos que deben ser regulados, contexto de la informática forense. Su estudio se encamina a esclarecer tales hechos (fraudes electrónicos y telefónicos, robo de identidad, desviaciones de fuertes cantidades de dinero, mal uso de información confidencial, entre otros) afectando los intereses de una organización o de una persona.

Las herramientas que utiliza el Cómputo Forense permiten realizar un análisis detallado de la evidencia y dar información de suma importancia para aclarar el hecho en el que se ha incurrido un delito, esclarece cualquier mal uso de los medios digitales, lo cual permite tener un avance en la forma en que los profesionales del derecho llevan a cabo una investigación de esta índole, por lo cual es importante que tengan conocimiento de esta disciplina.

En una entrevista realizada en el año 2006 al Doctor Jeimy J. Cano [9] sobre responder a la pregunta ¿Cuánto se puede tardar en reunir las suficientes pruebas o evidencias para dar con la persona que es autor de un ataque? responde al respecto:

“Es una pregunta complicada de responder, pues muchas veces el informático forense debe prepararse para fallar en identificar a la persona real que cometió el ataque. Pues la versatilidad que ofrece Internet para enmascarar direcciones IP, correos electrónicos, entre otros aspectos, sugiere un gran conocimiento técnico y paciencia por parte de los atacantes, los cuales también consideran estrategias “anti-forenses” que limitan las investigaciones y la efectividad de las mismas. La recolección de pista puede ser demorada, algunos casos pueden llevar años en esta labor.”

Por lo tanto, el tiempo que se tarda en encontrar una evidencia y a la persona que es autor de un ataque depende del tipo de evidencia y de las diferentes herramientas que se empleen para detectar la anomalía o intrusión en los sistemas informáticos.

El CF es una ciencia relativamente nueva, los países en los que tiene mayor auge son Argentina, Colombia, Bolivia, Venezuela, España y Estados Unidos, en estos países la IF ya está regulada dentro del marco de la ley y es válida como evidencia digital presentada en un caso legal, en estos casos el especialista de la IF posee sólidos conocimientos técnicos, prácticos, conoce las herramientas de uso a emplear y está al día en bugs (vulnerabilidades) de sistemas (sistemas operativos, software y hardware) [9].

La ciencia forense es sistemática y se basa en hechos premeditados para recabar pruebas para luego analizarlas. El término computer forensic, proceso forense, informática forense o forensic computing en el esquema de certificación CISSP (Certified Information Systems Security Professional) hace referencia a:

“Una ciencia, técnica o disciplina relacionada encargada de la investigación de los delitos y abusos relacionados que tienen que ver con los computadores y con las redes que los conectan, de una manera repetible y competente” [10].

Para Juan Carlos Guel, jefe del Departamento de Seguridad en Cómputo de la Dirección General de Servicios de Cómputo Académico y Coordinador del Equipo de Respuesta a Incidentes en Seguridad en Cómputo de la UNAM, señala:

“La Informática o cómputo forense es un conjunto de técnicas especializadas que tiene como finalidad la reconstrucción de hechos pasados basados en los datos recolectados, para lo cual se procesa la información que pueda ser usada como evidencia en un equipo de cómputo” [11].

Para G. Noblett, Michael [12] es:

“Disciplina de las ciencias forenses, que considerando las tareas propias asociadas con la evidencia, procura descubrir e interpretar la información en los medios informáticos para establecer los hechos y formular las hipótesis relacionadas con el caso; o como la disciplina científica y especializada que entendiendo los elementos propios de las tecnologías de los equipos de computación ofrece un análisis de la información residente en dichos equipos”.

Hikal [13] al respecto opina:

“...se enfoca a aquellos casos en los que se utiliza el equipo de cómputo como medio para cometer una conducta presuntamente delictuosa así, como cuando el equipo es violentado en sus partes lógicas (programas) o en sus partes físicas”.

Finalmente, con base en las definiciones anteriores se puede decir que el CF es una disciplina que se encarga de analizar evidencias digitales para reconstruir hechos pasados haciendo uso de sus herramientas forenses, ayuda a encontrar las pruebas clave para esclarecer un hecho en el que se ha incurrido en un delito en el uso de la tecnología.

El cómputo forense y el marco jurídico

La informática forense hace entonces su aparición como una disciplina auxiliar de la justicia moderna, para enfrentar los desafíos y técnicas de los intrusos informáticos, así como garantizar la verdad de la evidencia digital que se pudiese aportar en un procedimiento judicial [12].

Desde 1984, el Laboratorio del FBI y otras agencias que persiguen el cumplimiento de la ley, empezaron a

desarrollar programas para examinar evidencia en medios digitales y electrónicos.

Con la finalidad de averiguar qué ha ocurrido y quién ha sido el presunto autor relacionado con el uso de las tecnologías de información se establece una relación entre el marco jurídico y la tecnología denominada Informática Forense [14].

La prioridad del CF es preservar lo más íntegramente posible las evidencias del crimen, esto significa resguardar y proteger el hardware y las redes a las cuales se conecta, es decir colocar el sistema fuera de servicio (offline). Si el sistema, por parte del administrador o empresa, requiere que siga funcionando, la investigación forense no podrá seguir el rumbo correcto porque se podría seguir haciendo mal uso del hardware o la red, lo cual ocasionaría cualquier posibilidad de persecución del intruso que está haciendo mal uso de los sistemas de información.

La evidencia digital se puede modificar o eliminar fácilmente por la persona que está violando la información, sin embargo esto no es cierto, esto sólo es en apariencia, la IF permite reconstruir todos los archivos alterados o eliminados, se hace una copia exacta de la evidencia recabada y con el análisis forense se encuentran elementos suficientes para esclarecer un hecho.

Etapas y herramientas del cómputo forense

El cómputo forense lleva a cabo una serie de etapas para recabar la(s) evidencia(s) que permitan esclarecer algún hecho y determinar si se ha incurrido en un delito: identificación, preservación, análisis y presentación, las cuales se detallan a continuación [15]:

A) Identificación

Es muy importante conocer los antecedentes, situación actual y el proceso que se quiere seguir para tomar la mejor decisión con respecto a las búsquedas y la estrategia de investigación. Incluye muchas veces la identificación del bien informático, su uso dentro de la red, el inicio de la cadena de custodia (proceso que verifica la integridad y manejo adecuado de la evidencia), la revisión del entorno legal que protege el bien y del apoyo para la toma de decisiones con respecto al siguiente paso una vez revisados los resultados.

B) Preservación

Este paso incluye la revisión y generación de las imágenes forenses de la evidencia electrónica para realizar el análisis. Dicha duplicación se realiza utilizando tecnología de punta para mantener la integridad de la evidencia y la cadena de custodia que se requiere. El realizar una imagen forense se refiere al proceso que se requiere para generar una copia “bit-a-bit” de todo el disco, el cual permite recuperar, en el siguiente paso, toda la información contenida y borrada del disco duro. Para evitar la contaminación del disco duro, normalmente se ocupan bloqueadores de escritura de hardware, los cuales evitan el contacto de lectura con el disco, lo que provocaría una alteración no deseada en los medios.

C) Análisis

Proceso de aplicar técnicas científicas y analíticas a los medios duplicados por medio del proceso forense para encontrar pruebas de ciertas conductas. Se pueden realizar búsquedas de cadenas de caracteres, acciones específicas del o de los usuarios de la máquina como son el uso de dispositivos de USB (marca, modelo), búsqueda de archivos específicos, recuperación e identificación de correos electrónicos, recuperación de los últimos sitios visitados, recuperación del caché del navegador de Internet, entre otros.

D) Presentación

Es el recopilar toda la información que se obtuvo a partir del análisis para realizar el reporte y la presentación a los abogados, la generación (si es el caso) de un peritaje y de su correcta interpretación sin hacer uso de tecnicismos.

Para llevar a cabo estas etapas, existen diversas herramientas que utilizan los peritos forenses para realizar el análisis de hardware, software o las redes a las cuales se conectan y saber a ciencia cierta qué mal uso hicieron en el sistema, hardware o las redes a las que se conectan [14]. Algunas de estas herramientas a mencionar: Marco Digital Forensics¹, CAINE², Marco Digital Forensics³, X-Ways Forense⁴, SANS Investigación Forense Toolkit - SIFT⁵, entre otras⁶.

El aprovechamiento de las fallas tecnológicas sobre infraestructuras es un escenario perfecto para que se cultiven tendencias relacionadas con intrusos informáticos, esta información es analizada por la Informática Forense utilizando procedimientos y herramientas adecuadas para analizar y recabar la prueba verídica que dé fe de algún hecho sospechoso y tal motivo encuentre a la persona que es culpable.

Mediante el respaldo que se tenga de las leyes, esta disciplina crecerá y no permitirá los supuestos en un juicio debido a que las herramientas forenses que emplea la Informática Forense son exactas y precisas, por lo tanto no se altera la evidencia que está siendo analizada, son pruebas contundentes y confiables.

Ventajas

La informática forense, como nueva disciplina, tiene grandes ventajas, por ejemplo [16]:

- Se ayuda de herramientas que permiten dar de manera precisa los datos encontrados en cualquier dispositivo electrónico capaz de almacenar información.
- Es fiable por la objetividad y exactitud debido a que la información encontrada mediante sus herramientas no permite alterar información alguna.

- Un especialista en CF ofrece información exacta, completa, clara, precisa, veraz, objetiva y neutra, puesto que proviene de un elemento electrónico en el que no cabe subjetividad alguna.
- Es posible obtener información de manera veraz y rápida, imposible de obtener mediante cualquier otro procedimiento informático.
- La evidencia electrónica permite el esclarecimiento de ciertos delitos en los que estas pruebas son el único medio probatorio existente.
- Varios profesionales opinan que el uso de documentos y firmas electrónicas favorecen el desarrollo del comercio electrónico y con el procedimiento empleado por el CF permite encontrar cualquier evidencia en caso de algún fraude o infiltración a la red.

La informática forense en México

En los Estados Unidos de Norteamérica, la primera propuesta de legislar con este respecto, se presentó en 1977 por el senador Ribicoff en el Congreso Federal [17].

Años después, en 1983 en París, la OECD (Organización para la Cooperación y Desarrollo Económico) designó un comité de expertos para discutir el crimen relacionado con las computadoras y la necesidad de cambios en los códigos penales. El dictamen de esta organización, recomendó a los países miembros, la modificación de su legislación penal de forma que se integraran los nuevos delitos informáticos. Por tanto el borrador que elaboró el comité sirve para combatir los delitos que se cometen en computadoras a través de la redes y permite proteger a las personas que han sido objeto de un delito informático [17].

En México fue hasta el año de 1999 en que se incorporó a la legislación punitiva los delitos informáticos con el fin de combatir los delitos informáticos, sobre todo los cometidos a través de las redes de telecomunicaciones en Internet, por ejemplo, transacciones de fondos ilegales, la oferta de servicios ilegales, la violación de los derechos de autor, delitos que violan la dignidad humana y la protección de los menores [18].

En México existen leyes para delitos informáticos, pero no hay una ley o reforma que hable de la validez de la evidencia que emerge de un procedimiento forense realizado por la informática forense, ésta es una problemática que necesita ser regulada y tenga validez en caso de que se viole el uso de un dispositivo electrónico o un mal uso en una red.

Con base en lo anterior, también en México se han hecho esfuerzos y propuestas para legislar los delitos informáticos. Tal es el caso, de la propuesta metodológica de análisis forense para dispositivos de telefonía celular del Instituto Politécnico Nacional (IPN), la cual propone un método para la reconstrucción de eventos que se encarga de la creación, validación, prueba y aceptación/rechazo de la hipótesis relacionada con el incidente [19]; así mismo, el IPN tiene diferentes propuestas en el mismo sentido, para

¹ <http://www.digital-forensic.org/>

² <http://www.caine-live.net/>

³ <http://www.digital-forensic.org/>

⁴ <http://www.x-ways.net/forensics/>

⁵ <http://digital-forensics.sans.org/community/downloads>

⁶ <https://prezi.com/kcoki3kq5wsj/21-herramientas-mas-populares-de-informatica-forense/>

discos duros [20], para estudios de metodologías de análisis forense digital [21], cómputo forense en red [22] y para el uso de diferentes herramientas como Honeypots [23].

La IF en México, ha tenido desarrollos aplicados en diferentes áreas como lo es la financiera, tal es el caso de la Universidad Veracruzana 2012 con el trabajo intitulado “*La Auditoría Forense sobre el Fraude Financiero y el lavado de activos*” (2012) que presenta el cómo detectar y prevenir estos delitos con mecanismos efectivos mediante procedimientos que se llevan a cabo para realizar una Auditoría Forense [24].

En la Facultad de Derecho de la UNAM, se han generado diversas actividades y esfuerzos académicos a fin de acercar contenidos novedosos, actuales y con base en la legislación existente, los cuales abordan los delitos virtuales, tal es el caso del Segundo Encuentro Latinoamericano sobre Ciberseguridad: delitos cibernéticos e informática forense celebrado ahora en el 2015 [25].

Loredo [26] en la Universidad Autónoma de Nuevo León, da a conocer los principales delitos informáticos y los riesgos que estos generan para la sociedad, las empresas y los gobiernos, asimismo, las principales leyes que existen en México para tipificar este tipo de delitos y de los acuerdos internacionales de los países han firmado y desarrollado con el fin de combatir este problema, esto en el 2013.

Lo anterior, deja visto que en México existe un gran interés en trabajar e integrar el cómputo forense en las diferentes disciplinas.

El cómputo forense en las empresas

Actualmente la realidad competitiva de las empresas, hace imprescindible para ellas acoplarse a las tecnologías de seguridad de información disponibles, por lo que es prioritario que las empresas tomen medidas para proteger su información estratégica tanto de ataques internos como externos y a todos los niveles.

La informática forense va mucho más allá de verificar e identificar la intrusión o ataque en los sistemas informáticos de una empresa, una labor importante es adiestrar y concientizar al personal involucrado dentro de la red organizativa y en indicar las medidas preventivas a seguir para evitar que la información de la empresa sea vulnerable.

A medida que el internet crece, lo hace de igual manera el número de acciones ilegales contra la seguridad de las redes corporativas y de las personas, por ello hay que preguntarse:

- A) ¿Quién lleva a cabo estos incidentes?
- B) ¿Qué los posibilita?
- C) ¿Qué, quién y por qué los provoca?
- D) ¿Cómo se evitan?
- E) ¿Cómo se producen?
- F) ¿Qué medidas se deben implementar?

G) ¿Se pueden realizar acciones jurídicas legales?

Es necesario que las organizaciones y las personas protejan su seguridad, hoy en día el cómputo forense no sólo se aplica en nuestro país desde el punto de vista de persecución de delitos, también a nivel interno en las organizaciones, principalmente para detectar si alguien de la misma empresa ha realizado comunicación o transferencias de información de la empresa a un tercero.

III. CONCLUSIONES

El Cómputo Forense no sólo se ha transformado en una disciplina clave en la investigación, también es utilizada para asegurar los organismos de control y transparencia de las empresas, de tal manera que la seguridad de realizar una copia forense de la empresa sirva para la prevención de algún mal manejo.

La computadora o teléfono celular saben más de lo que uno cree, las contraseñas, visitas a Internet, programas que se ejecutan en la computadora, fotografías que se descargan, compras que se realizan en línea, todo queda registrado en los dispositivos electrónicos que se utilizan y aunque se borren los historiales de archivos o programas que utilicen, la evidencia siempre se queda en los dispositivos.

El CF es una disciplina muy novedosa y que puede dar mucho auge y ayudar a encontrar pistas de forma precisa, no importa que la información sea borrada, ya que el uso de sus herramientas permiten recuperar todo tipo de archivos, mostrando todos los datos, cuándo fue creado, cuántas modificaciones sufrió, cuándo se eliminó, fechas de acceso, cuántas veces se utilizó, además de mostrar las características de cada uno de los archivos o documentos que se encuentren dentro de la evidencia electrónica.

Sin duda alguna por ser una disciplina reciente y nueva, no es conocida y sobretodo no está amparada por leyes que respalden la veracidad de las mismas pruebas encontradas durante un procedimiento forense a menos de que ésta sea una orden girada por un juez.

Son muy pocos países los que ya tienen una reforma en el uso de CF; en México, es necesaria una reforma que ampare la evidencia digital encontrada por un informático forense.

La falta de una regulación propia y sistemática, así como por la falta de jurisprudencia, los profesionales de derecho muestran un temor a la vulnerabilidad y facilidad con que estas pruebas pueden ser manipuladas, dado su alto grado de volatilidad, lo que resulta uno de los principales inconvenientes a la hora de probar su autenticidad. Algunos opinan que se trata de pruebas muy técnicas que son desconocidas para jueces y fiscales y que resultan difíciles de explicar, de ahí el rechazo a admitirlas en juicio.

Se debe trabajar mucho en esta labor para que en México se aminoren ataques informáticos.

IV. REFERENCIAS

- Ureta, L.A., *Retos a superar en la administración de justicia ante los delitos informáticos en el Ecuador*, in *Maestría en Sistemas de Información Gerencial 2009*, Escuela Superior Politécnica del Litoral: Guayaquil - Ecuador.
- Álvarez, M.D. and V.A. Guamán, *Metodologías, estrategias y herramientas de la informática forense aplicables para la Dirección Nacional de Comunicación y Criminalística de la Policía Nacional*, in *Ingeniería de Sistemas CUE*. 2008, Universidad Politécnica Salesiana: Ecuador.
- Di Iorio, A., et al., *La recuperación de la información y la informática forense: Una propuesta de proceso unificado*. 2013, Universidad FASTA: Argentina.
- Semprini, G. and A. Bozzetti. *Informática Forense al Servicio de una Justicia Moderna*. in *Simposio Argentino de Informática y Derecho, SID*. 2013. Argentina: SID.
- Ariza, A., J. Ruíz, and J. Cano, *iPhone 3G: Un Nuevo Reto para la Informática Forense*. 2014, Universidad Pontificia Javeriana: Bogotá-Colombia.
- Rueda, J.S. and D. Rico, *Análisis forense digital en dispositivos móviles*. *Revista de Semilleros de Investigación*, 2014. **1**(1).
- Chaves, M.A., *Panorama general de la Informática Forense y de los delitos informáticos en Costa Rica*. *InterSedes*, 2011. **7**(12).
- Rodríguez, F. and A. Doménech, *La informática forense: El rastro digital del crimen*. *Quadernos de criminología: revista de criminología y ciencias forenses*, 2011(14): p. 14-21.
- Cano, J., *Introducción a la informática forense: Una disciplina técnico-legal*. *Sistemas*, Asociación Colombiana de Ingenieros de Sistemas (ACIS), 2006. **96**: p. 64-73.
- Bertolín, J.A., *Seguridad de la información. Redes, informática y sistemas de información*. 2008: Editorial Paraninfo.
- Gutiérrez, G., *Cómputo forense, tras las huellas del infractor*. *Enter@te* en línea: Internet, *Cómputo y Telecomunicaciones*, 2004. **3**.
- Noblett, M.G., M.M. Pollitt, and L.A. Presley, *Recovering and examining computer forensic evidence*. *Forensic Science Communications*, 2000. **2**(4): p. 1-13.
- Hikal, W., *Introducción al estudio de la Criminología*. 2009, México: Porrúa.
- Xombra. *Cómputo Forense o Informática Forense*. 2012 citado: 04 de Septiembre de 2012; Recuperado de: http://www.xombra.com/go_news.php?articulo=1942.
- Cano, J. and J. Pimentel, *Consideraciones Sobre el Estado del Arte del Peritaje Informático y los estándares de manipulación de pruebas electrónicas en el mundo*. *Derecho Comunicaciones y Nuevas Tecnologías*, Universidad de los Andes, 2007. **3**: p. 1-23.
- Ramírez, G., *Informática Forense*. Publicación Universidad San Carlos de Guatemala, 2008: p. 2.
- Chungata, A.M., *El fraude como delito informático*, in *Facultad de Jurisprudencia y Ciencias Políticas y Sociales 2015*, Universidad de Cuenca: Cuenca, Ecuador.
- Ruiz, J.E., *Delitos informáticos en México*. *Revista del Instituto de la Judicatura Federal*, 2009. **28**: p. 207-236.
- Santes, L., *Propuesta de una metodología de análisis forense para dispositivo de telefonía celular*, in *Ciencias de Ingeniería Microelectrónica*. 2009, Instituto Politécnico Nacional México.
- Castañeda, F.J., et al., *Evaluación de herramientas para análisis forense orientado a discos duros*, in *Escuela Superior de ingeniería mecánica y eléctrica, ESIME*. 2009, Instituto Politécnico Nacional: México.
- De León, J.F., *Estudio de metodologías de análisis forense digital*, in *Especialidad en Seguridad Informática y Tecnologías de Información*. 2009, Instituto Politécnico Nacional: México.
- Canizal, J.C., et al., *Esquema de forensia en red*, in *Ciencias de la Informática e Ingeniería en Comunicaciones y Electrónica*. 2010, Instituto Politécnico Nacional: México.
- Ángeles, J., *Cómputo forense mediante la Tecnología Honeypot*, in *Especialidad en Seguridad Informática y Tecnologías de Información*. 2010, Instituto Politécnico Nacional: México.
- Juárez, E.J., *La auditoría forense sobre el fraude financiero y el lavado de activos*, in *Facultad de*

Contaduría y Administración. 2012, Universidad Veracruzana: Xalapa, Veracruz.

25. García, D. *Actividad académica y tecno-científica sobre ciberseguridad, ciberdelincuencia y protección de los menores*. in *Segundo Encuentro Latinoamericano sobre Ciberseguridad: delitos cibernéticos e informática forense*. 2015. México: Universidad Nacional Autónoma de México.
26. Loredó, J.A. and A. Ramírez, *Delitos informáticos: su clasificación y una visión general de las medidas de acción para combatirlo*. Celerinet, 2013: p. 44-51.

AUTORES

Ing. Eladia García Garduño es Ingeniera en Computación por la Facultad de Ingeniería de la UAEMex.

M. en I. Mireya Salgado Gallegos es Ingeniera en Computación con Maestría en Informática por la Facultad de Ingeniería de la UAEMex y Candidata a Doctor por la Universidad Anáhuac.

M. en A. Silvia Edith Albarrán Trujillo es Ingeniera en Computación con Maestría en Administración por la UAEMex y Candidata a Doctor por la Universidad Anáhuac.

M. en I. Juan Carlos Pérez Merlos es Ingeniero en Electrónica y obtuvo la Maestría en Ingeniería en Informática en la Facultad de Ingeniería de la UAEMex, Maestría en Electrónica en el CICESE y Candidato a Doctor por la Universidad Anáhuac.

Dr. Eduardo Rodríguez Ángeles es Ingeniero en Control y Automatización por la ESIME-IPN, M en C. en Control automático y Doctor en Ciencias en control automático por el CINVESTAV-IPN.

Configuración PIC-Raspberry Pi® modelo B, con fines de monitoreo de señales analógicas

Juan Carlos Morelos Romero
Instituto Politécnico Nacional
ESIME Culhuacán
Distrito Federal, México
jcmorelos_mr@hotmail.com

Gustavo Delgado Reyes
Instituto Politécnico Nacional
ESIME Culhuacán
Distrito Federal, México
dengue_mgs4@hotmail.com

Asdrúbal López Chau
Universidad Autónoma del Estado de
México
CU Zumpango
Estado de México, México
alchau@uaemex.mx

Resumen— Raspberry Pi® es una computadora embebida cuyo auge ha aumentado día a día, en este sentido en muchos proyectos realizados con esta computadora resulta indispensable medir señales ambientales tales como tensión, temperatura, luz, corriente, etc., estas son señales analógicas que la Raspberry Pi® no puede registrar por sí sola. En este sentido, es importante proponer un arreglo que sea económico y práctico, comparado con dispositivos dedicados a convertir señales analógicas en señales digitales y con capacidad de implantarse en una Raspberry Pi®. En el presente trabajo se propone usar el convertidor Analógico/Digital (A/D) de un microcontrolador con una Raspberry Pi®. Lo que se busca medir es la tensión (señal analógica) a la salida de una resistencia variable (potenciómetro), posteriormente la señal analógica será convertida a una señal digital utilizando el microcontrolador y después el mismo microcontrolador enviará la información obtenida a la Raspberry Pi® utilizando comunicación serial.

Palabras clave—Raspberry Pi, PIC, Comunicación serial, convertidor A/D, computadora.

Abstract— Raspberry Pi® is an embedded computer whose rise has increased day by day, in this sense in many projects it is essential to measure environmental signals such as voltage, temperature, light, electric current, etc., these analog signals Raspberry Pi® can not measure itself. In this sense, it is important to propose an arrangement that is economical and practical, compared with dedicated converters, the main purpose of this arrangement is that it is able to be implemented on the embedded computer Raspberry Pi®. In this paper we propose using the Analog / Digital converter (A / D) of a microcontroller with a Raspberry Pi®. This is to measure the voltage (analog signal) at the output of a variable resistor (potentiometer), then the analog signal is converted to a digital signal using microcontroller and then this device will send the information obtained to the Raspberry Pi® using serial communication.

Keywords—Raspberry Pi, PIC, Serial communication, A/D converter, computer.

I. INTRODUCCIÓN

En la actualidad muchos de los aparatos electrónicos que tenemos en nuestros hogares cuentan por lo menos con un sistema computacional integrado, estos sistemas

computacionales o computadoras son conocidos como sistemas embebidos o computadoras embebidas. En [1], se define un sistema embebido como una combinación de circuitos de computadoras y software que se integran en un producto con fines tales como el control, la vigilancia y la comunicación sin intervención humana.

Una computadora embebida normalmente está integrada dentro de un sistema más grande. Actualmente, existen computadoras embebidas que pueden ser utilizadas como computadoras de escritorio tal es el caso de Raspberry Pi®. Este tipo de computadoras embebidas ofrecen una interfaz de usuario como lo es un sistema operativo con prácticamente todas las características y herramientas que tienen los sistemas operativos convencionales en computadoras personales.

A. Raspberry Pi®

Raspberry Pi® es un pequeño computador de bajo costo, del tamaño de una tarjeta de crédito que se puede conectar a un monitor o televisión; usa un teclado y ratón estándar, es una plataforma adecuada en la que se puede aprender lenguajes de programación como Scratch, Python y C. Es capaz de hacer todo lo que se espera que una computadora de escritorio haga, desde navegar en internet y reproducir videos de alta definición, hasta hacer hojas de cálculo o jugar videojuegos [2], adicionalmente Raspberry Pi® cuenta con procesador de textos, compilador propio, software para computación científica como octave® y mathematica®. En lo que a hardware se refiere esta computadora embebida cuenta con pines General Purpose Input Output (GPIO), los cuales se utilizan para interactuar con dispositivos electrónicos que no pertenecen a la misma Raspberry Pi®, en general estos pines sirven como interfaz para interactuar con prácticamente cualquier proceso que requiera automatización. Cabe destacar que es a través de estos pines de propósito general que la comunicación serial se lleva a cabo. La disposición de los GPIO's en la tarjeta es como se aprecia en la Fig. 1.

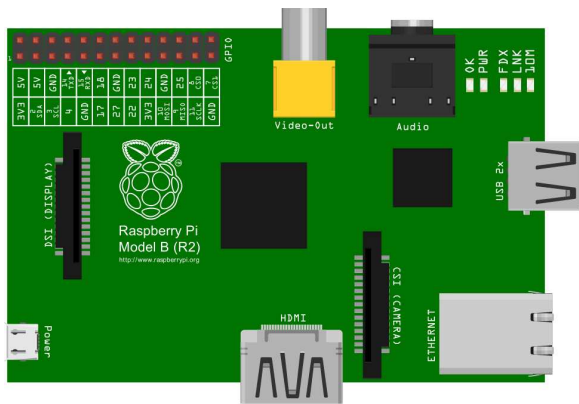


Fig. 1. Raspberry Pi® modelo B revisión 2.

B. Comunicación serial

El ser humano y otros seres vivos tiene la necesidad de comunicarse. A partir de la aparición de la electrónica y avances tecnológicos se ha ampliado la idea de que no solo los seres vivos necesitan comunicarse, también es necesaria la comunicación en aparatos electrónicos.

Actualmente existen distintos métodos para lograr la comunicación entre dispositivos electrónicos, uno de ellos es la comunicación serial.

En [3] y [4] se menciona que un puerto serie o puerto serial es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez.

Los tres tipos de comunicación en serie son:

- Simplex: En este caso el emisor y el receptor están perfectamente definidos y la comunicación es unidireccional.
- Duplex: En este caso ambos extremos del sistema de comunicación cumplen funciones de transmisor y receptor y los datos se desplazan en ambos sentidos pero no de manera simultánea.
- Full-Duplex: El sistema es similar al dúplex, pero los datos se desplazan en ambos sentidos simultáneamente.

Los tres protocolos en serie que se manejan son [5]:

- El protocolo Serial Peripheral Interface (SPI) es un enlace de datos síncrono, y precisamente como es síncrono requiere una señal de reloj. Es un protocolo full-duplex.
- El protocolo serie Circuito inter-integrado (I²C) es un enlace de datos síncrono. También conocido como interfaz de dos hilos (TWI). Requiere solo dos cables (excluyendo las conexiones a tierra y potencia). Otras normas se han ideado sobre la base de este marco, como el SMBus [6].

- Universal Asynchronous Receiver Transmitter (UART) es un protocolo en serie que utiliza dos pines. El protocolo UART no necesita ninguna señal de reloj, como bien indica el adjetivo *asíncrono* en su nombre.

Las interfaces definidas por normas son las siguientes [7]:

- RS-232: Se puede utilizar para comunicaciones seriales en distancias de hasta 50 pies.
- RS-422: Las ventajas en comparación con la interfaz 232 son que es más inmune al ruido y se puede comunicar a mayores distancias.
- RS-485: Es una mejora sobre RS-422 ya que incrementa el número de dispositivos que se pueden conectar (de 10 a 32).

La velocidad de transmisión indica el número de bits por segundo que se transfieren, y se mide en baudios (bauds) [7].

II. DESARROLLO

En esta sección del presente artículo se describe la manera en que se ha implantado un enlace serial entre un PIC de la familia de microcontroladores MICROCHIP®, cuyo modelo es PIC16F887 (el cual será referido solamente como PIC en el presente artículo), y la computadora embebida Raspberry Pi®, en este contexto se ha empleado el modelos B y cabe destacar que el enlace de comunicación que se ha implantado es de tipo simplex, donde el transmisor se define del lado del PIC y el receptor se define del lado de la Raspberry Pi®.

A. Programación del microcontrolador

Algunas de las características del PIC son las siguientes [8]:

- Arquitectura Reduced Instruction Set Computer (RISC).
 - El microcontrolador cuenta con solo 35 instrucciones diferentes.
 - Todas las instrucciones son uni-ciclo excepto por las de ramificación.
- Frecuencia máxima de operación 20MHz.
- Voltaje de la fuente de alimentación de 2.0V a 5.5V.
 - Consumo: 220 μ A (2.0V, 4MHz), 11 μ A (2.0V, 32Hz) 50nA (en modo espera).
- 35 pines de entrada/salida.
- Memoria ROM de 8K con tecnología Flash.
 - El chip se puede re-programar hasta 100000 veces.
- Módulo Universal Synchronous Asynchronous Receiver Transmitter (USART), soporta las comunicaciones seriales RS485, RS232 y LIN 2.0.

- Puerto Serie Síncrono Maestro (MSSP), soporta los modos SPI e I²C.
- Convertidor A/D.
 - 14 canales.
 - Resolución de 10 bits.
- 3 temporizadores/contadores independientes.

Debido a que en esta parte del artículo se describen los detalles de la implementación de la comunicación serial entre el PIC y Raspberry Pi®, vale la pena destacar que en este caso el potenciómetro varía los valores de tensión entre 0 y 5v, y que posteriormente estos valores analógicos serán monitoreados por el microcontrolador. El potenciómetro funciona como una resistencia variable y se comporta como un divisor de tensión.

Para este trabajo el potenciómetro ha servido como objeto de prueba para reproducir las señales analógicas que serán leídas por el PIC. En este caso se usa un potenciómetro con valor de 10kΩ, conectado a un voltaje de entrada de 5v.

Un divisor de tensión es una configuración para repartir la tensión de la fuente, como se aprecia en la Fig. 2.

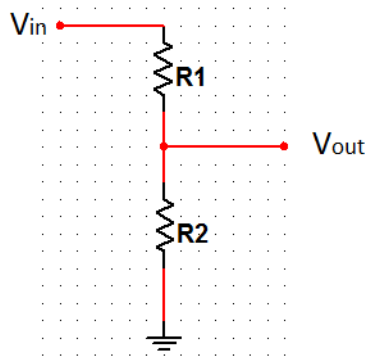


Fig. 2. Divisor de tensión.

El divisor de tensión es descrito por:

$$V_{out} = \frac{R_2 \times V_{in}}{R_1 + R_2} \quad (1)$$

Donde:

V_{out} es el voltaje de salida.

R₂ es el valor de la segunda resistencia.

R₁ es el valor de la primera resistencia.

V_{in} es el voltaje de entrada.

Aplicando (1) se puede calcular el valor de tensión mínima que se puede obtener del potenciómetro, se tiene un voltaje de entrada de 5v, R₂=0 y R₁=10kΩ, por lo tanto se obtiene un voltaje de salida de 0v. De una manera similar para obtener el

valor de tensión máxima, se tiene un voltaje de entrada de 5v, R₂=10kΩ y R₁=0, por lo que se obtiene un voltaje de salida de 5v.

Continuando con la programación del PIC, el compilador que se utilizó para desarrollar el código fue PIC C Compiler, en [9] se describe que un compilador en C contiene operadores estándar del lenguaje C y funciones incorporadas en librerías específicas para los PIC, contiene más de 307 funciones integradas que simplifican el acceso al hardware. Se incluyen funciones tales como:

- Temporizadores y módulos PWM.
- Convertidores A/D.
- Controladores de LCD.
- Entre otras.

Entre las características del PIC se observa que cuenta con 14 canales para convertidor A/D, para el caso de este trabajo solo uno es necesario para convertir los datos analógicos provenientes del potenciómetro.

En el desarrollo del código que se necesita para programar el PIC se define al pin que lea la señal analógica proveniente del potenciómetro y que sea convertida a una señal digital capaz de ser enviada vía serial, también en el mismo código se define al pin requerido para la comunicación serial. Entre los pines del PIC se encuentran dos pines dedicados para la comunicación serial: un transmisor y un receptor denominados TX y RX respectivamente. Para el caso del PIC es requerido el pin TX.

Utilizando la interfaz rs232 para el código del programa, se define la velocidad de transmisión a 9600 baudios y el pin 6 por el cual enviara la información.

```
#include <16f887.h>
#use rs232(baud=9600, xmit=pin_c6)
main()
{
  setup_adc_ports(SAN0);
  setup_adc(ADC_CLOCK_INTERNAL);
  while(true)
  {
    volt=5.0*analog/1023.0;
    sprintf (&cad, "%f", volt);
    puts(cad);
  }
}
```

Fig. 3. Segmento del código donde se definen los pines que se utilizan, desarrollado en el software PIC C Compiler.

En la Fig. 3 se observa el código que se utiliza para programar el PIC, en él se define al convertidor A/D (pin SAN0). Un valor convertido está como un determinado número de escalones y como el número de escalones con resolución de 10 bits es de 1023 (1024 en realidad pero como en el PIC comienza contando en 0, contara hasta 1023) se realiza una regla de tres, representando que cuando se lean 1023 escalones el valor del voltaje será de 5v, la expresión es la siguiente:

$$Volt = \frac{5 \times read}{1023} \quad (2)$$

Donde:

volt es valor de la tensión medida.
read es valor leído por el pin SAN0.

Posteriormente del cálculo de la regla de tres, en el código el valor numérico será convertido a una cadena de caracteres utilizando la función `sprintf()` y será enviado utilizando la función `puts()`. Como el PIC leerá varios datos lo anteriormente explicado estará dentro de un ciclo `while()`.

Se decide convertir los datos numéricos a cadena de caracteres ya que del lado de la Raspberry Pi® las librerías para la comunicación serial no leen datos numéricos, por el contrario si leen cadena de caracteres [10].

B. Programación de la Raspberry Pi®

Ya que se han abordado ciertas características de hardware del microcontrolador empleado, resta mencionar algunas propias que definen al modelo B revisión 2 de Raspberry Pi® (modelo utilizado):

- Usa un chip de tipo SoC (System on a Chip) el cual es un Broadcom BCM2835 que está compuesto por un microprocesador ARM 1176JZF -S a 700 MHz y un GPU (Graphics Processin Unit) Broadcom VideoCore IV.
- 512 MB de memoria RAM.
- Unidad de almacenamiento SD.
- 26 pines de GPIO.
- Puerto Ethernet.
- 2 puertos USB 2.0.
- Conector HDMI para salida de audio y video.
- Conector RCA para salida de video.
- Salida analógica de audio estéreo.
- Consumo aproximado de corriente de 700 mA.

En la Tabla I se presenta la descripción de los 26 GPIO's de la Raspberry Pi®, para la programación de los GPIO's hay dos formas de referirse a ellos en el código, de la forma física (BOARD) y la denominación por Broadcom (BCM).

El Sistema Operativo (SO) que se puede cargar a Raspberry Pi® es Raspbian® el cual es una de las distribuciones de Linux®. Vale la pena mencionar que este SO es gratis, y se ejecuta en el procesador ARM de la Raspberry Pi®, sin embargo la gran mayoría de los SO's funcionan solo en arquitectura de tipo Intel. Es importante destacar que no todas las distribuciones de Linux se ejecutaran en la Raspberry Pi®, porque algunos no soportan en particular el tipo de procesador de esta computadora embebida, por ejemplo no se puede usar Ubuntu® en esta Raspberry Pi® [11]. El SO que se desee utilizar se instala en la tarjeta SD, en este sentido, la instalación

es probablemente diferente a la habitual como se muestra en [12].

TABLA I. DESCRIPCIÓN DE LOS GPIO'S DE LA RASPBERRY PI MODELO B, REVISIÓN 2.

Descripción de GPIO	BOARD	BCM	Descripción de GPIO	BOARD	BCM
3.3 V	1	3V3	5 V	2	5V0
I ² C I SDA	3	GPIO2	5 V	4	5V0
I ² C I SCL	5	GPIO3	Ground	6	Ground
GPIO_GCLK	7	GPIO4	UART TXD	8	GPIO14
Ground	9	Ground	UART RXD	10	GPIO15
GPIO_GEN0	11	GPIO17	GPIO_GEN1	12	GPIO18
GPIO_GEN2	13	GPIO27	Ground	14	Ground
GPIO_GEN3	15	GPIO22	GPIO_GEN4	16	GPIO23
3.3 V	17	3V3	GPIO_GEN5	18	GPO24
SPI MOSI	19	GPIO10	Ground	20	Ground
SPI MISO	21	GPIO9	GPIO_GEN6	22	GPIO25
SPI SCLK	23	GPIO11	SPI_CE0_N	24	GPIO8
Ground	25	Ground	SPI_CE1_N	26	GPIO7

Raspbian® es un SO libre basado en Debian®, optimizado para el hardware de Raspberry Pi®. Raspbian® cuenta con más de 35000 paquetes, optimizados para el mejor rendimiento en Raspberry Pi®, Raspbian® fue completado en Junio de 2012, sin embargo, todavía está en activo desarrollo con un énfasis en la mejora de la estabilidad y el rendimiento de la mayor cantidad de paquetes como sea posible [13].

Es necesario decir que para lograr la comunicación serial entre el PIC y Raspberry Pi® se ha configurado el puerto serial de esta computadora embebida. La configuración que se realiza es la siguiente:

1. Se ejecuta el explorador de archivos con privilegios de Root.
2. En la ruta de acceso /etc se abre el archivo `inittab`.
3. En el archivo se comenta la siguiente línea agregando # al principio:
4. `#TO:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100`
5. Se guarda el archivo y se cierra.
6. En la ruta de acceso /boot se abre el archivo `cmdline.txt`.
7. En el archivo de la línea:
8. `dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p6 rootfstype=ext4 elevator=deadline rootwait`
9. Se corta una parte quedando como sigue:
10. `dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p6 rootfstype=ext4 elevator=deadline rootwait`
11. Se guarda el archivo y se cierra.
12. Se reinicia la Raspberry Pi®.

El código que se ha programado en la Raspberry Pi® se realiza en el editor de textos Leafpad® (se puede crear el código en cualquier editor de textos instalado en la Raspberry Pi). En este sentido por parte de la Raspberry Pi® se requiere ocupar el GPIO UART RXD, para ello las librerías que son necesarias son las desarrolladas por Gordon Henderson (se pueden ver en [14]), estas librerías son `wiringPi.h` y `wiringSerial.h`.

Para el desarrollo de la comunicación serial se usa la siguiente instrucción de Raspbian®:

```
fd = serialOpen (DEVICE, BAUD);
```

Donde *DEVICE* es el dispositivo para el puerto serial (`/dev/ttyAMA0`) y *BAUD* es la velocidad de transmisión. Al igual que el PIC la velocidad de transmisión que se define es de 9600 baudios y lograr la sincronía.

```
#include <wiringPi.h>
#include <wiringSerial.h>
int main ()
{
    fd = serialOpen ("/dev/ttyAMA0",
    9600);
    for ( ; ; )
    {
        read (fd, cad, sizeof (cad));
    }
}
```

Fig. 4. Segmento de código en la Raspberry Pi®.

En la Fig. 4 se muestra el segmento de código que se utilizó para la programación de la Raspberry Pi, se define el dispositivo y la velocidad de transmisión, además también en un ciclo infinito se leerán con la función `read()` los valores que provienen del microcontrolador, estos datos son recibidos como cadena de datos o string y se convertirán a datos numéricos con la función `sscanf()` y como último paso estos datos serán presentados en pantalla utilizando la función `printf()`.

III. RESULTADOS

Una vez lograda la programación en el PIC y haber realizado el código en la Raspberry Pi®, se procede a hacer la conexión completa entre el potenciómetro, el PIC y la Raspberry Pi® como se aprecia en la Fig. 5.

A continuación se describen las conexiones que se aprecian en la Fig. 5:

1. El potenciómetro está conectado a una tensión de 5v, esta tensión proviene de la conexión que provee uno de los GPIO's la Raspberry Pi®.
2. El potenciómetro se conecta al PIC en el puerto SANO que se definió en el código de programación para que los datos provenientes del potenciómetro sean convertidos a datos digitales.
3. El PIC transmite la información de manera serial por el pin TX, a la Raspberry Pi® que por su parte recibirá la información por el GPIO RXD.

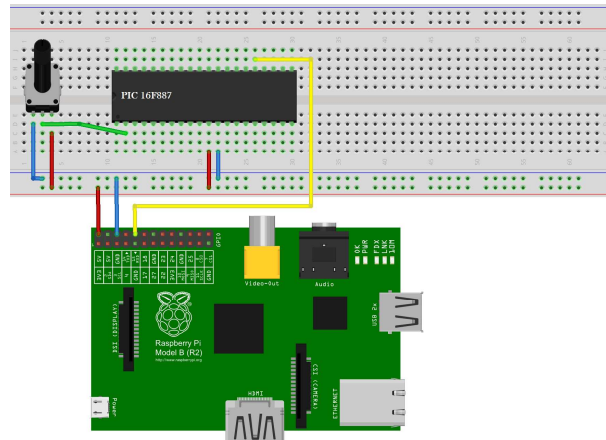


Fig. 5. Conexión para la comunicación serial entre PIC-Raspberry Pi®.

Cabe aclarar que Raspberry Pi® cuenta con un compilador GCC, un compilador en C, por ello para compilar el código del programa que se realizó en Raspberry Pi® se necesita usar un programa o terminal con el cual cuenta la misma Raspberry Pi® llamado LXTerminal®. La compilación se realizó de la siguiente manera, se guarda el código realizado en el editor de textos, por ejemplo `archivo.c`, posteriormente en la terminal LXTerminal® la compilación se hará de la siguiente manera:

```
cc -o archivo archivo.c -lwiringPi
```

`-lwiringPi` fue un requerimiento por parte de las librerías `wiringPi.h` y `wiringSerial.h` para compilar. Después de haber compilado se requiere ejecutar el archivo generado, lo cual se puede hacer en la misma terminal para observar los datos censados por el PIC y valorar el proyecto realizado, para ejecutar el programa se utiliza la siguiente función de Raspbian®:

```
sudo ./archivo
```

En la Fig. 6 se muestra la terminal de Raspberry Pi® que presenta los datos censados por el PIC, estos valores son los voltajes obtenidos por Raspberry Pi® mediante el enlace serial.

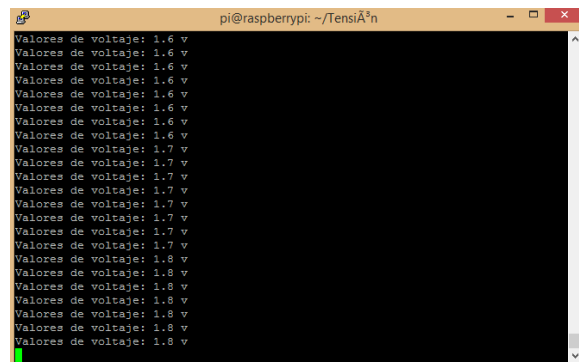


Fig. 6. Terminal en Raspberry Pi® mostrando distintos voltajes.

IV. CONCLUSIONES

Como una conclusión importante del presente trabajo se establece que el arreglo de comunicación descrito no solo sirve para monitorear el voltaje proveniente de un potenciómetro, este arreglo también es capaz de monitorear señales analógicas de otra índole, como por ejemplo temperatura u otras señales ambientales. Este trabajo se puede aplicar en proyectos más complejos en los que se requiera censar o monitorear con Raspberry Pi®. Puede resolver la problemática de no contar con un convertidor A/D en una Raspberry Pi®. Otra problemática que puede ser resuelta y que resultaría de mucho interés para muchos, es que si no se cuenta inmediatamente con un dispositivo con la característica de leer señales eléctricas y además no se cuenta con los recursos económicos de conseguir uno de esos dispositivos, este artículo plantea un método que resulta económicamente alcanzable para poder construir un pequeño osciloscopio capaz de monitorear señales eléctricas en una computadora (Raspberry Pi®) con la capacidad de poder exportar los datos obtenidos a otras computadoras.

REFERENCIAS

- [1] "Embedded system definition by The Linux Information Project (LINFO)." [Online]. Disponible: http://www.linfo.org/embedded_system.html. [Consultado: Mar-2015].
- [2] "What is a Raspberry Pi? | Raspberry Pi." [Online]. Disponible: <http://www.raspberrypi.org/help/what-is-a-raspberry-pi/>. [Consultado: Mar-2015].
- [3] "Puerto serie - Wikipedia, la enciclopedia libre." [Online]. Disponible: http://es.wikipedia.org/wiki/Puerto_serie. [Consultado: Mar-2015].
- [4] "What is Serial Port? Webopedia." [Online]. Disponible: http://www.webopedia.com/TERM/S/serial_port.html. [Consultado: Mar-2015].
- [5] Donald Norris, *12 Proyectos Raspberry Pi*. España: Estribor, 2013, pp. 13-14.
- [6] W. Gay, *Mastering the Raspberry Pi*. Apress, 2014, p. 167.
- [7] "Comunicación Serial: Conceptos Generales - National Instruments." [Online]. Disponible: <http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>. [Consultado: Mar-2015].
- [8] "Microcontrolador PIC16F887 - Microcontroladores PIC - Programación en C con ejemplos." [Online]. Disponible: <http://www.mikroe.com/chapters/view/81/capitulo-3-microcontrolador-pic16f887/#c3v1>. [Consultado: Mar-2015].
- [9] "PIC C Compiler - iElectroX." [Online]. Disponible: <https://sites.google.com/site/ielectrox/home/programas-1/pic-c-compiler>. [Consultado: Mar-2015].
- [10] "Raspberry Pi | wiringPi | Serial Library | Gordons Projects." [Online]. Disponible: <https://projects.drogon.net/raspberry-pi/wiringpi/serial-library/>. [Consultado: Mar-2015].
- [11] M. Schmidt and J. Carter, *Raspberry Pi: a quick start guide*. Dallas TX: The Pragmatic Bookshelf, 2012, p. 11.
- [12] "Raspberry Pi Documentation." [Online]. Disponible: <http://www.raspberrypi.org/documentation/installation/installing-images/README.md>. [Consultado: Mar-2015].
- [13] "FrontPage - Raspbian." [Online]. Disponible: <http://raspbian.org/>. [Consultado: Mar-2015].
- [14] "Raspberry Pi | Wiring | Gordons Projects." [Online]. Disponible: <https://projects.drogon.net/raspberry-pi/wiringpi/>. [Consultado: Mar-2015].

BIOGRAFÍAS

Juan Carlos Morelos Romero. Ingeniero Mecánico egresado de la Escuela Superior de Ingeniería Mecánica y Eléctrica del IPN. Actualmente es estudiante de la Maestría en Ciencias de Ingeniería en Sistemas Energéticos de la misma escuela.

Gustavo Delgado Reyes. Ingeniero en Comunicaciones y Electrónica y Maestro en Ciencias de Ingeniería en Microelectrónica egresado de la Escuela Superior de Ingeniería Mecánica y Eléctrica del IPN. Actualmente es estudiante del Doctorado en Comunicaciones y Electrónica en el IPN. Sus áreas de interés son: Sistemas en tiempo real, Sistemas embebidos y Teoría de control.

Asdrubal López Chau. Se graduó en 1997 como Ingeniero en Comunicaciones y Electrónica de la Escuela Superior de Ingeniería Mecánica y Eléctrica del IPN. Terminó su Maestría en Ciencias de Ingeniería en Computación en 2000 en el Centro de Investigación en Computación (CIC) del IPN. En el 2013, el Dr. Asdrubal se graduó como Doctor en Ciencias de la computación en el Centro de Investigación y Estudios Avanzados (CINVESTAV) del IPN. En la actualidad es profesor de investigación de la Universidad Autónoma del Estado de México, Centro Universitario Zumpango, en el programa de Ingeniería Informática. Sus áreas de investigación actuales son: Maquina de Aprendizaje, Big Data y Sistemas en Tiempo Real.

Efecto del Overclocking sobre los Tiempos de Ejecución en una Computadora Embebida con Diferentes Niveles de Carga

Erik Antonio Aguilar, Jose Luis Cano Rosas, Pedro Guevara López

Instituto Politécnico Nacional - ESIME Culhuacán

Av. Santa Ana 1000, San Francisco Culhuacan, Coyoacán, 04430 Ciudad de México, Distrito Federal, Tel. 57296000 ext. 73250.
flay_he@hotmail.com, lucksyrgmail.com, pguevara@ipn.mx

Resumen— En la actualidad, gran parte de la vida del ser humano depende de los avances tecnológicos, especialmente de las computadoras y últimamente de los dispositivos móviles y embebidos; no obstante la cantidad de aplicaciones que constantemente son actualizadas requieren más poder de procesamiento, ocasionando que el usuario tenga que adquirir un dispositivo con mayor poder de cómputo esto implica altos gastos iniciales aunado a esto una solución alternativa para incrementar el poder de cálculo es la técnica overclocking. En este sentido este trabajo está enfocado a presentar el efecto que tiene el overclocking en Raspberry Pi, una computadora embebida, cuando esta sometida a diferentes niveles de carga, para lo cual se a tomado como carga algoritmos computacionales de inversion de matrices 2×2 , 4×4 , 8×8 , 32×32 y 128×128 ; obteniendo una ganancia significativa en los tiempos de ejecucion en cada uno de los resultados obtenidos para la inversion de matrices.

Palabras clave: Overclocking, momento de probabilidad, tiempo de ejecución, computadora embebida, algoritmo.

Abstract— Currently, great part of human life depends on technological advances especially in computers and lately, in mobile and embedded devices. however, how many applications that are constantly updated, make necessary more processing power, causing the user purchase a device with greater computing power, that involves high initial costs however an alternative solution to increase computing power is the overclocking technique, in this sense, this paper is focused on presenting the effect having of the overclocking about Raspberry Pi an embedded device when subjected to different load levels, for which it, is taken as load, computational algorithms of inversion of matrices 2×2 , 4×4 , 8×8 , 32×32 and 128×128 ; with results an significant gain in execution times for each of the results obtained the inversion of matrices.

Keywords—Overclocking, moment, computing time, embedded computer, algorithm.

I. INTRODUCCIÓN

En la actualidad, la tecnología de la computación apunta hacia los sistemas embebidos; por ejemplo tablets, teléfonos celulares, consolas de videojuegos portátiles, computadoras embebidas, etc. Aunado a esto, las nuevas aplicaciones de

software exigen cada vez más poder de procesamiento y requiriendo más recursos de hardware. En este sentido, el objeto de estudio de este trabajo son las computadoras embebidas, específicamente la computadora Raspberry Pi modelo B. La solución más simple para incrementar el poder de procesamiento y obtener tiempos de respuesta más pequeños es cambiando la computadora embebida, su memoria o el propio procesador; sin embargo, esto genera altos costos iniciales y acumulación de equipos obsoletos.

Una alternativa viable a la demanda de poder de cómputo es el overclocking; este consiste en incrementar las frecuencias del reloj del procesador y de la memoria, e incrementar sus voltajes de operación. Sin embargo, se pueden presentar algunos riesgos: daño permanente del hardware, disminución de su tiempo de vida, “inestabilidad” del sistema o invalidación de la garantía del fabricante. Aunado a esto, el overclocking no es una práctica recomendada por los fabricantes; sin embargo en varias ocasiones los mismos proporcionan herramientas para ello bajo responsabilidad del usuario.

Para probar el efecto del overclocking en los tiempos de ejecución, se utilizó una computadora embebida Raspberry Pi debido a que es altamente configurable y cuenta con un sistema operativo de tiempo compartido basado en Linux (Raspbian), admite programación de alto nivel que permite realizar operaciones de alta complejidad temporal. A diferencia, otros sistemas como microcontroladores, DSPs, GPUs, FPGAs requieren programación especializada y cuentan con poca flexibilidad.

Como parte del banco de pruebas, para generar carga computacional se desarrolló un algoritmo para inversión de matrices basado en el método de Gauss- Jordan; esto es porque la mayoría de los cálculos computacionales que realizan los sistemas embebidos involucran operaciones matriciales que generalmente se usan para describir sistemas de ecuaciones y tienen mucha aplicación en áreas como control automático, procesamiento de imágenes, filtrado digital y en aplicaciones de computación científica. Por ello es

muy importante su estudio y tiempo de solución para determinar la capacidad de los sistemas embebidos al realizar procesamiento de datos con alta complejidad computacional temporal.

Con base en todo esto, el objetivo principal de este trabajo fue desarrollar un estudio experimental sobre el efecto de los niveles preestablecidos de overclocking para el primer y segundo momento de probabilidad de los tiempos de ejecución generados en la inversión de matrices de diferentes dimensiones por un método iterativo basado en Gauss-Jordan en la computadora embebida Raspberry Pi. Todo ello con la finalidad de determinar cuándo usar overclocking y bajo qué circunstancias.

A. Tiempos de Ejecución, Raspberry Pi y Overclocking

De acuerdo a [9], Los sistemas operativos del tipo POSIX (Portable UNIX), incluido Raspbian, tiene soporte para procesos concurrentes; un proceso es un programa en ejecución que consume recursos de memoria y procesamiento de una computadora y su tiempo de ejecución es $C_{i,k} \in \mathbf{R}^+$; el tiempo de ejecución es el tiempo en que la instancia o sección $J_{i,k}$ de un proceso J_i concluye sus operaciones, sin considerar los desajustes de ésta en el procesador. La variación de los Tiempos de ejecución $C_{i,k}$ para n instancias de un proceso donde $l_{i,k}$ es el tiempo de arribo, $s_{i,k}$ es tiempo de inicio. Y $f_{i,k}$ es tiempo de finalizado i número de tarea y k número de iteraciones, como se aprecia en la Fig. 1.

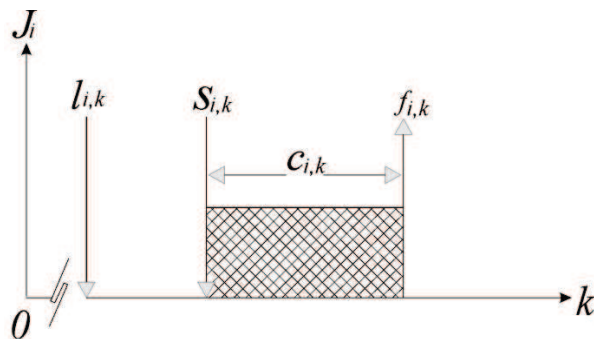


Fig. 1. Variación de los Tiempos de ejecución $C_{i,k}$ para n instancias.

El tiempo de respuesta del sistema computacional depende en gran parte del comportamiento de los tiempos de ejecución $c_{i,k}$ del proceso J_i ; sin embargo, este comportamiento fluctúa debido a varios factores como los expresados en *F. Stappert and P. Altenbernd* en [5] donde consideraron que el comportamiento fluctuante de los tiempos de ejecución se deben a los factores computacionales como son el *cacheing*, *pipeline*, la búsqueda de la ruta de ejecución considerando a la exclusión mutua, número de lazos, predicciones y otras interacciones.

La Raspberry Pi, ya mostrada en la figura 2, es una computadora del tamaño de una tarjeta de crédito de bajo costo que se conecta a un monitor o TV, utiliza un teclado y un monitor estándar, es un dispositivo que permite a las

personas de todas las edades explorar la computación para aprender a programar en lenguajes como C, Scratch y Python, es capaz de hacer todo lo que se espera de una computadora de escritorio desde navegar por internet, reproducir video de alta definición, hacer hojas de cálculo, con procesador de textos y jugar juegos desarrollada por Raspberry Pi Foundation [4].

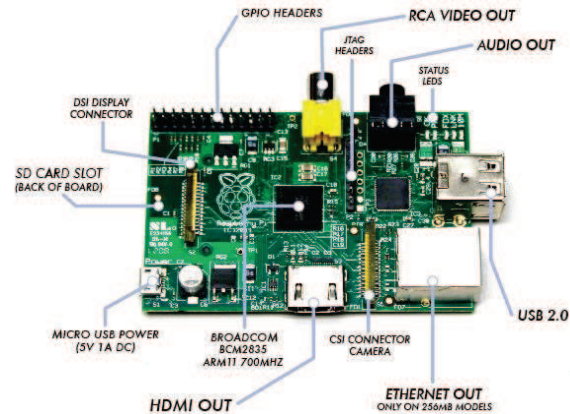


Fig. 2. Imagen de la computadora embebida Raspberry Pi que esquematiza el hardware mediante nombres el conjunto de componentes y puertos para periféricos e interfaces [7].

Basado en [6] y [1], desde un punto de vista personal una Raspberry Pi ya mostrada en la Fig. 2. Es una computadora embebida con arquitectura *system on a chip*, que integra la mayoría de los componentes en un solo circuito integrado. Esta tarjeta de desarrollo está compuesta con un chip BCM2835 que contiene un procesador ARM1176JZF-S de 700MHz (el cual se puede someter a overclock), procesador gráfico Video-Core IV y 512 MB de memoria RAM en su versión B, con sistema operativo Raspbian como una distribución de GNU/Linux 7 conformado por el sistema de archivos de Debian.

El overclocking de acuerdo a [3], es el término en inglés que hace referencia al aumento de la frecuencia de reloj con que opera el hardware de un sistema de cómputo, por medio de configuraciones específicas establecidas por el usuario, fuera del rango que el fabricante predetermina en sus dispositivos, para aumentar la velocidad de procesamiento de datos en el sistema; por ello en la mayoría de los casos los fabricantes invalidan la garantía de los equipos al momento de realizar esta práctica, si los mismos fabricantes proporcionan herramientas y especificaciones de hardware y software que permiten configurar el equipo para hacer overclocking. En este sentido, la computadora Raspberry Pi no cuenta con un sistema básico de entradas y salidas (BIOS) convencional y a diferencia de las computadoras comunes que establecen y guardan la configuración de los parámetros del sistema mediante el BIOS, ahora se almacena en un archivo de texto llamado "config.txt", localizado en la ruta `/boot/config.txt` de Raspbian; en donde se especifica el modo de funcionamiento y los parámetros a utilizar para el arranque del

sistema, este archivo es leído por el sistema después de que el ARM Core es inicializado.

La selección de los niveles de overclocking se realiza a través del script `raspi-config` en la configuración de Raspbian, esto se muestra en la Fig. 3.

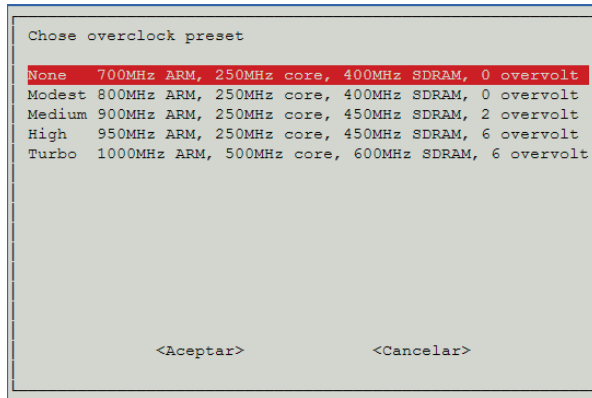


Fig. 3. Captura de pantalla del script `raspi-config` de Raspbian.

II. DESARROLLO

A. Desarrollo del banco de pruebas

Una parte fundamental en el desarrollo de este trabajo para experimentar el efecto del overclocking, es establecer, integrar y conformar un banco de pruebas que permita tener contacto con esa parte intangible del sistema que concibe dicho efecto y para ello se ha desarrollado software de ejecución como shell-script compilado sobre lenguaje de programación ANSI C e implementado en el sistema operativo Raspbian desde una computadora embebida Raspberry PI, denominado `matrices.h` descrito en esta sección como una biblioteca de funciones para la inversión de matrices basada en los fundamentos teórico prácticos que caracteriza a la matriz y su inversión en los diferentes algoritmos principalmente en el método de Gauss-Jordan, elaborado con el propósito de generar carga computacional para generar tiempos de ejecución. Las funciones desarrolladas fueron: `iniciar_matriz()`, `llenar_matriz()`, `sumar_matriz()`, `producto_matriz()` e `inversa()`.

Se eligió la inversión de matrices como método para generar carga computacional debido a su nivel de complejidad; esto permitirá medir tiempos de ejecución de magnitud considerable de acuerdo a la dimensión de la matriz a invertir. Por otra parte, las operaciones matriciales tienen muchas aplicaciones en diferentes áreas.

B. Medición de los tiempos de ejecución

En la literatura se pudieron encontrar algunas técnicas para medir los tiempos de ejecución que se basan en la obtención de

un tiempo de referencia inicial y un final cuya diferencia es el valor del tiempo transcurrido al ejecutarse el segmento de código en estudio. La función `clock()` es una herramienta presente en la biblioteca `time.h` de los sistemas POSIX, pero no es la única función que se puede usar para este propósito. Dentro del sistema operativo Raspbian, se pueden encontrar otras como la función `time()`, `ClockTime()` y `clock_gettime()`, siendo esta última la que se usará en esta investigación. La diferencia de esta última con la función `clock()` es que la anterior trabaja con múltiplos enteros de “ticks” de reloj y registra valores en el orden de los microsegundos, en cambio con la función `clock_gettime()`, la resolución puede ser ajustada permitiendo obtener valores más precisos de los tiempos de ejecución medidos en el orden de los nanosegundos [2]. La Fig. 4 se contempla un segmento de código utilizando la función `clock_gettime()`.

```
for (k=0; k<ITERACIONES; k++){
    matriz= iniciar_matriz(M, M);

    clock_gettime(CLOCK_REALTIME, &inicio);

    matriz= perturbacion(matriz, M, M,
    mag_ruido_ext);

    matriz_inversa=inversa(matriz, M);

    imprimir_matriz(matriz, M, M);
    imprimir_matriz(matriz_inversa, M, M);

    clock_gettime(CLOCK_REALTIME, &fin);

    milisegundos_i= (inicio.tv_sec*1000.0) +
    ((double)inicio.tv_nsec/1000000.0);
    milisegundos_f= (fin.tv_sec*1000.0) +
    ((double)fin.tv_nsec/1000000.0);
    ejecucion=
    milisegundos_f-milisegundos_i;

    fprintf(f_c, "%.3f\n", ejecucion);
    printf("c(%d)= %f milisegundos\n",
    k, ejecucion);

    system("sudo /opt/vc/bin/
    vcgencmd measure_temp >>
    temperatura.txt");
}
```

Fig. 4. Medición de tiempos de ejecución usando la función `clock_gettime()`.

Se propuso medir el tiempo de ejecución para 500 inversiones de matrices de 2×2 , 4×4 , 8×8 , 16×16 , 32×32 , y 128×128 ; sin embargo, se considera el uso del primer y segundo momento de probabilidad debido a que estos dan información de su comportamiento global.

III. RESULTADOS

A. Resultados Experimentales

Formando un banco de pruebas de hardware y software con los recursos mencionados, se realizaron 35 pruebas obteniendo como resultados estrictamente experimentales, diferentes conjuntos de 500 muestras de datos correspondientes a los ya mencionados tiempos de ejecución en milisegundos de acuerdo el tamaño de la matriz y el nivel de overclocking testado, para generar esquemas gráficos utilizando MATLAB como software auxiliar donde se visualiza y examina el comportamiento de estos por medio de un análisis estadístico como lo es el primer y segundo momento de probabilidad, dando lugar a las gráficas correspondientes. Todos los experimentos fueron realizados en un ambiente conocido y descrito en la Tabla I, se observa las precisas condiciones ambientales al momento de realizar las pruebas correspondientes. Para reducir el número de páginas de este trabajo, solo se reportan algunas graficas del conjunto total de experimentos.

TABLA I. CONDICIONES AMBIENTALES DE PRUEBA PARA MEDICIÓN DE TIEMPOS DE EJECUCIÓN EN LA COMPUTADORA RASPBERRY PI.

<i>Temperatura:</i> 21 - 23.5 °C	<i>Presión atmosférica:</i> 781 hpa
<i>Altitud:</i> 2182 msnm	<i>Nivel de iluminación:</i> 100 Lux
<i>Latitud:</i> 19.252058	<i>Longitud:</i> -99.102109

B. Tiempos de ejecución en Matrices 2x2

En la Fig. 5 se muestra la gráfica representativa de los tiempos de Ejecución en diferentes niveles de Overclocking en matrices de 2x2. Se observa que con el nivel de overclocking Turbo las mediciones de los tiempos de ejecución y su valor medio son los menores tiempos de ejecución.

La Fig. 6 representa una gráfica representativa de la varianza en los tiempos de ejecución con diferentes niveles de overclocking en matrices de 2x2. Cabe resaltar que los sobresaltos se deben a que otro proceso usó el procesador incrementando el tiempo de ejecución del experimento.

En la Fig. 7 se muestra una gráfica de los Tiempos de Ejecución Acumulados en distintos niveles de Overclocking para matrices de 2x2. Contemplando una notable diferencia, al usar el modo None con respecto al nivel Turbo.

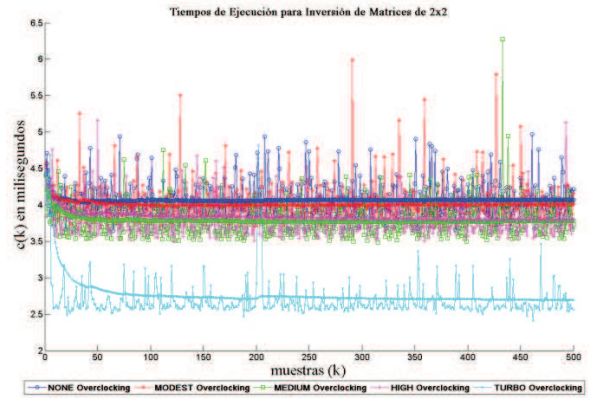


Fig. 5. Tiempos de Ejecucion para Inversión de Matrices de 2x2.

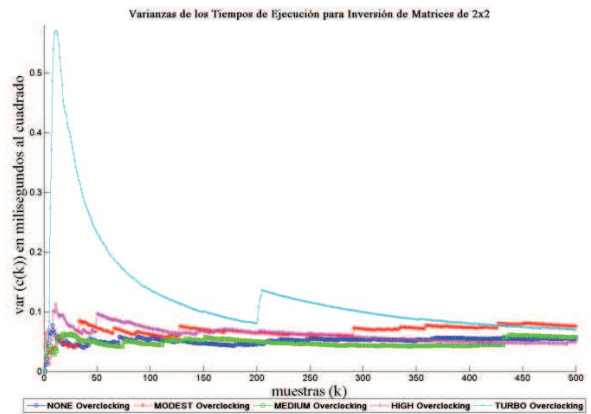


Fig. 6. Varianza de los Tiempos de Ejecución en Matrices de 2x2.

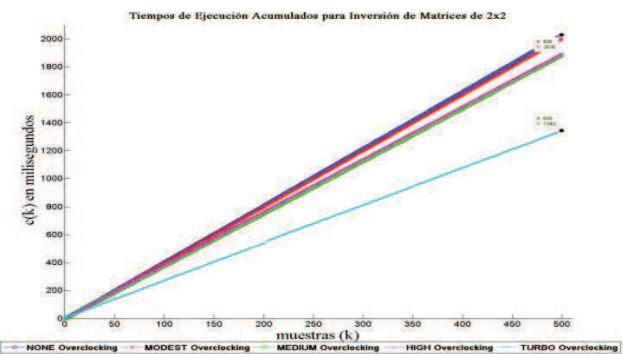


Fig. 7. tiempos de ejecución acumulados para inversión de matrices de 2x2

C. *Tiempos de ejecución en Matrices 128x128*

En la Fig. 8 se muestra la gráfica representativa de los tiempos de ejecución en diferentes niveles de Overclocking en matrices de 128x128.

En la Fig. 9 Se muestra la gráfica particular de los tiempos de ejecución acumulados en distintos niveles de Overclocking para matrices de 128x128.

En la Fig. 10 se muestra una gráfica de los Tiempos de Ejecución Acumulados en distintos niveles de Overclocking para matrices de 128x128. Contemplando una notable diferencia, al usar el modo None con respecto al modo Turbo.

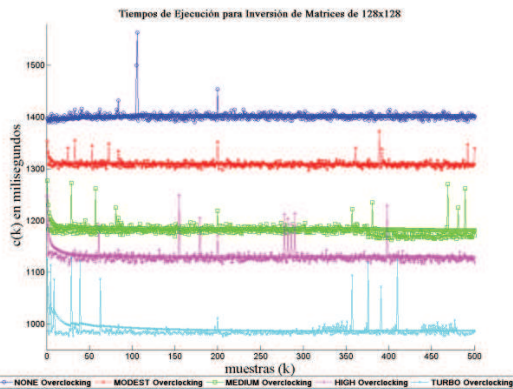


Fig. 8. Tiempos de Ejecución para inversión de Matrices 128x128.

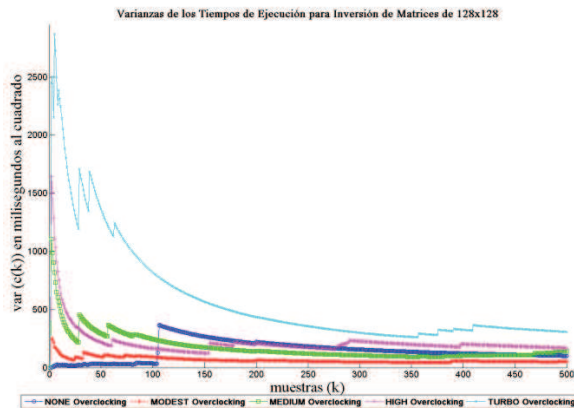


Fig. 9. Varianzas de los Tiempos de Ejecución para inversión de matrices de 128x128.

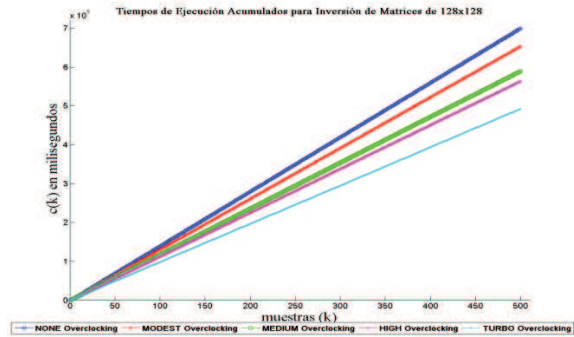


Fig. 10. tiempos de ejecución acumulados para inversión de matrices de 128x128.

D. *Momento de Probabilidad para Inversión de Matrices*

En la Fig. 11 se presentan los gráficos representativos del valor final del primer momento de probabilidad de los tiempos de ejecución en diferentes niveles de overclocking generados por la inversión de matrices desde 2x2 hasta 128x128, los valores intermedios fueron interpolados linealmente por Matlab.

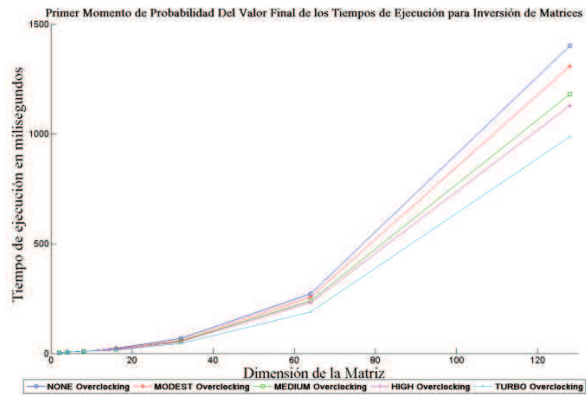


Fig. 11. Primer Momento de Probabilidad del valor final de los tiempos de ejecución para inversión de Matrices.

En la Tabla II se presenta la descripción comparativa de los tiempos de ejecución en milisegundos de la inversión de matrices desde 2x2 hasta 128x128, en los distintos niveles de overclocking.

TABLA II. TABLA DE TIEMPOS DE EJECUCIÓN EN MILLISEGUNDOS.

Dimensión	2x2	4x4	8x8	32x32	128x128
None	2030	3193	4858	33930	699500
Modest	1998	3130	4739	29500	653100
Medium	1893	3020	4495	28420	589300
High	1885	3008	4483	27380	563300
Turbo	1343	2344	4224	22510	492500

En la Tabla III se presenta la descripción comparativa del porcentaje de tiempo alcanzado para la inversión de matrices en base al valor final del promedio de los tiempos de ejecución acumulados en diferentes dimensiones de matriz (desde 2×2 hasta 128×128) y niveles de overlocking tomando como punto de referencia al 100% el nivel None.

TABLA III. PORCENTAJES DE GANANCIA EN TIEMPO ALCANZADO

Dimensión	2x2	4x4	8x8	32x32	128x128
None	0.000%	0.000%	0.000%	0.000%	0.000%
Modest	1.576%	1.973%	2.450%	13.056%	6.633%
Medium	6.749%	5.418%	7.472%	16.239%	15.754%
High	7.143%	5.794%	7.719%	19.304%	19.471%
Turbo	33.842%	26.589%	13.051%	33.658%	29.593%

E. Interpretación de resultados

De las gráficas obtenidas puede observarse lo siguiente:

- Las mediciones comienzan en un valor muy alto debido a que al inicio de la ejecución de cada uno de los procesos se lleva a cabo la asignación de los recursos en memoria, ejecutándose las funciones `iniciar_matriz()` y `llenar_matriz()`, que de acuerdo a los experimentos realizados consumen un mayor tiempo de ejecución comparado con la función `inversa()`.
- Las mayores variaciones en los tiempos de ejecución en las mediciones en todos los experimentos se deben a que no siempre se requiere el mismo número de iteraciones para invertir una matriz, por ejemplo si en algún caso se deseara invertir la matriz identidad, se requiere el menor número de iteraciones, reduciéndose el número de operaciones aritméticas.
- De acuerdo a las gráficas del primer momento de probabilidad de los tiempos de ejecución en todos los experimentos, se observa que en general la dinámica de estos tiempos tiene un comportamiento estacionario convergiendo a un valor final.
- De las gráficas del primer momento de probabilidad para inversión de matrices de $n \times n$ se observa que los tiempos de ejecución se incrementan de forma exponencial de acuerdo a n . Esto representa una complejidad computacional temporal no polinomial (NP).

IV. CONCLUSIONES

A medida que se incrementa el nivel de overlocking, el primer momento de probabilidad disminuye, mientras que el segundo momento de probabilidad se incrementa; esto quiere decir que a mayor nivel de overlocking se obtienen mejores

resultados, pero el sistema tiende a volverse “inestable” desde el punto de vista computacional.

Con base en los resultados obtenidos, se afirma que el efecto del overlocking consigue una ganancia sustancial en tiempo de ejecución en todos y cada uno de los resultados obtenidos para la inversión de matrices, si bien que para el caso del nivel Modest se disminuyó los tiempos de ejecución en un 1.549% respecto al nivel None, mientras que para el nivel Turbo se disminuyó hasta el 33.751% respecto al nivel None esto representa hasta 3.45 minutos de tiempo ahorrado. Con matrices de mayores dimensiones a las probadas en este trabajo, el efecto del overlocking disminuye al aumentar su complejidad computacional temporal de la inversión de la matriz.

REFERENCES

- [1] Furber, S. B. (2000). “*ARM System-on-chip Architecture*”. Addison-Wesley.
- [2] QNX’s Helpviewer (2012) “*Tick, tock: Understanding the Neutrino Microkernel’s concept of time*” Sistema Operativo en tiempo real QNX 6.5 Neutrino
- [3] Wainner, S., & Richmond, R. (2003). “*The Book of Overclocking: Tweak Your PC to Unleash Its Power*”. No Starch Press.
- [4] “What is a Raspberry Pi? | Raspberry Pi.” [Online] (n.d.). Retrieved from <http://www.raspberrypi.org/help/what-is-a-raspberry-pi/>. [Consultado Mar-2015]
- [5] Stappert, F., & Altenbernd, P. (2000). “*Complete Worst-Case Execution Time Analysis of Straight-line Hard Real-Time Programs*”. V.46, N.4. Pp.339-355.
- [6] “System-on-chip technology comes of age | EE Times. (n.d.)”. [Online] Retrieved [March 13, 2015], from http://www.eetimes.com/document.asp?doc_id=1279975 The Great Debate: SOC vs. SIP | EE Times. (n.d.). Retrieved [March 13, 2015], from http://www.eetimes.com/document.asp?doc_id=1153043
- [7] (<http://www4.pcmag.com/media/images/351321-raspberry-pi.jpg?thumb=y>), [online] (<http://www4.pcmag.com/media/images/351321-raspberry-pi.jpg?thumb=y>). [Consultado: Mar-2015]
- [8] J.Luis Cano Rosas, “*Efecto del Overclocking sobre los tiempos de ejecución generados por inversión de matrices en una computadora embebida*”. Tesis M.S”. Dept. Sección de Estudios de Posgrado e Investigación, ESIME Culhucan., México, 2015
- [9] Jorge Salvador Valdez Martínez, Pedro Guevara López, Luis de Jesús Beltrán Sosa, “*Métodos de Medición de Tiempos de Ejecución en el Sistema Operativo de Tiempo Real QNX 6.5 Neutrino*”, XIII Congreso Nacional de Ingeniería Electromecánica y de Sistemas, ESIME Zacatenco, México D.F., Noviembre de (2012)

Técnicas de Generación y Solución de Laberintos

E. A. Requena-Vega, V. M. Landassuri-Moreno, H. R. Orozco-Aguirre

Centro Universitario UAEM Valle de México (CUAEM-VM)

Universidad Autónoma del Estado de México (UAEMex)

Bld. Universitario s/n Predio San Javier, Atizapán de Zaragoza, Estado de México, México. CP 54500.

ultrik@hotmail.com, vmlandassurim@uaemex.mx, hrorozcoa@uaemex.mx

Resumen— Un laberinto es un recinto que tiene una entrada y una salida ubicados en distintos lugares, los que están constituidos por pasajes parecidos que se entrecruzan, con el fin de desorientar y hacer difícil el llegar a la salida. En la actualidad se han diseñado y elaborado distintas estrategias para resolverlos mediante grafos, sistemas multi-agente pero estos métodos pueden consumir demasiado tiempo en dar una solución. Éste trabajo muestra un panorama general de las diversas técnicas de generación y solución de laberintos.

Palabras clave—Algoritmos Evolutivos, Algoritmos Genéticos

I. INTRODUCCIÓN

Existe evidencia por parte de la cultura Griega, donde ellos usaban los laberintos para poner en competencia a los jóvenes, ya que en el centro de estos los esperaba una hermosa mujer que los aceptaría; a su vez, en Egipto tenían o conocían los laberintos dados los descubrimientos arqueológicos de los últimos siglos. Posteriormente, diversa culturas como la Escandinava los usaban antes de salir a pescar, ya que creían que los malos vientos quedaban atrapados entre sus enredados y complejos pasajes, consecuentemente, en Finlandia los pastores los utilizaban para protegerse de los malos espíritus, que los seguían hasta el centro del laberinto y quedaban atrapados.

En la actualidad, los laberintos se han usado para generar juegos donde el intelecto y creatividad se ponen en duda, al generarlos o al resolverlos. Por lo tanto, se han generado laberintos de múltiples formas y áreas del conocimiento, por ejemplo, desde jardines decorativos hasta en los videojuegos.

Desde otro ángulo, usualmente nos enfrentamos a los laberintos para resolverlos, sin embargo existe su contraparte que puede llegar a ser igual de difícil que resolverlos, y aquí nos referimos, a la generación de laberintos. Así, durante la búsqueda de la solución [1], esta puede ser dada por un jugador o un algoritmo.

De esta forma, se tiene a un laberinto “perfecto” el cual puede tener o no tener bucles (ciclos), además se asegura que exista un camino libre desde el inicio al final del laberinto. Estos tipos de laberintos (perfectos) también tienen corredores o pasajes no muy extensos y en su mayoría existen varios callejones sin salidas [2]. Por consiguiente, se espera que no sean de un crecimiento de tipo árbol, es decir, que el crecimiento de la solución no sea exponencial y crezca a un nivel de mayor tamaño al esperado.

En la actualidad, los laberintos han sido enfocados hacia el área de video juegos por computadora. No obstante, muchas de las escenas de los videojuegos pueden ser vistos como laberintos, al analizar en su totalidad video juego, a gran o pequeña escala, basados en paisajes naturales o paisajes artificiales. De esta forma, el objetivo de éste trabajo es mostrará un panorama general de los diferentes tipos de algoritmos para la generación y solución de laberintos de diversos tipos.

Este trabajo está organizado de la siguiente manera: en la sección II se mostrará los algoritmos actuales en la generación de laberintos. En su contraparte, la sección III presenta los algoritmos para la solución de laberintos, mientras que la sección IV trata acerca de Computo Evolutivo. La sección V presenta una técnica avanzada para la solución de laberintos mediante el uso de un AG, y la sección VI la conclusión alcanzada para este trabajo.

II. GENERACIÓN DE LABERINTOS

La tarea de generación de un laberinto [8] puede ser llevada a cabo de manera manual o de forma automática por medio de un algoritmo computacional. En este artículo, solo interesan aquellos métodos en los cuales es empleado un algoritmo. Al respecto, existen distintas variantes de ellos, algunas los generan en menor tiempo, otras más complejos o elaborados. Sin embargo, para poder tener una idea más clara de cómo trabajan estos algoritmos, a continuación se describen los más conocidos:

A. Algoritmo de Prim

El algoritmo de Prim es necesario tener un almacenamiento proporcional al tamaño del laberinto, lo cual puede ser una desventaja para algunos de gran tamaño. Éste algoritmo consiste en generar el laberinto mediante el uso de Spanning Tree (ST), donde existen tres tipos de celdas:

- Celdas “dentro”. Son aquellas celdas que pertenecen al laberinto.
- Celdas “frontera”. Como su nombre lo dice son celdas fronterizas o vecinas a las celdas “dentro”, pero que aun no pertenecen al laberinto.

- Celdas “fuera”. Son celdas que no pertenecen al laberinto aun y tampoco son vecinas a las celdas “dentro”.

El algoritmo consiste en crear y seleccionar una celda “dentro” y se ajustan todas sus celdas vecinas las cuales son, celdas “frontera”. A continuación se selecciona aleatoriamente una de esas celdas “frontera”, y se crea dentro de ella una celda “dentro”, repitiendo el procedimiento. El algoritmo termina cuando ya no existan más celdas “fuera” o “frontera”, quedando solamente celdas “dentro”. El laberinto resultante tendrá muchos callejones aunque muy pequeños y sin salida [2]. A continuación se muestra el pseudocódigo del Algoritmo de Prim:

Prim (Grafo G)

```

/* Inicializamos todos los nodos del grafo.
La distancia la ponemos a infinito y el padre de cada nodo a
NULL. Encolamos, en una cola de prioridad donde la prioridad es la
distancia, todas las parejas <nodo,distancia> del grafo*/
por cada u en V[G] hacer
    distancia[u] = INFINITO
    padre[u] = NULL
    Añadir(cola,<u,distancia[u]>)
distancia[u]=0
mientras !esta_vacia(cola) hacer
    u = extraer_minimo(cola)
    por cada v adyacente a 'u' hacer
        si ((v ∈ cola) && (distancia[v] > peso(u, v)))
            entonces
                padre[v] = u
                distancia[v] = peso(u, v)
                Actualizar(cola,<v,distancia[v]>)

```

En la Fig. 1 Se puede observar el resultado de aplicar éste algoritmo.

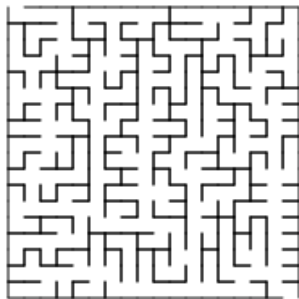


Fig. 1. Laberinto creado usando el algoritmo de Prim

B. Algoritmo de Regreso Recursivo

En el algoritmo de regreso recursivo, se crea una nueva celda cada vez que exista una sección vacía, siguiente a la celda actual en la que se está posicionado, al ser creada la celda siguiente, el algoritmo se posiciona en esa nueva celda y la celda anterior es mandada a la pila, en caso de no existir una celda vacía, se saca la celda de la posición anterior de la pila.

El laberinto termina cuando se vacía la pila. Desafortunadamente, el laberinto resultante suele contener callejones largos y sin salidas, con una solución muy enredada y larga, contraria al algoritmo de Prim [2].

C. Algoritmo de Kruskal

Este es un algoritmo que puede crear laberintos perfectos, pero se requiere un almacenamiento proporcional al tamaño del laberinto, lo cual puede ser una desventaja, al igual que el Algoritmo de Prim.

La construcción del laberinto consiste en que no crezca como un árbol, ya que se crea el paisaje completo aleatoriamente, donde que cada celda tiene un único identificador, el cual inicialmente forma un conjunto con ese único elemento. Un bucle de recorrido, se posiciona en orden aleatorio sobre los bordes.

En cualquiera de los dos lados del laberinto existen celdas con diferente identificador, el algoritmo elimina los muros y se genera un conjunto de mayor tamaño al absorber a los identificadores vecinos (los que muestran un camino de acceso entre ellos), haciendo más sencillo el camino entre ellos, y eliminando el número de bucles que se tendrán que hacer.

La fusión de estos conjuntos puede ser una operación lenta, donde aquí se utiliza un árbol por conjunto, con el identificador en la raíz y a continuación se empalman los diferentes árboles para terminar el laberinto. Este método se ejecuta de forma rápida pero no tanto como los ya mencionados anteriormente, ya que se tienen que ejecutar las lista de conjuntos [2] las cuales son las uniones o combinaciones que va haciendo conforme elimina muros.

En esta sección se puede ver pseudocódigo del Algoritmo de Kruskal para un mejor entendimiento de su funcionamiento:

function Kruskal(G)

```

Para cada v en V[G] hacer
    Nuevo conjunto C(v) ← {v}.
Nuevo heap Q que contiene todas las aristas de G, ordenando
por su peso.
Defino un árbol T ← ∅
// n es el número total de vértices
Mientras T tenga menos de n-1 vértices hacer

```

```

(u,v) ← Q.sacarMin()
// previene ciclos en T. agrega (u,v) si u y v están diferentes
componentes en el conjunto. Nótese que C(u) devuelve la componente
a la que pertenece u.
Si C(v) ≠ C(u) hacer
  Agregar arista (v,u) a T.
  Merge C(v) y C(u) en el conjunto
Return árbol T

```

La Fig. 2 Muestra un ejemplo de aplicar el algoritmo de Kruskal en la generación de laberintos.

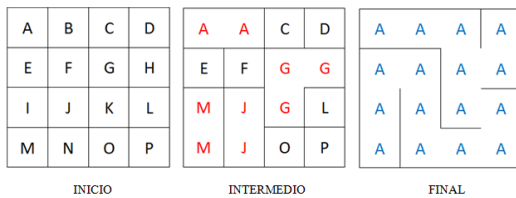


Fig. 2. Creación de un laberinto usando el Algoritmo de Kruskal

D. Algoritmo de Aldous-Broder

Es un algoritmo interesante que no requiere almacenamiento extra o una pila, ya que inicia su proceso de una celda seleccionada y se mueve aleatoriamente a una posición vecina, creando una celda dentro de ella. Se mantiene en movimiento hasta que todas las celdas vecinas ya hayan sido creadas.

La desventaja que pudiera tener (dependiendo del contexto) es que es un algoritmo lento, aparte de que no existe algún escaneo inteligente de las últimas celdas, con garantía de que no exista un término. En promedio se tarda en ejecutar 7 veces más de tiempo que los anteriores algoritmos, aunque en casos graves la creación de números aleatorios se mantiene, ocasionando no llegar a las últimas celdas [2].

E. Algoritmo de Wilson

Este es una versión mejorada del Algoritmo Aldous-Broder, pero contrario al algoritmo anterior, requiere un almacenamiento para el laberinto. El algoritmo de Wilson consiste en seleccionar una celda que será la celda inicial, y se selecciona otra celda que será la celda final, después se crea un camino aleatorio entre estas dos celdas ya creadas, una vez hecho esto se rellena ese camino con celdas creando un muro, evitando el crear bucles en el laberinto, y finalmente se anexan al laberinto.

El tiempo de ejecución de este algoritmo es 5 veces más rápido que el algoritmo Aldous-Broder y tardar menos de 2

veces más que los mejores algoritmos siendo tomado como un adiccionario de muros [2].

F. Algoritmo Caza y Mata

Es parecido al algoritmo de Regreso Recursivo, pero éste algoritmo no requiere almacenamiento extra y ni una pila. Su ejecución es más rápida que el algoritmo de Kruskal, pero un poco más que el algoritmo de Regreso Recursivo. Consiste en ubicarse en una celda ya creada y a partir de ella se activa el modo de “caza” o “hunting”, el cual escanea todo el laberinto hasta encontrar una celda deseada o creada.

Su ejecución es lenta debido a que consume tiempo de ejecución por el modo de “caza” al tratar de encontrar las últimas celdas del laberinto. Este algoritmo es adecuado para la creación de algoritmos más grandes o como un adiccionario de muros simplemente [2].

G. Algoritmo de Árbol incremental

Este algoritmo requiere tanto almacenamiento como sea el tamaño del laberinto, pero lo interesante de él, es que permite crear laberintos con texturas diferentes. El algoritmo consiste en seleccionar una celda ya creada, la cual se introduce a una lista, posteriormente se ve si la celda en la siguiente posición está deseada, si es este caso, la celda nueva es creada, pero si no existe una celda deseada se quita de la lista la celda actual. El comportamiento de este algoritmo puede cambiar conforme al tipo de selección de las celdas [2].

H. Algoritmo de Eller

Es un algoritmo más rápido que los ya mencionados, ya que la creación del laberinto no requiere mucha memoria, requiriendo solamente la memoria proporcional a la de un renglón del laberinto.

Su ejecución consiste en crear un renglón a la vez, cuando un renglón ya es creado el algoritmo ya no lo toma encuentra para crear el siguiente renglón, conforme crea los nuevos renglones, los va juntando o ligando con los anteriores, si crea los renglones de forma horizontal los liga tanto verticalmente como inversamente. El único problema con este algoritmo es que al no observar los diferentes renglones, puede crear bordes o manchas en el laberinto [2].

I. Algoritmo de División Recursiva

Es similar al algoritmo de Regreso Recursivo, solamente que el enfoque es de muros en vez de paisaje. Consiste en crear un muro vertical u horizontal aleatorio viendo la disponibilidad del área en el laberinto, con una apertura a lo largo del muro, y repite recursivamente este proceso en dos sub-áreas generadas

por el muro dividido. Para un mejor resultado se usa un bias para la selección de un muro vertical u horizontal aleatorio [2].

III. SOLUCIÓN DE LABERINTOS

En primera instancia, para resolver un laberinto se necesita recurrir a una técnica que permita hacerlo en el menor tiempo posible y en caso de poder lograrlo encontrar la solución óptima o más cercana a ella. En la literatura, existe distintos trabajos que han abordado el cómo conseguirlo.

Por otra parte, algunas técnicas son buenas para laberintos pequeños y otras son preferibles para laberintos grandes. En pocas ocasiones las técnicas existentes pueden dar todas las soluciones posibles en un tiempo aceptable, esto es debido a que un laberinto entre más grande y complejo sea, se dificulta resolverlo de manera óptima.

En esta sección, se dan a conocer y explican los diversos algoritmos que han ofrecido resultados aceptables en la práctica. Estos se introducen como sigue:

A. Algoritmo Seguidor de Muros

Su manera de trabajar es parecida a cuando un humano coloca cualquiera de sus manos sobre un muro o pared y avanza sin despegar la mano del muro, puede trabajar hacia la dirección derecha (mano derecha) o la dirección izquierda (mano izquierda).

Éste algoritmo puede guardar las celdas ya recorridas, a su vez, también guarda las celdas en las que ya ha estado dos veces, lo cual puede provocar que no encuentre la solución rápidamente, así como no siempre encuentra el camino o solución más corta [3]. La Fig. 3 muestra un ejemplo con éste algoritmo en la resolución de un laberinto.

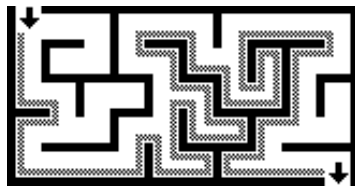


Fig. 3. Solución de un laberinto con el Algoritmo Seguidor de Muros con la mano derecha.

B. Algoritmo Relleno de Inundaciones

Es uno de los algoritmos más famosos y eficientes, para resolver laberintos con un punto de destino preestablecido [3]. Su manera de trabajar consiste en una matriz la cual tomará como un plano con valores numéricos, se ubica en la celda de inicio a la cual se le asigna los valores numéricos iguales a la cantidad de celdas que este alejada de la celda final, la cual tiene los valores numéricos del origen, es decir (0,0). Al iniciar

su ejecución el algoritmo escanea su vecindad para moverse y les asigna valores numéricos respecto a la celda final, éste proceso lo hace cada vez que se mueve a una nueva posición. Las celdas vecinas a la celda final tienen el valor numérico de 1 mientras conforme se alejan su valor numérico aumenta de 1 en 1.

C. Algoritmo de Promesa

Es una versión mejorada del Algoritmo Seguidor de Muros, ya que este puede saltar hacia un obstáculo distinto o pared. Su funcionamiento es similar a él, y puede ir a la derecha o izquierda, pero al ver que ya paso por esa celda, este continua por la dirección inicial tomando otro camino.

Si no guarda las celdas ya recorridas, puede provocar que en más ocasiones entre de nuevo a esos pasajes ya recorridos, haciendo que el tiempo de resolución aumente.

Una ventaja es que este algoritmo [2] puede solucionar un laberinto no importando su posición, es decir, puede solucionarlo inclusive desde el centro del laberinto, siempre y cuando el camino vaya al exterior y no al interior del laberinto.

D. Algoritmo Buscador del Camino más Corto

Este algoritmo como su nombre lo dice busca el camino o la solución más corta para un laberinto dado. Su manera de trabajar consiste en que conforme avanza en las celdas las va llenando, al encontrar una intersección o cruce con otro pasaje llena simultáneamente los pasajes, y así continuamente hasta que la celda final sea llenada. Una vez que es llenada la celda final hace un retroceso por ese camino hasta la celda de inicio.

A este camino se le conoce como la Ruta A, y es la solución o camino más corta si es que existieran más caminos en el laberinto [4].

Finalmente, existen otras alternativas más elaboradas en cuanto a su funcionamiento para permitir el que se calcule más de una solución a la vez. Estas alternativas recaen en el uso de sistemas multi-agente y algoritmos genéticos, de manera independiente, o bien, trabajando como una mezcla de ellos. La siguiente sección ofrece un panorama de los Algoritmos Genéticos (AG).

IV. CÓMPUTO EVOLUTIVO

El cómputo evolutivo o computación evolutiva (CE), es una rama de la computación y la inteligencia artificial (IA), en donde se trata de extraer indicios de la evolución de las especies para aplicarlo a algoritmos que puedan resolver problemas, donde otros algoritmos fallan. Por ejemplo, [5] comprende métodos de búsqueda y aprendizaje automatizado, tomando como base la evolución natural. Existen diversos enfoques del CE, los cuales son importantes comentar:

Los Algoritmos Evolutivos (AEs) son algoritmos que intentan obtener la solución a un problema complejo, imitando

la evolución Darwiniana, creando ellos mismos seres artificiales los cuales son llamados “individuo” [6]. El trabajo o labor de estos individuos es llegar a un área o solución óptima que sea capaz de resolver el problema planteado.

Existen diferentes variaciones de ellos, entre los que están los Algoritmos Genéticos (AG) [6], Programación Evolutiva (PE) [6], Programación Genética (PG) [6], entre otros. En la siguiente sección se explicará brevemente los Algoritmos Genéticos para tener un panorama de cómo trabajan y ver como pueden contribuir a la construcción y solución de laberintos.

A. Algoritmos Genéticos

Son algoritmos que trabajan para llegar a un valor óptimo o resultado, estos algoritmos están basados o tratan de imitar la evolución, trabajan mediante valores los cuales llamaremos cromosomas [7]. A un conjunto de cromosomas se le puede llamar individuo y a su vez el conjunto de individuos lo denominaremos población, un ejemplo se puede ver en el ejemplo de la Fig. 4.

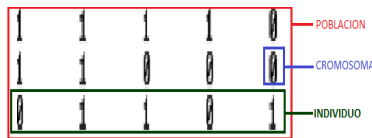


Fig. 4. Ejemplo de una población de individuos y sus cromosomas

Dependiendo del problema, los cromosomas puede agrupar un conjunto de genes, los cuales pueden comprender una característica específica del los individuos. Por ejemplo: el color de cabello de un persona. Para el caso de la Fig. 4, estamos asumiendo como ejemplo que el cromosoma está construido por un solo gen.

Los individuos son una parte esencial de este tipo de algoritmos, puesto que son aquellos que sufrirán cambios conforme se acerquen o alejen de la solución óptima. Algunos de los cambios que sufren los individuos son:

- Selección: Este proceso se encarga de seleccionar a los individuos de la población, para después mutados o cruzarlos entre ellos (primer paso de la reproducción de individuos).
- Cruzamiento: Es la acción entre dos individuos los cuales intercambian parte de su información genética, para crear nuevos individuos llamados hijos.
- Mutación: Es el proceso por el cual a un individuo se le cambia un o una cierta cantidad de genes, por valores que lo pueden acercar o alejar de la solución.

Estas son las tres operaciones básicas de los AGs, y a partir de ellas se pueden generar nuevos individuos que exploren el espacio de búsqueda, en miras de obtener a un

individuo lo suficientemente adaptado en su entorno, que resuelva de forma precisa un problema.

Así, como se menciona anteriormente, al conjunto completo de los individuos se le conoce como Población, la cual se espera que tenga una diversidad alta, es decir, que los individuos sean diferentes de los otros individuos en la población. Además de aplicar la Selección (método para escoger que individuos se cruzan y/o mutan), Cruzamiento o Mutación, se cuenta con la operación de elitismo, la cual es la acción por la cual se selecciona a uno o varios individuos con mayor adaptabilidad para pasarlos a la nueva población sin cambio alguno. Esto ayuda a preservar las mejores soluciones encontradas, a lo largo de todas las interacciones (generaciones) del Algoritmo Genético.

V. ALGORITMOS GENÉTICOS EN LA RESOLUCIÓN Y GENERACIÓN DE LABERINTOS

Como se puede ver, los Algoritmos Evolutivos pueden ser vistos como una herramienta sumamente potente que nos de prestaciones que otros algoritmos no tienen. De ahí que se pueda asumir que el uso de AGs nos permitirá resolver de forma precisa las tareas de generación y resolución de laberintos, dando posiblemente una mayor variedad al tipo de laberintos generados.

Si nos enfocamos a la generación de laberintos, podemos pensar en que cada individuo sea un laberinto, que potencialmente tiene las características que deseamos desarrollar o encontrar en una cierta clase de laberintos. Después de la evolución de la población inicial, podríamos escoger al mejor individuo encontrado y esperar que cuente con lo deseado, de forma muy próxima a ello.

Si nos enfocamos a la resolución de laberintos con AGs, podemos pensar en tener un laberinto principal a resolver, fijo durante toda la evolución, donde este puede ser leído desde un archivo, o bien generado en ese momento a partir de otro método. Posteriormente, cada individuo en la población, representaría un camino aleatorio, el cual tiene que ser optimizado para encontrar un camino de inicio a fin. Donde los caminos los podemos codificar de forma binaria, entera, inclusive de forma booleana.

Para ambos casos, se considera que se pueden aplicar las tres operaciones de las AGs comentadas anteriormente, donde la función de adaptación estará determinada por el tipo de tarea a resolver.

Existe una gran gama de AGs que son capaces de encontrar la solución óptima en base a las diferentes técnicas que puedan ser usadas en el algoritmo, una de las desventajas de este tipo de técnicas, es el alto consumo de tiempo, ya que bien, si el problema crece el tiempo de resolución, tardando más de lo esperado. A su vez el tiempo requerido para encontrar la solución óptima, estará dado por todas las evaluaciones que se realicen durante la evolución, por ejemplo, si se tienen varias tareas a optimizar, estas se pueden ingresar al AG a través de restricciones, las cuales consecuentemente requerirán más

computo que uno que no las requiera. Sin embargo, se espera que este tipo de algoritmos, pueda encontrar laberintos, y sus soluciones de mejor forma que los comentados en este artículo.

VI. CONCLUSIONES

El presente trabajo, mostro un panorama de los diversos tipos de algoritmos que existen para generar y solucionar laberintos automáticamente, dando una descripción de cada uno de ellos e indicando aspectos buenos y malos. Así como se mostro de forma general algunas de las características que tienen los Algoritmos Evolutivos (AEs), particularmente los Algoritmos Genéticos (AGs) que pueden ser considerados como candidatos a experimentar de forma teórica y empírica, en cuanto a la efectividad que estos pueden tener para resolver y generar laberintos.

Así, se espera poder encontrar el diseño de un AG que genere laberintos, con un grado de dificultad mayor. Por el otro lado, se espera que los AG puedan resolver el problema aun más rápido y más óptimo. Aparte de ello, se ha considerado el uso de sistemas multi-agente [9] de forma independiente o en conjunto con los AGs para buscar nuevos caminos que permita mejores soluciones en este ámbito.

AGRADECIMIENTOS

Se agradece las facilidades prestadas por parte del Clúster del Centro Universitario UAEM Valle de México (CLU-CUUAEM-VM), de la Universidad Autónoma del Estado de México (UAEM).

REFERENCIAS

1. Metin Turan, Kenan Aydin, "A dynamic terrain-spaced maze generation algorithm"
2. <http://www.astrolog.org/labyrinth/algrithm.htm>.
3. Sayuti Saman, Abdramane, "Solving a Reconfigurable Maze using Hybrid Wall Follower Algorithm", 2013.
4. "A new Shortest Path Finding Algorithm for a Maze Solving Robot with Simulation", 2011.
5. Ochoa G. "Introducción a la Computación Evolutiva y la Morfogénesis Artificial".
6. Gareth Jones "Genetic and Evolutionary Algorithms", University of Sheffield, UK.
7. Mat Buckland "AI techniques for Game Programming", Premier Press, 2002.
8. Lee, Ki Sang, "Maze – Generating Algorithms". Apuntes de clase de la materia Physical Mathematics I, impartidas en 2009 por el professor Ewan D. Stewart en High Energy Physics Group del departamento de Física, KAIST. South Korea.
9. Méndez García, Angeles E. Arquitectura para sistemas multiagente en aplicaciones distribuidas usando tecnología Java. 2015. Tesis de Licenciatura. Centro Universitario UAEM Valle de México.

Arquitectura de un Sistema de Control para una Fuente Mágica

Dr. José Martín Flores Albino
Ingeniería en Sistemas y Comunicaciones
Centro Universitario UAEM Valle de México.
Atizapán de Zaragoza, Estado de México.
jmfloresa@uaemex.mx

Dr. Victor Manuel Landassuri Moreno
Ingeniería en Computación
Centro Universitario UAEM Valle de México.
Atizapán de Zaragoza, Estado de México.
vmlandassurim@uaemex.mx

Luis Felipe Romero Morales
Ingeniería en Sistemas y Comunicaciones
Centro Universitario UAEM Valle de México.
Atizapán de Zaragoza, Estado de México.

Resumen—: *En este trabajo se presenta información sobre la arquitectura y componentes de los sistemas utilizados en el diseño y control de fuentes dedicadas para espectáculos públicos (conocidas también como Fuentes Mágicas) El objetivo es proporcionar conocimientos sobre sus elementos tecnológicos que son necesarios para su operación. Además como parte del proyecto se incorporan la Inteligencia Artificial para el desarrollo de secuencias automáticas.*

Palabras clave — *Sistemas de Control, Inteligencia Artificial, Fuente mágica.*

I. INTRODUCCIÓN

A. El agua

Desde el principio de los tiempos los humanos hemos buscado estar en contacto con el agua porque es indispensable para nuestra vida. Compuesta por dos moléculas de hidrógeno y una de oxígeno. El agua es inodora, incolora e insípida que a su vez la podemos encontrar en sus tres estados moleculares sólida (glaciares y polos), líquida (en ríos, lagos y mares) y gas (evaporación y niebla), también sabemos que su punto de fusión es a los 0°C y el de ebullición es a los 100°C, siendo un elemento que tienen en común todos los seres vivos.

Civilizaciones antiguas se asentaron en importantes ríos, lagos o mares; por ejemplo: los Egipcios, Aztecas, Mayas, Chinos y Romanos todos ellos empezaron como un pequeño pueblo que sólo buscaba sobrevivir, pero el dominio del agua les permitió llegar a ser potenciales civilizaciones que a través del tiempo dejaron huella en la historia.

A los humanos no sólo nos preocupa todas las características y propiedades del agua, también a lo largo de nuestra existencia terrestre, la humanidad ha mostrado gran interés por su control, manipulación y almacenamiento, lo demuestra los miles de inventos desarrollados para el manejo de este vital líquido. El agua nos limpia, alimenta y ayuda a nuestro desarrollo, es sinónimo de vida, salud, belleza y

pureza, es un recurso natural renovable que nos toca cuidar a todos. (Agua, 2009).

B. Las fuentes ornamentales

Por otro lado tenemos la propia historia de las Fuentes ornamentales, que para el presente trabajo es importante. Como ejemplo de importantes fuentes, tenemos: a la famosa fuente en Roma llamada “La fontana de Trevi” edificada en 1743. Es famosa por encontrarse en un lugar turístico donde cada año tiene miles de visitantes, su historia se remonta a la construcción de los antiguos acueductos. En la edad media fue dañada, muchos papas católicos intentaron restaurarla pero no se pudo lograr la restauración, en ese entonces la perfección era sinónimo de vida o muerte, por esta razón ninguno quería tomar la batuta para llevar a cabo el proyecto. Desde el 2007 cuenta con un circuito de agua reciclada que se renueva constantemente (Malissard, Alain, 2001).

Las fuentes reflejan un estado de abundancia, poder o bienestar social debido a que son la cúspide de obras magnas de un elemento esencial para la civilización. En un principio las fuentes se construían sobre los pozos o en los ojos de agua, y se convertían en una señal para indicar la existencia de agua potable y muchas ocasiones a quien pertenecía. Los griegos consideraban a las fuentes de agua sagradas. El filósofo griego Thales considero a las fuentes como la génesis de pueblos. Las fuentes de agua públicas datan desde los romanos donde se construían a partir de bases cuadradas sobre las que se colocaban esculturas de animales, humanos o dioses.

Las fuentes de agua son un elemento de arquitectura que transforma el lugar donde se encuentra. El agua estimula los sentidos, el movimiento del agua produce efectos visuales y acústicos que hacen que las personas se tranquilicen y se relajen, es por eso, que las fuentes en las ciudades o grandes concentraciones de personas como en edificios, logra un efecto tranquilizador de la ansiedad, ya no solo forman parte del paisaje, sino como una “terapia alternativa a la ansiedad”.

En este trabajo se presenta los antecedentes del proyecto para un sistema de control de una fuente ornamental. Las fuentes mágicas pasaron de ser construcciones estáticas, a ser complejos sistemas que utilizan actuadores, sensores que las hace más atractivas. El campo de la Inteligencia Artificial puede a portar a las fuentes ornamentales un comportamiento basado en la información de los sensores, piezas musicales o de otro tipo.

La organización de este artículo es como sigue: la Sección II presenta las arquitecturas de las fuentes, mientras que la Sección III las tecnologías en ellas. Así, pasamos a la sección IV donde se muestra el sistema de control de fuentes y en la Sección V la técnicas de Inteligencia artificial usadas en el control de ellas. Para finalizar, tenemos las conclusiones en la Sección VI.

II. ARQUITECTURA DE LAS FUENTES

La tecnología en las fuentes ornamentales es muy antigua. Desde sistemas puramente hidráulicos, hasta sistemas que incluyen elementos de control, sensores eléctricos y electrónicos. Básicamente una fuente es una construcción que emana agua, está cae en un depósito, y posiblemente un sistema de recirculación.

A. Arquitectura Física

Físicamente las fuentes ornamentales utilizan un pozo donde emite el agua, misma, que es tomada para hacerla circular o transportar. Los pozos pueden ser naturales o artificiales. Los naturales son los: manantiales, ojos de agua, etc. El agua puede estar impulsada o estática. El agua que sale es impulsada de los pozos, debido a la presión a la que está sometida en el interior es mayor que la atmosférica. También ocurre que los depósitos de agua están a temperaturas muy altas lo que hace elevar la presión y la hacen fluir; como en los géiseres.

Generalmente las fuentes ornamentales requieren de un taque de agua, o cisterna para contener un repositorio de líquido para regular el abastecimiento de la fuente. Este depósito requerirá de válvulas de control para el ingreso/salida del agua. Será necesario filtros para el líquido que ingresa evitando el paso de objetos extraños a la cisterna. En el caso de que la cisterna este en un nivel de altura más alto podría ocurrir que la presión del agua produzca un efecto *sifón*, lo que vaciaría la cisterna, en vez de llenarla. Para evitar que esto ocurra, se requerirá una válvula anti-retorno al ingreso del depósito.

En la salida del sistema se requerirá una válvula. Para que el agua sea impulsada hacia afuera se utiliza la gravedad o un sistema de bombeo, en ambos casos se hace a través de elevar la presión del agua en la cisterna con respecto al punto de salida del agua. Los sistemas por gravedad son económicos siempre y cuando sea posible construir la cisterna a una altura mayor que la de la fuente. Si no es posible usar el sistema por gravedad se requerirá de un sistema de bombeo.

Los sistemas de bombeo se pueden dividir en dinámico y por desplazamiento; Los sistemas de bombeo dinámico se basan en darle energía al agua continuamente y no requiere de un volumen almacenado, por ejemplo: las bombas centrífugas son de este tipo, que utilizan una turbina para incrementar la presión hidrostática produciendo un efecto de succión (The Centrifugal Pump). Los sistemas de bombeo por desplazamiento le proporcionan energía al agua contenida en un volumen; ejemplo de este tipo de bombas son las de diafragma o pistón, estas bombas proporcionan un volumen de agua constante y debe tenerse cuidado no operar cuando la salida de agua está cerrada. También podemos encontrar sistemas de bombeo que pueden estar sumergidos, en este caso el sistema de impulso está contenido en un receptáculo que puede sumergirse y elimina la necesidad de tubería de suministro entre el pozo y la bomba. En todos los sistemas hidráulicos se produce el efecto de cavitación, es cuando se forman burbujas de vapor en el líquido que fluye, estas burbujas se destruyen colapsando sobre sí mismas provocando un choque hidrostático que produce calor y daña los materiales que están cerca, una posible solución a este problema es una válvula expulsora de aire.

Para distribuir el agua con la presión y volúmenes suficientes para la operación de las fuentes se utiliza tubería de metal o sintético (PVC). Este debe calcularse para proporcionar el volumen de agua suficiente. Por ejemplo una tubería de ½ pulgada de diámetro requiere proporcionar un flujo o gasto de 5 galones (3.78 litros) por minuto, significa que la velocidad de salida del agua de 98 pulgadas/segundo.

$$D = \sqrt{\frac{4F}{\pi}}; v = \frac{4f}{\pi D^2}; f = \frac{1}{4}\pi D^2 v$$

donde D=diámetro interno de la tubería en *pies*; f= volumen del flujo en *pies cúbicos/segundo*; v es la velocidad de salida del agua en *pies/segundo*.

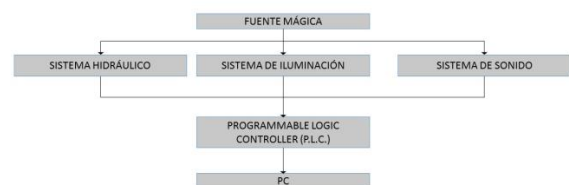


Figura 2: Esquema general de los subsistemas de una fuente mágica.

La tubería abastece a los elementos de salida finales que son boquillas que tienen como tarea dar forma al agua saliente. Los efectos de salida del agua es variada. Podemos

clasificar algunos de estos efectos en dos clases: Spray o pulverizadores, y efecto de colimación. Los primeros dispersan el agua en un amplio patrón y mayor del de salida, se parece al efecto de las lentes divergentes de luz. Las boquillas de colimación crean un flujo concentrado de agua, se forma un flujo uniforme que concentra el agua y permite crear una estructura que sorprende por su consistencia y claridad. En (Eggers & Villermaux, 2008) se encuentra una descripción de la física de estos elementos.

III. TECNOLOGÍA EN LAS FUENTES

Las fuentes que operan por medio de gravedad son las más comunes y las primeras que se construyeron en las primeras ciudades del mundo. Estas fuentes eran activadas por el mismo flujo de suministro de agua. Desde los 1920 con la disposición de bombas se implementaron más elementos tecnológicos a las fuentes.

El sonido del agua es tranquilizante. Esto lo notaron grandes autores de música orquestal como George Frideric Handels con su concierto llamado "Música Acuática" que fue un encargo del rey Jorge I para ser interpretada sobre el río Támesis, así que la unión de música y agua brotando de fuentes, potencia las emociones de los espectadores.

En fuentes altamente tecnificadas como las del Hotel Bellagio en Las Vegas, son un espectáculo por si solo y recibe una gran cantidad de visitantes. Tecnológicamente es impresionante la comunión entre agua, luz, música y video. Sobre un lago de poco más 3 hectáreas, que contienen 27 millones de galones de agua, y en algún momento hay 17,000 galones de agua en el aire (Shakerin, Whater Fountains with Special Effects, 2005).

Los avances tecnológicos en el control de fuentes son impresionantes. Esto ha permitido que las fuentes mágicas tengan dinamismos muy interesantes, por ejemplo, las fuentes interactivas: Son fuentes donde el usuario o espectador puede iniciar una secuencia de acciones en la fuente, ejemplo de esto es lanzar una moneda a su interior y ver como se activa el movimiento, asociado a esto hay una cantidad de sensores que permita que la fuente responda interactuando con el espectador o espectadores.

Fuentes danzantes: en 1975, Gunter Przystawik, patento un mecanismo con el cual podía mover las boquillas de agua y sincronizarlos con música. Tomas Simons en 1996 patento un sistema de flujos en oposición para producir formas que podían variar en dirección dependiendo de la presión y los flujos en los surtidores. Robinson y Fuller inventaron un sistema de fuente que ilumina flujos de agua a través de llamas de colores (patente US 4858826)

Por el contrario, las fuentes iluminadas aprovechan el efecto de refracción (la desviación de la luz al pasar de un medio a otro) y la reflexión (la luz es devuelta al medio

cuando trata de pasar a otro de diferente), el agua puede ser iluminada. A través del Experimento de Thyndall-Colladon (1820-1893) se puede observar como un flujo de agua en movimiento puede ser el conducto para transmitir la luz y al hacerlo queda iluminado. Lo que dio origen a la trasmisión de información a través de medios que conducen la luz (fibra óptica). En la actualidad los sistemas de iluminación han evolucionado y han impactado en la iluminación de las fuentes ornamentales. Los dispositivos de alto rendimiento hacen que la transformación de energía eléctrica a luz sea más eficiente reduciendo la cantidad de calor emitido en el proceso. Si se usa una fuente de tipo incandescente de luz (foco), se genera mucho calor y resulta un problema dado que el cambio de temperatura al estar en contacto con el agua hace que su vida se reduzca. Formas para aislar la fuente de luz del agua es a través de fibra óptica de vidrio, de esta forma la fuente de luz queda aislada del agua. Con la tecnología Led, que producen luz por una recombinación de electrones en la franja de contacto de dos materiales semiconductores de diferente clase. Se ha logrado producir luz con mejor rendimiento y sin producir tanto calor en el proceso. Las fuentes de luz led han incluso emitido luz coherente, tipo láser que la hace más concentrada.

IV. SISTEMA DE CONTROL DE FUENTES

Los sistemas de control han evolucionado desde los sistemas puramente mecánicos o hidráulicos hasta los computacionales. Un sistema de control requiere de: sensores, actuadores y un procesador. Los sensores son los responsable de comunicar los cambios en las variables que influencia al sistema que es controlado. En el caso de las fuentes, se requieren sensores que detecten que los subsistemas están en condiciones para operar, para que en el caso de falla se alarme o incluso proteja al sistema evitando que se active. También se necesitan sensores que permitan ejecutar las acciones del sistema para determinar su comportamiento; por ejemplo, si una descarga debe depender de un evento anterior, se debe detectar su término para que el siguiente efecto se ejecute. Algunos sensores podrán incluso interactuar con el público y tal vez responder a los aplausos de las personas en una fuente interactiva. En conclusión los sensores son los sentidos del sistema completo, que lo protege y le permite funcionar.

Los actuadores son los dispositivos que permiten que la fuente realice las acciones deseadas: Desde las bombas y válvulas, hasta motores que permitan movimiento, altavoces que generen música o sonidos sincronizados y video proyectado sobre las formas de agua.

Los sensores y actuadores son vistos como los ojos y brazos, respectivamente del sistema. Pero se necesita un procesador que de las reglas de comportamiento y permita sincronizar los eventos externos de entrada y salida del sistema.

El control de una fuente interactiva demanda una sincronización de varios sistemas: hidraulico, iluminacion y

sonido. Esto hace que la demanda sea intensa para el procesador. El equipamiento de sensores y actuadores permite que la información sea enviada por una red de datos común. Son redes de datos que llevan la información de sensores y actuadores hasta los equipos de procesamiento, esto simplifica la transmisión de ida y vuelta. Redes como Ethernet industrial podrían utilizarse.

En (Blaho, Bielko, Farkas, & Fodrek, 2014) habla de un sistema de control para requerimientos de precisión y confiabilidad, donde se requiere repuesta en tiempo real del sistema, examinan ambientes de ejecución en tiempo real. Comparan sistema operativo Linux RTAI, y en java en tiempo real y lo ejemplifican con una plataforma de control, que entre ellas está la de una fuente interactiva.

En (Shakerin, Microcontroller Water Fountain: a Multidisciplinary Project, 2004) se presenta una diseño de una fuente con nueve jets, que forman las letras del alfabeto y algunas formas geométricas. Todo ello con un sistema controlado por un microcontrolador y aun costo muy bajo. Su objetivo es dirigido a un ámbito de la enseñanza. No obstante se habla de la complejidad de este tipo de sistemas.

V. INTELIGENCIA ARTIFICIAL EN EL CONTROL DE FUENTES.

La Inteligencia Artificial es un área de las ciencias de la computación que tiene como meta replicar la capacidad de razonamiento y discernimiento huma de forma artificial. Las computadoras son sistemas de cálculo que han permitido reproducir paradigmas de los mecanismos de razonamiento. Modelos de razonamiento como son: Aprendizaje basado en Reglas (Rule-Based Learning); Lógica Inductiva Programada (Inductive Logic Programming); Aprendizaje por Instancias (Instance-Based Learning) han sido resultados de este trabajo. En apoyo de los sistemas que modelan en razonamiento, los sistemas evolutivos del conocimiento han trabajado para permitir que los datos sean procesados de forma interactiva para lograr extraer patrones que proporcionan información. Como ejemplo: Los algoritmos genéticos, que imitan el proceso de evolución genética. Comienzan con una población inicial y a través de procesos de mutación aleatoria o de cruzamiento, sólo algunos individuos podrán pasar a la siguiente generación, en cada iteración se evalúa una premisa y sólo aquellos que mejor la cumplan podrán volver a evolucionar. Aprendizaje reforzado. Es un paradigma que modela la comprensión del ambiente a través de permitir experimentar con las posibles respuesta de un agente, a través de la experiencia, el agente, ajusta su comportamiento hasta logra un objetivo, de este modelo se desprenden los paradigmas de agentes múltiples (Multi-agents Systems). Por último consideremos las redes neuronales artificiales. A través del modelo Neuronal de Santiago Ramón y Cajal (1857-1934) que en biología nos permitió entender la estructura el cerebro como una red de elementos de procesamiento formado por las celular neuronales, un tipo particular de células que compartes una enorme cantidad de conexiones entre ellas compartiendo e intercambiando información, y en función de estos

intercambios modifican su comportamiento, permitiendo fijar información en nuestro cerebro, además de responder a situaciones desconocidas pero familiares de manera adecuada.

En el Artículo (Yoo & Lee, 2009) proponen el uso de la Inteligencia Artificial para la ceración de secuencias para una fuente musical. En este trabajo presentan la hipótesis que a través del uso de la Inteligencia Artificial es posible reducir costos al crear un sistema que genere automáticamente los escenarios de comportamiento de las fuentes. Utilizan redes Bayesianas para el procesamiento de la música y la generación de los patrones de comportamiento en la fuente.

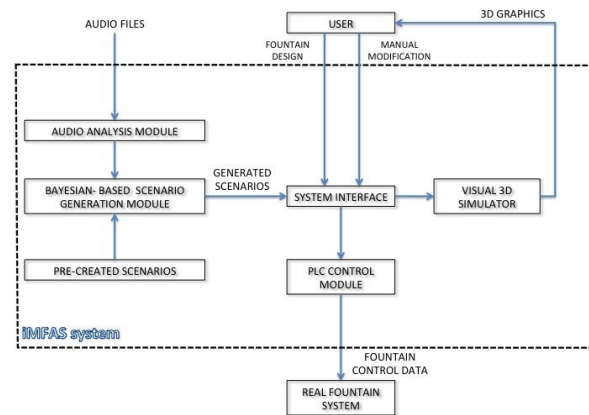


Figura 2: Esquema general del Sistema para el control de la Fuente música (Yoo & Lee, 2009).

En el trabajo citado anteriormente a través de la transformada de Fourier de la señal de sonido o pieza musical se construyen una red Bayesiana que tiene como elementos a cada uno de los surtidores que través de la probabilidad de que el surtidor sea encendido o no, a partir del nivel del espectro de la señal, además de incluir correlación entre ellos, el modelo propuesto después les servirá para construir nuevos escenarios al cambiar la señal musical y pasarlo por la red.

En (Vempala & Russo, 2012) hacen uso de dos Redes Neuronales, una de tipo Feedforward y otra de tipo Elman para predecir la clase de emoción que una pieza musical expresa. Clasifican 12 piezas musicales con 13 características, la red proporciona un índice Valence/Arousal que en términos psicológicos el primero expresa la emoción atrayente y el segundo una emoción de exaltación. Estos índices son tomados de una muestra de 45 personas y se entrena a la red con esta información. Para el trabajo con la fuente es interesante esta investigación porque permitiría que la reacción de la fuente correspondiera al tipo de emoción que se esperaría que una pieza musical se perciba.

VI. CONCLUSIONES

En este trabajo se presenta un panorama general de la importancia del agua para los seres vivos, así como el control y manipulación del vital líquido influyó para que las civilizaciones antiguas dejaran vestigios al paso de los siglos; La historia y relevancia propia de las fuentes de agua, no sólo vistas como un elemento ornamental, más como una importante muestra del progreso, poder y bienestar de una sociedad. La tecnología ha impactado sus posibilidades en los sistemas modernos de fuentes para darles más dinamismo y atractivo, tanto así que ya hay ejemplos de importantes fuentes que atraen a miles de personas por su impacto visual. La tecnología de la Inteligencia Artificial puede aportar de forma directa en el diseño de escenarios de comportamiento dando autonomía y apoyo para su operación, para que el diseñador del espectáculo se vea favorecido con un asistente inteligente y que el espectáculo se vuelva una embriaguez para los ojos y oídos del espectador en turno.

VII. REFERENCIAS

Agua, C. N. (2009). Semblanza Histórica del Agua en México. En C. N. Agua, *Semblanza Histórica del Agua en México* (págs. 1-82). México: Secretaría de Medio Ambiente y Recursos Naturales.

Blaho, M., Bielko, S., Farkas, L., & Fodrek, P. (2014). Computer based control with real-time capabilities. *WEAS transaction on Systems and Control* , 16-2.

Eggers, J., & Villermaux, E. (2008). Physics of liquid jets. *Reports on Progress in Physics* , 1-79.

Malissard, Alain. (2001). Los romanos y el agua. La cultura del agua en la Roma antigua. Barcelona: Herder.

Shakerin, S. (2004). Microcontroller Water Fountain: a Multidisciplinary Project. *Int. Journal Engng* , 654-659.

Shakerin, S. (2005). Whater Fountains with Special Effects. *American Scientis* , 444-4451.

The Centrifugal Pump. (s.f.). *Research and Technology* .

Vempala, N. N., & Russo, F. A. (2012). Predicting Emotion from Music Audio Features Using Neural Networks. *9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)* , 19-22 J.

Yoo, M.-J., & Lee, I.-K. (2009). Creating Musical-Fountains Shows. *IEEE Computer Society* , 6-13.

Prototipo para el uso y control de dispositivos en el hogar utilizando tecnologías de la información y comunicación.

Christian Gómez-Simón¹, Maleny Ramírez-López², Carlos Omar González-Morán³

¹⁻²Departamento de Ingeniería en Computación ³Área de investigación de Ingeniería Industrial

Centro Universitario UAEM Valle de México

Atizapán de Zaragoza, 54466, México.

¹christian.gomz@outlook.com, ²maleny.ramirez.92@gmail.com, ³coglez@gmail.com

Resumen

Hasta el día de hoy la necesidad de mejorar y hacer más cómoda la vida cotidiana, ha triado como consecuencia implementar tecnología para facilitar las tareas en el hogar. Para automatizar los dispositivos de un hogar se desarrolló un prototipo llamado XDESIMA que permite manejar y controlar la iluminación de una maqueta, el cual contiene leds alimentados por 5 Volts de corriente directa y a su vez bombillas alimentadas con corriente alterna de 125 Volts, con la finalidad de simular la instalación eléctrica de un hogar. Bajo este principio el primer prototipo puede controlar el encendido y apagado de cualquier dispositivo electrónico que funcione con alimentación de 125 Volts de corriente alterna, esto con una aplicación en un dispositivo móvil con sistema operativo Android y un circuito electrónico compuesto esencialmente por Arduino, modulo Bluetooth HC-05, TRIAC y MOC3010.

Palabras clave: *Domótica, aplicaciones móviles, Android, Arduino, automatización del hogar.*

1. Introducción

Hasta el día de hoy la necesidad de mejorar y hacer más cómoda la vida cotidiana, ha traído como consecuencia implementar tecnología para facilitar las tareas en el hogar. En la sociedad actual una de las necesidades más concurrentes es el control de la iluminación

de un hogar por y para personas en sillas de ruedas o adultos mayores, actualmente esta necesidad ha sido atacada por diferentes empresas e instituciones dando diversas soluciones gracias al uso de la domótica, de esta forma la construcción de casas automatizadas da una gran solución a este problema.

Stefan Junestrand, Xavier Passaret y Daniel Vázquez indican que la domótica en el mercado actual ha sido adoptada como la capacidad de automatizar y controlar remotamente las instalaciones domesticas con el beneficio de tener un ahorro energético, el confort y la seguridad de los hogares [2].

Con este concepto además de dar un mayor confort a la sociedad, se puede tener una mejor calidad de vida para las personas de edad adulta y/o en silla de ruedas. En base a esta necesidad se desarrolló un prototipo de control para la iluminación de los hogares llamado XDESIMA. Con XDESIMA de entrada se tendrá el control de encendido y apagado de las bombillas de un hogar mediante una aplicación móvil (Smartphone, Tablet) que cuente con sistema operativo Android. Con este principio a futuro se contempla utilizar la base de este prototipo para implementar un sistema de control más amplio, es decir, se podrá tener el control de diversos dispositivos (audio, apertura de persianas, rutinas en la iluminación, escenas configurables) y la

lectura de datos en tiempo real de sensores (humo, temperatura, humedad, entre otros).

2. Resumen del Sistema

El funcionamiento principal es controlar la iluminación de las bombillas y leds a partir de una aplicación móvil, para cada una de las estancias que componen un hogar. El proceso es sencillo: se tiene una aplicación instalada en un dispositivo móvil con Android que contiene un conjunto de botones, Fig. 1, y una programación por eventos, es decir que la aplicación está a la espera de lo que haga el usuario dentro de la aplicación, por ejemplo al pulsar el botón para encender o apagar una bombilla, este enviará un carácter al módulo Bluetooth y lo reenviará al Arduino para ejecutar un conjunto de instrucciones acorde al carácter recibido. Dentro de las instrucciones a realizar se encuentra el manejo de la señal de control, donde el circuito electrónico con esta señal de 5V de corriente directa finaliza el proceso de encendiendo o apagado del dispositivo conectado. Este es un prototipo integral porque incluye tanto electrónica analógica como electrónica digital controlado por una aplicación móvil. La Fig. 2 permite observar el proceso y la comunicación que hay entre los dispositivos de Hardware y Software de forma general.



Figura 1: Aplicación en Android.

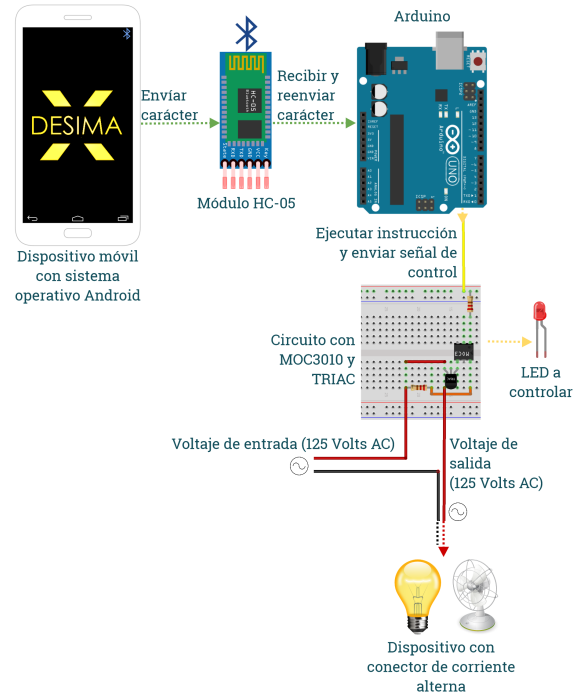


Figura 2: Comunicación entre componentes del prototipo XDESIMA.

El párrafo anterior describe una solución práctica y sencilla, teniendo como base anteriores trabajos descritos en artículos internacionales tales como: 1) el sistema HAS (Sharon Panth & Mahesh Jivani); 2) el sistema Ubiquitous Smart Home (Shiu Kumar, 2014). Ambos autores realizan un par de sistemas capaces de resolver la problemática abordada en este trabajo.

1. Sistema HAS: es un sistema electrónico con una aplicación para dispositivos móviles con sistema operativo Android, la gran ventaja que ofrece el sistema HAS es la compatibilidad de poder utilizarse en la mayoría de dispositivos móviles con Android (celulares, tabletas, dispositivos multimedia equipados en los aviones, entre otros) y sin importar la versión que se tenga instalada. Este sistema utiliza la tecnología Bluetooth equipados en los dispositivos móviles para la comunicación con el microcontrolador AT89C51 el cual se encarga de enviar una señal de interrupción para el encendido o apagado de un Led.

Este sistema fue probado en diferentes dispositivos móviles y diferentes versiones de Android, y fue expuesta la gran ventaja de portabilidad de utilizar este tipo de sistema operativo para realizarlo [4].

2. Este sistema es muy similar al anterior, sin embargo no utiliza la tecnología Bluetooth como medio de transmisión de datos, este utiliza Internet como medio de comunicación entre una tarjeta Arduino y dispositivos móviles con sistema operativo Android. Una de sus características es que el usuario puede utilizar una aplicación móvil para controlar dispositivos electrónicos y obtener lecturas de sensores pero a su vez también puede ser utilizado desde cualquier navegador web a través de una página Web. La lectura de datos a través de sensores (gas, humo, movimiento, temperatura y humedad) son enviados del Arduino al dispositivo móvil. Una vez que el usuario monitorea los valores de los sensores, se pueden tomar decisiones para controlar la iluminación, aire acondicionado, ventiladores, cerraduras y alarmas desde la aplicación y la misma tarjeta Arduino.

La principal ventaja de este sistema radica en su algoritmo, donde se implementó un componente de autenticación, es decir, cada que el usuario desee controlar algún dispositivo electrónico del hogar, ya sea desde la aplicación o algún navegador Web, deberá ingresar la dirección IP de la tarjeta de red conectada al Arduino y además una contraseña previamente establecida, esto con la finalidad de evitar que alguien ajeno lo utilice [6].

Los componentes de hardware más importantes que integran este sistema son: Tarjeta Arduino, Bluetooth HC-05, TRIAC y MOC3010, la función de cada uno de estos componentes se describe a continuación:

- Tarjeta Arduino: es una placa que consta de un microcontrolador ATmega encargado de ejecutar instrucciones programadas y procesar datos recibidos por el dispositivo móvil. Para funcionar solo necesita de un adaptador de corriente alterna o conectarse a una batería de corriente directa.

Para programar o reprogramar el Arduino se requiere conectarse a una computadora mediante cable USB.

- Módulo Bluetooth HC-05: recibe datos del dispositivo móvil y los transmite al Arduino para que este ejecute las instrucciones correspondientes. La característica principal es que puede funcionar como un módulo Maestro-Esclavo lo cual refiere a que además de recibir conexiones también es capaz de generar conexiones a otros dispositivos que cuenten con Bluetooth.
- TRIAC: permite abrir o cerrar un circuito en donde haya un flujo de corriente variable, para tener comunicación con el Arduino se requiere un dispositivo intermedio llamado MOC (optoacoplador, fotoacoplador o fotointerruptor).
- MOC 3010: es un material semiconductor el cual contiene un led interno; cuando este dispositivo se activa con corriente directa permitirá que el TRIAC cierre el circuito de corriente alterna, ocasionando el encendido o en su defecto el apagado de la bombilla de 100 Watts.

3. Diseño del hardware

Esta sección puede llevar a debatir el uso de los dispositivos para controlar el flujo de corriente alterna, debido a que existen diferentes alternativas al uso de un circuito con TRIAC y MOC. Sin embargo se implementó este circuito de forma que solo requiera un cable extra (cable de control) para la instalación de una bombilla.

Para controlar la apertura o el cerrado de un circuito de corriente alterna se requiere un par de dispositivos intermedios (TRIAC y MOC), donde el MOC3010 recibe una señal de control con corriente directa a 5V de uno de los pines de salida del Arduino y con esta señal se incita al TRIAC a cerrar o abrir el circuito de corriente alterna. El circuito de la Fig. 3 muestra que además se tiene conectado un led que debe encenderse o apagarse al mismo tiempo que la bombilla, para que esto ocurra el led tomará la señal de control como

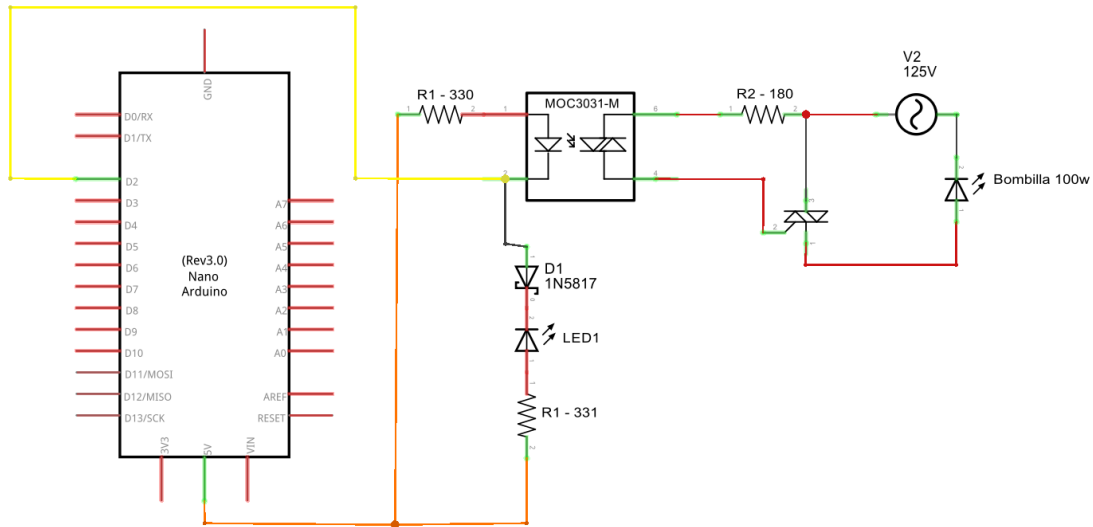


Figura 3: Circuito del prototipo XDESIMA para encender una bombilla y un led.

que recorra todas las tareas programadas durante cada bucle, cuando esta variable cambie de valor se buscará la tarea asociada al carácter recibido y al finalizar la ejecución de las tareas esta variable se reiniciará.

Tabla 1: Caracteres asociados en el algoritmo de Arduino.

Estancia	Encender	Apagar
Entrada	a	1
Planta baja	b	2
Estancia 1	c	3
Estancia 2	d	4
Escaleras	e	5
Aire Acondicionado	f	6
Anuncio	g	7
Encender todo	x	
Apagar todo		y

MOC para controlar el flujo de corriente alterna y finalmente en la parte inferior izquierda se encuentra el Arduino conectado al módulo Bluetooth HC-05.

El proceso para utilizar el prototipo es el siguiente:

- El usuario inicia la aplicación en el dispositivo móvil.
- La aplicación realiza una conexión al módulo Bluetooth HC-05.
- La aplicación se quedará a la espera de eventos (esperando que el usuario pulse un botón).
- Cuando el usuario pulse un botón, la aplicación enviará un carácter para que el Arduino ejecute el encendido o apagado de una bombilla.

5. Implementación

La integración del diseño de software y hardware trajeron como consecuencia el prototipo integral XDESIMA ilustrado en la Fig. 6, y la implementación del mismo. En la parte superior de la imagen se observa la maqueta el cual contiene la instalación de los leds y las bombillas a ser controlados. En la parte inferior derecha está el circuito eléctrico con TRIAC y

6. Conclusión

Los resultados de desarrollar el prototipo llamado XDESIMA fueron positivos, permitiendo tener el control de la iluminación en una distancia menor a 10 metros, este prototipo funcionó de forma correcta durante lapsos de 24 horas permitiendo la instalación de este prototipo para el control de una habitación. La instalación y el funcionamiento

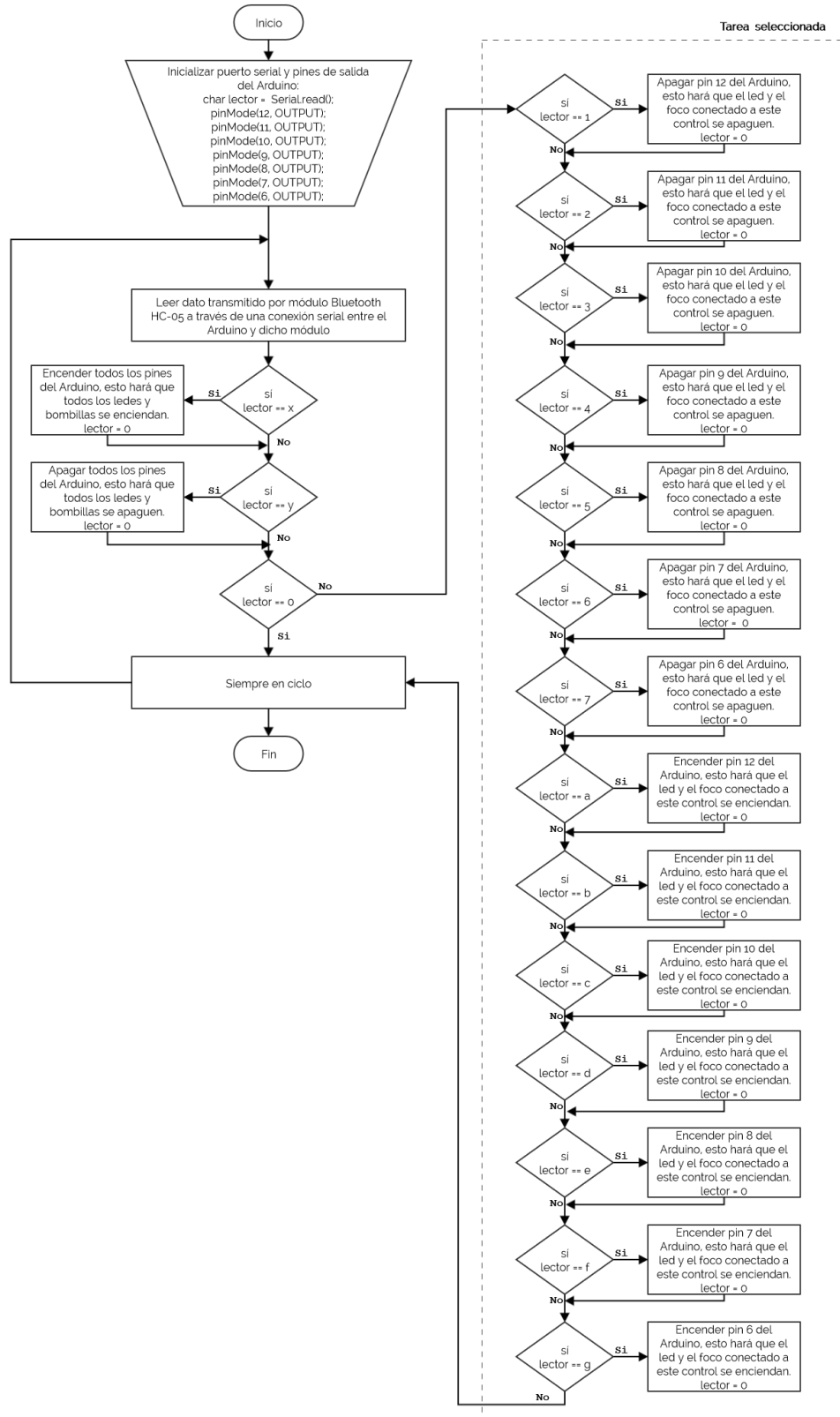


Figura 5: Diagrama de flujo de la aplicación móvil.

de este prototipo arrojo los mismos resultados positivos en un ambiente real. Donde el usuario que lo utilizó afirma haber obtenido grandes ventajas en cuanto a un mayor confort.

En el proceso de pruebas, la especificación del módulo HC-05 indica que el alcance máximo es de 10 metros, pero esto solo es verdad cuando se encuentra en un ambiente libre de interferencias. En el caso particular de un hogar, las paredes son un caso de interferencia el cual puede reducir considerablemente la distancia para obtener una conexión.

En trabajo a futuro se pretende integrar un sistema de control el cual nos permitirá saber quién y a qué hora a tocado el timbre de un hogar, el -¿Quién? lo resolveremos mediante la fotografía de la persona que ha tocado, ya que inmediatamente al tocar el timbre se tomará una fotografía, la cual será almacenada y el -¿A qué hora? se resolverá mediante una base de datos la cual contendrá la hora de la visita. Todo esto se pretende sea manejado vía Internet cuando el usuario este fuera del hogar y dentro del hogar será vía Bluetooth, es decir, se enviará una notificación con los datos antes mencionados.

Finalmente se pretende de igual forma controlar motores en las persianas, los cuales tendrán como finalidad abrir o cerrar dichas persianas, permitiendo así el paso de luz exterior, también se desea integrar el encendido de un boiler con el cual traería consigo un número de soluciones ya que levantarse a encender el boiler consume tiempo por las mañanas. Finalmente en la aplicación se podrá programar rutinas para iluminación de un hogar, con

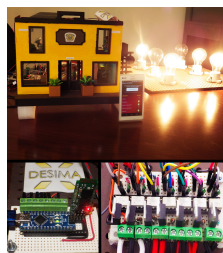


Figura 6: Implementación del prototipo XDESI-MA.

la opción de tener escenas configurables, por ejemplo fiesta, cita romántica, cine, estudio, entre algunos otros.

Referencias

- [1] Stefan Junestrand, Xavier Passaret, Daniel Vázquez: Domótica y hogar digital. Editorial Paraninfo, 2005.
- [2] (2011) Power PDA website [En línea]. Disponible: www.poderpda.com/editorial/el-sistema-operativo-movil-preferido-por-los-usuarios-de-poderpda
- [3] Somak R. Das and Silvia Chita: Home Automation and Security for Mobile Devices. 1st IEEE PerCom Workshop on Pervasive Communities and Service Clouds.
- [4] Sharon Panth and Mahesh Jinani: Home Automation System (HAS) using Android for Mobile Phone. International Journal of Electronics and Computer Science Engineering.
- [5] Syed Anwaarullah and S.V Altaf: RTOS based Home Automation System using Android. International Journal of Advanced Trends in Computer Science and Engineering, 2013.
- [6] Shiu Kumar: Ubiquitous Smart Home System using Android Application, International Journal of Computer Networks & Communications (IJCNC), Vol. 6, Enero 2014.
- [7] Claire Maternaghan: The Homer Home Automation System. ISSN 1460-9673, 2010.

Vuelo Autónomo con Programación Visual y Redes Neuronales Artificiales de un Parrot AR-Drone

Juan C. Rodríguez-Sánchez, Miguel A. Moreno-Martínez, Victor M. Landassuri-Moreno, José M. Flores Albino
Centro Universitario UAEM Valle de México
Universidad Autónoma del Estado de México C.P. 54400, México.
jcsylatp@yahoo.com, mluigiz12_z@hotmail.es, vmlandassurim@uaemex.mx, jose_martin_70@yahoo.com

Resumen— Este trabajo presenta una propuesta para desarrollar un algoritmo de vuelo autónomo, el cual es implementado en el lenguaje de programación visual, LabVIEW, como interfaz entre la computadora y un Drone de la marca Parrot (AR-Drone 1.0 o 2.0). Por otro lado, se cuenta con Redes Neuronales Artificiales, las cuales son generadas por un Algoritmo Evolutivo (para adaptarlas automáticamente), para hacer la clasificación de formas, las cuales son captadas por el Drone con una cámara frontal o bien, con una cámara inferior. Así, se tienen diversas figuras las cuales al ser captadas y reconocidas por el Drone, le permiten tomar acciones como despegar, aterrizar, girar, avanzar o retroceder. Como experimento particular, se repite el experimento del problema “What and Where” donde una red neuronal permite resolver dos problemas al mismo tiempo, es decir, donde está el objeto reconocido, y que objeto es; también se muestra experimentos realizados con programación visual (LabView) para la sección de segmentación y posible reconocimiento de imágenes. Los resultados obtenidos muestran clasificaciones precisas por parte de las Redes Neuronales para resolver el problema, lo que comprueba el problema del “What and Where”. El avance del proyecto se encuentra en etapas terminales y se espera pronto tener resultados definitivos.

Keywords—Clasificación, Control, UAV.

I. INTRODUCCIÓN

El uso de Drones ha ido en aumento en los últimos años, teniendo un sin fin de aplicaciones. En este trabajo se muestra una propuesta de vuelo autónomo para un Drone de la Marca Parrot 1.0 y 2.0, donde el objetivo es captar una imagen con la cámara frontal del Drone (o bien con la cámara inferior), para que él pueda tomar acciones y decisiones de forma autónoma. En un principio, el trabajo está propuesto para reconocer figuras básicas las cuales sirva de comandos para el Drone, y donde claramente está pensado a futuro para hacerlo con imágenes más complicadas, por ejemplo el poder esquivar a una persona u obstáculo y seguir la trayectoria planteada, imágenes con baja luz y lluvia entre otras.

En una primera instancia, es necesario captar la imagen y procesarla, es decir, segmentar la imagen para extraer la parte de interés [1-8]. Posteriormente, usar una técnica de reconocimiento de patrones para determinar el símbolo o carácter de interés, por ejemplo Redes Neuronales Artificiales (RNAs).

Para la captura de la imagen, su segmentación y extracción, se propone en este trabajo usar el lenguaje de programación visual LabVIEW, mientras que las Redes Neuronales Artificiales, adaptadas durante evolución, serán usadas para realizar el reconocimiento de caracteres. Donde en un principio, es reproducido el problema “What and Where” [9]. Así, el objetivo de este algoritmo es identificar que figura se encuentra en que posición, esto con una sola red neuronal que resuelva ambos casos.

El resto del artículo está repartido de la siguiente forma: La sección II muestra técnicas de segmentación de imágenes, la sección III describe el problema What and Where. La sección IV presenta LabView en el procesamiento de imágenes mientras que la Sección V muestra la configuración experimental; posteriormente, se muestra la sección VI con los resultados y la sección VI con las conclusiones alcanzadas hasta este momento.

II. PROCESAMIENTO DIGITAL DE IMÁGENES

En [1] se usa una técnica llamada Sliding Concentric Window para hacer la segmentación de la imagen y posteriormente enviarla a una RNA para realizar la tarea de clasificación. Otro trabajo similar es el presentado en [2], donde el objeto es dividido en varias regiones antes de proceder a la clasificación de caracteres.

También podemos encontrar publicaciones donde el objetivo es poder reconocer objetos, por ejemplo, de placas de automóviles, a distintas horas, climas, con obstáculos que interfieran la visibilidad, ángulos de toma, entre otros [3]. Otros trabajos se enfocan a hacer imágenes más nítidas para los algoritmos de clasificación, donde [4] es un claro ejemplo de ello.

En otro contexto, se tienen algoritmos evolutivos para generar programas los cuales automáticamente resuelvan una problemática, para nuestro interés, la segmentación de imágenes [5]. Así [6, 7] presenta una mejora al artículo anteriormente mostrado para poder hacer una selección de parámetros más adecuados. En su contraparte, [8] usa un algoritmo simple del mencionado anteriormente, para sentar las bases y proponer nuevas mejoras.

III. PROBLEMA WHAT AND WHERE

El problema de clasificación presentado por Rueckl et. al. [9] es un caso especial que pone a prueba el sistema visual básico, donde se tiene que identificar una serie de objetos, los cuales pueden aparecer en una variedad de posiciones.

Claramente se tienen dos tareas por resolver, el qué se está reconociendo (What) y en qué posición de la imagen se encuentra dicho objeto (Where). Previos experimentos de [9] demuestran que una red neuronal con nodos ocultos separados para cada tarea puede aprender de forma más rápida que si los nodos están mezclados entre ellos. Sin embargo, para casos prácticos de este trabajo, es interesante dado que el Drone puede ver objetos en diferentes posiciones. De ahí la necesidad de replicar dichos experimentos para poder determinar si es realmente útil para este trabajo.

También cabe destacar que el tener dos tareas presentes para una red neuronal, puede implicar que las entradas de una tarea inducen ruido en los nodos ocultos de la segunda tarea.

IV. LabVIEW

Como se comentó anteriormente, aquí es requerido el procesamiento de imagen con el objetivo de facilitar el reconocimiento de formas básicas capturadas mediante la cámara del Parrot A.R. Drone. Estas imágenes forman un conjunto de elementos que una vez procesadas, equivaldrán a instrucciones de encendido, avance frontal, giro y aterrizaje. Cada imagen contiene una forma (inicialmente las generadas para el problema what and where) donde una forma representa el encendido, y otra el avance, y así sucesivamente para todos los controles del Drone. Las formas son genéricas y pueden ser intercambiadas por otros símbolos o incluso números o letras. Donde cabe resaltar que las formas del What and Where son fácilmente distinguibles unas de otras. Por otro lado, se pretende que al final de este trabajo, se pueda determinar cual podría ser la mejor forma a usar para los comandos del Drone.

En diversos trabajos de la literatura se realiza la búsqueda de formas en ambientes con ruido, por ejemplo en [10] se busca una línea recta en un ambiente con variaciones de parámetros que afectan la identificación de formas simples mediante técnicas de detección de bordes. Estos algoritmos definen posiciones dentro del plano cartesiano mediante la Transformada Hough o el Filtro de Sobel que se basa en el método de los gradientes.

Por otra parte en [11] se aplican filtros de ruido para contrarrestar las variaciones en el procesamiento de imágenes, mediante el uso de un conjunto de técnicas matemáticas basadas en superficies de nivel formadas por pares (x, y) de tal manera que se obtienen funciones de la forma $u = u(x, y)$. A su vez se utilizan funciones de Sobolev y modelos de variación total. Lo anterior es debido a que en entornos con un elevado nivel de variación, el análisis computacional no es suficiente, para eliminar de manera eficiente el ruido presente en una imagen.

Otras técnicas se basan en la reconstrucción de imágenes que utilizan deconvolución mediante aproximaciones como filtros de Weiner para minimizar el error cuadrático medio, el método de Jansen-van Cittert y el de Maximum Likelihood Estimation para restauración de una imagen [12]. Como primera etapa, los experimentos se llevarán a cabo en un entorno cerrado y disminuyendo los niveles de ruido, i.e. sin obstáculos, o variaciones bruscas en los niveles de luminosidad. En la siguiente sección se muestran las herramientas disponibles en el entorno de programación de LabVIEW.

A. Herramientas de LabVIEW

Para llevar a cabo el tratamiento de imágenes, LabVIEW cuenta con diferentes barras de herramientas, las cuales incluyen la conversión de una imagen a matrices de unos y ceros (lógicas), manejo de umbrales, escalas de grises e histogramas, así como el reconocimiento básico de patrones que se basa en técnicas de resta de píxeles. Los formatos de imagen con los que puede trabajar LabVIEW son estándares, tales como mapas de bits, jpg, gif, png, entre otros. Para lograr lo anterior, se recurre al uso del toolbox IMAQ, que permite reducir tiempo y esfuerzo en lo que respecta a la programación mediante el uso de herramientas de procesamiento digital. En la Fig. 1 se muestra un ejemplo de código para almacenar una imagen tomada con una web-Cam en una carpeta.

Es necesario a su vez enlazar la cámara del Drone con la cámara web. En la Fig. 2 se muestra el código que permite la captura de imágenes usando la cámara del Parrot A.R. Drone.

En la Fig. 3, se muestra una imagen capturada mediante la cámara del drone y almacenada en un archivo mediante código de LabVIEW.

La imagen puede ser recuperada de la carpeta que la contiene para su procesamiento. Las técnicas utilizadas en la programación de estas herramientas son variadas y de acuerdo a [13] incluyen transformaciones de imágenes, detección de umbrales, Histogramas de imagen, Segmentación, Filtros de suavizado, Reconocimiento de patrones, entre otros.

Por otro lado, [14] muestra como una imagen puede visualizarse como una función que a su vez debe ser digitalizada espacialmente en sus coordenadas (x, y) mediante técnicas de muestreo y también en amplitud modificando escalas de grises. Al ser una imagen una matriz de puntos, se pueden realizar sobre ella diferentes operaciones. Por ejemplo, para éste proyecto, se tiene contemplado el uso de Segmentación para extraer de toda la imagen captada, una sección útil y de interés para el Drone, es decir, el área que contiene al comando de interés para este caso. También, se considera la Binarización, la cual elimina toda la información no necesaria y la deja en un formato al similar al What and Where, donde la imagen a reconocer es binaria (similar a la Fig. 3). Posterior a la Binarización, es requerido redimensionar la imagen para dejarla al tamaño de entrada de la RNA que haya sido designada para tal tarea. Estas operaciones se describen a continuación.

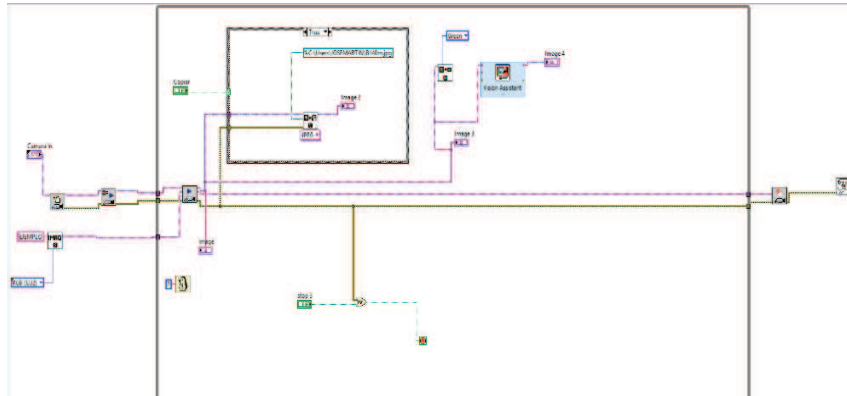


Fig. 1. Captura de imagen vía cámara web

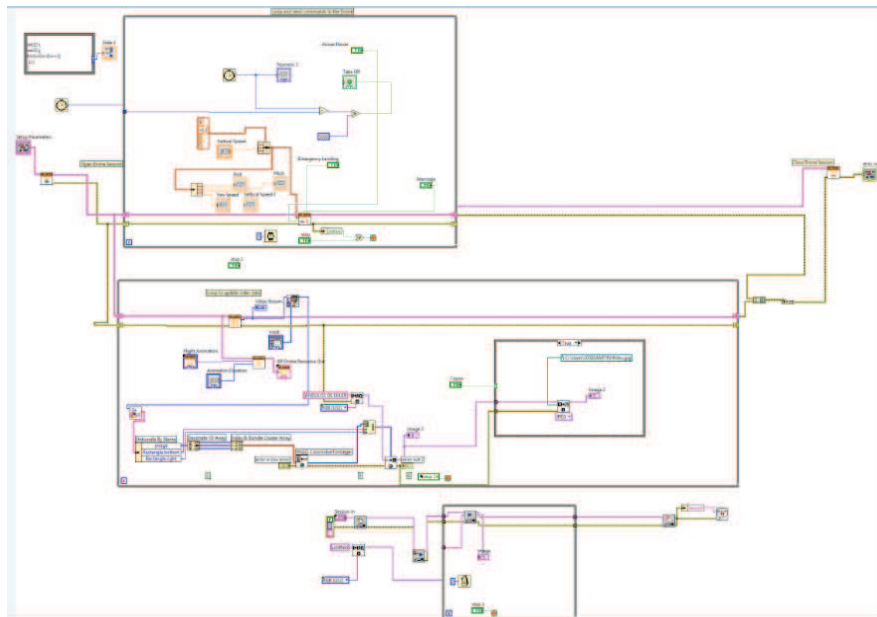


Fig. 2. Código para captura de imagen mediante la cámara del Parrot A.R. Drone



Fig. 3. Imagen de una forma simple del problema What and Where representada por bordes rectangulares capturada con la cámara del Parrot A.R. Drone

B. Detección de Umbral

Una vez recuperada la imagen, de la carpeta, se puede aplicar un umbral mediante la herramienta Threshold, tal y como se muestra en la Fig. 4.

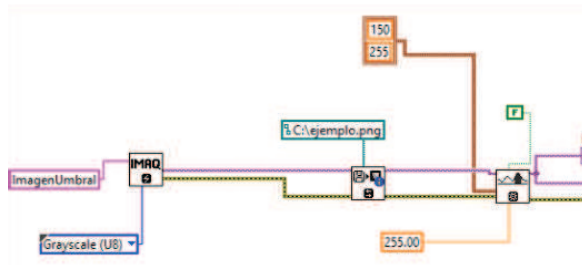


Fig. 4. Aplicación del Filtro de umbral

C. Conversión a escala de grises

El procesamiento de la imagen requiere de ciertas operaciones previas, tal es el caso de la conversión a escala de grises. En la Fig. 5 se muestra una imagen a la cual se le aplicó esta operación.



Fig. 5. Ejemplo de una imagen en escala de grises.

D. Segmentación

Consiste en la separación de una imagen digital en grupos de píxeles mediante el uso de técnicas como la detección de discontinuidades, la detección de puntos, líneas y bordes [3-6].

E. Binarización

Para binarizar la imagen (mostrar su equivalente en formato de Unos y Ceros), es conveniente utilizar herramientas para obtener umbral, ya que una operación que facilita la conversión de la imagen a formato digital es precisamente la de obtención del umbral que se hace mediante herramientas de LabVIEW (se coloca enseguida la imagen de la herramienta de threshold). En la Fig. 6 se muestra la imagen binarizada.

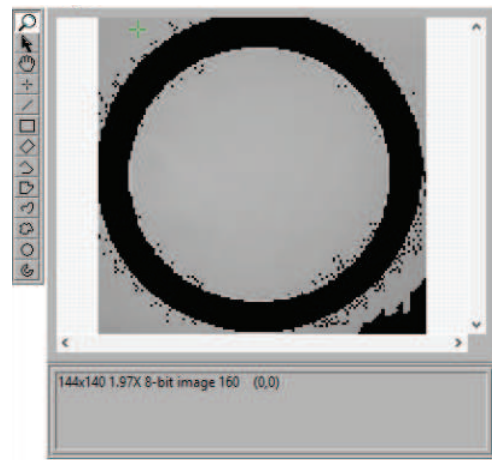


Fig. 6. Obtención de los píxeles de la imagen capturada mediante cámara

En la Fig. 7 se muestra el código para obtener los píxeles en blanco y negro de la imagen.

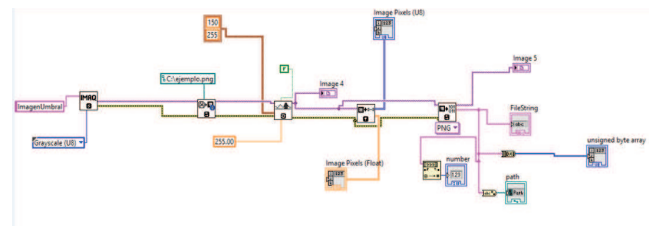


Fig. 7. Código LabVIEW para obtención de píxeles

En la Fig. 8 se muestra la imagen cuyos píxeles han sido convertidos a ceros. Cabe señalar que para procesar los bits se tienen contempladas herramientas de Matlab y/o Python, en caso de no obtener buenos resultados con LabVIEW.



Fig. 8. Pixeles de Imagen convertidos a ceros y unos.

F. Primer Experimento de reconocimiento de imagen para encendido del Parrot A.R Drone

Dado los controles anteriormente mostrados, se genero un primer experimento para ver las prestaciones que se pueden obtener con éste lenguaje visual. En lo que se refiere al reconocimiento de formas, LabVIEW cuenta con un conjunto de herramientas para formas, colores, tamaños, entre otras características dentro de una imagen. En el primer experimento de reconocimiento de formas se logró encender el Parrot A.R. Drone enviando un bit de encendido una vez que se reconoce la forma mediante la cámara. En la Fig. 9 se muestra el código generado.

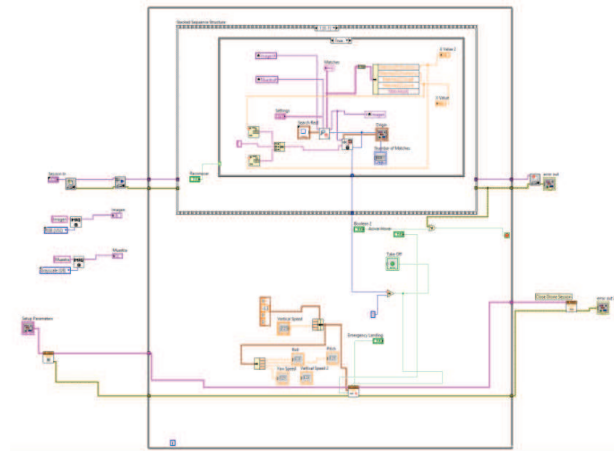


Fig. 9. Código para el reconocimiento de forma circular

En la interfaz de usuario se especifica la cámara y al seleccionar el botón de Reconocer, se activa la señal que enciende al Drone. La Interfaz de usuario se muestra en la Fig. 10.

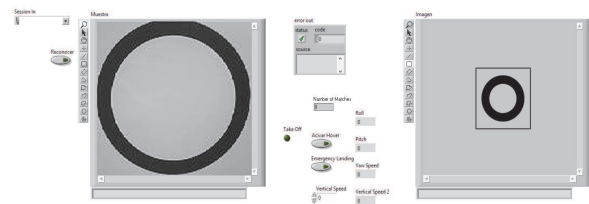


Fig. 10. Interfaz de Usuario para reconocimiento de forma a través de cámara y encendido del Drone

V. CONFIGURACIÓN EXPERIMENTAL

Para la parte de las RNAs, se realizaron 10 ejecuciones independientes con un algoritmo evolutivo para generar RNAs, las cuales resolvieran el problema del What and Where, durante 100 generaciones de evolución como máximo. La población del algoritmo evolutivo fue de 20 individuos y en las entradas de las redes fueron fijadas así como el números de salidas, dado que la tarea de calificación no cambia (no es dinámica).

Así, solo se permitio evolucionar a los nodos ocultos y la conectividad de ellos (conexiones) para brindar mayor flexibilidad en la búsqueda de la solución, la cual, en este caso, es una arquitectura adecuada, con un entrenamiento que birnde generalización, que resuelva el problema.

VI. RESULTADOS

La Fig. 11 muestra la evolución de las conexiones promedio de las redes neuronales artificiales en la población, las cuales son evolucionadas con un algoritmo evolutivo. Como se puede observar existe una gran cantidad de conexiones al inicio y éstas van en aumento. Hay que considerar que este problema cuenta con una matriz de entrada de 8x8, dando 64 neuronas de entrada y 18 salidas, las primeras 9 para el What y las restantes para el Where. El mismo comportamiento puede observarse para la Fig. 12 donde los nodos ocultos tambien van en aumento. A pesar de ello, se nota claramente que el error de la red neuronal (Fig. 13) decrementa al aumentar los nodos.

La Fig. 14 muestra una ejecución independiente, para el peor, promedio y mejor individuos de la población, donde desde generaciones muy tempranas, el mejor individuo llega a un error de cero, y alrededor de la generación 60, el resto de la población alcanza un error similar.

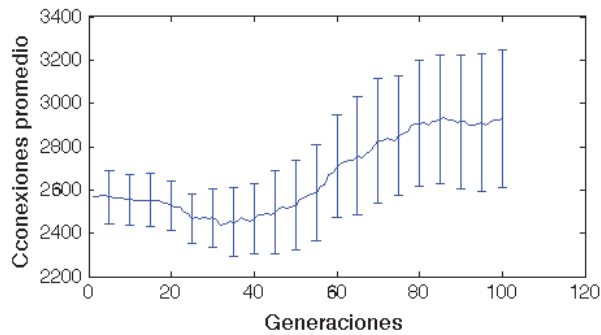


Fig. 11. Conexiones promedio de 10 repeticiones independientes para el problema what and where

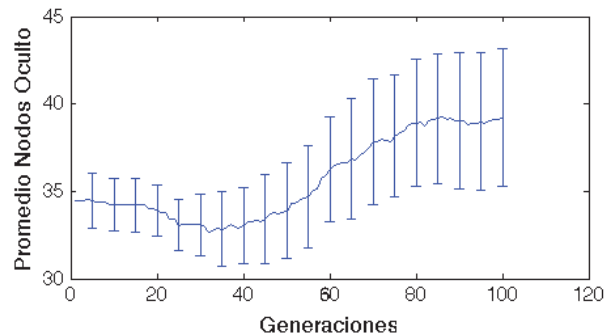


Fig. 12. Nodos ocultos en promedio para de 10 repeticiones independientes para el problema what and where

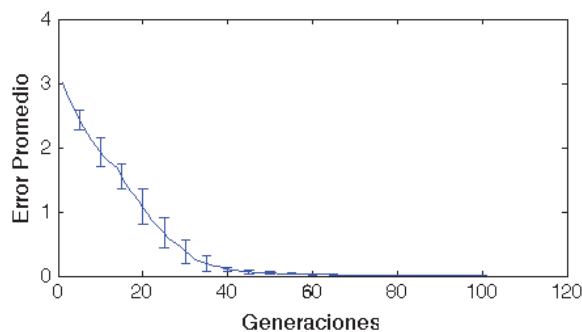


Fig. 13. Error promedio de 10 repeticiones independientes para el problema what and where

VII. CONCLUSIONES

En este trabajo se presentó la propuesta de análisis de imágenes mediante el uso conjunto de técnicas de programación visual y evolución de Redes Neuronales Artificiales.

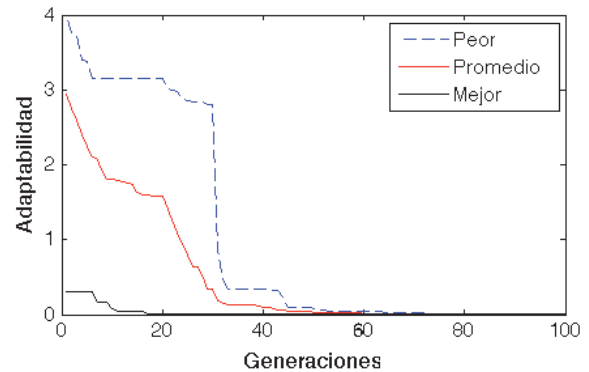


Fig. 14. Mejor, promedio y peor individuos para una ejecución independiente del problema what and where

De ahí, se concluye que el algoritmo What and Where puede ser aplicado sin mayor problema a éste proyecto para ser usado con el Drone durante el procesamiento de imágenes obtenido a través de cámaras, esto para lograr el control de Hardware en base a los experimentos realizados.

En una segunda etapa del proyecto, se buscará eliminar el ruido provocado por variaciones en el entorno ya sea aplicando las técnicas de filtrado de imagen mencionadas aquí u otras como las señaladas en [15], donde se menciona que las RNAs son eficientes para operaciones de segmentación de imágenes irregulares, que con otros métodos serían difíciles de procesar.

Se buscará también mejorar la adaptabilidad del Drone en entornos cambiantes, aplicando por ejemplo, técnicas que involucran el diseño de Arquitecturas Cognitivas mediante el uso de RNAs para la navegación autónoma de robots [16].

Se espera que a mediano plazo y del seguimiento de los resultados aquí obtenidos, las RNAs al llevar a cabo el procesamiento de imagen para detección de formas mediante el entrenamiento supervisado, provean de las herramientas adecuadas para lograr el vuelo autónomo del Parrot A.R. Drone en un entorno cerrado.

AGRADECIMIENTOS

Se agradece el tiempo de cómputo brindado por parte del Clúster del Centro Universitario UAEM Valle de México (CLU-CUUAEM-VM), de la Universidad Autónoma del Estado de México (UAEM) para llevar a cabo los experimentos de éste trabajo.

REFERENCIAS

- [1] Kaushik Deba, Md. Ibrahim Khana, Anik Saha, and Kang-Hyun Job. An Efficient Method of Vehicle License Plate Recognition Based on Sliding Concentric Windows and Artificial Neural Network. *Procedia Technology* 4 (2012) 812 – 819.
- [2] H.Erdinc Kocera, K.Kursat Cevikb. Artificial neural networks based vehicle license plate recognition. *Procedia Computer Science* 3 (2011) 1033–1037.

- [3] Fikriye Öztürka, Figen Özena. A New License Plate Recognition System Based on Probabilistic Neural Networks. *Procedia Technology* 1 (2012) 124 – 128.
- [4] Xiaofeng Zhang, Fengchang Xu, Yan Su. Research on the License Plate Recognition based on MATLAB. *Procedia Engineering* 15 (2011) 1330 – 1334
- [5] Henrik Berga,, Roland Olssona, Thomas Lindblad, José Chiloc. Automatic design of pulse coupled neurons for image segmentation. *Neurocomputing* 71 (2008) 1980–1993
- [6] Chao Gao, Dongguo Zhou n, Yongcai Guo. Automatic iterative algorithm for image segmentation using a modified pulse-coupled neural network. *Neurocomputing* 119 (2013) 332–338.
- [7] Yunfeng Lu , Jun Miao, Lijuan Duan, Yuanhua Qiao, Ruixin Jia. A new approach to image segmentation based on simplified region growing PCNN. *Applied Mathematics and Computation* 205 (2008) 807–814.
- [8] Shuo Wei, Qu Hong, Mengshu Hou. Automatic image segmentation based on PCNN with adaptive threshold time constant. *Neurocomputing* 74 (2011) 1485–1491.
- [9] Rueckl, Jay G. and Cave, Kyle R. and Kosslyn, Stephen M. Why are What and Where' Processed by Separate Cortical Visual Systems? A Computational Investigation. *Journal of Cognitive Neuroscience* 1(2). 171-186, 1989.
- [10] G.Bergues, G.Ames, L.Canali, C. Schurrer and A.G. Flesia. Detección de líneas en imágenes con ruido en un entorno de medición de alta precisión. 3-4 (2014)
- [11] Carlos Brito Loeza, Víctor Uc Cetina, Anabel Martín-González. Introducción a los Métodos Variacionales en Procesamiento de Imágenes: Filtros de Ruido. 2,3,11 (2014).
- [12] José M. Cecilia, Francisco J. Moreno, José M. García. Acelerando algoritmos de deconvolución para imágenes de microscopía usando GPUs. *Proceedings of the II Workshop en Aplicaciones de Nuevas Arquitecturas de Consumo y altas prestaciones (Madrid, Spain, 2009), ANACAP09, URJC1, CD-ROM.*
- [13] Rubén Posada-Gómez, Oscar Osvaldo Sandoval-González, Albino Martínez Sibaja, Otniel Portillo-Rodríguez and Giner Alor-Hernández. *Digital Image Processing Using LabView.* 299-311
- [14] Fu, K. S., R.C. González, C.S.G. Lee. *Robótica: Control, Detección, Visión e Inteligencia.* New York: McGraw-Hill. 312-315
- [15] Ramírez Q. Juan A., Chacón M. Mario. *Redes Neuronales Artificiales para el procesamiento de imágenes, una revisión de la última década.* 1,13,14 (2011)
- [16] Análisis del Comportamiento de Robots Móviles con RNA. Un acercamiento desde el paradigma reactivo. Alejandro Hossian, Gustavo Monte, Verónica Olivera. 19-23 (2013).

Revisión de la transición del protocolo IPv4 a IPv6 en Latinoamérica

Miriam Olvera Cueyar¹, Alma Delia Vite Rojo

¹Cuerpo académico Desarrollo de Software Aplicado
Universidad Politécnica de Tulancingo
Tulancingo, Hgo. México
miriam.olvera@upt.edu.mx, alma.vite@upt.edu.mx

Luis Bryan Oroxón Ramírez

Estudiante de Ingeniería en Sistemas Computacionales
Universidad Politécnica de Tulancingo
Tulancingo, Hgo. México
bryan.oroxon@gmail.com

I. INTRODUCCIÓN

El internet ya es una parte de nuestras vidas, somos una sociedad global, nos podemos conectar de diferentes maneras, para satisfacer necesidades de comunicación, de información, entretenimiento, etc. Cuando se creó el internet se confió en el protocolo IPv4, con una capacidad de 4.29 billones de direcciones IP, pero en menos de 10 años la demanda de direcciones superó las expectativas de crecimiento, el agotamiento de direcciones IPv4 para asignar se alcanzó en menos tiempo del que se esperaba, para solventar esta carencia y resolver las necesidades exponenciales de direcciones, se creó el protocolo, IPv6. "El protocolo de Internet versión 6, es una tecnología que entrará poco a poco a reemplazar a la versión 4, su desarrollo contribuye en la búsqueda de soluciones a las problemáticas existentes en IPV4, está fue diseñada como la evolución del IPv4 presentando características de seguridad y mejor funcionamiento." [1] [2]

II. ANTECEDENTES

Internet está basada en el protocolo de internet, desde su inicio comercial se ha utilizado la versión 4 de dicho protocolo IPv4.

Este protocolo fue diseñado como un experimento, en su momento no se llegó a pensar que llegarían a agotarse sus direcciones únicas. IPv4 utiliza direcciones de 32 bits, con lo cual se permite direccionar de forma única 2^{32} (4,294,967,296) direcciones.

En sus inicios, el internet con IPv4 pasa por una etapa inicial en la que se entregaban direcciones a cualquier persona o entidad que las solicitaba, sin mayores justificaciones.

En una segunda etapa, se organiza una entidad IANA (Internet Assigned Numbers Authority), actualmente parte de ICANN (Internet Corporation for assigned Names and Numbers), conjuntamente con los registros regionales (RIRs o Regional Internet Registers), son los encargados de la asignación de recursos de numeración (Direcciones IP).

En 1992, el IETF, observa que es necesario "ampliar" el número de direcciones, debido al inesperado crecimiento del

internet, en ese momento las direcciones disponibles no permitían asignar una única dirección a cada persona del planeta y mucho menos asignar direcciones para varios dispositivos a una persona.

El crecimiento de internet es exponencial, donde antes sólo se conectaban algunas computadoras ahora se conectan cientos, miles y millones de dispositivos de todo tipo.

Para permitir el crecimiento continuo de internet, se crea el mecanismo NAT (Network Address Translation), que permite que cada red conectada a internet, ya sea un usuario o una organización, utilice una dirección IPv4 pública en su punto de conexión a internet, y se conecte a un número determinado y limitado de direcciones IP privadas en el interior de dicha red. NAT realiza una traducción de las múltiples direcciones privadas a la única dirección IP pública del punto de conexión.

Aún con el uso de NAT, el crecimiento continuo y explosivo de internet a llevado a un agotamiento de dicho espacio de direcciones en almacenamiento central IANA.

El Agotamiento final de este recurso, se produce por una petición de nuevos bloques por parte de APNIC (concretamente 2 prefijos /8), en febrero de 2011. En ese momento se reparte los últimos 5 prefijos /8 (ya que así lo indicaba una política global) entregándose uno de ellos a cada una de las 5 regiones. Bajo la demanda de ese momento las regiones de RIPE, ARIN y APNIC, agotarían sus remanentes en 6 u 8 meses. LACNIC y AfriNIC posiblemente tendrían recursos para unos 12 meses adicionales. [1]

III. ¿QUÉ ES IPV6?

Internet Protocol versión 6 (IPv6) fue diseñado para reemplazar a Internet Protocolo versión 4 (IPv4) por IETF (Grupo de Trabajo de Ingeniería de Internet).

Debido a la falta de direccionamiento IP, se desarrolló IPv6, que a inicios de su creación se llamó como: IPng (Protocolo de Internet de Nueva Generación), que promete dar solución a

los problemas del direccionamiento, dándole un mejoramiento en capacidad del envío de la información, la seguridad, la facilidad y el rendimiento en los equipos.

La nueva versión del protocolo usa direcciones de 128 bits lo cual equivale a tener $2^{128} = 340.283.366.920.938.463.374.607.431.768.211.456$ direcciones IP, que es representado en formato hexadecimal, esta cantidad de nuevas direcciones, podrán ser utilizadas por miles de millones de usuarios que requieran servicios en las diferentes plataformas que necesitan las direcciones IP como, las páginas Web, los dispositivos móviles como teléfonos celulares, PDA's, dispositivos de consumo, vehículos, etc. De esta manera esta versión del protocolo permite solucionar el grave problema de direccionamiento que hoy en día se debe enfrentar con la IPv4. [1]

A. ¿Por qué se da la transición de IPv4 a IPv6?

Para satisfacer las necesidades de escasas direcciones IPv4 en el mundo y demanda de mejores servicios en el protocolo de internet en nivel de seguridad y configuración.

B. Características principales de IPv6 con respecto a IPv4

Las características principales de IPv6 son:

1) Capacidades de direccionamiento extendida

“IPv6 incrementa el tamaño de dirección IP de 32 bits a 128 bits, para dar soporte a más niveles de direccionamiento jerárquico, un número mucho mayor de nodos direccionables, y una autoconfiguración más simple de direcciones. La escalabilidad del enrutamiento multienvío se mejora agregando un campo "ámbito" a las direcciones multienvío. Y se define un nuevo tipo de dirección llamada "anycast", usada para enviar un paquete a cualquiera de un grupo de nodos. [1]

2) Simplificación del formato de cabecera

Se modificó la cabecera de IPv4, en IPv6 es más simple, pues posee menor cantidad de campos, sus estructuras y contenidos han sido mejorados; optimizando los recursos que utiliza, pues se han eliminado algunos campos repetitivos que ya se representaban anticuados, incrementando algunas características para hacer frente a las nuevas necesidades de las redes actuales, como comunicación en tiempo real y seguridad. [1]

3) Soporte mejorado para las extensiones y opciones

Los cambios en la manera en que se codifican las opciones de la cabecera IP permiten un reenvío más eficiente, límites menos rigurosos en la longitud de opciones y mayor flexibilidad para introducir nuevas opciones en el futuro. [1]

4) Capacidad de etiquetado de flujo

Una nueva capacidad se agrega para permitir el etiquetado de paquetes que pertenecen a "flujos" de tráfico particulares para lo cual el remitente solicita tratamiento especial, como la calidad de servicio no estándar o el servicio en "tiempo real". [1]

5) Capacidades de autenticación y privacidad

Extensiones para utilizar autenticación, integridad de los datos, y (opcional) confidencialidad de los datos, se especifican para el IPv6. [1]

6) Características de movilidad.

IPv6 permite que la comunicación pueda llevarse a cabo en cualquier momento y lugar con un óptimo grado de operatividad, así como de forma transparente al usuario, permitiéndole realizar su propia gestión y control. Cuestiones de gran importancia si queremos disfrutar de servicios multimedia en los terminales móviles de última generación (VozIP y Video). Protocolos como MPI (Mobile IP) o HMIP (Hierarchical MIP) posibilitan la implantación y explotación real de estos servicios. [1]

7) Autoconfiguración de los nodos.

La autoconfiguración de direcciones es más simple. Especialmente en direcciones Globales Unicast, los 64 bits superiores son asignados por un mensaje desde el router (Router Advertisement) y los 64 bits más bajos son asignados con la dirección MAC (en formato EUI-64). En este caso, el largo del prefijo de la subred es 64, por lo que no hay que preocuparse más por la máscara de red. Además el largo del prefijo no depende del número de hosts por lo tanto la asignación es más simple. [1]

8) Calidad de servicio y clases de servicios.

Capacidad de una red para sostener un comportamiento adecuado del tráfico que transita por ella, cumpliendo con parámetros relevantes para el usuario final. IPv6 fue diseñado con soporte extendido a QoS. En el encabezamiento se han incluido dos campos relacionados a QoS, éstos son: clase de tráfico e identificador de flujo. [1]

9) Multihoming

Es la facilidad que se da a las empresas o instituciones que desean realizar el cambio de un proveedor a otro por algún motivo, no necesitaría cambiar de dirección, ni la realización de una nueva configuración de los equipos de comunicación. Es decir cumple con la facilidad de realizar el cambio de ISP's (Proveedor de Servicios de Internet). [1]

IPv4

Presenta soporte inadecuado para las aplicaciones del siglo XXI:

Las nuevas aplicaciones son más demandantes, requieren garantías en:

- Los tiempos de respuesta.
- La disponibilidad de ancho de banda.
- Seguridad.
- Difícil de adecuar a las nuevas aplicaciones.
- En IPv4 la seguridad es opcional, no fue diseñado para ser seguro. [3]

IV. ¿COMO SE VA A IMPLEMENTAR LA TRANSICIÓN?

No se trata de una migración como erróneamente se indica en muchas ocasiones, sino que ambos protocolos, IPv4 e IPv6, existirán durante algún tiempo, es decir se produce una coexistencia.

Ya que Internet se implementó en su totalidad en el protocolo IPv4. El Internet está funcionando en todo momento, no puede ser apagado, no sería posible su sustitución inmediata, es decir no es posible apagar la red, ni siquiera por un par de minutos para realizar la transición a IPv6.

La actualización no es posible de forma instantánea, es necesario que se actualicen los equipos, que los proveedores y organizaciones se preparen para este cambio.

Para este propósito, la organización encargada de la estandarización de los protocolos de Internet (IETF, Internet Engineering Task Force), diseñó junto con el propio IPv6, una serie de mecanismos que llamamos de transición y coexistencia.

Los mecanismos de transición y coexistencia permiten que ambos protocolos, IPv4 e IPv6, existan durante algún tiempo, es decir se produce una coexistencia.

La mayoría de contenidos existentes son IPv4, pero poco a poco, gracias a esta coexistencia, conforme más contenidos y servicios estén disponibles con IPv6, IPv6 será predominante. Esto es lo que llamamos la transición.

El diseño del protocolo IPv6 da preferencia al contenido de paquetes IPv6 frente a paquetes IPv4, si ambos están disponibles.

Es importante recordar que el 3 de Febrero de 2011 se agotaron las direcciones IPv4 en el registro central de IANA (Autoridad de Asignación de Números en Internet), por lo que los proveedores de servicios de Internet están acelerando el despliegue de IPv6 en sus redes para que tanto los nuevos usuarios como los existentes sigan disfrutando de un uso

habitual y continuo de Internet con las nuevas características que ofrece IPv6.

V. CARACTERÍSTICAS DE IPV6

- Arquitectura jerárquica de direcciones
- Autoconfiguración de direcciones
- Seguridad e integridad de datos
- Calidad del servicio, QoS.
- Soporte a tráfico multimedia en tiempo real.
- Computación móvil.
- Aplicaciones multicast y unicast.
- Mecanismos de transición gradual de IPv4 a IPv6.

Las direcciones IPv6 se asignan a interfaces lógicas, una interfaz puede tener muchas direcciones. Las direcciones tiene ámbito de acción:

[3]



Fig. 1 Tipos de direcciones IPv6. [3]

A. Tipos de direcciones IPv6

Unicast.
Anycast.
Multicast.

No hay broadcast en IPv6. [3]

VI. TRANSICIÓN DE IPV4 A IPV6

Podemos hacer una clasificación general entre los mecanismos de transición de acuerdo al tipo de técnica que se utiliza: [4]

- Dual stack
- Túneles
- Traducción

A. Dual stack:

Es el método propuesto originalmente para tener una transición suave hacia IPv6. Requiere que los hosts y los

routers soporten ambas versiones de IP y, por tanto, servicios y aplicaciones tanto IPv4 como IPv6. Actualmente es el mecanismo principal que permite introducir IPv6 en arquitecturas IPv6. Su desventaja es que necesita contar con direcciones IPv4 que se están agotando para funcionar.

En esta red, operan de forma simultánea IPv4 e IPv6. En una red dual stack, ambos protocolos son desplegados completamente y los protocolos de enrutamiento deben llevar los prefijos correspondientes a cada tecnología, de manera transparente. [4] [5]

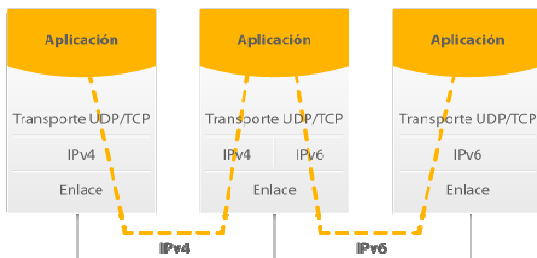


Fig. 2 Dual Stack. o pila doble. [4] [5]

B. Túneles/Encapsulamiento

Utilizando esta técnica, se pueden enviar paquetes IPv6 dentro de paquetes IPv4, el tráfico IPv6 viaja encapsulado en paquetes IPv4. Permite de esta forma atravesar redes que no manejan IPv6, pero también podemos encontrar la situación inversa. Los paquetes originales son transportados hasta un punto de la red por medio del protocolo original, luego encapsulados para atravesar la porción de red que no lo soporta y luego son des-encapsulados en el otro extremo para ser enviados al destino final en forma nativa. Los túneles más habituales son los túneles manuales y los túneles automáticos. Los túneles manuales se deben configurar explícitamente en algún equipo de la red, mientras que en el segundo caso se configuran automáticamente en algunos sistemas operativos. En el caso de los primeros, podemos citar los túneles manuales entre dos equipos o mediante "tunnel brokers". En el segundo caso, los más conocidos son 6to4 y Teredo. [4] [5]

C. Traducción

Esta técnica consiste en utilizar algún dispositivo en la red que convierta los paquetes de IPv4 a IPv6 y viceversa. Ese dispositivo tiene que ser capaz de realizar la traducción en los dos sentidos de forma que se permita la comunicación.

Dentro de esta clasificación podemos mencionar NAT64/DNS64: la red es IPv6 nativa y para llegar a sitios que son sólo IPv4 se realiza una traducción al estilo NAT, mediante un mapeo entre los paquetes IPv6 e IPv4. Se utiliza un prefijo especial para mapear direcciones IPv4 a IPv6: 64:ff9b::/96. [4] [5]

Es necesario también utilizar una modificación al DNS, llamada DNS64, que permite generar un registro AAAA aún cuando el destino no tenga dirección IPv6 (es decir, el DNS responde sólo con registros de tipo A). [4] [5]

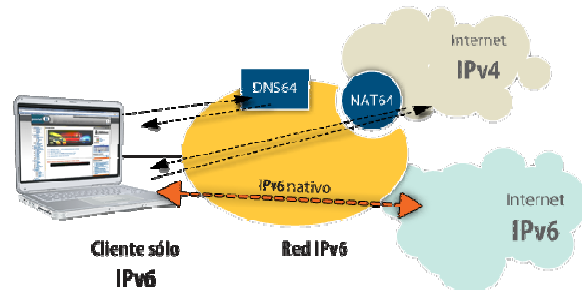


Fig. 3 Traducción. [4] [5]

En el inicio del Internet en 1983 existían disponibles 3.7 billones de direcciones disponibles para asignar, sin embargo estas direcciones en el año 2014, en su mayoría ya fueron asignadas, este agotamiento se muestra en la Fig. 4, crecimiento de IPv4.



Fig. 4 Crecimiento de IPv4. [6]

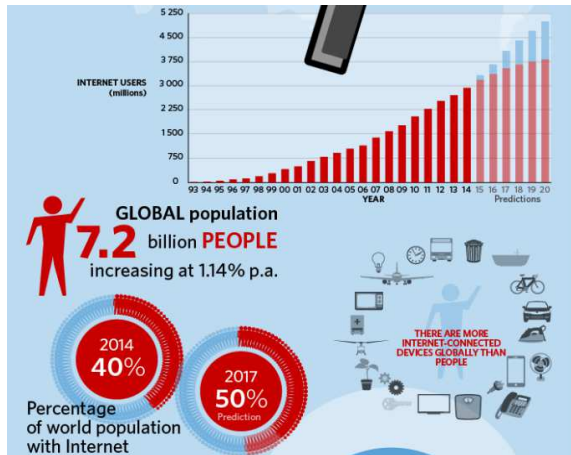


Fig. 5 Estimación del crecimiento de la población para 2017. [6]

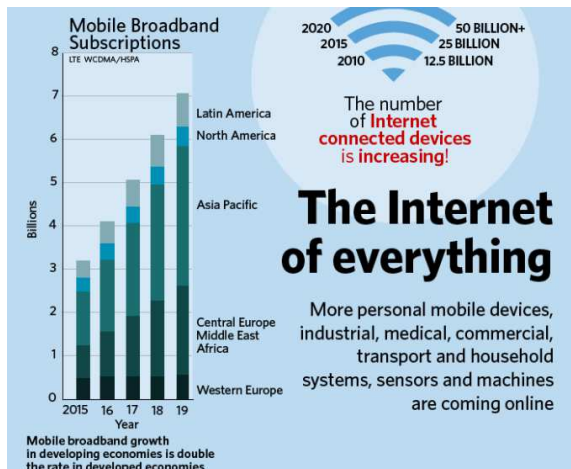


Fig. 6 Crecimiento de dispositivos móviles para el 2020. [6]

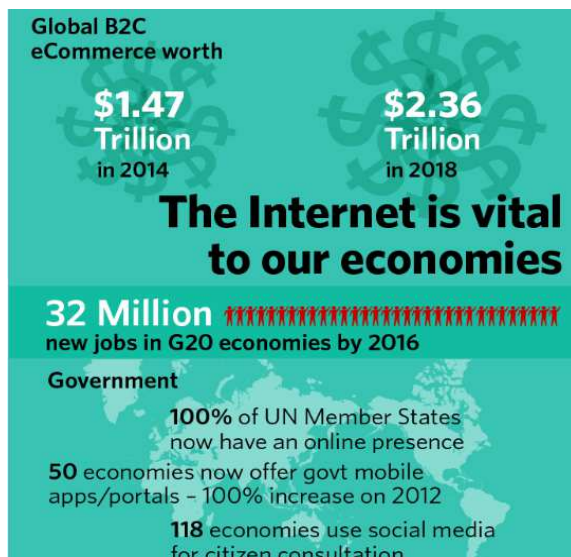


Fig. 7 eCommerce. [6]

VII. TRANSICIÓN DE IPV6 EN LATINOAMÉRICA

A. Regional Internet Registries

Un Registro Regional de Internet o Regional Internet Registry (RIR) es una organización que supervisa la asignación y el registro de recursos de números de Internet dentro de una región particular del mundo. Los recursos incluyen direcciones IP (tanto IPv4 como IPv6) y números de sistemas autónomos (para su uso en encaminamiento BGP).

Hay actualmente 5 RIRs en funcionamiento: [7]

1. American Registry for Internet Numbers (ARIN) para América Anglosajona.
2. RIPE Network Coordination Centre (RIPE NCC) para Europa, el Oriente Medio y Asia Central.
3. Asia-Pacific Network Information Centre (APNIC) para Asia y la Región Pacífica.
4. Latin American and Caribbean Internet Address Registry (LACNIC) para América Latina y el Caribe.
5. African Network Information Centre (AfrinIC) para África

Las direcciones restantes de los RIR de su Pool /8 son para APNIC: 0.7541, RIPE NCC: 1.0276, LACNIC: 0.1868, ARIN: 0.2942, AFRINIC: 2.6330. Estos datos se pueden observar en la fig. 8.

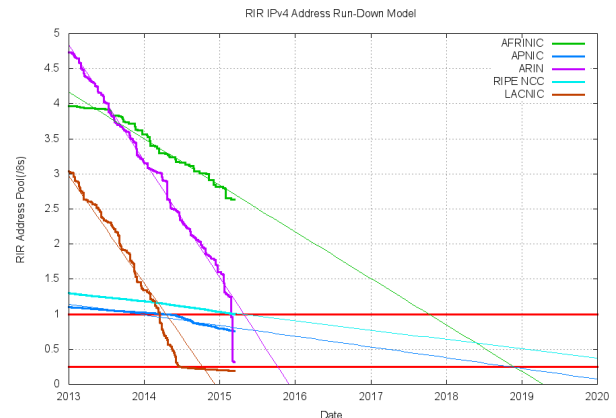


Fig. 8 Proyección del consumo de restos de los Pools de Direcciones RIR. [8]

B. Proporción de prefijos enrutables en Latinoamérica IPv6

PAÍS	Prefijos enrutables IPv6
Canadá	46.59%
Estados Unidos	37.72%
México	34.48%
Guatemala	15.79%
Belice	20%
El Salvador	15.38%
Honduras	43.27%
Nicaragua	22.22%
Costa Rica	32.79%
Panamá	14.63%
Colombia	27.03%
Venezuela	43.48%
Guyana	33.33%
Surinam	0%
Guyana Francesa	0%
Brasil	20.59%
Ecuador	33.33%
Perú	22.22%
Bolivia	21.05%
Paraguay	13.64%
Uruguay	41.67%
Chile	17.20%
Argentina	14.38%
Cuba	33.33%
Bahamas	0%
Jamaica	16.67%
Haití	0.00%
República Dominicana	23.81%
Puerto Rico	38.46%
Cuba	33.33%
Bahamas	0%
Jamaica	16.67%
Haití	0%

Datos recabados de APNIC al 27 de abril de 2015 indican que el porcentaje de usuarios IPv6 en Latinoamérica es del 0.1469% y el índice de adopción IPv6 global es de 3.2418% [9].

La proporción de adopción de IPv6 se muestra en el mapa con la Fig. 9 IPv6 en Latinoamérica. [10].



Fig. 9 IPv6 en Latinoamérica. [10]

Actualmente existen 7 grupos de trabajo constituidos en América Latina y el Caribe [11]

Cuba <http://www.cu.IPv6tf.org/>
 Brasil <http://www.br.IPv6tf.org/>
 México <http://www.mx.IPv6tf.org/>
 Perú <http://www.pe.IPv6tf.org/>
 Argentina <http://www.ar.IPv6tf.org/>
 Colombia <http://www.co.IPv6tf.org/>
 Panamá <http://www.pa.IPv6tf.org/>

VIII. TRANSICIÓN DE IPV6 EN MÉXICO

La UIT ha adoptado medidas en varios foros para fomentar la transición del protocolo IPv4 al protocolo IPv6. La Resolución 180 de la Conferencia de Plenipotenciarios, aprobada en Guadalajara (México) en 2010, exhorta a que se desplieguen esfuerzos para facilitar la transición de IPv4 a IPv6.

La UIT ha adoptado medidas en varios foros para fomentar la transición del protocolo de Internet versión 4 (IPv4) al protocolo de Internet versión 6 (IPv6). México aprobó el 8 de junio del 2011, el nuevo protocolo IPv6 para combatir el problema de escasas direcciones.

El 6 de junio 2012 en México los dos proveedores que en conjunto concentran el 80 por ciento del tráfico de internet (Alcatel y Telmex) están listos en su totalidad para ofrecer servicios del protocolo (IPv6). [12]

A. La Universidad Nacional Autónoma de México (UNAM)

Inició investigaciones en la materia desde el mes de diciembre de 1998, fecha en la que se constituye el proyecto IPv6, y durante el segundo semestre del año 1999 la UNAM es líder en el ámbito nacional. Dentro del proyecto IPv6 se estableció un amplio programa de pruebas y trabajos con temas como: implementaciones, stacks IPv4/IPv6, túneles, software de conexión, aplicaciones multimedia, servidores para Web y DNS, autoconfiguración, calidad de servicio, IPv6 sobre ATM, conexión con redes internacionales de IPv6 (6Bone, 6REN), IPv6 en Internet2, etc.

Dentro de las primeras pruebas realizadas, destaca la de conexión a 6Bone, la cual fue una red mundial experimental utilizada para probar los conceptos y la puesta en operación de IPv6. Al final en 6Bone participaron en el ámbito mundial 47 países, entre ellos México, donde la UNAM fue el primer nodo en el país, registrándose en junio de 1999.

Posteriormente en septiembre de 1999 fue aceptada como uno de los 68 nodos de Backbone que en esa fecha operaban en 6Bone, obteniendo un rango de direcciones tipo pTLA: 3ffe:8070::/28. Cabe destacar que con este hecho la UNAM fue el primer nodo, y hasta ese momento el único, de este tipo en México, y el tercero en Latinoamérica. Adicionalmente ha podido delegar direcciones y configurar túneles a instituciones en México y en el mundo interesadas en realizar pruebas con IPv6.

En octubre del 2000 se obtuvo un bloque del tipo sTLA (2001:0448::/35), adjudicado por ARIN, la entidad de registro para norteamérica y que en aquel entonces daba servicio también a Latinoamérica, que se ha utilizado por ejemplo en la RedCUDI, la red de Internet2 de México.

Posteriormente, en junio de 2005 se obtiene otro bloque de direcciones IPv6 adjudicado por LACNIC, la entidad de registro para Latinoamérica y el Caribe.

Para contar con una red de pruebas en una primera etapa, y posteriormente con una red de producción, se instaló la Red IPv6 de la UNAM, la primera red IPv6 instalada en México y que inició operaciones en agosto de 1999. Esta red contó con varios túneles hacia otros nodos de Backbone de 6Bone: SPRINT, FIBERTEL, MERIT, BAY NETWORKS, JANET e ISI-LAP, y hacia los hosts que ha tenido corriendo con distintos sistemas operativos como Windows 2000, 2003, Vista y 7, Solaris, así como varias distribuciones de Linux y de BSD.

Actualmente se sigue trabajando con instituciones mexicanas y de América Latina para realizar su conexión IPv6 hacia la

UNAM. Entre las instituciones mexicanas han destacado: Instituto Politécnico Nacional, Universidad Autónoma Metropolitana, Instituto Tecnológico de Estudios Superiores de Monterrey, Universidad Autónoma de Chiapas, Universidad Autónoma de Guerrero, Universidad Autónoma del Estado de Hidalgo, Universidad Autónoma de Nuevo León, Instituto Tecnológico de Oaxaca, Instituto Tecnológico de Mérida, Instituto Tecnológico Autónomo de México, PEMEX, STYX, ASTER, etc.

Entre las instituciones latinoamericanas han estado: Instituto de Informática de la Universidad Austral de Chile y las universidades UBio-Bio, UFRO y UDLA; ex-RETINA ahora InnovaRed, y las universidades LINTI-UNLP, UBA, de Argentina; EAFIT y las universidades UdeA, UniCauca y UniPamplona de Colombia; INICTEL, NITCOM, y la UNI de Perú, etc.

En enero del 2010 se pone en funcionamiento un servidor de Túneles para ofrecer conexión automática con IPv6 en RedUNAM y salir a Internet también con IPv6.

El 8 de junio de 2011, declarado como "Día Mundial de IPv6" o "IPv6 World Day" por la Internet Society (ISOC, por sus siglas en inglés), durante 24 horas la UNAM participó en este evento mundial mediante la habilitación de IPv6 en su portal principal (www.unam.mx), al cual pudieron acceder todos los usuarios de manera transparente tanto por IPv4 como por IPv6.

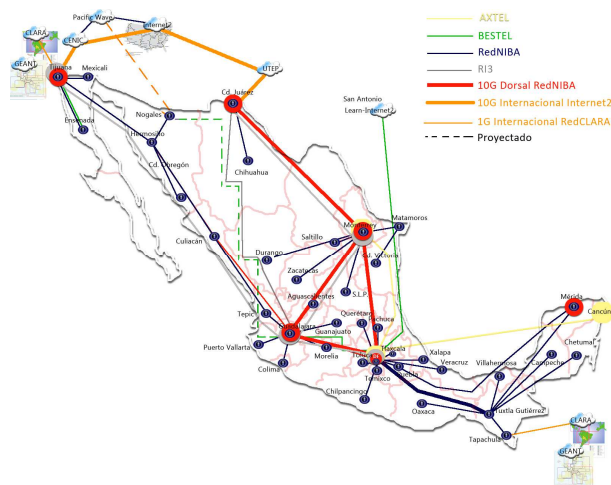
En abril del 2012, se configura y habilita la primer conexión nativa con IPv6, por Internet comercial, desde la UNAM con un ISP mexicano, previo a la participación en el "Lanzamiento Mundial de IPv6" organizado también por la Internet Society (ISOC) el 6 de junio; cuando se configura y habilita otra conexión nativa con IPv6, desde la UNAM, con un segundo ISP. [13]

B. CUDI

Asociación civil sin fines de lucro que gestiona la Red Nacional de Educación e Investigación (RNEI) para promover el desarrollo de nuestro país. Fue fundada en abril de 1999.

Con el objetivo principal de implementar aplicaciones avanzadas de Internet, así como brindar conectividad con otras iniciativas similares en otros países de Latinoamérica y el Caribe.

(RedCUDI) ha funcionado con IPv4, actualmente ya se tiene soporte, en el Backbone mediante IPv6; siendo necesario desarrollar y utilizar aplicaciones con soporte para IPv6 e IPv4. [14]



Actualizado el 24 de octubre de 2013 NOC-CUDI

Fig. 10 Backbone de la Red CUDI [15]

IX. OBSTÁCULOS DE LA ADOPCIÓN IPv6

Aunque ambos protocolos funcionan simultáneamente durante algunos años, IPv4 e IPv6 no son compatibles, por lo que se requiere generar esfuerzos para asegurarse que los servidores (web, correo y aplicaciones) estén disponibles para ambos protocolos, pero también implica que deben tomarse medidas de seguridad para cada protocolo, ya que una medida de seguridad IPv4 puede ser saltada si se utiliza un mecanismo de Transición como 4TO6.

Para implementar IPv6 debe contar un espacio de direcciones IPv6, conectividad IPv6 (nativa o por túneles), sistemas operativos, software, y herramientas de administración de redes, realizar upgrades de sus router, firewalls, y cualquier otro hardware que utilice IPv6, entrenar a su equipo de TI invirtiendo en capacitación.

Actualmente muchos ISP's (proveedor de soluciones de internet) no están preparados para soportar IPv6, muchas veces porque sus clientes no lo solicitan, o no aceptan los costos de actualizarse, o por no contar con el equipo o personal capacitado.

Muchos piensan que si no están usando IPv6 no deben preocuparse, pero es un hecho que muchas aplicaciones están preparadas para utilizar IPv6: Linux, Mac OS X, BSD, Microsoft Vista/Windows7/Windows8 tiene soporte IPv6, y en muchos esta opción esta activa por defecto, así que darán preferencia a tráfico IPv6 antes que a IPv4, al no proteger los nodos, está abierta una gran puerta trasera que puede ser explotada.

El costo de la transición es la principal razón por la baja adopción de IPv6.

Obstáculos técnicos y de diseño: IPv6 es una tecnología muy compleja comparada con IPv4, necesita mayor poder de procesamiento y memoria para soportar todas las aplicaciones. La mayoría de los routers de los ISP's son más que capaces de soportar IPv6 pero no todas las compañías tienen los dispositivos de red necesarios, con la capacidad suficiente para soportar IPv6.

Las direcciones IPv4 siguen operativas gracias a la ayuda de NAT (Network Address Translation), sin la ayuda de NAT todas las direcciones IPv4 se habrían agotado hace mucho tiempo atrás.

También existen muchas otras tecnologías que mantienen vivo a IPv4, como voz sobre IP (VoIP) todo el código escrito es para el formato de mensajes de IPv4, por defecto no es para IPv6, el código es estable en IPv4 pero no lo es en IPv6, así que existen problemas cuando se utiliza IPv6.

Otro problema fuerte es la falta de contenido IPv6, actualmente el tráfico IPv6 solo es el 5%, y la mayoría de ese tráfico es mediante mecanismos de túneles.

X. CONCLUSIONES

Las direcciones IPv4 actualmente ya se han agotado, por lo que nuevas implementaciones de redes se deberán realizar desde un enfoque organizado hacia el despliegue de IPv6, Latinoamérica debe afrontar el reto de implementar IPv6, donde se creen y apoyen las políticas concretas que exigen la adopción de redes IPv6, con incentivos que promuevan su adopción. Dicha transición deberá ocurrir sin problemas de interconexión con el mundo IP existente. Las organizaciones que estén preparadas para entrar en el mundo IPv6 serán consideradas como las primeras en adoptar una tecnología más nueva y superior. Quienes no estén preparados para este cambio, perderán el segmento de clientes IPv6.

Aunque los costos de la transición pueden ser un obstáculo inicial hacia el objetivo final, se debe comprender que la transición es inevitable. Con el transcurso del tiempo los costos de la implementación de NAT se incrementarán significativamente, por lo que los costos de mantenimiento de IPv4 superan con creces los costos de la transición hacia IPv6, lo que lo volverá IPv6 más atractivo.

IPv6 ofrece un mundo lleno de oportunidades de negocio y una solución técnica completa a muchos problemas de Internet que actualmente enfrentamos. Invertir en IPv6 tempranamente permitirá ahorrar una gran cantidad de recursos en el futuro además que asegurara su conexión con internet, ya que paulatinamente el soporte IPv4 se perderá y quienes no estén preparados para este cambios quedaran aislados.

XI. REFERENCIAS

[1] Estrategia de migración de IPv4 a IPv6 para las pymes en Colombia

Fuente: (2011) Arias Pulgarín, Héctor Fabio
<http://ribuc.ucp.edu.co:8080/jspui/bitstream/handle/10785/958/completo.pdf?sequence=1>

[2] Agotamiento de IPv4 o Transición de atiempo a IPv6 para el adecuado crecimiento de la Banda Ancha

Fuente: (2011) Palet, Jordi.
http://www.consulintel.es/pdf/ipv6_spain.pdf

[3] Tutorial de IPv6

Fuente:(2010) Fernández Alcántara, Azael
<http://www.ipv6.unam.mx/documentos/Tutorial-IPv6-UNAM.pdf>

[4] Mecanismo de transición.

Fuente: LACNIC. (2015). *PortalIPv6*. Obtenido de PortalIPv6.
<http://portalIPv6.lacnic.net/mecanismos-de-transicion/>

[5] Tecnologías de Transición IPv4- IPv6

FuentePortal IPv6 Cuba.
<http://www.cu.ipv6tf.org/transicionipv6.htm>

[6] The Story of IPv6.

Fuente: APNIC(2014) Smith, Tony.
<http://blog.apnic.net/2014/11/04/the-story-of-ipv6/>

[7] Regional Internet Registries.

Fuente: (2015) ARIN
<https://www.arin.net/knowledge/rirs.html>

[8] Proyección del consumo de restos de los Pools de Direcciones RIR

Fuente: (2015). Hustonn, Geoff. IPv4 Address Report.
<http://www.potaroo.net/tools/ipv4/>

[9] Labs.APNIC.NET - IP Resource Per Country Distribution Report.

Fuente: APCNIC. (2015). APCNIC. Obtenido de APCNIC.
<http://labs.apnic.net/dists/v6dcc.html>

[10] IPv6 en Latinoamérica.

Fuente: LACNIC. (2015). *PortalIPv6*. Obtenido de PortalIPv6.

<http://portalIPv6.lacnic.net/reporte-de-terminacion-de-direcciones-IPv4/>

[11] Transición IPv4 – IPv6

Fuente: LACNIC. (2011). PortalIPv6. Obtenido de PortalIPv6.
<http://www.cnnn.cl/documentos/documentos-2007/ipv4-ipv6-conferencia-cl.ppt>

[12] Adopción de IPv6 y transición desde IPv4

Fuente: (2014) International Telecommunication Union
<https://itunews.itu.int/Es/4207-Adopcion-de-IPv6-y-transicion-desde-IPv4.note.aspx>

[13] Historia UNAM IPv6

DTD DGSCA-UNAM
<http://www.ipv6.unam.mx/>

[14] Acerca de CUDI

Fuente: Corporación Universitaria para el Desarrollo de Internet A.C.
<http://www.cudi.edu.mx/acerca-de-cudi>

[15] Backbone CUDI

Fuente: Corporación Universitaria para el Desarrollo de Internet A.C.
<http://www.cudi.edu.mx/conexion/backbone>

Situación Actual de los Repositorios Abiertos en México

Ana María González Mendoza
Centro Universitario UAEM Texcoco
Av. Jardín Zumpango s/n, Fraccionamiento El Tejocote,
Texcoco, Estado de México, México
merlinalove@hotmail.com

Irene Aguilar Juarez
Centro Universitario UAEM Texcoco
Av. Jardín Zumpango s/n, Fraccionamiento El Tejocote,
Texcoco, Estado de México, México
ireneico@gmail.com

Resumen Después de dos décadas de publicación masiva de contenidos digitales en la Web, ahora la comunidad científica se enfrenta al reto de aprovechar de forma fácil y eficiente toda la información valiosa disponible. Para ello entre otras estrategias se ha centralizado la producción académica en repositorios abiertos, los cuales representan la fuente de información de mejor calidad para el desarrollo de la educación y de la investigación científica pues en ellos se almacena todo tipo de documentos académicos sobre cualquier tema del conocimiento humano. En este trabajo se describe el software de licencia libre que permite la implementación de estos repositorios, haciendo un análisis comparativo de las herramientas de desarrollo disponibles, posteriormente se describe la situación actual de los repositorios abiertos en el mundo y finalmente se describe la situación de estos repositorios en México

Keywords— *EPrints, DSpace, Fedora, Greenstone, Omeka, OAI-PMH, Repositorios Mexicanos.*

I. INTRODUCCION

El movimiento de Acceso Abierto se basa en el convencimiento de que la libre circulación de las ideas no sólo es un derecho humano, sino que se trata de un instrumento de mejora de la propia ciencia. Surge de la necesidad imperiosa de hacer accesibles los conocimientos a todos los que puedan utilizarlos, aplicarlos o aumentarlos, su finalidad es hacer un mejor uso de los recursos, minimizar la duplicación de esfuerzos, encontrar fuentes que permitan fortalecer el discurso de la investigación. Se originó formalmente en el año 2001 y actualmente múltiples organizaciones reconocen y favorecen este movimiento, entre ellos el European Research Council y el European Research Advisory Board, los consejos de investigación científica del Reino Unido y el Irish Research Council.

Los repositorios de acceso abierto son portales que permiten el acceso a publicaciones científicas o técnicas, de forma gratuita, y de esta forma ser leídos, descargados, copiados, distribuidos, impresos, buscados o enlazados y utilizados con propósitos legítimos ligados a la investigación científica, a la educación o a la gestión de políticas públicas, sin otras barreras económicas, legales o técnicas que las que suponga Internet.

En los últimos 15 años han surgido una gran variedad de herramientas para la implementación de bibliotecas digitales y repositorios. Si bien esto trae consigo un amplio abanico de posibilidades para buscar la que mejor se adapte a nuestras necesidades; puede resultar una tarea ardua el seleccionar una de ellas.

En este trabajo se han analizado 5 de estas herramientas: Eprints, DSpace, Fedora, Greenstone y Omeka; todas ellas con dos características en común: Ser de software libre y compatible con el protocolo de interoperabilidad OAI (Open Access Initiative).

En la primera parte de este documento, hablamos de estas herramientas y sus características. Como resultado se pueden apreciar sus coincidencias y particularidades.

Posteriormente se muestra un análisis de repositorios desde una perspectiva global, conoceremos cual es la herramienta líder a nivel mundial y cuál es el país que cuenta con más repositorios; para terminar con una aproximación a Latinoamérica y finalmente la situación en la que se encuentra México. Cerramos este documento presentando las conclusiones.

II. HERRAMIENTAS DE SW LIBRE PARA LA CREACION DE REPOSITARIOS

Anteriormente otros autores como George Pirounakis,

A. EPrints

El software EPrints se convirtió en el primero y uno de los más utilizados softwares de acceso libre y gratuito para la implementación de repositorios institucionales y ha inspirado desde su desarrollo a otros software que cumplen propósitos similares.

EPrints es utilizado para la creación de repositorios y es compatible con el protocolo OAI (Open Archives Initiative) para la recolección de metadatos. Comparte muchas de las características comúnmente observadas en los sistemas de gestión de documentos, pero ha sido utilizado, principalmente, para la implementación de repositorios institucionales y revistas científicas. EPrints fue creado en la University of Southampton School of Electronics and Computer Science y liberado bajo una licencia GPL, General Public License por sus

siglas en inglés. Su primera versión es lanzada en el año 2000 como resultado de una reunión en el año 1999 en Santa Fe que puso en marcha lo que eventualmente se convirtió en el protocolo OAI-PMH (Open Archives Initiative - Protocol for Metadata Harvesting).

EPrints es una aplicación web con arquitectura LAMP (donde la inicial P corresponde a Perl) y es multiplataforma (Linux, Solaris, Mac OS X y Windows).

La configuración de un repositorio EPrints implica la modificación de los archivos de configuración escritos en Perl o XML. La publicación del repositorio es en plantillas HTML, hojas de estilo CSS e imágenes en línea. Es una herramienta multilingüe; ha sido traducido a otros idiomas a través de archivos frase XML específicos del idioma, incluyen búlgaro, francés, alemán, húngaro, italiano, japonés, ruso, español y ucraniano.

Existen más de 300 repositorios de acceso abiertos registrados que han sido implementados con EPrints alrededor del mundo. [1]

B. DSpace

DSpace es un software de código abierto que provee herramientas para la administración de colecciones digitales, el uso más común es en la creación de repositorios institucionales.

La primera versión de DSpace la 1.0 fue liberada el cuatro de noviembre de 2002, como resultado de un esfuerzo conjunto por parte del Instituto Tecnológico de Massachusetts MIT y los laboratorios Hewlett-Packard, bajo una licencia GPL que permite a los usuarios personalizar o extender el software según se necesite. El uso de esta licencia no tiene ningún costo y en caso de hacer una mejora o modificación del código se sugiere compartirla a la comunidad DSpace como forma de retribución; los *committers* de esta comunidad son desarrolladores voluntarios seleccionados en un grupo específico y se encargaran de depurar y añadir estas mejoras.

DSpace está escrito en Java. Emplea Base de Datos Oracle o PostgreSQL. Es multiplataforma compatible con Linux, Mac OSX y Windows. Soporta los protocolos: OAI-PMH, OAI-ORE, SWORD, WebDAV, OpenSearch, OpenURL, RSS, ATOM. La interface es totalmente personalizable. Reconoce los tipos de archivo más habituales como son textos, imágenes, videos y audio; además de poder agregar nuevos formatos. Utiliza metadatos por default Dublin Core, pero es posible usar MARC Y MODS. Sus campos para realizar búsqueda son personalizables y se encuentra disponible en más de 20 idiomas.

DSpace enfatiza entre sus características, ser una herramienta para preservar a través del tiempo los contenidos, por lo que al utilizarla le concedemos la libertad de realizar copias de los ítems con fines de conservación y respaldo; por su parte DSpace siempre hará mención del autor y no hará modificaciones a los contenidos.

DSpace es el software libre para la implementación de repositorios más popular a nivel mundial; es utilizado por más de 1000 repositorios de acceso abierto en todo el mundo. [2]

C. Fedora

FEDORA (Flexible Extensible Digital Object Repository Architecture), también llamado Fedora Commons, es liderado por el Fedora Project Steering Group bajo la administración de DuraSpace. Fue originalmente desarrollado en 1997 por Sandy Payette and Carl Lagoze en la Universidad de Cornell y fue pensado para almacenar, gestionar y tener acceso a objetos digitales. En el 2000 Thornton Staples y Rosser Wayland, crearon el primer prototipo y para el 16 de mayo de 2003 es liberada la primera versión al público de Fedora; la 1.0.

Fedora define un conjunto de abstracciones para expresar a los objetos digitales; definiendo sus relaciones y sus comportamientos. Es de código abierto y ofrece un servicio central basado en APIs; además de poseer otras características como búsquedas, protocolo OAI-PMH, administración, etc. Una característica clave de Fedora es su flexibilidad para soportar todo tipo de contenido digital.

Ofrece soporte RDF y el software repositorio está integrado con la tecnología de almacenamiento de triple semántica, incluyendo la base de datos Mulgara RDF. Fedora ayuda a garantizar que el contenido digital es duradero, proporcionando características que apoyan la conservación digital.

Cuenta con una comunidad que ha puesto herramientas adicionales, aplicaciones y utilidades que aumentan las posibilidades del repositorio de Fedora incluyendo Hydra, Islandora y Avalon. Fedora es usado para la gestión de objetos digitales en colecciones digitales, bibliotecas digitales, repositorios institucionales, preservación de archivos, entre otros.

La arquitectura del servidor FEDORA se basa en 4 APIs principales: gestión, acceso, búsqueda y el Open Archival Initiative Service (para obtención de metadatos).

Existen más de 30 repositorios de acceso abiertos registrados alrededor del mundo que usan Fedora. [3]

D. Greenstone

Greenstone es un conjunto de programas para crear y distribuir colecciones de bibliotecas digitales en Internet o en CD-ROM. Es un software libre, multilingüe, publicado bajo los términos de la Licencia GPL. Greenstone lo produce el New Zealand Digital Library Project en la Universidad de Waikato, y ha sido desarrollado y distribuido en cooperación con la Unesco y la ONG Human Info de Bélgica.

Los desarrolladores de Greenstone recibieron el Premio Namur 2004 de la International Federation for Information Processing por contribuciones a la concienciación de las implicaciones sociales de las tecnologías de la información, y la necesidad de un enfoque holístico en el uso de tecnologías de la información que tenga en cuenta sus implicaciones sociales.

Greenstone se puede usar para crear grandes colecciones de documentos digitales en las que se pueden hacer búsquedas. Además de las herramientas en línea de órdenes para la construcción de bibliotecas digitales, Greenstone tiene la Interfaz para Bibliotecarios de Greenstone (GLI), gráfica, que se usa para construir colecciones y asignar metadatos.

Por medio de complementos seleccionados por el usuario, Greenstone puede importar documentos digitales en diversos formatos, incluyendo texto, HTML, jpg, tiff, MP3, PDF, video, y Microsoft Word, entre otros. El texto, PDF, HTML y documentos similares se convierten al Formato de Archivo de Greenstone (GAF) que es un formato XML equivalente. Greenstone cuenta con una interfaz en PHP llamada EmeraldView. [4]

Existen poco más de 50 repositorios abiertos registrados que han sido implementados con Greenstone.

E. Omeka

Omeka es un software libre, flexible y de código abierto pensado para la publicación en web de colecciones de bibliotecas digitales, archivos, museos o cualquier otra institución que desee difundir su patrimonio cultural. Omeka ha sido desarrollado por el Rosenzweig Center for History and New Media de la Universidad George Mason quien también es responsable del gestor bibliográfico Zotero. Se trata de una plataforma interoperable, que cumple con el estándar OAI-PMH.

La gran diferencia con otros programas pensados para construir repositorios digitales, es que Omeka no sólo es capaz de almacenar y mostrar colecciones digitales, sino que también permite crear exposiciones virtuales en torno a éstas. El sistema permite asociar una plantilla (aparición) diferente a cada exposición. El sitio Europeana www.europeana.eu utiliza Omeka para distribuir sus exposiciones virtuales.

Entre sus características soporta cualquier tipo de archivo, usa metadatos Dublin Core, creación de diferentes perfiles de usuarios.

Cuenta con una gran variedad de plug-ins para añadir funcionalidad. Permite personalizar las búsquedas. Cuenta con servicios 2.0 como Google Translate para que nuestros visitantes puedan traducir la interfaz de Omeka o Tweetster for Omeka que permite publicar tweets automáticamente cada vez que se publique un nuevo ítem en el repositorio.

Se pueden crear árboles jerárquicos de las exposiciones o personalizar el orden de aparición de los ítems.

Además se permite añadir geolocalización a nuestros ítems y situarlos en un mapa o crear líneas de tiempo. Entre otras muchas funcionalidades. [5]

III. COMPARATIVA

Una vez que hemos conocido algunos aspectos importantes de estas herramientas vale la pena realizar las siguientes observaciones.

Los primeros esfuerzos para crear repositorios abiertos no se dio en el continente americano y en contraste una de las herramientas que terminaría siendo la más utilizadas en el mundo sí; nos referimos a DSpace. Al tratarse de licencias GPL o GNU (General Public License) por sus siglas en inglés; todas estas herramientas cuentan con comunidades que dan soporte. La primera herramienta para desarrollar repositorios

de acceso abierto fue Eprints aunque en la actualidad no es la más popular y uno de las herramientas para la creación de repositorios de acceso abierto más actuales es Omeka.

Todas estas herramientas soportan los tipos de datos más comunes: .doc, .pdf, .jpg, .gif, .mp4, .mov, etc. y muchos de ellos pueden agregar nuevas extensiones de archivo para que nuevos formatos puedan ser almacenados. Todas ellas cuentan con soporte multilinguaje. Se pueden personalizar las búsquedas y permite registro y autenticación de usuarios. Además de contar con interfaces personalizables.

En la Tabla I podemos observar que existe diversidad en los lenguajes en los cuales han sido desarrolladas estas herramientas; así como las Bases de Datos que manejan. Por lo que debemos estar seguros que podemos encontrar una que se adapte a nuestros requerimientos específicos. Otra característica que vale la pena resaltar es que todas ellas son compatibles con el protocolo OAI-MPH y los metadatos Dublin Core. Una cosa a tener en cuenta al momento de pensar en la interoperabilidad.

TABLE I. COMPARATIVA

	EPrints	Green stone	DSpace	Fedora	Omeka
Lanzamiento	EPrints 1.0 Noviembre 2000	Green stone 2001	DSpace 1.0.1 4 de noviembre de 2002	Fedora 1.0 2003	Omeka 1.0 2 de Junio de 2009
Ultima version	EPrints 3.3.13	Green stone 2.86	DSpace 5.0	Fedora 4.1.0	Omeka 2.2.2
Almacena	Data Object	Docu ment	Item	Digital Object	Item
Langu age	Pearl	Java y C++	Java	Java	PHP
Base de Datos	MySQL , Oracle y Postgre SQL	Gestor propio	Oracle y Postgre SQL	Depen de del sw utilizad o en conjunt o	MySQL y PostgreS QL
Protoc olo OAI-PMH	Si	Si	Si	Si	Si
Metad atos Dublin Core	Si	Si	Si	Si	Si

IV. REPOSITARIOS A NIVEL MUNDIAL

Existen varios sitios web donde se pueden realizar consultas de los repositorios implementados con distintos softwares, sin embargo los datos presentados en este documento se obtuvieron de dos fuentes: Los sitios webs oficiales de los repositorios y datos referidos en la página OpenDOAR.

OpenDOAR es un prestigioso directorio de repositorios institucionales de acceso abierto; es un proyecto liderado por la Universidad de Nottingham, en Reino Unido. Cada repositorio que se agrega en el directorio es previamente analizado por un equipo de especialistas del proyecto, quienes evalúan rigurosamente y en forma manual la calidad de la información que contiene cada uno de ellos. Es importante hacer énfasis en esto, ya que muchos otros repositorios, si bien existen; no se encuentran incluidos este directorio, y por lo tanto queda fuera de este análisis.

OpenDOAR permite realizar búsquedas tomando en cuenta distintos criterios, como son: Materia, Tipo de contenido, Tipo de repositorio, País, Idioma y Software. Y nos permite ordenar los resultados por: Nombre, País, Número de ítems, OAI URL y Software. Cuenta con una sección para visualizar los datos en gráficas y proporciona información en profundidad sobre cada repositorio que puede ser utilizada para su análisis.

OpenDOAR fue identificado por KB Oliver y R. Swain en 2006 como el líder de directorios de repositorios en un estudio realizado por la Universidad de Johns Hopkins. Y fue uno de los servicios que contribuyeron para que SHERPA fuera galardonado con el Premio Europa SPARC 2007. [6]

Como podemos apreciar en la Figura 1, el software más utilizado a nivel mundial es DSpace dejando muy por detrás a Eprints ubicado en el segundo lugar. En tercer lugar se ubican repositorios que no especifican el software con el que han sido desarrollados; como por ejemplo el sitio de Redalyc en México el cual cuenta con más de 200,000 mil ítems.

En la cuarta posición está ubicado Digital Commons desarrollado por Bepress, su licencia es de tipo comercial y es líder en ofrecer repositorios con servicio de hosting. Es compatible con el protocolo OAI-PMH. Utiliza metadatos Dublin Core. Esta desarrollado en Pearl y utiliza una base de datos PostgreSQL. [7]

En el quinto lugar de esta grafica encontramos a OPUS; originalmente se desarrolló con el apoyo de la Deutsches Forschungsnetz en 1997 y 1998 en la Universidad de Stuttgart, desde 2010 su desarrollo quedo a cargo de Cooperative Library Network of Berlin-Brandenburg (KOBV), en Berlín Alemania. Está escrito en PHP, utiliza Base de Datos MySQL, es compatible con el protocolo OAI-PMH y utiliza metadatos XmetaDiss. [8]

En Alemania, el software OPUS es el más comúnmente utilizado para la implementación de repositorios de acceso abierto. Cabe señalar que de acuerdo a OpenDOAR en el continente Americano no hay repositorios registrados desarrollados con OPUS.

Otro dato interesante, que podemos observar en la Figura 2, es que de 2849 repositorios registrados en OpenDOAR el 45.5% de los repositorios a nivel mundial están en Europa.

En la Figura 3 se observar que el país que tiene más repositorios es Estados Unidos, y el país latinoamericano que sobresale es Brasil.

La mayoría de los repositorios a nivel mundial son de tipo institucional y el mayor tipo de documentos que almacenan son artículos científicos, seguidos por documentos de tesis. Mas de

la mitad de estos repositorios están en inglés, y el segundo idioma que predomina es el español. [6].

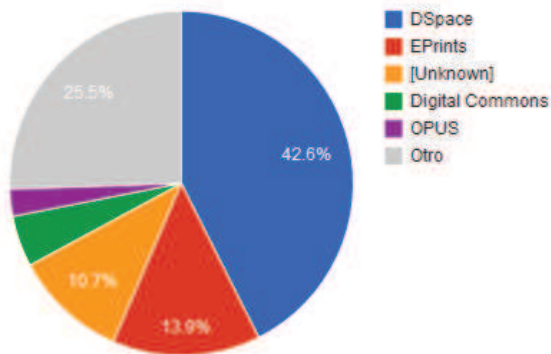


Fig. 1. Software utilizado a nivel mundial para la creación de repositorios de acceso abierto

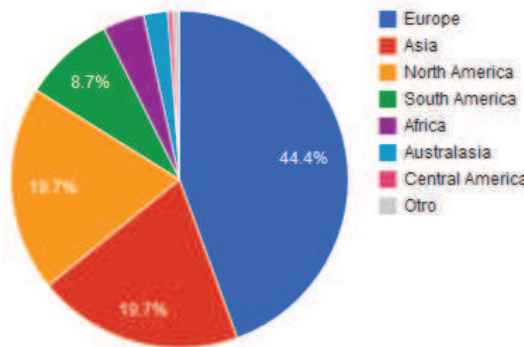


Fig. 2. Gráfica de los repositorios registrados de acuerdo al continente.

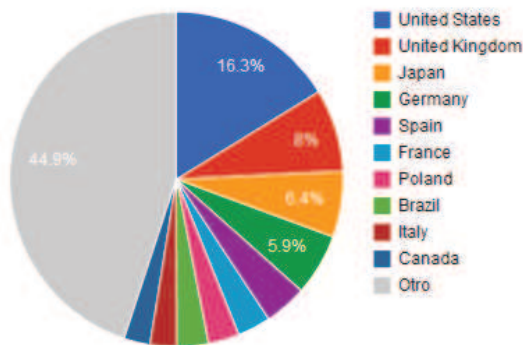


Fig. 3. Grafica de repositorios registrados por país.

V. REPOSITARIOS EN MEXICO

Siguiendo esta pauta quisimos ver la situación de los repositorios en Latinoamérica, podemos observar que Brasil es

el país latinoamericano con más repositorios. México se ubica en la quinta posición como podemos apreciar en la Figura 4.

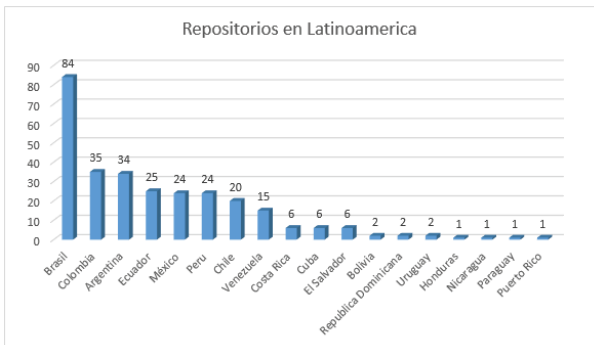


Fig. 4. Repositorios de acceso abierto en Latinoamérica.

Como se puede apreciar en la Figura 5 en México más del 50% de estos repositorios están desarrollados en DSpace repitiendo la tendencia que hemos visto reflejada a nivel mundial; 5 de ellos no proporcionan datos del software con el que se han desarrollado, 2 están implementados con EPrints, 1 con Catia, 1 con HTML, 1 con Open Repository, 1 más con SciELO y 1 desarrollo propio. La mayoría de estos repositorios manejan una base de datos PostgreSQL.

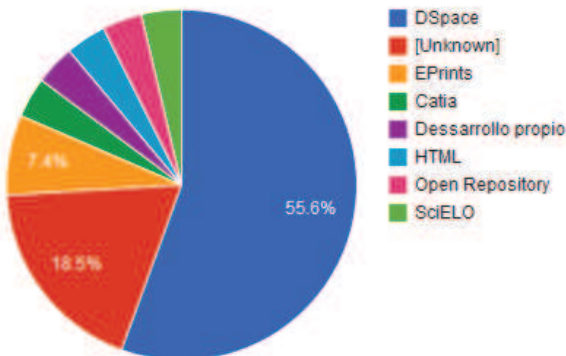


Fig. 5. Herramientas utilizadas en México para crear repositorios de acceso abierto.

En la Figura 6 podemos observar que la mayoría de los contenidos que albergan los repositorios en México consiste en artículos, seguido por los trabajos de tesis y disertaciones. Esto se debe a que los encargados de estos repositorios son las instituciones de educación superior. No es de extrañarse que estos contenidos sean los que prevalecen en los repositorios.

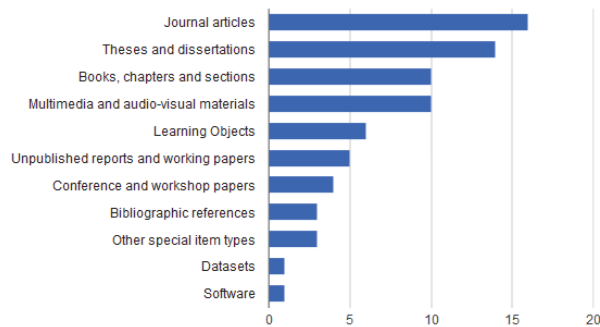


Fig. 6. Tipo de contenidos de los repositorios mexicanos.

En la Figura 7 observamos los temas que predominan en los repositorios mexicanos. En primer lugar encontramos los repositorios multidisciplinarios, seguidos por ciencias en general. Podemos intuir que conforme va madurando la idea de repositorios se van volviendo más especializados, no es una casualidad que los primeros repositorios al surgir en instituciones de educación superior fueran pensados para albergar toda la propuesta académica de los mismos.

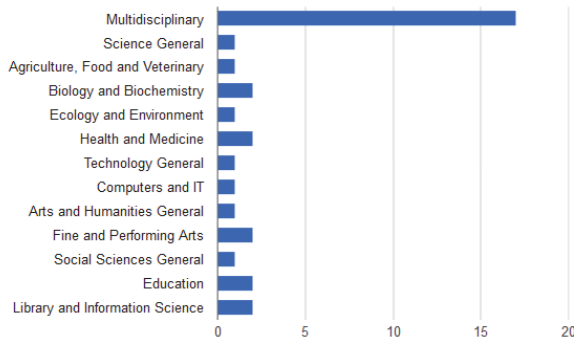


Fig. 7. Temas predominantes en los repositorios mexicanos.

Otro dato interesante y que vale la pena resaltar es que en 16 de estos repositorios, es posible recolectar datos mediante el protocolo OAI-PMH. Y que de 2010 a la fecha hubo un repunte en la implementación de repositorios pasando de 12 a 27 que son los que se encuentran registrados en OpenDOAR al día de hoy.

CONCLUSIONES

Como podemos observar es un tema muy amplio con muchas vertientes, existen más de 100 herramientas para la implementación de repositorios, todas ellas desarrolladas en distintos lenguajes, compatibles con distintos protocolos, tipos de metadatos y bases de datos.

Cada uno de ellos con distintos enfoques y filosofías. Es importante tener en cuenta que la tendencia apunta a crear repositorios de acceso abierto, sin restarle importancia a todas las implicaciones legales que pudiese traer consigo la

distribución de estos contenidos sin el pleno consentimiento de su creador. El reto ya no es únicamente implementar un repositorio, ya que con estas herramientas se facilita mucho la parte técnica; el verdadero reto está en hacerlo correctamente, empleando las técnicas adecuadas para el correcto meta etiquetado y más allá de eso que los contenidos estén correctamente almacenados, garantizar su preservación y su posterior recuperación, sin dejar de lado que estos contenidos deben ser de calidad. Solamente así es posible acercarnos al verdadero ideal de la interoperabilidad y alcanzar el verdadero objetivo de los repositorios de acceso abierto. Quizás es por ello que Alemania se ha preocupado tanto en que todos sus repositorios sean desarrollados con una misma herramienta. Ya no se trata de almacenar por almacenar. En el sitio Repositories Support Project [9], podemos encontrar algunas directrices para crear y gestionar repositorios exitosos.

La implementación de un repositorio que cumpla con los ideales para los que fue creado no es una tarea sencilla, por lo que se debe tener en claro los objetivos que se pretenden alcanzar y de este modo enfocarse y buscar las opciones que mejor cumplan nuestras expectativas; de lo contrario podría convertirse en un largo y complejo sendero.

Como trabajo futuro sería interesante aplicar métricas que nos permitan evaluar ciertas características de los repositorios implementados actualmente en México y de esta forma analizar qué tan cerca nos encontramos de alcanzar el ideal de interoperabilidad.

REFERENCIAS

- [1] EPrints, "EPrints," 2015. [Online]. Available: <http://www.eprints.org>. [Accessed 2015].
- [2] DSpace, "DSpace," 2015. [Online]. Available: <http://www.dspace.org/>. [Accessed 2015].
- [3] F. Commons, "Fedora Commons," 2015. [Online]. Available: <http://www.fedorarepository.org>. [Accessed 2015].
- [4] Greenstone, «Greenstone,» 2014. [En línea]. Available: www.greenstone.org. [Último acceso: 2014].
- [5] Omeka, "Omeka," 2015. [Online]. Available: www.omeka.org.
- [6] OpenDOAR, "OpenDOAR Directory of Open Access Repositories," 2015. [Online]. Available: www.opendoar.org. [Accessed 2015].
- [7] Bepress, "Digital Commons," 2015. [Online]. Available: <http://digitalcommons.bepress.com/>. [Accessed 2015].
- [8] K.-Z.-Z. f. I. B. ZIB, «OPUS 4,» 2015. [En línea]. Available: <http://www.kobv.de/opus4>. [Último acceso: 2015].
- [9] Jisc, "Repositories Support Project," 2015. [Online]. Available: <http://www.rsp.ac.uk/>. [Accessed 2015].
- [10] M. N. George Pirounakis, "Comparing Open Source Digital Library Software," Grecia, 2015.
- [11] I. P. D. S. D. P. A. R. G. Ileana Dayamina de la Cruz Santos, «Biblioteca virtual de la EcuRed: nueva fuente de información en la red nacional cubana,» 2011. [En línea]. Available: <http://www.redalyc.org/articulo.oa?id=181418901007>.
- [12] M. Á. G. M. Argueta, «LA INTEROPERABILIDAD Y EL INTERCAMBIO DE METADATOS EN LA RED,» 2012. [En línea]. Available: <http://www.revista.unam.mx/vol.13/num9/art95/index.html>.
- [13] R. S. B. M. Kathleen B. Oliver, "Directories of Institutional Repositories: Research Results & Recommendations," 2006.

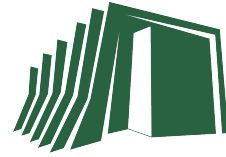
Primer Congreso Internacional Mexicano en Sistemas y Computación
MICCS 2015



Clasificación de comportamientos riesgosos entre peatón y conductor utilizando algoritmos de Minería de Datos.

Rebeca Roldán López
Maricela Quintana López

Centro Universitario UAEM Valle de México, Universidad Autónoma del Estado de México.
Blvd. Universitario s/n Predio San Javier Atizapán de Zaragoza. C.P. 54500. Estado de México, México.



INTRODUCCIÓN

Los seres humanos estamos expuestos a sufrir algún accidente toda nuestra vida, sobre todo en cuestiones viales. De acuerdo con la Organización Panamericana de la Salud, en todas las subregiones de América tanto peatones, motociclistas y ciclistas son principales víctimas fatales a causa del tránsito, representando el 23%, 15% y 3% de las muertes, respectivamente. Los ocupantes de los automóviles tampoco son exentos de ser víctimas. Es necesario identificar los posibles factores que producen interacciones nocivas entre peatón y conductor para prevenir accidentes no fatales como fatales, para mejorar la calidad de vida.

OBJETIVOS

- 1 Simular el comportamiento de conductor y peatón en un escenario virtual para generar datos.
- 2 Identificar variables que afecten la interacción entre conductor y peatón que den lugar a un percance.
- 3 Utilizar distintos algoritmos de minería de datos para clasificar los accidentes según la gravedad de ocurrencia.

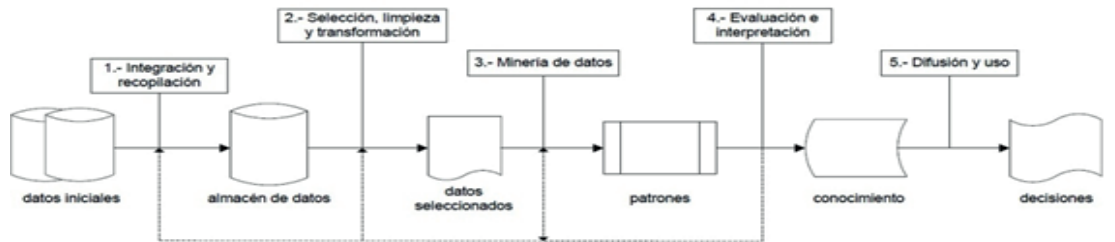


HIPÓTESIS

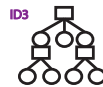
Con el uso de minería de datos se puede contruir un clasificador que determine, basado en el comportamiento del conductor y del peatón, si la interacción entre ellos conduce a un accidente.



PROCESO KDD KNOWLEDGE DISCOVERY IN DATABASES



Herramienta de simulación



Peatón	Conductor	Clasificación		
Cruza por cebras	Usa puente peatonal	No respeta semáforo	Exceso velocidad	Daño material
Cruza por cebras	Atiende celular	No respeta semáforo	Atiende celular	Lesionado
Ignora puente peatonal	Cruza por cebras	No respeta semáforo	Atiende celular	Grave
Estado inconveniente	Cruce prohibido	Estado inconveniente	Exceso velocidad	Fatal

USO

Proponer un diseño vial que reduzca la ocurrencia de accidentes de una vía pública con altos índices de incidentes.

REFERENCIAS

1. García-Valdecasas Medina, J. I. (2011). La simulación basada en agentes: una nueva forma de explorar los fenómenos sociales. *Reis. Revista Española de Investigaciones Sociológicas*, (136) 91-109.
2. Wang, H., Parrish A. et. al. (2005). Variable Selection and Ranking for Analyzing Automobile Traffic Accident Data. *Association for Computing Machinery. ACM Symposium on Applied Computing*. (32-37).
3. Jain, S., Borgiattino, C. et. al. (2014). On the Limits of Positioning-based Pedestrian Risk Awareness. *ACM*. (23-28).
4. Fray, D., Hui, P. et. Al. (2013). Knowledge Discovery and Causality in Urban City Traffic: A study using Planet Scale Vehicular Imagery Data. *ACM*. (67-72).



Segmentación de Imágenes Aplicado a Visión Artificial con Drones

Miguel Angel Moreno-Martínez, J. C. Rodríguez-Sánchez, V. Landassuri-Moreno, José Martín Flores Albino, Saúl Lazcano Salas
Centro Universitario UAEM Valle de México

Universidad Autónoma del Estado de México C.P. 54400, México.

mluigiz12_z@hotmail.es, jrcsylatp@yahoo.com, vmlandassurim@uaemex.mx, jose_martin_70@yahoo.com, slazcanos@uaemex.mx



Resumen: En este trabajo se propone un proyecto el cual tiene por objetivo abordar la segmentación de imágenes para drones utilizando la cámara frontal e inferior que ya trae integradas. Esto para lograr vuelo autónomo de dichos dispositivos



1. Introducción:

Los algoritmos de segmentación de imágenes son necesarios para separar información útil de una imagen. Estos son aplicados en diferentes áreas:

- Área de exploración, con técnicas de filtro digital de convolución espacial [1]
- Área deportivas, técnica de la clasificación de la silueta pictórica [2]
- Automotriz, técnica bottom-hat, Osu's [3]

2. Drones

Hay muchos modelos, marcas, los drones o UAV en sus siglas en inglés Vehículo aéreo no tripulado, tiene lineamientos de vuelo que no se abordaran en este artículo

Sus usos:

- Se puede usar en filmaciones
- Expediciones o explorar ruinas
- Equipos de rescate
- Otros



Fig. 1 Ejemplo de Segmentación de herramientas por medio de fotos, lo cual puede ser aplicado a Drones para detección de formas y objetos que obstaculicen su paso

3. Segmentación de imágenes:

Mediante la segmentación de imágenes es posible reconocer formas y objetos que permitan indicar un comando al Dron, por ejemplo:

- Subir
- Bajar
- Girar
- Despegar/aterizar
- etc.

Esto puede hacerse con filtros que nos ofrecen las librerías del software o paquetes de estos programas como el filtro de partículas, que es usado para que el dron siga a una persona por el color de su ropa, el filtro guassiano, gradiente, rainbow etc. Se puede procesar las imágenes para registrar placas de los automóviles [4]. El reconocimiento de caracteres o características del objeto(placas, gestos, etc., en diferente horario y distancia [5].

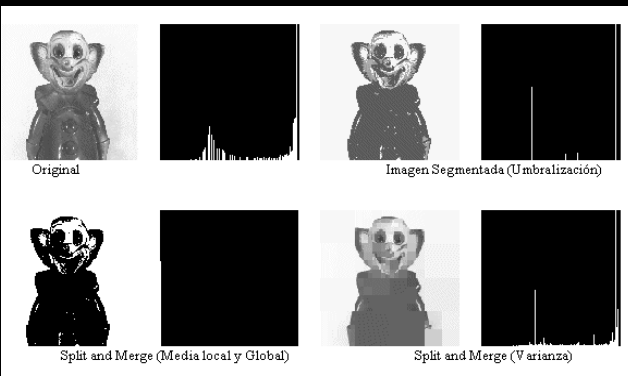


Fig. 2. Espectro de la segmentación

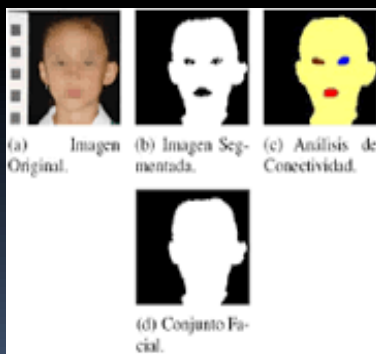


Fig. 3. título de la figura

4. Trabajo futuro y Resultados esperados

La siguiente parte es iniciar con el procesamiento de imágenes, implementando un algoritmo de segmentación.

Se espera obtener un Dron que vuele autonomamente, siguiendo objetos, como por ejemplo, al reconocer cierta placa de automovil, este lo seguira. Así como habrá otras plicaciones que se podrán generar.

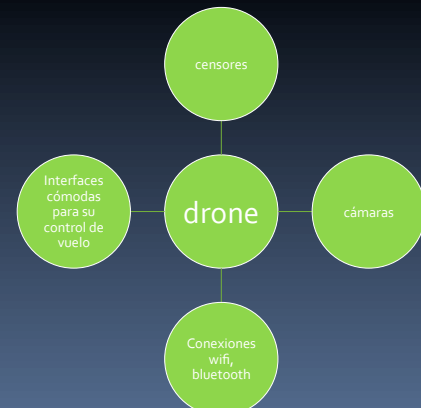


Fig. 4 componentes de los drones

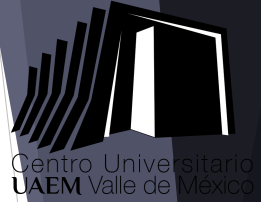
REFERENCIAS

- [1] Jorge Márquez Flores [Procesamiento y Análisis Digital de Imágenes] (2013) 1-86
- [2] Daniel Soto Guerrero, (Dr. José Gabriel Ramírez Torres /director de tesis/) (2012) 153
- [3] Fikriye Öztürka, Figen Özena [A New License Plate Recognition Based on Probabilistic Neural Networks] (2012) 1-5
- [4] H.Erdinc Kocera, K.Kursat Cevikb [Artificial neural networks based vehicle license plate recognition] (2011) 1033-10370
- [5] Kaushik Deba, Md. Ibrahim Khana, Anik Sahaa, and Kang-Hyun Job [An Efficient Method of Vehicle License Plate Recognition Based on Sliding Concentric Windows and Artificial Neural Network] (2012) 812 – 819



Sistema de Reconocimiento de Caracteres de Placas de Automóviles en México

Edgar Ricardo Palacios Rojas, Víctor Manuel Landassuri Moreno, Saúl Lazcano Salas
Centro Universitario UAEM Valle de México, Universidad Autónoma del Estado de México
Bld. Universitario s/n Predio San Javier Atizapán de Zaragoza, C.P. 54500, Estado de México.
epr242016@gmail.com, vmlandassurim@uaemex.mx, slazcanos@uaemex.mx



Resumen

En este trabajo se empleará Segmentación imágenes y Redes Neuronales Artificiales (RNAs) para realizar la clasificación de caracteres de placas de auto, con el objetivo de contar con un sistema automático para ello. Así, se aplicarán diversas técnicas de segmentación para poder definir el método más óptimo, junto con RNAs del tipo Multi capa con entrenamiento Backpropagation, las cuales han mostrado un nivel aceptable de clasificación.



Objetivo

Realizar un Sistema de Reconocimiento de Caracteres de placas de automoviles que sea funcional para cualquier estado de la república mexicana.

Objetivos Particulares:

- Generar una Base de datos de placas de automóvil
- Desarrollar el algoritmo de segmentación más óptimo para éste caso
- Desarrollar el reconocimiento de caracteres empleando las RNAs
- Diseño de una Interfaz gráfica (GUI)



Introducción

- La elección del reconocimiento de caracteres de placas de automóviles sigue representando un gran reto hoy en día, dado los diferentes factores que pueden interferir a la hora de hacer el reconocimiento, como neblina, lluvia entre otros
- Por otro lado, las placas de autos en algunos estados del país están estructuradas así: "AAA-99 99", en su contraparte, el Distrito Federal tienen el siguiente formato "AAA-999", lo que implica diseñar algoritmos e interfaces de software que sean genéricas
- Este trabajo se enfocará en explicar cómo hacer reconocimiento de patrones de forma genérica con RNAs



Problemática

Por mencionar algunos de los aspectos a los que se enfrentará este trabajo son:

- Naturaleza
- Luminosidad
- Movimiento
- Tamaño
- Distancia
- Actualización en formatos placas en los distintos y estados



Fig.1 Pre-procesamiento Operaciones para adaptar la información de una imagen y tener mejor análisis en pasos posteriores.

Tipos de segmentación[1]:

- Segmentación de forma (SS)
- Segmentación de bordes (CS)
- Segmentación de color (CrS)
- Segmentación de movimiento (VS)
- Segmentación de textura (TS)

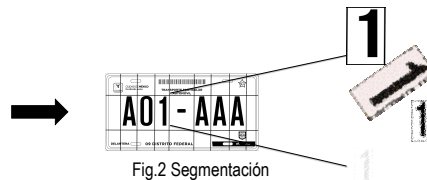


Fig.2 Segmentación

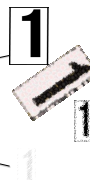


Fig.3 Entrada

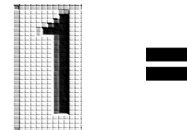


Fig.4 Detección y clasificación

ES UN "1" ✓

ERROR ✗

Fig.6 Análisis de la imagen

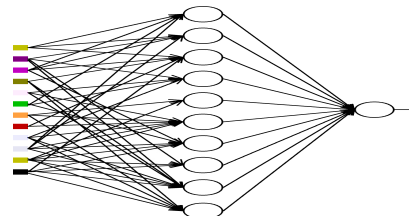


Fig.5 Modelo neuronal artificial



Resultados Esperados

- Hacer una predicción correcta de la placa de automóvil con imágenes sin ruido
- Permitir una calificación correcta de la placa, para cualquier estado de la república



Trabajos futuros

- Diseñar un algoritmo de segmentación de imágenes
- Dejar un Framework de dichas técnicas en el CUUAEM-VM
- Diseñar RNAs que permitan la clasificación de caracteres alfanuméricos



Referencias

- [1] Ramirez Q. Juan A. y Chacón M. Mario I., "Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década", RIEE&C, REVISTA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y COMPUTACIÓN, Vol. 9 No. 1, JULIO 2010
- [2] Egmont-Petersen, Ridder, Handels, "Image processing with neural networks—a review", Pattern recognition, vol. 35, no. 10, pp. 2279-2301, Oct 2002
- [3] Ernesto Varela Arregocés y Edwin Campbells Sánchez, "Redes Neuronales Artificiales: Una Revisión del Estado del Arte, Aplicaciones Y Tendencias Futuras", {evarelabqool, ecampbells}@hotmail.com, Universidad Simón Bolívar Barranquilla – Atlántico
- [4] Daniel Soudry, Itay Hubara, Ron Meir, "Expectation Backpropagation: Parameter-Free Training of Multilayer Neural Networks with Continuous or Discrete Weights"
- [5] C Baldassi, A Braunstein, N Brunel, and R Zecchina, "Efficient supervised learning in networks with binary synapses", PNAS, 104(26):11079–84, 2007